SYM-AM-21-042

PROCEEDINGS
OF THE
EIGHTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY, MAY 13, 2021 SESSIONS
VOLUME III

Acquisition Research:
Creating Synergy for Informed Change

May 11–13, 2021

Published: May 5, 2021

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Table of Contents

SYM-AM-21-042

Proceedings
of the
Eighteenth Annual Acquisition Research Symposium

Thursday, May 13, 2021 Sessions
Volume III

Acquisition Research: Creating Synergy for Informed Change

May 11-13, 2021

Published: May 5, 2021

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

THIS PAGE INTENTIONALLY LEFT BLANK

WELCOME: DAVID H. LEWIS, VICE ADMIRAL, U.S. NAVY (RET), ACQUISITION CHAIR, ACQUISITION RESEARCH PROGRAM

Vice Admiral David H. Lewis, USN (Ret.) took the helm as the Naval Postgraduate School Chair of Acquisition. As chair, he will lead the Acquisition Research Program (ARP) in the Graduate School of Defense Management and connect NPS with leaders and policymakers in the acquisition community.

Lewis graduated from NPS in 1988 with a Master of Science in Computer Science, and we’re pleased to welcome him back to campus in this leadership role. Lewis is replacing the founding Chair of Acquisition, Rear Admiral, USN (Ret.) Jim Greene, who retired this June.

Most recently, Lewis served as Director of the Defense Contract Management Agency, managing over $7 trillion in defense contracts. In this role, he oversaw the agency’s efforts to ensure that supplies and services contracted for by the Department of Defense are delivered on time and in line with contract performance requirements.

During his career at sea, Lewis served as a communications officer, fire control and missile battery officer, and combat systems officer aboard destroyers and guided-missile cruisers.

Upon selection to flag rank in 2009, Lewis served as Vice Commander, Naval Sea Systems Command and then served four years as Program Executive Officer, Ships, where he directed the delivery of 18 ships and procurement of another 51 ships. From 2014-2017 he served as Commander, Space and Naval Warfare Systems Command where he led a global workforce of 10,300 civilian and military personnel who design, develop and deploy advanced communications and information capabilities.

Lewis’s extensive experience in shipbuilding has given him a unique understanding of the full acquisition lifecycle. He has delivered ships as a program manager and program executive officer, then later sustained and modernized them as a fleet engineer and systems commander. He will bring valuable perspective to NPS students and faculty, as well as the broader acquisition innovation community working to get superior capabilities into the hands of our warfighters.

Lewis’s expertise in product delivery will amplify ARP’s ability to execute its mission of delivering the real-time information and analytical capabilities needed by today’s acquisition professionals and policymakers. Adding VADM Lewis to the team also demonstrates NPS’s continued commitment to providing world-class defense-focused education and research...

KEYNOTE SPEAKER: VICE ADMIRAL JON HILL, USN, DIRECTOR, MISSILE DEFENSE AGENCY

Vice Admiral Jon Hill, USN is a native of Texas, born and raised on Fort Bliss. A Surface Warfare Officer, designated as an Engineering Duty Officer, he is a graduate of Saint Mary's University. He earned his Master of Science in Applied Physics and Ordnance Engineering from Naval Postgraduate School.

In June 2019, Hill became the 11th director of the Missile Defense Agency (MDA). In this capacity, he oversees the MDA's global mission to develop, deliver, and sustain layered capabilities to defend deployed forces, the United States, allies and friends against ballistic missile attacks in all phases of flight.

Admiral Hill's first Flag Officer tour was Program Executive Officer for Integrated Warfare Systems (PEO IWS). In this role, he was accountable for developing and certifying the deployment of all surface ship combat control systems, radars, missiles, launchers, electronic warfare, naval gunnery systems, and surface and subsurface anti-submarine warfare mission capabilities within the Fleet and joint force.

Hill previously served as the Deputy Director, Missile Defense Agency. Other leadership and acquisition engineering positions include AEGIS Shipbuilding (PMS 400), Naval Surface Warfare Center (NSWC) Dahlgren Division and Port Hueneme Division, PEO Theater Surface Combatants, and on the Assistant Secretary of the Navy staff for Research, Development and Acquisition (ASN RD&A).

He also served on the JOINT Staff (J-6), U.S. Army Staff for Missile Systems, and as a senior fellow on the Chief of Naval Operations Strategic Studies Group (CNO SSG XXVII). He served as Technical Director for AEGIS Ballistic Missile Defense then as AEGIS Combat Systems Major Program Manager (MPM) responsible for delivering Naval Integrated Fire Control and Counter Air (NIFC-CA) and Integrated Air and Missile Defense (IAMD) capabilities to forces afloat.

Personal awards include the Navy Distinguished Service Medal, Defense Superior Service Medal (two awards), the Legion of Merit (two awards), the Defense Meritorious Service Medal (two awards), the Meritorious Service Medal (three awards), the Joint Service Commendation Medal, the U.S. Army Commendation Medal, the Navy & Marine Corps Commendation Medal (two awards) and the Navy Achievement Medal (two awards).

PANEL 17. DOD ACQUISITION RESEARCH

Thursday, May 13, 2021	
7:45 a.m. – 9:00 am.	**Chair: David H. Lewis, Vice Admiral, U.S. Navy (Ret),** Acquisition Chair, Acquisition Research Program at Naval Postgraduate School **Panelists:** **Dr. Arun Seraphin,** Professional Staff Member, United States Senate Committee on Armed Services **Dr. Dinesh Verma,** Professor; Executive Director, Systems Engineering Research Center (SERC), Stevens Institute of Technology **Dr. Keith F. Snider,** Dean and Professor of Public Administration and Management, Graduate School of Defense Management at Naval Postgraduate School

David H. Lewis, Vice Admiral, U.S. Navy (Ret)—took the helm as the Naval Postgraduate School Chair of Acquisition. As chair, he will lead the Acquisition Research Program (ARP) in the Graduate School of Defense Management and connect NPS with leaders and policymakers in the acquisition community.

Lewis graduated from NPS in 1988 with a Master of Science in Computer Science, and we're pleased to welcome him back to campus in this leadership role. Lewis is replacing the founding Chair of Acquisition, Rear Admiral, USN (Ret.) Jim Greene, who retired this June.

Most recently, Lewis served as Director of the Defense Contract Management Agency, managing over $7 trillion in defense contracts. In this role, he oversaw the agency's efforts to ensure that supplies and services contracted for by the Department of Defense are delivered on time and in line with contract performance requirements.

During his career at sea, Lewis served as a communications officer, fire control and missile battery officer, and combat systems officer aboard destroyers and guided-missile cruisers.

Upon selection to flag rank in 2009, Lewis served as Vice Commander, Naval Sea Systems Command and then served four years as Program Executive Officer, Ships, where he directed the delivery of 18 ships and procurement of another 51 ships. From 2014-2017 he served as Commander, Space and Naval Warfare Systems Command where he led a global workforce of 10,300 civilian and military personnel who design, develop and deploy advanced communications and information capabilities.

Lewis's extensive experience in shipbuilding has given him a unique understanding of the full acquisition lifecycle. He has delivered ships as a program manager and program executive officer, then later sustained and modernized them as a fleet engineer and systems commander. He will bring valuable perspective to NPS students and faculty, as well as the broader acquisition innovation community working to get superior capabilities into the hands of our warfighters.

Lewis's expertise in product delivery will amplify ARP's ability to execute its mission of delivering the real-time information and analytical capabilities needed by today's acquisition professionals and policymakers. Adding VADM Lewis to the team also demonstrates NPS's continued commitment to providing world-class defense-focused education and research

Dr. Arun Seraphin— Dr. Arun A. Seraphin is a Professional Staff Member on the staff of the United States Senate Committee on Armed Services. His areas of responsibility include acquisition policy, Pentagon management issues, Department of Defense's science and technology programs, information technology systems, technology transition issues, defense laboratories, Small Business Innovation Research program, manufacturing programs, and test and evaluation programs. As such he assists Senators in their oversight of DOD policies and programs, including in the authorization of budgets, civilian nominations, legislative initiatives, and hearings. He rejoined the committee staff in 2014, after previously serving there between 2001 and 2010. In 2009, he was named one of ten Defense "Staffers to Know" by Roll Call, a Capitol Hill newspaper.

Dr. Dinesh Verma—served as the Founding Dean of the School of Systems and Enterprises at Stevens Institute of Technology from 2007 through 2016. He currently serves as the Executive Director of the Systems Engineering Research Center (SERC), a US Department of Defense sponsored University Affiliated Research Center (UARC) focused on systems engineering research. During his fifteen years at Stevens he has successfully proposed research and academic programs exceeding $150m in value. He has a curtsey appointment as a Visiting Professor in the Department of Biochemistry in the School of Medicine at Georgetown University. Verma served as Scientific Advisor to the Director of the Embedded Systems Institute in Eindhoven, Holland from 2003 through 2008. Prior to this role, he served as Technical Director at Lockheed Martin Undersea Systems, in Manassas, Virginia, in the area of adapted systems and supportability engineering processes, methods and tools for complex system development.

Dr. Keith F. Snider—is Dean and Professor of Public Administration and Management in the Graduate School of Public Administration and Policy at the Naval Postgraduate School, Monterey, CA. He graduated from the United States Military Academy at West Point and served on active duty for twenty years, the last several as a member of the Army Acquisition Corps. He received his PhD in Public Administration and Public Affairs from Virginia Polytechnic Institute and State University. His teaching and research interests lie in the areas of defense acquisition policy, defense project management, public organizations, and public administration theory and history.

PANEL 18. THE DIGITAL TRANSFORMATION OF ENGINEERING

Thursday, May 13, 2021	
9:15 a.m. – 10:00 a.m.	**Chair: Rear Admiral Jason Lloyd, USN,** Deputy Commander for Ship Design, Integration and Engineering ***A Framework to Categorize the Benefits and Value of Digital Engineering*** Tom McDermott, Stevens Institute of Technology ***Competencies, Education, and Training for Digital Engineering in the DoD*** Nicole Hutchison, Stevens Institute of Technology ***A Study of MBSE Through the Development of Modeling and Data Exchange Processes*** William Emeny, NSWC PHD Dustin Talley, NSWC PHD Michael Rubow, NSWC PHD Ryan Robar, NSWC PHD Lynn Nguyen, NSWC PHD Lance Lowenberg, NIWC Pacific

Rear Admiral Jason Lloyd, USN—is a native of Maryville, Tennessee and a graduate of Florida State University with a Bachelor of Science in Mechanical Engineering. He earned his commission through the Naval Reserve Officers Training Corps program and also holds a Master of Science in Mechanical Engineering from the Naval Postgraduate School.

His shipboard tours include USS Bainbridge (CGN 25) and USS Nimitz (CVN 68). From May 2013 until June 2016, he served as the first Reactor Officer on USS Gerald R. Ford (CVN 78) in Newport News, Virginia. During these assignments, Lloyd conducted deployments to the Mediterranean Sea and North Atlantic Ocean as well as executed a Refueling Complex Overhaul and a New Construction testing program.

Ashore, Lloyd served as a Deputy Project Superintendent at Norfolk Naval Shipyard, Maintenance Coordinator at Commander Naval Air Forces, Principal Assistant Program Manager at PEO Carriers, Program Manager Representative for Refueling Complex Overhaul and Program Manager Representative for New Construction Carriers at the Supervisor of Shipbuilding, Newport News. Following his Reactor Officer tour, Lloyd served as the Executive Assistant to Commander, Naval Sea Systems Command. From June, 2017 until March, 2020 Lloyd served as the Commanding Officer of the Supervisor of Shipbuilding, Newport News, Virginia. In May 2020, Lloyd assumed duties as Chief Engineer and Deputy Commander for Ship Design, Integration and Engineering, Naval Sea Systems Command.

As the Navy's chief engineer and NAVSEA deputy commander for Ship Design, Integration and Naval Engineering, Lloyd leads the engineering and scientific expertise, knowledge and technical authority necessary to design, build, maintain, repair, modernize, certify and dispose of the Navy's ships, aircraft carriers, submarines and associated combat and weapons systems. He is authorized to wear the Legion of Merit (three awards), Meritorious Service Medal (four awards), the Navy and Marine Corps Commendation Medal (four awards), Navy Achievement Medal (two awards) in addition to various unit awards.

A Framework to Categorize the Benefits and Value of Digital Engineering

Tom McDermott—Stevens Institute of Technology [tmcdermo@stevens.edu]

Additional authors: Alejandro Salado, Eileen Van Aken, Kaitlin Henderson: Virginia Tech

Abstract

The Department of Defense (DoD) envisions that digital engineering information exchange, system modeling, and data driven system engineering processes will become core to product and process development. As this transformation occurs, it will change the way Systems Engineering (SE) is measured and valued. Over the past 3 years, the Systems Engineering Research Center (SERC) has studied the Digital Engineering (DE) transformation processes and progress. This work has focused on DoD acquisition and program office activities but is applicable to all enterprises undergoing DE and Model-Based Systems Engineering (MBSE) transformations. A previous SERC research task created an Enterprise System-of-Systems Model for DE-enabled acquisition, conceptually modeling the potential future DoD acquisition enterprise. This research helped to understand the structure of future DoD/contractor program enterprises when the five goals of the DoD DE strategy were achieved, and the expected outcomes of that transition. That research cited the need for the community to standardize and implement measures that reflect success at the enterprise level. A second research task was completed to define metrics that represent value, benefits, and change progress in enterprise DE transformation. A third task is currently underway to design and implement measures that quantify DE benefits.

Introduction

DE is defined as "an integrated digital approach that uses authoritative sources of systems' data and models as a continuum across disciplines to support life cycle activities from concept through disposal" (DAU, 2020). A DE ecosystem is an interconnected infrastructure, environment, and methodology that enables the exchange of digital artifacts from an authoritative source of truth. MBSE is a subset of DE, defined as "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" (OMG, 2020).

MBSE has been a popular topic in the SE community for over a decade, but the level of movement toward broad implementation has not always been clear. With the release of the DoD DE strategy, a clear set of high-level goals are defined for the DoD acquisition community and its industry base.

Digital transformation is a change process heavily rooted in "how we train and shape the workforce to use those processes," as noted by Goal 5 of the DoD Digital Engineering Strategy (DoD, 2018). Each of the DoD's goals implies that an enterprise, organizational unit, or multi-organizational program has a means to define the outcomes of a DE strategy, performance metrics, measurement approaches, and leading indicators of change in the transformation process.

This research sought to define a comprehensive framework for DE benefits and expected value linked to the ongoing development of DE enterprise capabilities and experienced transformation "pain points," enablers, obstacles, and change strategies. Using a combination of literature review, broad surveys, and government program office visits, we found that the DE and MBSE communities, across government, industry, and academia, are not sufficiently mature at this point in their DE transformations to standardize on best practices and formal success metrics. Pockets of excellence exist, but experience and maturity vary widely.

We also found that government lags industry in maturity and should look to both their industry partners and the broader swath of commercial industry for best practices. The differing levels of DE capability across a government acquisition enterprise, prime contractors, and support contractors will be an obstacle to successful DE transformation. Programs, particularly legacy programs that have established non-digital processes, must invest effort in program-wide development and maturation of DE.

In addition, MBSE and an Authoritative Source of Truth (ASOT), as the core DE strategies for managing the complexity of large complex systems and systems-of-systems (SoS), lag in maturity to other DE strategies, such as Agile software development, product line engineering/product life-cycle management (PLM/PLE), and integrated supply chain management (ICSM). Pilot efforts that integrate MBSE and the ASOT across other more established disciplinary DE areas are necessary, but they should be executed broadly across all of these areas (many current pilots focus only on selected disciplinary areas or life cycle stages). Lessons learned from these efforts should inform best practices and success metrics for the full DE transformation.

In this research to date, we have only been able to document two instances where actual measurement approaches for DE processes had been developed and used (McDermott et al., 2020a). Based on this research, we were able to create a framework that categorizes DE benefits and adoption metrics. Efforts are now underway to pilot the most frequently cited DE benefits and build measurement models for them. This guidance is still being sought after by government agencies. The long-term goal of this research is to advance the practice of DE and MBSE through definition of enterprise value.

Research Results

A DE transformation process needs to assess both adoption of the methods and tools into the workforce in terms of number of users, resources, etc., and also the drivers of adoption that are linked to user experience with the methods and tools. To understand productivity indicators and areas of new value, the previous SERC study, “Enterprise System-of-Systems Model for Digital Thread Enabled Acquisition,” was used as the base digital enterprise transformation model (SERC, 2018). This study linked digital enterprise transformation to outcomes related to improved quality, improved velocity/agility, and better knowledge transfer. Knowledge transfer is a unique value of DE/MBSE that can be distinguished from other digital enterprise transformation metrics. A primary goal of MBSE and the associated data collected in an Authoritative Source of Truth is communication, sharing, and management of data, information, and knowledge.

Based upon this background research, we created a general categorization of DE/MBSE organizational change metrics linked to quality, velocity/agility, user experience, knowledge transfer, and adoption. Using literature reviews and a broad survey of DE/MBSE benefits, obstacles, and enablers, as well as government and industry discussions, the research produced an initial “top 10” list of metrics. A key result of the research is the development and definition of two frameworks that categorize DE benefits and adoption strategies that can be universally applied to a formal enterprise change strategy and associated performance measurement activities. The first framework is linked to the benefits of DE and categorizes 48 benefit areas linked to four digital transformation outcome areas: quality, velocity/agility, user experience, and knowledge transfer. The second framework addresses enterprise adoption of DE and provides a categorization of 37 success factors linked to organizational management subsystems encompassing leadership, communication, strategy and vision, resources, workforce, change strategy and processes, customers, measurement and data, workforce, organization DE processes relate to DE, and the organizational and external environments. The study conducted background research on literature discussing the benefits and values of DE/MBSE, a benchmark survey to assess the current state of maturity across enterprises currently implementing DE/MBSE, and interviews and discussions with government and industry.

The study found the systems engineering community perceives significant benefit from DE and MBSE transformation, but specific benefits have not yet been translated to organizational value drivers and success measures. Organizations appear to be searching for guidance on measuring the value and benefits of DE/MBSE usage. The study documented 10 top-cited metrics categories from literature and survey data. Seven of these were classed as benefits: increased traceability of requirements, design, and testing; reduced errors/defects in program phases; reduced activity times in development processes; improved consistency from phase to phase and project to project; increased capacity for reuse of data and models; higher support for automation; and better communication and information sharing. Three were classed as adoption measures: maturity of DE/MBSE methods and processes; training; and people willing to use DE/MBSE tools. Current efforts are underway to build causal models and data collection and analysis approaches to address the seven benefit measures.

Enterprise Metrics Categorization

Digital engineering is a subset of the larger aspects of enterprise digital transformation. Gartner (2019) reported four common characteristics for good enterprise-level digital transformation metrics: *adoption*, *usability*, *productivity*, and *new value*. This research developed five metrics areas relevant to DE: **adoption**, **user experience** (*usability*), **velocity/agility** (*productivity*), **quality** and **knowledge transfer** (*both new value*). These are shown in Figure 1.

A DE transformation process needs to assess both **adoption** of the methods and tools into the workforce in terms of number of users, resources, etc., and also the drivers of adoption that are linked to **user experience** with the methods and tools. To understand *productivity* indicators and areas of *new value*, the previous SERC study, "Enterprise System-of-Systems Model for Digital Thread Enabled Acquisition," was used as the base digital enterprise transformation model (SERC, 2018). This study linked digital enterprise transformation to outcomes related to improved **quality**, improved **velocity/agility**, and better **knowledge transfer**. Knowledge transfer is a unique value of DE/MBSE that can be distinguished from other digital enterprise transformation metrics. A primary goal of MBSE and the associated data collected in an Authoritative Source of Truth (ASOT) is communication, sharing, and management of data, information, and knowledge. Based upon this background research, we created a general categorization of DE/MBSE organizational change metrics linked to quality, velocity/agility, user experience, knowledge transfer, and adoption. Using literature reviews and a broad survey of DE/MBSE benefits, obstacles, and enablers, as well as government and industry discussions, the research produced an initial "top 10" list of metrics described in Table 1.

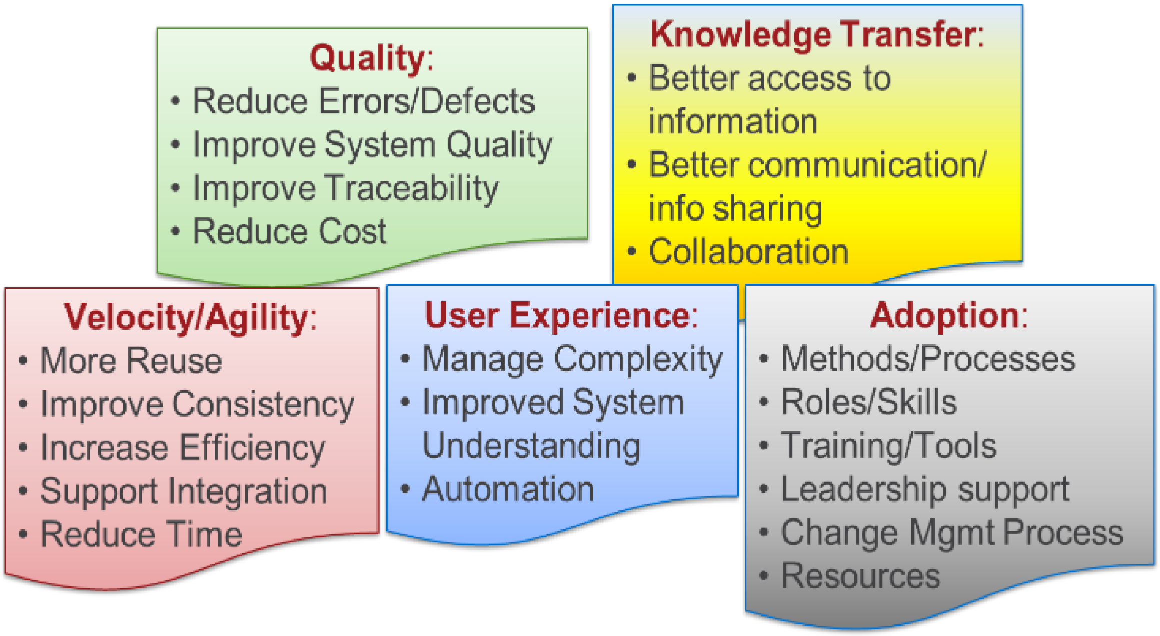


Figure 1. Top-Level Metrics Framework

It is important to note that measurement of DE/MBSE is a complex process that must be integrated with the entirety of enterprise measurement strategies across all enterprise functions. DE/MBSE cannot be isolated to a small group or limited set of programs if the goal is to understand and track enterprise value. Generally pilot efforts are recommended to start the adoption process, but maturity in DE/MBSE must become enterprise strategy and a component of enterprise performance measurement. This list is a starting point; a full list of 55 metrics categories derived from the research is provided later in Table 2.

Table 1. Top 10 Collected Enterprise Metric Definitions (McDermott et al., 2020a)

Metric Area	Metrics Category	Inputs	Ex. Processes	Ex. Outputs	Outcomes
Quality	Increased traceability	User needs and system requirements are in a modeling tool and linked to truth data & models	• MBSE: reqs., structure, use cases, traceability tools • ASOT: all reqs. at each level are linked with data	• Decreasing number of reqs. changes • Improving requirement volatility trends	• Fully digital traceability of reqs., design, test, and information • Available from one source of truth
Quality	Reduced defects/errors	Data, models, reqs., design artifacts	• Peer review and technical review in models • Design automation • Test automation	• Defects/errors discovered and corrected earlier in development phases • Less total defects/errors • Error-free deployments	• Reduced total errors/defects in each program phase • Reduced errors/defects that escape from one phase to the next • Increased number of saves in each phase
Velocity/ Agility	Reduced time	Historical estimated effort, planned effort, resourced schedules, milestone schedules	• Estimation processes: COCOMO, COSYSMO, etc. • Schedule tracking or EVMS	Program schedule durations trending toward reduced total or activity times	Time reduction trend data: • total project schedule • average across projects • total and average per activity • response time to need • delays from plan

Metric Area	Metrics Category	Inputs	Ex. Processes	Ex. Outputs	Outcomes
	Improved consistency	Planning schedules and resource loading, prioritization of needs, development and delivery processes, and stable resources	More regular and frequent development and implementation planning periods	• More predictable scope and cycle time for capability releases • More consistent content and schedule for production deployments	• Processes produce consistent results from project to project • Data or models have consistent use from project to project • Practitioners apply consistent work processes and instructions
	Increased capacity for reuse	Standards, data, models, search tools, CM tools, certifications, data/model managers	• Data and functional modeling • Patterns • Standards • CM • Compliance testing	• Pay once for data = reuse everywhere • Standard reusable capabilities or sub-functions • Compliance	•Models/dataset s reused project to project • Percent direct use/modification /change • Related cost/schedule estimation and actuals
User Experienc e	Higher level support for automation	Investment resources for automation, data collection, and automation tools	Automated: • document generation • test • data search, etc.	• New processes • Reduced labor hours • Reduced time	• Automated v. manual activities • Investment in automation • Automation strategy
Knowledg e Transfer	Better communicatio n/ info sharing	Investment resources for collaboration and communication tools, IT infrastructure, and data and libraries	• Teams interact around shared data • Participation in model-based reviews • Data/model desktop availability	• Number of employees and disciplines communicating and sharing information • Number of events held in the toolsets	• Processes and tools to share and jointly assess information • Opportunities to share knowledge and learn in process around common tools and representations

Metric Area	Metrics Category	Inputs	Ex. Processes	Ex. Outputs	Outcomes
Adoption	DE/MBSE methods and processes	Enterprise strategy and investment, experience with DE/MBSE	• Periodic assessment via survey and scoring	• Attainment of "level 4" capabilities	Availability and maturity of MBSE capabilities (refer to the INCOSE MBSE Capabilities Matrix (INCOSE, 2020) for a full assessment)
	Training	Curricula, classes, mentoring, assessment	• Training • Learning management	• Availability of training • Investment in training • Number trained • Effectiveness of training	Appropriately trained and experienced workforce and customer
	Increased willingness to use DE/MBSE tools	Vision/mission, leadership support, incentives, tools, methods/processes, training	• Change management strategy	Number of: • people actively using the tools • tool experts • people actively working with tool artifacts	Models and tools produce communication media to all general users in an accessible form

Descriptive Summary of Top-Cited Metrics Areas

Table 2 provides a full descriptive summary of 55 candidate metrics derived from the benefit and adoption categories. These are grouped into the five metrics areas of Table 1 and ranked by number of literature or survey citations in each area. The table includes example descriptive phrases of each metrics category developed in textual analysis of the literature and survey data. The table also lists examples of potential outcome metrics for each metrics category.

Table 2. Descriptive Summary of Top-Cited Metrics Areas (McDermott et al., 2020a)

Metrics Category	**Example descriptive phrases**	**Example outcome metrics**
Metric Area: Quality		
Increased traceability	requirements, design, information traceability	• Full digital traceability of requirements, design, test, and information • Availability from one source of truth
Reduce cost	cost effective, cost savings, save money, optimize cost	• Lower total cost compared to similar previous work
Improve system quality	higher quality, quality of design, increased system quality, first time quality, improve SE quality, improve specification quality	• Improved: total quality (roll-up of quality measures); first time quality (deployment success)
Reduce risk	reduce development risk, reduce project risk, lower risk, reduce technology risk, reduced programmatic risk, mitigate risk, reduce design risk, reduce schedule risk, reduce risk in early design decisions	• Risks identified and risk mitigations executed via DE enterprise processes • New risks uncovered by system modeling
Reduce defects/ errors	reduce error rate, earlier error detection, reduction of failure corrections, limit human errors, early detection of issues, detect defects earlier, early detection of errors and omissions, reduced specification defects, reduce defects, remove human sources of errors, reduce requirements defects	• Reduced: total errors/defects in each program phase; errors/defects that escape from one phase to the next • Increased number of saves in each phase
Improved system design	improved design completeness, design process, design integrity, design accuracy, streamline design process, system design maturity, design performance, better design outcomes, clarity of design	• Design outcomes show improvement and the design process is more effective compared to similar programs (rollup measure)
Better requirements generation	requirements definition, streamlining process of requirements generation, requirements elicitation, well-defined set of requirements, multiple methods for requirements characterization, more explicit requirements, improved requirements	• Measurement of requirements quality factors in the DE process: correctness, completeness, clarity, non-ambiguity, testability, etc.

Metrics Category	Example descriptive phrases	Example outcome metrics
Improved deliverable quality	improve product quality, better engineering products	• Reduced deliverable defects • Improved deliverables acceptance rate
Increased effectiveness	effectively perform SE work, improved representation effectiveness, increased effectiveness of model, more effective processes	• Effectiveness of a process is how relevant the output is to the desired objective
Improved risk analysis	earlier/ improved risk identification, identify risk	• Risks identified by phase
Better analysis capability	better analysis of system, tradespace analytics, perform trade-offs and comparisons between alternative designs, simulation	• Decisions balance cost, schedule, risk, performance, & capabilities • Improved affordability, efficiency & effectiveness of tradespace processes
Strengthened testing	model based test and evaluation, increased testability, improved developmental testing	• Improved: test coverage; automated tests; number of errors found by automation versus manual means; efficiency & effectiveness of test process • Reduced number of defects/errors in each phase
Increased rigor/ Improved predictive ability	rigorous model, rigorous formalisms, more rigorous data, better predict behavior of system, predict dynamic behavior, predictive analytics	• Increased: level of difficulty/complexity of project; number of alternatives analyzed; subject matter experts involved • Improved: exhaustiveness of data collection; consistency of analysis processes; predictive links between design & capabilities

Metrics Category	Example descriptive phrases	Example outcome metrics
More stakeholder involvement	easy way to present view of system to stakeholders, better engage stakeholders, quick answers to stakeholder's questions, share knowledge of system with stakeholders, stakeholder engagement, satisfy stakeholder needs	• Improved: process efficiency & effectiveness for stakeholder involvement in modeling; number of stakeholders contributing; stakeholder access to tools, models, data
Metric Area: Velocity/Agility		
Improved consistency	consistency of info, consistency of model, mitigate inconsistencies, consistent documentation, project activities consistent, data consistency, consistent between system artifacts	• Processes produce consistency from project to project in: results; data; models used; work processes & instructions applied by practitioners
Reduce time	shorter design cycles, time savings, faster time to market, ability to meet schedule, reduce development time, time to search for info reduced, reduce product cycle time, delays reduced	• Time reduction trend data: total project schedule; average across projects; total & average per activity; response time to need; delays from plan
Increased capacity for reuse	reusability of models, reuse of info/designs	• Models/datasets reused project to project • percent direct use/modification/change; • related cost/schedule estimation & actuals
Increased efficiency	efficient system development, higher design efficiency, more efficient product development process	• More efficient process time, resources per unit output, flow • Reduced waste
Increased productivity	gains in productivity	• Effort per unit of production
Reduce rework	reduce rework	• Reduced: number of rework cycles; percent rework; errors causing rework; size of rework effort; technical debt
Early V&V	early verification and/or validation	• Formal testing: credited in earlier phases; done in models and simulation vs. system

Metrics Category	Example descriptive phrases	Example outcome metrics
Reduce ambiguity	less ambiguous system representation, clarity, streamline content, unambiguous	• Higher levels of specificity; decisions based on data; application of uncertainty quantification methods
Increased uniformity	uniformity	• Application of standards: technical, process, work & effort, etc.
Easy to make changes	easier to make design changes, increased agility in making changes, changes automatically across all items, increased changeability	• Improved ability to: implement changes; change management process automation
Reduce waste	reduce waste, save resources	• Lean processes: waste removal and flow (pull)
Better requirements management	better meet requirements, provide insight into requirements, requirements explicitly associated with components, coordinate changes to requirements	• Process effectiveness demonstrated by how relevant output is to desired objective: # requirements, requirements volatility, requirements satisfaction, etc.
Higher level of support for integration	integration of information, providing a foundation to integrate diverse models, system design integration, support for virtual enterprise/supply chain integration, integration as you go	• Developmental testing credited in earlier phases; testing done in models and simulation vs. system; reuse of data & models in integration activities
Increased precision	design precision, more precise data, correctness, mitigate redundancies, accuracy	• Six Sigma processes • Reduced standard deviation
Increased flexibility	flexibility in design changes, increase flexibility in which design architectures are considered	• Time- and cost-effective incorporation of: new requirements; sensitivity analysis to change vs. a reference
Metric Area: User Experience		

Metrics Category	Example descriptive phrases	Example outcome metrics
Improved system understanding	reduce misunderstanding, common understanding of system, increased understanding between stakeholders, understanding of domain/behavior/system design/requirements, early model understanding, increased readability, better insight of the problem, coherent	• Assessments from activities such as technical reviews and change processes, standard models or patterns of SE and domain, common understanding of architecture/abstractions (architectural quality/risk assessment), etc.
Better manage complexity	simplify/reduce complexity, understand/specify complex systems, manage complex information/ design	• Improved: data/model integration & management; distribute control; empowerment across data/between disciplines; ability to iterate/experiment
Higher level support for automation	automation of design process, automatic generation of system documents, automated model configuration management	• Increased: automated vs. manual activities; investment in automation; automation strategy
Better data management/ capture	representation of data, enhanced ability to capture system design data, manage data	• Improved data management architecture, automation • Reduced technical debt
Better decision making	make early decisions, enables effective decision making, make better informed decisions	• Visualizing different levels of specificity; more decisions based on data and analysis, access to and visualization of data
Reduce burden of SE tasks	reduce complexity of engineering process	• Reduce time spent on or waiting for SE artifacts
Reduce effort	reduce cognitive load, reduction in engineering effort, reduce formal analysis effort, streamline effort of system architecture, reduce work effort, reduce amount of human input in test scoping	• Process efficiency demonstrated by relevancy of output to desired objective: effort per unit of production; total effort vs. similar programs; effort vs. plan
Metric Area: Knowledge Transfer		

Metrics Category	**Example descriptive phrases**	**Example outcome metrics**
Better communication/info sharing	communication with stakeholders/team/designers/developers/different engineering disciplines, information sharing, knowledge sharing, exchange of information, knowledge transfer	• Improved: processes and tools to share and jointly assess information; opportunities to share knowledge and learn in process around common tools & representations
Better accessibility of info	Ease of info availability, single source of truth, centralized/unique/single source of info, simpler access to info, synthesize info, unified coherent model, one complete model	• Develop: tools that support access to and viewing of data/models; widely shared models; executable models
Improved collaboration	simplify collaboration within team	• Develop: tools that support human collaboration around shared data & models
Better knowledge management/capture	knowledge capture of process, better information capture, early knowledge capture, more effective knowledge management	• Develop: tools that support wide diversity of information; integration across domains; methods to build and enter knowledge
Improved architecture/Multiple viewpoints of model	help develop unambiguous architecture, rapidly define system architecture, faster architecture maturity, accurate architecture design; shared view of system, more holistic representation of system/models, dynamically generated system views	• Develop tools that support intuitive structuring of model views, story-telling, interface management
Metric Area: Adoption (Ranked separately from the other four metrics areas)		
Leadership support/Commitment	Demonstrating commitment and general support for MBSE implementation by senior leaders through communication, actions, and priorities	• Demonstrate messaging, awareness of DE/MBSE • Participation in reviews, performance management incentives, succession planning
Workforce knowledge/skills	Developing a workforce with the knowledge, skills, and competencies needed to support MBSE adoption	• Availability and maturity of MBSE competencies (refer to the INCOSE MBSE Capabilities Matrix in the complete report for a full assessment)

Metrics Category	Example descriptive phrases	Example outcome metrics
DE/MBSE methods and processes	Developing and deploying consistent, systematic, and documented processes for MBSE throughout the relevant parts of the organization, including steps/phases, outputs, and roles/responsibilities	• Availability and maturity of MBSE capabilities (refer to the INCOSE MBSE Capabilities Matrix in the complete report for a full assessment)
Training	Investing in and providing the education/training required to develop the workforce knowledge/skills needed to support MBSE implementation	• Appropriately trained & experienced workforce, and customer
DE/MBSE Tools	Ensuring MBSE tools have sufficient quality, have sufficient maturity, are available, and are common	• Tools: availability, investment in, experience with, and stability
Demonstrating benefits/results	Creating "quick wins" to demonstrate results (benefits and outcomes) from applying MBSE	• Develop DE/MBSE growth strategy, pilot efforts, publications, lessons learned
Change management process design	Defining and implementing a systematic change approach to implement MBSE, with clear actions, timeline, roles, resources needed, staged deployment steps/phases for experimentation (where relevant), and outcomes expected	• Revised and relevant vision, mission, change strategy, engagement plan, feedback plan, etc.
General resources for DE/MBSE implementation	Ensuring financial and other resources are available to support MBSE implementation	• Funding, IT support, training support, Internal R&D, etc.
People willing to use DE/MBSE tools	Willingness and motivation of people in SE roles across organization to use MBSE tools	• Communicate models and modeling tools output to all of the general users in an accessible form
Alignment with customer requirements	Identifying how MBSE adoption supports meeting customer needs and requirements	• Implement: customer engagement plan; customer requirements elicitation; involvement of customer; participation with customer
MBSE terminology/ontology/libraries	Clearly identifying a common terminology, ontology, and libraries to support MBSE adoption	• Investment in enterprise data development and management, shared libraries, stability of data definition and stores

Metrics Category	Example descriptive phrases	Example outcome metrics
Champions	Defining and creating the role of champion to use expertise to advocate for and encourage others' use of MBSE	• Create evangelist role, and enlist number of evangelists • Demonstrated leadership support
People in SE roles	Quality of and support from people holding SE roles across the organization	• Defined SE role • Develop plan integrating SE and DE, scope of SE teams/organization, etc.
Communities of Practice	Creating a community of practice within the organization to provide guidance, expertise, and other resources as MBSE is deployed	• Investment in CoP • Established number of participants

Figure 2 provides a full summary of the top DE benefit areas from the literature review and survey conducted in the research on DE benefits. The figure depicts the percentage of literature review papers or survey respondents citing each benefit area. This was used to define the top metric categories related to benefits of DE. Figure 3 provides a summary of the top enablers, obstacles, and areas of change based on survey data. This was used to derive the top metrics categories related to DE adoption.

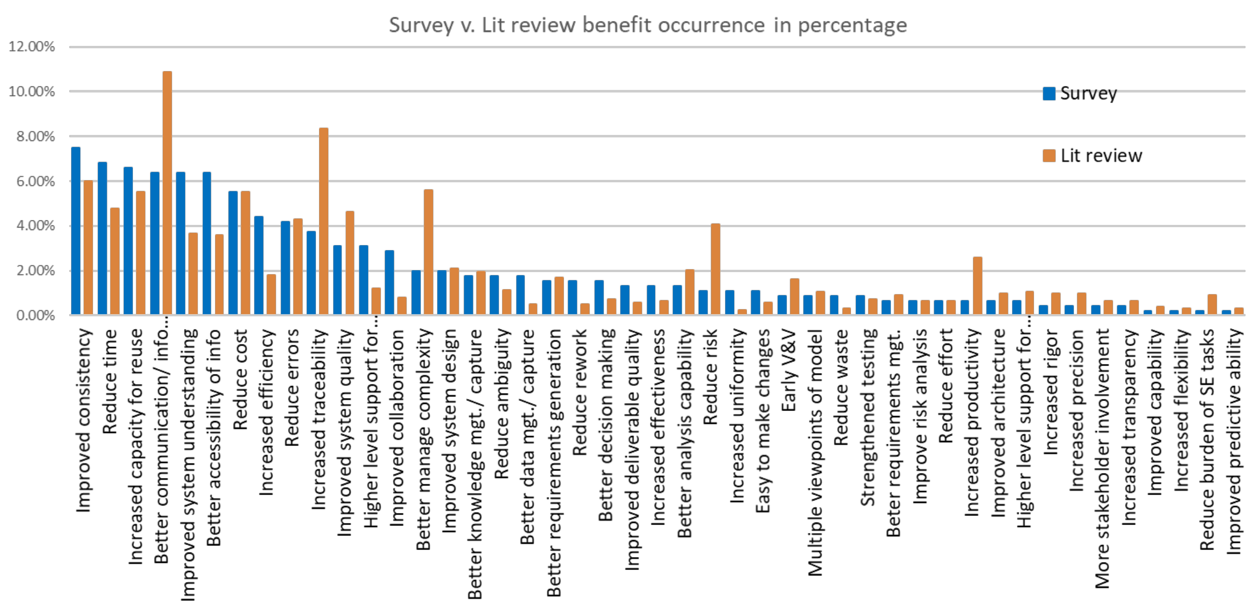


Figure 2. Top Cited DE Benefits Areas from Literature and Survey Results (McDermott et al., 2020b)

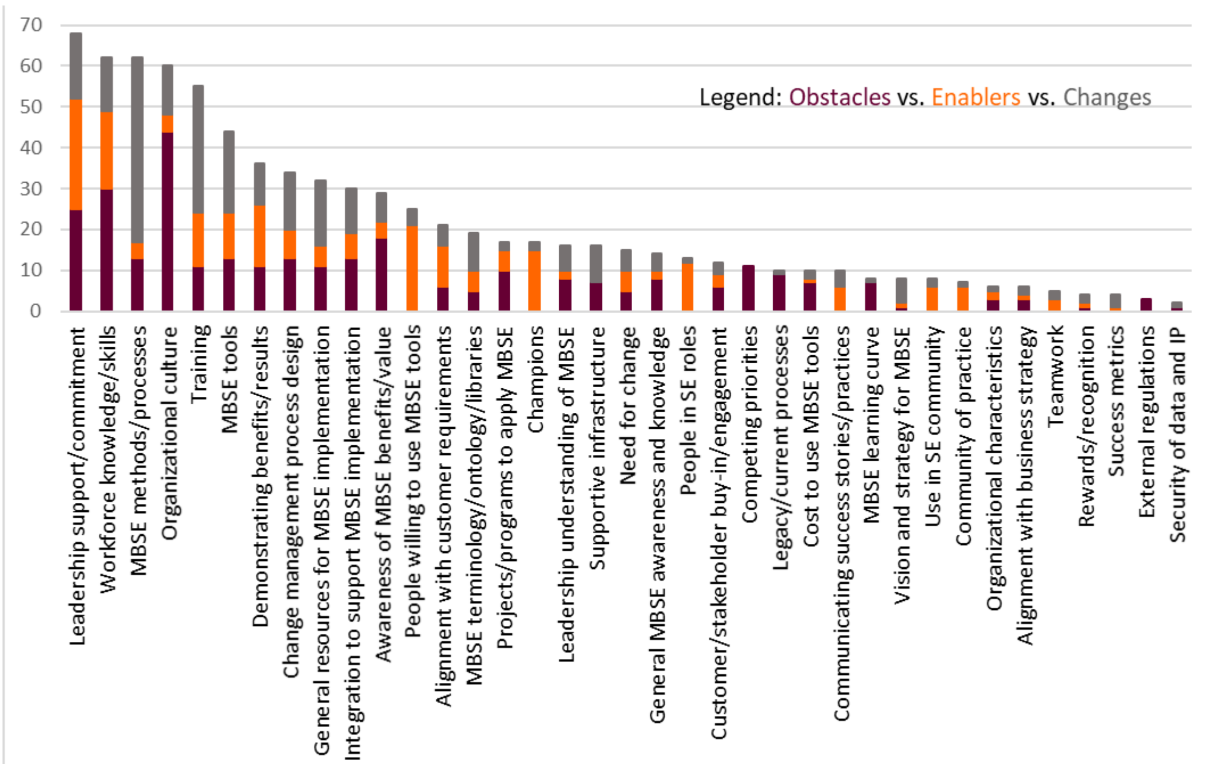


Figure 3. Obstacles, Enablers, and Changes for DE Adoption, ranked by Frequency of Mention (McDermott et al., 2020b)

Findings

This research task used the following four guiding questions:

1. What would a “Program Office Guide to Successful DE Transition” look like?
2. How can the value and effectiveness of DE be described and measured?
3. Are there game-changing methods and/or technologies that would make a difference?
4. Can an organizational performance model for DE transformation be described?

At the start of the research effort, the hope was to identify and document best practices across the DoD, defense industry, and other industries related to measurement of the DE enterprise transformation, metrics for success, and standard success guidance. It quickly became clear that best practices do not yet exist in the DE and MBSE community, and the transformation process is not yet mature enough across the community to standardize best practices and success metrics. Given the state of the practice, the research shifted to a set of efforts to define a comprehensive framework for DE benefits and expected value linked to the ongoing development of DE enterprise capabilities and experienced transformation “pain points,” enablers, obstacles, and change strategies.

A key result of this research is the development and definition of two frameworks that categorize DE benefits and adoption strategies that can be universally applied to a formal enterprise change strategy and associated performance measurement activities. The first framework is linked to the benefits of DE and categorizes 48 benefit areas linked to four digital transformation outcome areas: quality, velocity/agility, user experience, and knowledge transfer. This framework identifies a number of candidate success metrics. A test application to an ongoing DoD pilot project was completed and is documented in this report. The second framework addresses enterprise adoption of DE and provides a categorization of 37 success factors linked to organizational management subsystems encompassing leadership, communication, strategy and vision, resources, workforce, change strategy and processes, customers, measurement and data, workforce, organization DE processes relate to DE, and the organizational and external environments. The following summarizes the findings based on the four research questions:

What would a program office successful DE transition look like?

1) The DE and MBSE communities, across government, industry, and academia, are not sufficiently mature at this point in their DE transformations to standardize on best practices and formal success metrics. Pockets of excellence exist, but experience and maturity vary widely.

2) Government lags industry in maturity and should look to both their industry partners and the broader swath of commercial industry for best practices. The differing levels of DE capability across a government acquisition enterprise, prime contractors, and support contractors will be an obstacle to successful DE transformation. Programs, particularly legacy programs that have established non-digital processes, must invest effort in program-wide development and maturation of DE.

3) MBSE and the ASOT, as the core DE strategies for managing the complexity of large complex systems and systems-of-systems (SoS), lag in maturity to other DE strategies, such as Agile software development, product line engineering/product life-cycle management (PLM/PLE), and integrated supply chain management (ICSM). Pilot efforts that integrate MBSE and the ASOT across other more established disciplinary DE areas are necessary. Lessons learned from these efforts should inform best practices and success metrics for the full DE transformation.

4) Organizations should continue to share lessons learned from their pilot efforts.

5) The community should share their implementation and measurement strategies, and future surveys should assess maturity and best practices.

6) More effort is necessary to pilot draft guidance and to test and validate results. The next phase of this research is working with a government/industry/academia effort to standardize key practices and metrics.

How can the value and effectiveness of DE be described and measured?

7) The community perceives significant benefit from DE and MBSE transformation, but specific benefits have not yet been translated to organizational value drivers and success metrics. In fact, organizations appear to be searching for guidance on measuring the value and benefits of DE/MBSE usage. Based on extensive literature review and survey data, this research presents a guiding framework for benefits and metrics. Based on this work, the DoD should provide common guidance to program offices on data collection and should track several top-level measures that are consistently used across those offices. Table 1 of this report makes recommendations based on categories of metrics most frequently reported in literature and from survey data, but further work is needed to evaluate these metrics in practice—few examples exist today.

Are there game-changing methods and/or technologies that would make a difference?

8) Technology in the DE and MBSE ecosystem is evolving rapidly. Tools and infrastructure, based on survey data, are becoming more mature and less of an obstacle to DE success. However, enterprises must continue to focus on their unique DE innovation strategies to build successful infrastructure and practices, focus resources and people on the unique aspects of

the DE infrastructure as part of the DE transformation team (not general IT), and create programs to invest in and evaluate evolving technologies and standards.

9) The transformative aspect of DE/MBSE will succeed based on how technology enables automation of SE tasks and human collaboration across all disciplines across a full model-centric engineering process. The DoD should fund research and incentivize tool vendors to introduce more automation into the DE/MBSE processes.

Can an organizational performance model for DE transformation be described?

10) Successful DE and MBSE are inseparable from good systems engineering. DE/MBSE is just an extension of existing systems engineering roles and skills. DE presents newer roles related to the data science aspects of MBSE, particularly data management, data integration, and data analysis. Also, there is more emphasis on tool experts: roles focused exclusively on the use and maintenance of tools to support DE/MBSE. Workforce development is a critical component of DE/MBSE adoption, and this research provides an initial survey-based framework for DE roles and skills. The results of the MBSE Maturity Survey conducted with this effort capture this framework (McDermott et al., 2020b).

11) In a transformation program, one would start with a high-level description of program adoption practices linked to the benefits of DE/MBSE, then use these to design a set of organizational capabilities for doing DE/MBSE, measure the performance of the organization within each of these capabilities, and use this to produce results that enable new value to the organization. This starts with leadership and strategy; is implemented across enterprise operations and workforce capabilities; and should produce customer value and enterprise-wide results. This is the core of the Baldrige Criteria for Performance Excellence (NIST, 2019). Although this research was not able to produce a "cookbook" for program office success, it does provide a set of frameworks for a program office or enterprise to evolve that guide.

11) Finally, there appears to be a strong top-to-bottom leadership commitment to DE transformation at this point in time, but the perception of progress and success differs greatly between leadership and the workforce using the methods, processes, and tools. In terms of the Gartner Hype Cycle (Gartner, 2020), the community is just starting up the "Slope of Enlightenment" where benefits start to crystallize and become widely understood. A strong understanding of adoption obstacles and enablers must exist and be tracked at all enterprise levels.

Figure 4 suggests an overall program leadership and measurement model presented as a concept map.

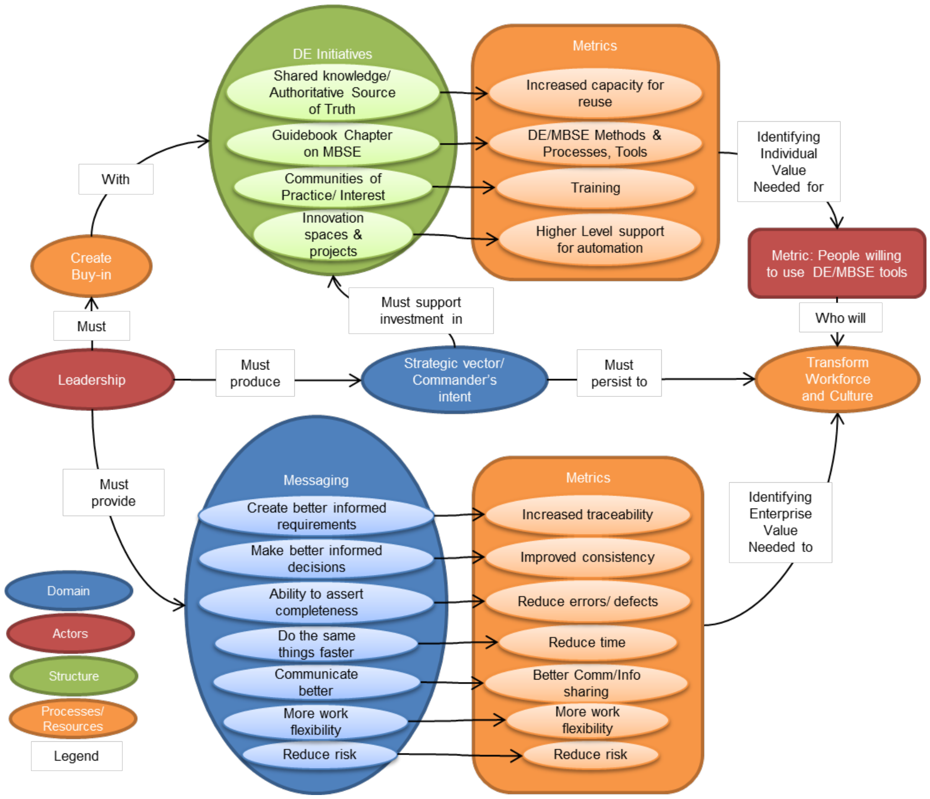


Figure 4. Top Level Organizational Performance Model (McDermott et al., 2020a)

Summary

A key result of this research is the development and definition of two frameworks: a DE benefits framework and an enterprise adoption framework, which can be universally applied to a formal enterprise change strategy and associated performance measurement activities. From these, we derived an additional metrics framework and captured, at this point, 10 primary categories of metrics around which to start a measurement program. The primary value of this research is in these comprehensive frameworks.

References

Defense Acquisition University. (2020). *DAU glossary*. https://www.dau.edu/glossary/Pages/GlossaryContent.aspx?itemid=27345#:~:text=An%20integrated%20digital%20approach%20that,activities%20from%20concept%20through%20disposal

DoD. (2018). *Digital engineering strategy*.

Gartner. (2019). *How to measure digital transformation progress*. https://www.gartner.com/smarterwithgartner/how-to-measure-digital-transformation-progress/

Gartner. (2020). *Gartner hype cycle*. https://www.gartner.com/en/research/methodologies/gartner-hype-cycle

INCOSE. (2020). *Model-based enterprise capability matrix and user's guide, version 1.0*.

McDermott, T., Van Aken, E., Hutchison, N., Salado, A., Henderson, K., & Clifford, M. (2020a). *Digital engineering metrics* (Technical Report SERC-2020-TR-002). Stevens Institute of Technology. https://sercuarc.org/wp-content/uploads/2020/06/SERC-TR-2020-002-DE-Metrics-6-8-2020.pdf

McDermott, T., Van Aken, E., Hutchison, N., Salado, A., Henderson, K., & Clifford, M. (2020b). *Benchmarking the benefits and current maturity of model-based systems engineering across the enterprise: Results of the MBSE maturity survey* (Technical Report SERC-2020-SR-001). Stevens Institute of Technology. https://sercuarc.org/wp-content/uploads/2020/03/SERC-SR-2020-001-Benchmarking-the-Benefits-and-Current-Maturity-of-MBSE-3-2020.pdf

NIST. (2019*). Baldrige excellence framework: Proven leadership and management practices for high performance*. Department of Commerce, National Institute of Standards and Technology. https://www.nist.gov/baldrige/publications/baldrige-excellence-framework

Object Management Group. (2020). *MBSE Wiki*. https://www.omgwiki.org/MBSE/doku.php

Systems Engineering Research Center. (2018). *Enterprise system-of-systems model for digital thread enabled acquisition* (Technical report SERC-2018-TR-109). Stevens Institute of Technology. https://sercuarc.org/publication/?id=197&pub-type=Technical-Report&publication=SERC-2018-TR-109-Enterprise+System-of-Systems+Model+for+Digital-Thread+Enabled+Acquisition

Digital Transformation for Defense Acquisition: Digital Engineering Competency Framework (DECF)

Nicole Hutchison—is a research engineer with the SERC and serves as the Digital Engineering Competency Framework Principal Investigator. Other research projects include Helix, the Mission Engineering Competency Model, Digital Engineering Metrics. She is the Managing Editor of the Systems Engineering Body of Knowledge (SEBoK). Before coming to Stevens, Hutchison worked for Analytic Services Inc. primarily focused on public health, biodefense, and full-scale disaster exercises.

Hoong Yan See Tao—is a researcher with the SERC. See Tao received her PhD in systems engineering from Stevens Institute of Technology in 2017. Previously, she was a postdoctoral fellow with the SERC (2017–2020). She was president of the Stevens INCOSE Student Division (2013–2014) and an instructor for the systems engineering course in a Pre-College Program at Stevens in 2016. She earned her BS and MS in electrical engineering from the University of Texas at El Paso (UTEP), where she was the team leader for two NASA and TSGC design challenges.

Kara Pepe—is Chief of Staff for the Systems Engineering Research Center (SERC) within the School of Systems and Enterprises at Stevens Institute of Technology. She received her BE in Engineering Management and ME in Systems Engineering from Stevens and is currently a PhD student. Her research focuses on digital engineering in general and population health in particular. She is a member of SWE, ASEM, and INCOSE. Pepe volunteers with Geeks Rule, an organization that promotes the study and engagement with STEM among underserved youth in New York City.

Mark Blackburn—is a Senior Research Scientist with Stevens Institute of Technology. Blackburn has been the Principal Investigator (PI) on 14 System Engineering Research Center (SERC) research tasks for U.S. Navy NAVAIR, U.S. Army ARDEC, and DoD on Systems Engineering Transformation through Model-Centric Engineering. He is a member of the SERC Research Council, and core member of the Semantic Technologies for Systems Engineering initiative. Prior to joining Stevens, Blackburn worked in industry for more than 25 years. Blackburn holds a PhD from George Mason University.

Clifford Whitcomb—is a Distinguished Professor of Systems Engineering at the Naval Postgraduate School (NPS). He is the co-author of *Effective Interpersonal and Team Communication Skills for Engineers* and has published many textbook chapters and journal articles. He has been a principal investigator for the U.S. Navy Office of Naval Research, Office of the Joint Staff, Office of the Secretary of the Navy, and the Veterans Health Administration. He is a Fellow of the Society of Naval Architects and Marine Engineers (SNAME) and the International Council on Systems Engineering (INCOSE). Whitcomb is a retired U.S. naval officer.

Rabia Khan—is a Faculty Research Associate at the Systems Engineering Department at NPS. Khan has extensive experience in RDT&E, including projects related to systems engineering, defense energy, modeling and simulation, and operations research. She earned her bachelor’s degree in Neurobiology, Physiology, and Behavior from the University of California, Davis, and a master’s degree in Engineering Systems from NPS. Most recently, Khan received a certificate in the field of Modeling and Simulation in Healthcare, from a joint Uniformed Services University of the Health Sciences and NPS program.

Russell Peak—is a Senior Researcher at Georgia Tech in the Aerospace Systems Design Lab (ASDL) where he is MBSE Branch Chief. After 6 years in industry, he joined the research faculty at Georgia Tech. Since 1996, he has been principal investigator on more than 40 projects and has authored more than 135 publications. He currently leads the Digital Ecosystems Challenge Team (DECO) in the INCOSE MBSE Initiative. Peak represents Georgia Tech on the OMG SysML task force, is a Content Developer for the OMG Certified Systems Modeling Professional (OCSMP) program, and holds the highest OCSMP certification, Model Builder–Advanced (MBA).

Adam Baker—is a PhD candidate at the Aerospace Systems Engineering Laboratory at the Georgia Institute of Technology. He formerly received his BS in Mechanical Engineering from Lehigh University. His research topics have spanned a variety of systems engineering topics including the application of Model-Based Systems Engineering to the acquisition process, Digital Thread Architecture, and the incorporation of structural analysis into the early stages of the design process.

Dinesh Verma—is the Executive Director of the Systems Engineering Research Center (SERC) and the Acquisition Innovation and Research Center (AIRC). He is a Professor at Stevens Institute of Technology and the Scientific Advisor to the Director of the Embedded Systems Institute in Eindhoven, Holland. Verma received his PhD (1994) and an MS (1991) in Industrial and Systems Engineering from Virginia Tech. He served as the Founding Dean of the School of Systems and Enterprises at Stevens from 2007 through 2016. During his 15 years at Stevens, he successfully proposed research and academic programs exceeding $150 million in value.

Abstract

Digital transformation is fundamentally changing the way acquisition and engineering are performed across a wide range of government agencies, industries, and academia and is characterized by the integration of digital technology into all areas of a business, changing fundamental operations and how results are delivered in terms of new value to customers. It includes cultural change centered on alignment across leadership, strategy, customers, operations, and workforce evolution. Digital engineering (DE) is ‘‘an integrated digital approach that uses authoritative sources of systems' data and models as a continuum across disciplines to support life-cycle activities from concept through disposal. A DE ecosystem is an interconnected infrastructure, environment, and methodology that enables the exchange of digital artifacts from an authoritative source of truth.” (ODASD [SE], 2017) The purpose of the Digital Engineering Competency Framework (DECF) is to provide clear guidance for the DoD acquisition workforce, in particular the engineering (ENG) acquisition workforce, through clearly defined competencies that illuminate the knowledge, skills, abilities, and behaviors (KSABs) required for DE professionals. Though the DECF includes considerations specific to the Defense acquisition workforce, data was also gathered from outside the defense community.

Introduction

Based on evidence across the Services and industry, digital engineering (DE) has been affirmed as a vital practice necessary to support acquisition in an environment of increasing global challenges, dynamic threats, rapidly evolving technologies, and increasing life expectancy of systems currently in operation and future systems. In order for the DoD to provide the best advantage for successful acquisitions and sustainment, the DoD must continue to practice systems engineering efficiently and effectively. DE updates the systems engineering practices to take full advantage of the digital power of computation, visualization, and communication to take better, faster actions throughout the life cycle.

Methodology

The Digital Engineering Competency Framework (DECF) was created in compliance with DoD Instruction 1400.25, volume 250, “DoD Civilian Personnel Management Systems: Civilian Strategic Human Capital Planning” (DoD, 2016). DoDI 1400.25, vol. 250 outlines five tiers of competencies:

- *Tier 1*. Core Competencies, which apply across DoD regardless of DoD component or occupation.
- *Tier 2*. Primary Occupational Competencies, which apply across discrete occupational series and or functions.
- *Tier 3*. Sub-occupational Specialty Competencies, which are unique to sub-occupational specialties.
- *Tier 4*. DoD Component-Unique Competencies, which are so unlike any of the other competencies identified that they exist at the component level and are unique to the context or environment in which the work is performed.

- *Tier 5.* Position-Specific Competencies, which are required for a particular position within an occupation and are not addressed in tiers above. (DoD, 2016)

DECF version 1.1. (DECF v. 1.1) addresses competencies in Tiers 2–5, with Tier 2 for acquisition professionals generally and Tier 3 specifically for acquisition ENG professionals being the primary focus. Though focused on the DoD, the overarching framework is intended to be relevant to a wide variety of stakeholders across government and industry and should provide critical insights for any organization looking to successfully implement DE.

Key Terminology

- **Competency Group**. Top-level grouping of related competencies that represents a core area of expertise in DE.
- **Competency Subgroup.** Subgroups contain related/like competencies.
- **Competency**. Major grouping of related KSABs; each competency is identified by its title and includes a description that succinctly encompasses the general knowledge and skills related to said competency.
- **KSAB**. A brief statement of *knowledge, skill, ability, or behavior* related to a competency and associated with a specific proficiency level in said competency.
- **Proficiency Level**. For each competency, there will be five possible levels of attainment or proficiency: awareness, basic, intermediate, advanced, and expert.

The entire DECF rests on a foundation of general digital competencies that are required for any individual who may have tasks within a digital environment.

Building on Existing Competency Models

Existing competency models and guidance from the practicing acquisition community were used to develop the DECF. For baseline terminology, the DoD competency models (DAU ENG and PM and U.S. Department of the Navy Systems Engineering Career Competency Model [SECCM]) are used, as these are already in use within the DoD for systems acquisition. The non-DoD competency models were mapped against the DoD models and existing competencies were updated to reflect a digital environment in lieu of the traditional acquisition environment where appropriate.

The existing competency models examined included:

- DAU ENG and PM competency models (DAU, 2016a, 2016b)
- INCOSE Systems Engineering Competency Framework (INCOSE, 2018)
- MITRE Systems Engineering Competency Model (MITRE, 2007)
- NASA SE/PM Competency Model (NASA, 2019)
- Helix *Atlas* Proficiency Model (Hutchison et al., 2020)
- IEEE Software Engineering Competency Model (IEEE, 2014)
- U.S. Department of Labor Engineering Competency Model (U.S. Department of Labor, 2017)
- Mission Engineering Competency Framework (Vesonder et al., 2018)
- U.S. Department of the Navy Systems Engineering Career Competency Model (SECCM; Whitcomb et al., 2017)

The team gathered materials from the DE community related to competencies, including in-progress competency frameworks such as the Naval Digital Engineering Body of Knowledge. Paired with the assessment of existing models, experts and practitioners in DE and Model-Based Systems Engineering (MBSE) provided insights into their common activities and current training programs. These inputs were collected and compared to the competency models to determine where they fit in the existing frameworks and where new competencies needed to be created to account for them.

Modeling the Framework

The team utilized a Model-Based Systems Engineering (MBSE) approach to create a SysML model of the DECF, its context, and its use cases. This approach leveraged the capabilities of DE and demonstrated the value proposition of their adoption. The primary focus of the model was to capture the structure and content of the DECF. Block definition diagrams (BDDs), such as that shown in Figure 1, are functional and visual representations of the structure of the DECF. The actual content of the model—the multitude of groups, competency areas and KSABs—is fulfilled by specified instances of each of these blocks.

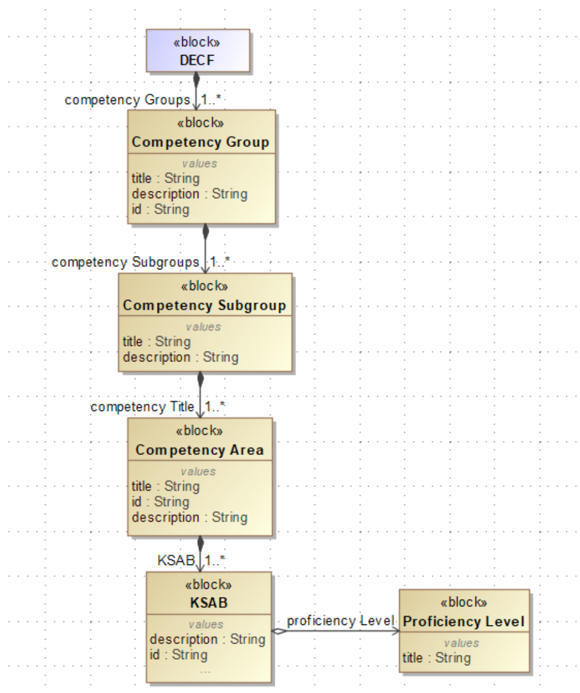


Figure 1. BDD of the DECF Structure

The primary purpose of the model was to capture the DECF structure and content, but initial efforts were made into exploring the use cases of the DECF. First the context of the relationship of the DECF to a defense organization was modeled as shown in Figure 2.

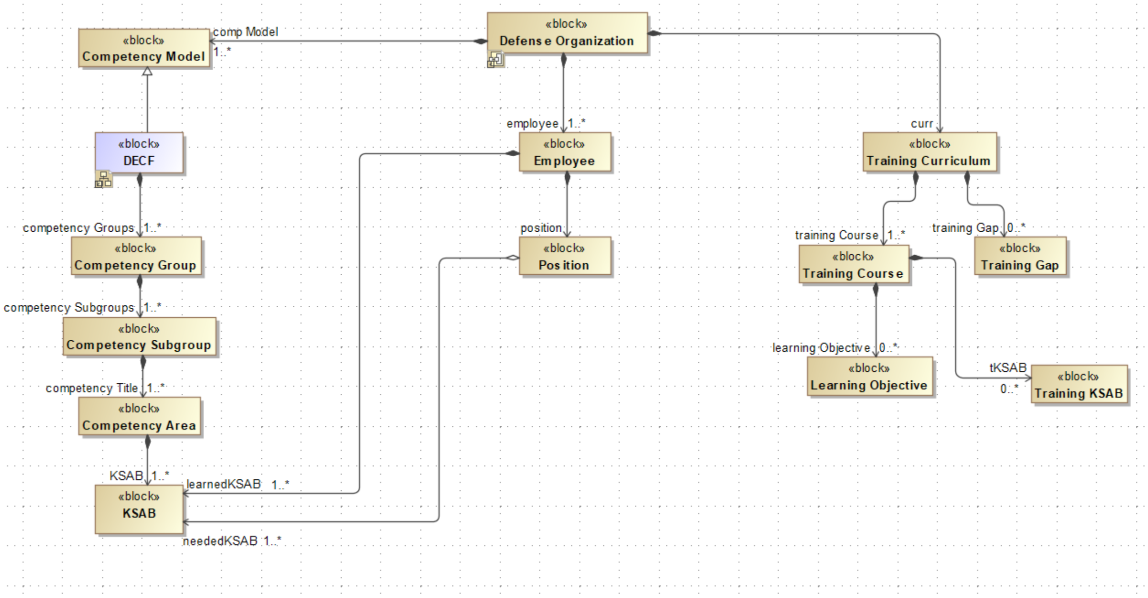


Figure 2. BDD of the DECF Context Within a Defense Organization

Although images captured from the SysML model may yield some insights to any person, attempting to browse model information natively is often inefficient and infeasible. Instead, an open-source tool called OpenMBEE was incorporated to generate text-based documents directly from model elements. (OpenMBEE, 2020). These documents were generated in, and accessible through, a web-based interface called View Editor. The View Editor interface not only allowed for easier sharing of model documents, but it also for direct commenting and modification of model elements. This modeling approach allowed reviews to take place in a DE environment using the online View Editor interface—giving potential users real-life experience using the required competencies—and it improved the visibility and collection of review feedback and facilitates its implementation. The model established an authoritative source for the DECF and followed DE practices for model management.

Digital Engineering Competency Framework v. 1.1

The DECF begins with the data foundations that are required for an effective digital environment ("Data Engineering"). The competency groups constitute how DE takes full advantage of the digital power of computation, visualization, and communication to take better, faster actions throughout the life cycle. These competency groups can be seen as supporting the four elements of John Boyd's OODA loop: Observe, Orient, Decide, and Act. Data Engineering guides how to observe, ensuring that data is acquired, curated, compressed, secured, and prepared. Next, Modeling and Simulation provide the ability to orient this data to describe and understand a phenomenon of interest. Decision Making utilizes analysis tools and techniques to make appropriate decisions. Finally, Engineering Methods are used to transform these decisions into engineering actions (Figure 3).

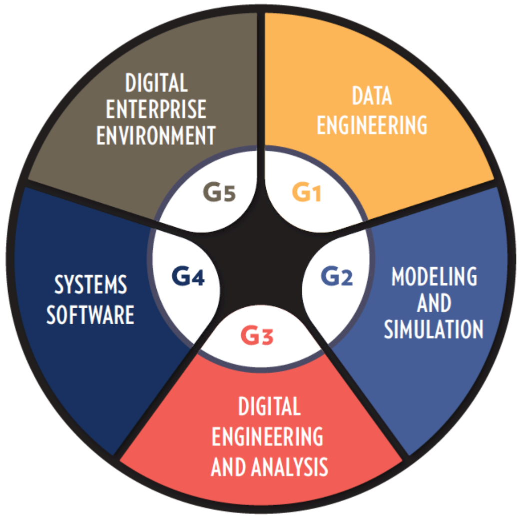


Figure 3. DECF Competency Groups

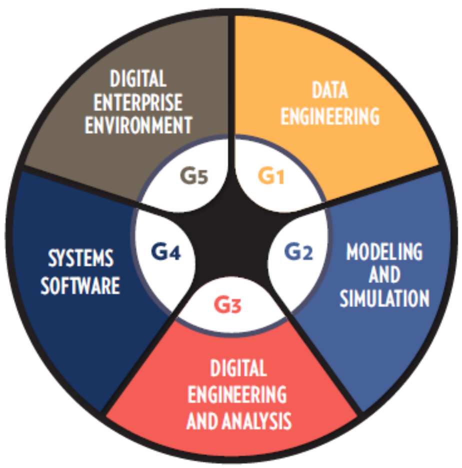
There are five competency groups identified for the DECF, as shown in Figure 4, along with a foundation of general digital competencies. Competency groups provide a logical structure for the individual competencies, making the DECF easier for users to understand and utilize. The groups in Figure 4 are intended to provide a holistic perspective of all the skills required to provide value in a digital environment. The five competency groups are defined in Table 1.

Table 1. Descriptions of DECF v. 1.1 Competency Groups

ID	Competency Title	Competency Description
G1	Data Engineering	Apply knowledge on how to acquire, curate, compress, secure, and prepare data resulting from a DE environment. Create or support data-focused processes. Data could originate from modeling and simulation, or from sensors in the physical world.
G2	Modeling and Simulation	Use of digital models to describe and understand phenomena of interest from initiation of the effort through the entire life cycle maturation. Model literacy—understanding what models are and how they work—is required to move into more advanced skills, from the ability to build a model using appropriate tools, standards, and ontology to creating a modeling environment.
G3	Digital Engineering and Analysis	Apply traditional engineering methods and processes in a digital environment. Create new engineering processes and methods for a digital environment. Create digital artifacts throughout the project or system life cycle. Use engineering methods, processes, and tools to support the engineering and system life cycle.
G4	Systems Software	Apply technical knowledge in various software or coding languages to create, support, and maintain applications. This includes the abilities to understand, apply, problem solve, create, and critique software in pursuit of particular learning and professional goals.
G5	Digital Enterprise Environment	Use the foundations of data, modeling, and software to create and maintain the digital enterprise. This requires creating the environment in which digital engineers, discipline and domain engineers, program managers, and decision-makers work.

The five competency groups are broken into nine competency subgroups and 31 competencies (including six foundational digital competencies). The competency hierarchy includes competency groups (G#), subgroups (S#), and individual competencies (C#). Where appropriate, the competency groups are divided into subgroups. Subgroups contain related like competencies. The competency hierarchy provides a logical structure for the individual competencies, making the DECF easier for users to understand and utilize. The hierarchy structure provides an overview of all the skills required to provide value in a DE environment regardless of specific roles. The overarching structure is shown in Figure 4.

DIGITAL ENGINEERING COMPETENCY FRAMEWORK (DECF) VERSION 1.1



FOUNDATIONAL DIGITAL COMPETENCIES	
F1	Digital Literacy
F2	Digital Engineering Value Proposition
F3	DoD Policy/Guidance
F4	Coaching and Mentoring
F5	Decision Making
F6	Software Literacy

LEGEND:
C# - Competency Title
F# - Foundational Competency Title
G# - Competency Group
S# - Competency Subgroup

G1 DATA ENGINEERING			
S1	Data Engineering	C1	Data Governance
		C2	Data Management

G2 MODELING AND SIMULATION			
S2	Modeling and Simulation	C3	Modeling
		C4	Simulation
		C5	Artificial Intelligence/Machine Learning
		C6	Data Visualization
		C7	Data Analytics

G3 DIGITAL ENGINEERING AND ANALYSIS			
S3	Digital Systems Engineering	C8	Digital Architecting
		C9	Digital Requirements Modeling
		C10	Digital Validation and Verification
		C11	Model-Based Systems Engineering Processes
S4	Engineering Management	C12	Digital Model-Based Reviews
		C13	Project and Program Management
		C14	Organizational Development
		C15	Digital Engineering Policy and Guidance
		C16	Configuration Management

G4 SYSTEMS SOFTWARE			
S5	Systems Software	C17	Software Construction
		C18	Software Engineering

G5 DIGITAL ENTERPRISE ENVIRONMENT			
S6	Digital Enterprise Environment Development	C19	Digital Environment Development
S7	Digital Enterprise Environment Management	C20	Management
		C21	Communications
		C22	Planning
S8	Digital Enterprise Environment Operations and Support	C23	Digital Environment Operations
		C24	Digital Environment Support
S9	Digital Enterprise Environment Security	C25	Digital Environment Security

Figure 4. Overview of DECF v. 1.1

Any acquisition role is likely to require some skills from each of these competencies in a digital environment; however, no role should require all the skills in every competency in each group at top proficiency level. The specific competency, proficiency levels, and KSABs required for an individual are dependent upon the role(s) being played.

Each of the competencies listed in Figure 4 is defined in the addendum at the end of this paper.

Proficiency Levels

In compliance with DoD Instruction 1400.25, each competency is broken down into KSABs relevant to proficiency levels within each competency (DoD, 2016). Not every individual will fully obtain all the competencies. The individual's proficiency level must be assessed to understand the true status of the workforce with respect to DE. Proficiency is the level to which an individual attains a competency, as illustrated in Table 2.

Table 2. Proficiency Levels of DECF v. 1.1

Proficiency Level	0	1	2	3	4	5
	None	**Awareness**	**Basic** General Knowledge (Entry)	**Intermediate** General Knowledge (Junior)	**Advanced** Detailed Knowledge (Senior)	**Expert** In-Depth Knowledge (SME)
Definition	No experience with or knowledge of the competency.	Applies the competency in the simplest situations.	Applies the competency in somewhat difficult situations.	Applies the competency in difficult situations.	Applies the competency in considerably difficult situations.	Applies the competency in exceptionally difficult situations.
		Requires close and extensive guidance.	Requires frequent guidance.	Requires occasional guidance.	Generally requires little or no guidance.	Serves as a key resource and advises others.
		Demonstrates awareness of concepts and processes.	Demonstrates familiarity with concepts and processes.	Demonstrates understanding of concepts and processes.	Demonstrates broad understanding of concepts and processes.	Demonstrates comprehensive, expert understanding of concepts and processes.

DECF v. 1.1 contains 962 KSABs divided among these competency areas. Each represents a unique and important aspect of what will enable a successful digital transformation and productive DE practices. The distribution of these KSABs in terms of both their competency area and respective proficiency level is shown in Table 3.

Table 3.Distribution of KSABs in DECF v. 1.1

Competency Group	Competency	Total KSABs	Proficiency Level				
			Awareness	Basic	Intermediate	Advanced	Expert
G1 Data Governance	C1 Data Governance	48	3	11	7	14	13
	C2 Data Management	30	2	7	1	14	6
G2 Modeling and Simulation	C4 Modeling	122	11	25	36	35	15
	C5 Simulation	56	8	8	16	16	8
	C6 Artificial Intelligence/Machine Learning	32	2	19	8	3	0
	C7 Data Visualization	22	2	4	12	2	2
	C3 Data Analytics	47	2	5	12	17	11
G3 Digital Engineering and Analysis	C8 Digital Architecting	55	3	14	18	18	2
	C9 Digital Requirements Modeling	24	1	3	15	4	1
	C10 Digital Validation and Verification	13	2	2	6	3	0
	C11 Model-Based Systems Engineering	108	11	33	17	35	12
	C12 Digital Model-Based Reviews	15	2	1	6	5	1
	C13 Project and Program Management	42	2	18	12	7	3
	C14 Organizational Development	18	1	2	1	4	10
	C15 Digital Engineering Policy and Guidance	23	1	3	2	7	10
	C16 Configuration Management	19	1	3	5	8	2
G4 Systems Software	C17 Software Construction	18	1	8	3	5	1
	C18 Software Engineering	47	3	5	5	24	10
G5 Digital Enterprise Environment	C19 Digital Environment Development	47	1	15	3	15	13
	C20 Management	28	2	2	1	10	13
	C21 Communications	12	1	2	1	3	5
	C22 Planning	11	1	2	2	3	3
	C23 Digital Environment Operations	27	3	4	8	10	2
	C24 Digital Environment Security	42	6	2	7	16	11
	C25 Digital Environment Support	56	2	5	12	22	15
		962	74	203	216	300	169

The KSABs within each competency are specifically targeted to DE. Therefore, a broader competency area like Communications has a relatively low number of KSABs related to DE, while a specific area like Model-Based Systems Engineering Processes, that is intrinsically linked with DE, has many KSABs.

It is also interesting to note the distribution of the KSABs across the five proficiency levels. KSABs at the Awareness and Basic levels represent broad fundamentals within a competency area, while KSABs at the Advanced and Expert levels include the practice of specific techniques that make up the various applications of the competency area. The DECF is established to be a general framework that can be used to create specific competency models that will be tailored based on component implementation of the Digital Engineering Strategy. As a result, the KSABs must cover the breadth of potential DE practices, even if all these practices are not utilized within every organization.

Cross-Cutting Elements in the DECF

Because there is not a specific competency surrounding them, it could be perceived that some relevant key elements in DE are not included in the DECF. The DECF is structured in a manner that concisely captures the breadth of DE enabling skills with a minimal number of competencies. Most technologies are inherently multidisciplinary in practice, and so creating distinct competencies for each potentially relevant technology would create significant overlaps in the framework. To demonstrate this potential overlap, three noteworthy cross-cutting DE elements were identified and analyzed:

- **Digital Twin:** An integrated multiphysics, multiscale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin.
- **Digital Thread:** An extensible, configurable, and component enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative technical data, software, information, and knowledge in the enterprise data-information-knowledge systems, based on the Digital System Model template, to inform decision-makers throughout a system's life cycle by providing the capability to access, integrate, and transform disparate data into actionable information.
- **Digital Artifact:** An artifact produced within, or generated from, the digital engineering ecosystem. These artifacts provide data for alternative views to visualize, communicate, and deliver data, information, and knowledge to stakeholders.

These elements are all enablers of the digital transformation of the DECF. An analysis of the DECF showed that each of these technologies had several dozen corresponding KSABs across various competencies. There were 55 KSABs related to Digital Artifacts spread across ten different competencies, 32 KSABs related to Digital Twin found in seven competencies, and another 32 KSABs related to Digital Thread in 11 different competencies. Each of these elements has more associated KSABs than several of the individual competencies included in the DECF. However as shown in Figure 6, the spread of these KSABs across the variety of competencies demonstrates the inherent interdisciplinary nature of these elements and how much overlap would occur if they were uniformly included as their own competencies. The exception to this rule is the competency for AI/ML, which was deemed both vital for inclusion and unique in its content. In the future there may be some additional unique, self-contained elements that may also warrant their own competencies in future iterations of the DECF.

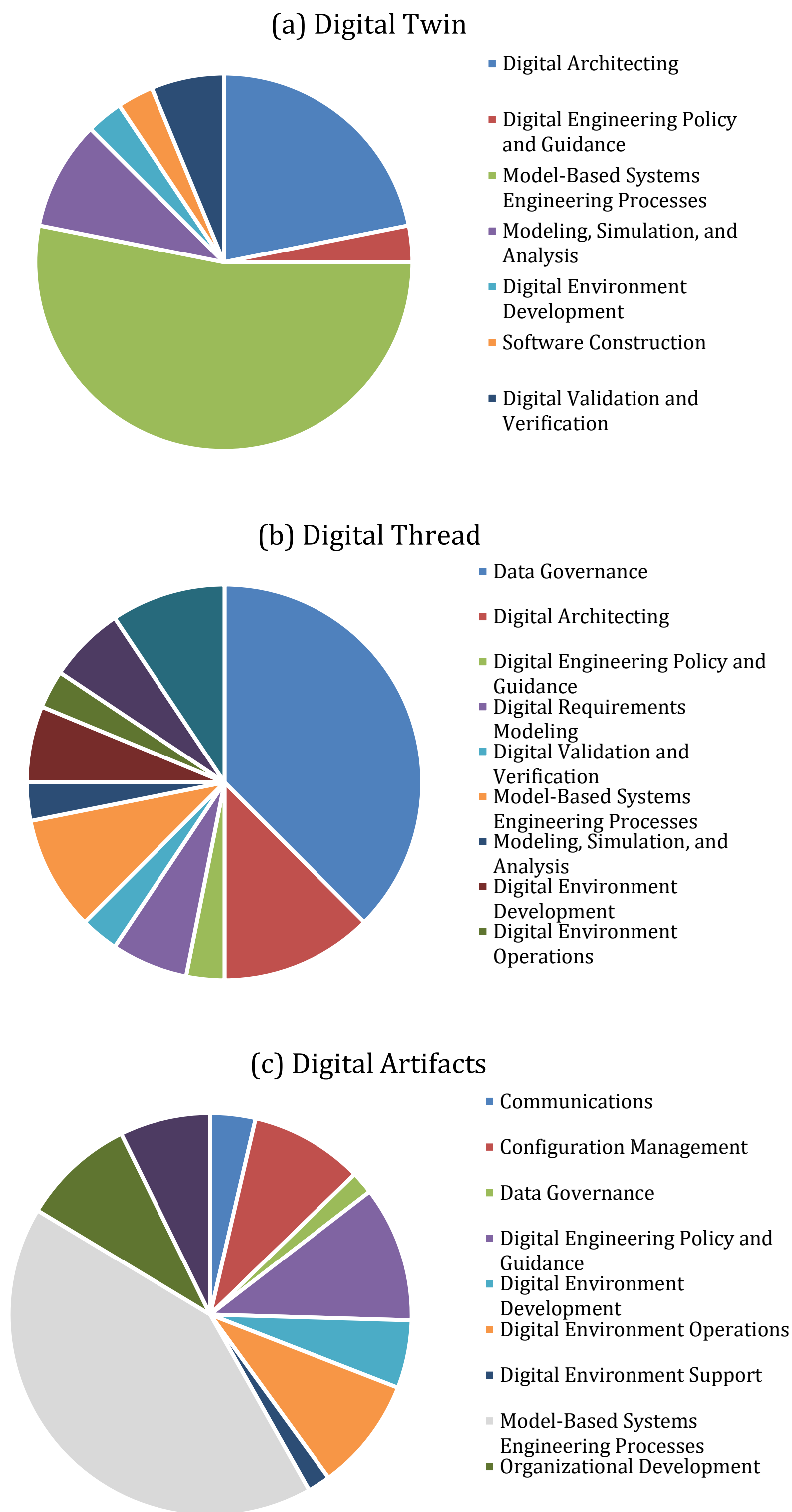


Figure 5. Distributions of Cross-Cutting Elements in the DECF: (a) Digital Twin; (b) Digital Thread; and (c) Digital Artifacts

Digital Engineering Competencies in DoD Resources

Using the DECF as a baseline, the team has been conducting a gap analysis using existing curriculum for digital engineering training in the DoD. Specifically, the team has investigated three courses:

- Defense Acquisition University (DAU) CLE084: Models, Simulations, and Digital Engineering;
- Coursera MBSE: Model Based Systems Engineering (currently utilized by the DAU); and
- MIT Architecture and Systems Engineering Program (utilized by NAVAIR), which includes relevant courses such as Models in Engineering and MBSE Documentation and Analysis.

The method used was to review course materials and attend the course as appropriate. Researchers captured the knowledge, skills, and abilities (KSAs) illustrated in the curriculum as well as the stated learning objectives. These were then compared against the KSABs in the DECF to determine the level of coverage and any major gaps. In addition, materials covered in the courses but not reflected in the DECF were flagged by the team for consideration and, where appropriate, incorporated into the DECF.

The key recommendations based on this analysis are outlined here. Additional details on the analysis can be found in (Hutchison et al., 2021).

The current courses in use by the Defense Acquisition University (DAU; DAU's internal course CLE084 and Coursera's MBSE course) are useful for providing a common foundation and orientation to the terminology and benefits of MBSE and DE. They would help individuals attain "Awareness" or "Basic" proficiencies—understanding of the concepts and in some instance the ability to apply them with heavy guidance or supervision. They are particularly helpful for individuals who have no background in systems engineering or modeling or who are firmly entrenched in a document-based acquisition approach. However, seasoned systems engineers and engineers would require substantially deeper training to become practitioners of DE.

Based on mapping of the DECF to the limited course materials available online for the MIT Architecture and Systems Engineering Program, the team found that the courses provided a solid fundamental understanding of models in engineering, MBSE, systems engineering, and digital architecting. In some cases, individuals can obtain proficiencies as high as the "Advanced" level. This program is recommended by NAVAIR for individuals who are facilitating change in their departments or organizations as they transition towards digital transformation.

Overall, the Department needs to:

- **Screen for foundational skills.** The DECF assumes a foundation of skills on which to build digital engineering competency. However, when it comes to training, this foundation cannot be assumed. The DoD needs to implement some basic screening approaches to ensure that individuals who do not have this foundation are offered opportunities to build it. Courses like DAU's CLE084 and Coursera's MBSE are already in use by the department and provide some of the necessary foundations. But it is important that only individuals who need these foundations utilize the resources, while individuals who will not gain proficiency through these courses be directed to other courses.
- **Introduce courses using modeling and simulation projects and problems as part of the curriculum.** As currently structured, CLE084 and Coursera MBSE training focus

on the lowest levels of proficiency. While Awareness level skills are a critical foundation on which to build skills for practical application of DE, they are not sufficient in and of themselves. Because up to 70% of learning is gained through experience (rather than classroom instruction), creating models and simulations that students can use to practice the skills of the task is paramount.

Ideally, implementation of modeling and simulation projects would impact not only DE-focused curriculum but could help enable a variety of disciplines for digital transformation. A modeling environment that spans, for example, the ENG and PM training courses at DAU would enable systems engineers and program managers to improve their familiarity with working in a digital environment before they are exposed to DE-specific training.

- **Include coaching and mentoring as part of the longer-term curriculum.** While coaching and mentoring can be applied within a single course, it would be most beneficial if longer-term coaching relationships were established specifically around the transition to DE in the DoD. The DAU may be in a unique position to broker such coaching opportunities, allowing individuals the opportunities to apply their DE knowledge on the job with expert guidance.

Future Work

The DECF provides only a starting point for updating the skills of the Defense acquisition workforce. In order to move forward, the DECF must be integrated into efforts such as the Office of the Under Secretary of Defense for Research and Engineering (OUSD [R&E]) and in particular, the Engineering and Technical Management (ETM) competency framework. The DECF and related competencies need to be understood and embraced by the workforce, including not only individuals playing primary roles in acquisition but their management and leadership as well. A baseline assessment against these competencies will be a critical first step to understanding how far along the path of digital transformation the workforce has come and in developing a plan to help the workforce continue to grow along this path.

References

Academy of Program/Project & Engineering Leadership. (n.d.). *NASA's systems engineering competencies.* National Aeronautics & Space Administration.

DAU. (2016a). *ENG: Engineering career field competency model* (Version 2.0).

DAU. (2016b). *PM: Project management career field competency model.*

DoD. (2016, June). *DoD civilian personnel management system: Civilian strategic human capital planning (SHCP)* (DoD-I 1400.25, Volume 250).

Hutchison, N., Pepe, K., Blackburn, M., See Tao, H. Y., Verma, D., Whitcomb, C., Khan, R., Peak, R., & Baker, A. (2021). *WRT-1006 technical report: Developing the digital engineering competency framework (DECF)—Phase 2* (SERC-2021-TR-005). Systems Engineering Research Center, Stevens Institute of Technology.

Hutchison, N., Verma, D., Blackburn, M., Pepe, K., See Tao, H. Y., Peak, R., Baker, A., Whitcomb, C., & Khan, R. (2021). *Developing the digital engineering competency framework (DECF)—Phase 2*. Systems Engineering Research Center, Stevens Institute of Technology.

Hutchison, N., Verma, D., Burke, P., Clifford, M., Giffin, R., Luna, S., & Partacz, M. (2018). *Atlas 1.1: An update to the theory of effective systems engineers*. Systems Engineering Research Center, Stevens Institute of Technology.

Hutchison, N., Verma, D., Burke, P., See Tao, H. Y., Giffin, R., Yu, Z., Makwana, D., Chen, X., & Xiao, Y. (2020a). *WRT-1004: Helix.* Systems Engineering Research Center, Stevens Institute of Technology.

Hutchison, N., Verma, D., Wade, J., Blackburn, M., Pepe, K., Peak, R., Baker, A., Whitcomb, C., & Khan, R. (2020). *Developing the digital Engineering Competency Framework (DECF)—Phase 1*. Systems Engineering Research Center, Stevens Institute of Technology.

Hu, E. (2021, March 27). Defense acquisition reform is critical going forward. *The Hill*. https://thehill.com/opinion/national-security/545228-defense-acquisition-reform-is-critical-going-forward

IEEE. (2014). *Software engineering competency model (v. 1.0)*. IEEE Computer Society.

INCOSE. (2018). *Systems engineering competency framework.*

ISO/IEC/IEEE. (2010). *Systems and software engineering—system and software engineering vocabulary* (ISO/IEC/IEEE 24765:2010).

MITRE. (2007). *Systems engineering competency model*.

Office of the Deputy Assistant Secretary of Defense (Systems Engineering). (2017). *DAU glossary: Digital engineering.* Defense Acquisition University.

OpenMBEE. (2021). *Connected engineering information for a connected world*. Openmbee.org.

NASA. (2009). *NASA's systems engineering competencies*. http://www.nasa.gov/pdf/303747main_Systems_Engineering_Competencies.pdf

NASA. (2020). *aPPEL knowledge services: PM&SE career development framework.* https://appel.nasa.gov/career-resources/development-framework/

NAVAIR/NAVSEA Working Group. (2020). *Digital competency framework overview: NAVAIR/NAVSEA working group* [Presentation].

Whitcomb, C., Khan, R., & White, C. (2017). *The systems engineering career competency model, version 1.0*. Naval Postgraduate School.

Addendum

Tables 4–8 display the specific competencies contained within the competency hierarchy shown in Figure 6.

Table 4. Competencies in the Data Engineering Group in DECF v. 1.1.

Subgroup	#	Competency	Description
S1-Data Engineering	C1	Data Governance	Data governance is a collection of practices and processes that help to ensure the formal management of data assets within a digital enterprise. Data governance practices help an enterprise gain better control over its data assets, including methods, technologies, and behaviors around the proper management of data. Data governance also entails security and privacy, integrity, usability, integration, compliance, availability, roles and responsibilities, and overall management of the internal and external data flows within an organization.
	C2	Data Management	Data management is applying policies, procedures, and information technology to plan for, acquire, access, manage, protect, and use data of a technical nature to support the total life cycle of the system. Data management includes verifying that all the data are secure, collected, documented, and archived along with descriptions of data to ensure completeness of data collected. Data management also ensures the distribution of data is in accordance with the data management plan for analysis.

Table 5. Competencies in the Modeling and Simulation Group in DECF v. 1.1.

Subgroup	#	Competency	Description
S2 Modeling and Simulation	C3	Modeling	Modeling is essential to aid in understanding complex systems and system interdependencies, and to communicate among team members and stakeholders.
	C4	Simulation	Simulation provides a means to explore concepts, system characteristics, and alternatives; open the trade space; facilitate informed decisions and assess overall system performance.
	C5	Artificial Intelligence/ Machine Learning	Artificial intelligence (AI) is the ability of machines to perform tasks that normally require human intelligence. Machine Learning (ML) is the application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.
	C6	Data Visualization	Data visualization is the creation of graphic representations of data, particularly to improve communication about that data. Data visualization is also the ability to identify patterns, trends, and correlations in the data and place them in a visual context to describe their importance. This entails building and managing data visuals, models, and artifacts.
	C7	Data Analytics	This is the process of inspecting, cleansing, transforming, modeling, and simulating data with the goal of discovering useful information, informing conclusions, and supporting decision making.

Table 6. Competencies in the Digital Engineering and Analysis Group in DECF v. 1.1.

Subgroup	#	Competency	Description
S3 Digital Systems Engineering	C8	Digital Architecting	Digital architecture activities use digital models to define a comprehensive digital system model based on principles, concepts, and properties logically related to and consistent with each other. Digital architecture has features, properties, and characteristics that satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission/business and stakeholder requirements) and life cycle concepts (e.g., operational, support) and which are implementable through digital enterprise related technologies. Digital architecture competencies relate to the ability to create system digital models and required architectural products and digital artifacts for a system or system-of-systems in accordance with applicable standards and policies.
	C9	Digital Requirements Modeling	Digital requirements modeling refers to being able to capture stakeholder high-level requirements by documenting stated needs in the form of a model, assist in the clarification and translation of need statements into a more digital engineering-oriented basis, create and derive system requirements, that are related to the system architecture definition. It is also used to establish requirements traceability throughout the digital model architecture; examine the relationships of requirements to digital artifacts, and trace design solutions to requirements; and ensure designs can be traced to the system capabilities and requirement sets within digital enterprise environment.
	C10	Digital Validation and Verification	Digital verification is the process for determining whether or not a product fulfills the requirements or specifications established for it, by using digital models and artifacts for testing and verification. Enabling this practice is important to ensure that digital practices correlate well with their real-world projects.
	C11	Model-Based Systems Engineering Processes	Model-based systems engineering is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.
S4 Engineering Management	C12	Digital Model-Based Reviews	Digital model-based reviews define the series and sequence of model-based systems engineering activities that bring stakeholders to the required level of commitment, prior to formal reviews. It utilizes system models, artifacts, and products for analysis of design and technical reviews to execute trade-off and design analyses, prototyping, manufacturing, testing, and sustainment of the system.
	C13	Project and Program Management	Project management is the planning, coordinating, and monitoring of the work activities needed to deliver a product, service, or enterprise endeavor within the constraints of schedule, budget, resources, infrastructure, and available staffing and technology.
	C14	Organizational Development	Organizational development focuses on developing organizational policies, standards, and guidelines for model-based systems engineering methods and artifacts.

Subgroup	#	Competency	Description
	C15	Digital Engineering Policy and Guidance	Digital engineering policy and guidance focuses on identifying process improvements to model-based engineering methods and contributing to organization of system life cycle development standards and definition of best practices. It includes defining strategy and approach to be used for modeling and analysis of complex systems.
	C16	Configuration Management	Configuration management refers to the development of configuration management strategies, policies, standards, and guidelines for digital engineering related artifacts in accordance with model-based systems engineering methods.

Table 7. Competencies in the System Software Group in DECF v. 1.1.

Subgroup	#	Competency	Description
S5 Systems Software	C17	Software Construction	Software Construction refers to the detailed creation of working software through a combination of coding, verification, unit testing, integration testing, and debugging.
	C18	Software Engineering	Software Engineering is the systematic application of digital engineering approaches to the development of software.

Table 8. Competencies in the Digital Enterprise Environment Group in DECF v. 1.1.

Subgroup	C#	Competency	Description
S6 Digital Enterprise Environment Development	C19	Digital Environment Development	A digital enterprise environment is an integrated digital development framework in which digital models and representations are interconnected such that the content and activities within it are managed to accomplish the organizational objectives of the enterprise.
S7 Digital Enterprise Environment Management	C20	Management	Management in the digital enterprise environment aims to deliver a framework that ensures transformational processes in enterprises occur with pace, high-quality, and security. This is achieved through a set of IT solutions that are designed to make digital businesses fast, seamless, and optimized at every level.
	C21	Communications	Communications include using digital model artifacts from the digital enterprise environment to investigate and manage the adoption of appropriate model-based tools, techniques, and processes for the operation of digital enterprise environment systems and services. Communications also establishes the appropriate guidance to enable transparent decision-making to be accomplished, allowing senior leaders to ensure the needs of principal stakeholders are understood, the value proposition offered by digital enterprise environment is accepted by stakeholders and the evolving needs of the stakeholders and their need for balancing benefits, opportunities, costs, and risks is embedded into strategic and operational plans.
	C22	Planning	Planning in the digital enterprise environment includes establishing strategies to monitor and manage the performance of digital artifacts and services, in respect to their contribution towards enterprise performance goals. Planning ensures that a framework of policies, standards, processes, and practices is in place to guide provision of

Subgroup	C#	Competency	Description
			digital enterprise environment services, and that suitable monitoring of the governance framework is in place to report on adherence to these obligations.
S8 Digital Enterprise Environment Operations and Support	C23	Digital Environment Operations	Operations within the digital enterprise environment include creating digital models and simulation artifacts and technology roadmaps, and sharing knowledge and insights from processes and results, with others. It encourages adoption to changes in the digital enterprise environment process or technology. It includes setting parameters for the prioritization of digital resources and the changes to be implemented and the configuration of digital engineering methods and tools to address the project needs.
	C24	Digital Environment Support	Support within a digital enterprise environment includes abilities to develop, mature, and implement methods and processes to support digital enterprise environment activities across the enterprise and life cycle.
S9 Digital Enterprise Environment Security	C25	Digital Environment Security	Digital Environment Security includes developing policies, standards, processes, and guidelines to ensure the physical and electronic security of digital environments and automated systems. This includes performing security risk and vulnerability assessments, and business impact analyses related to security and information assurance in the digital enterprise environment. It is intended to provide advice and guidance on the application and operation of digital environment physical, procedural, and technical security controls.

A Study of MBSE Through the Development of Modeling and Data Exchange Processes

William Emeny—graduated from Virginia Tech with a bachelor's degree in mechanical engineering and from the Naval Postgraduate School (NPS) with a master's degree in systems engineering. Emeny supports the Model Based Product Support NAVSEA program of record at the Naval Surface Warfare Center in Port Hueneme, CA. Emeny leads the effort to implement model-based systems engineering concepts and tools to support model driven design objects and to manage program requirements. Emeny is very active in the Department of Navy's MBSE community; he is an active member of the Naval Digital Engineering Body of Knowledge (NDEBOK) and the MBSE community of interest. [william.emeny1@navy.mil]

Ryan Robar—started his professional career with the Department of Homeland Security. From the Department of Homeland Security, he transitioned to Naval Sea Systems Command. Robar has spent the last 7 years holding different positions within supportability of the AEGIS Weapon System. Robar recently transitioned to focus his professional career to the cradle-to-grave supportability and integration of the Missile Defense Agency's Sea Based Weapon System Ballistic Missile Defense capabilities within the AWS. Robar is a recent graduate of NPS with a master's degree in Engineering Sciences and holds a bachelor's degree from California Lutheran University in Business. [ryan.robar@navy.mil]

Dustin Talley—is a Mechanical Engineer working at the Naval Surface Warfare Center, Port Hueneme Division. Talley started his career in 2015 working on research and development efforts related to renewable energy. In 2017, he transitioned his career and began working in test and evaluation. His primary role is planning and executing developmental testing of Directed Energy Weapons for naval platforms. Talley graduated with a Bachelor of Science in Mechanical Engineering from California State University, Long Beach. Talley also recently graduated with a Master of Science in Systems Engineering from NPS. [dustin.talley@navy.mil]

Michael Rubow—is an Engineer who works at the Naval Surface Warfare Center, Port Hueneme Division (NSWC PHD) as part of the In-Service Engineering Agent (ISEA) branch for the Harpoon and Over-The-Horizon Weapon Systems (HWS/OTH-WS). He earned his Master of Science degree in Systems Engineering from NPS in December 2020 and his Bachelor of Science degree in Mechanical Engineering from the University of California, Merced (UCM) in May 2014. Rubow enjoys participating in outdoor adventures such as hiking, skiing, and beach activities with his wife Caitlin. [Michael.rubow@navy.mil]

Lance Lowenberg—is a Robotics Engineer at the Naval Information Warfare Center (Pacific) who specializes in unmanned systems development. In addition to his work, he enjoys restoring classic cars and tinkering with other mechanical projects. He and his wife love camping, sailing, hiking, and traveling with their currently one but soon to be two furry four-legged pals. Lowenberg has a BS in Mechanical Engineering from UC Santa Barbara and an MS in Systems Engineering from NPS. [lance.lowenberg@navy.mil]

Lynn Nguyen—is a logistician at the NSWC PHD supporting readiness engineering for the Program Executive Office for Unmanned and Small Combatants (PEO USC) for the Littoral Combat Ship (LCS) program. Nguyen is involved in efforts to support system reliability, maintainability, and availability for LCS. She has a background in Integrated Product Support, DMSMS management, and task management. She completed her undergraduate education at UCSD and graduated from NPS with an MS in Engineering Systems. [lynn.p.nguyen@navy.mil]

Abstract

The DoN is undergoing a digital transformation that is set to address the needs of sustaining fleet assets for extended periods of time, while maintaining a superior lethality. Within the engineering domain, the DoN is starting to identify model-based systems engineering (MBSE) tools and concepts to streamline processes and enhance capability. The capstone looked to lay the foundation for a conceptual system model development process that utilizes SysML and OOSEM

to produce system model data and artifacts derived from a single scenario. During the digital transformation, communication of system model data to stakeholders was identified as a need, and a SysML tool was used to generate model-based documentation from a formatted Microsoft Word document. With incoming digital product support capabilities from the MBPS program, communication from a MBSE environment is critical and requires XML-formatted data. Using the information collected in the completion of the scenario, it was discovered that SysML elements will lose their SE-specific stereotypes when converted directly into XML format. To counter this, the capstone developed UML instances derived from the S3000L UML class-based data model to be converted into XML format. The findings and developments of this capstone support the ability for organizations to standardize the way system modeling data is developed, collected, and communicated to other systems external to the engineering domain.

Executive Summary

Currently, there is an initiative to transform legacy logistics information technology (IT) systems to use a model-centric approach to support products that aims to increase system uptime and reduce support costs. Model Based Product Support (MBPS) is a single piece of a larger digital readiness vision that includes new capabilities, such as predictive analytics, data-as-a-service, platform-as-a-service, process automation, and the integration of data across multiple platforms (SEA06L, 2019). This vision of a logistics digital transformation is shown in Figure 1. The new integrated product life-cycle management (PLM) platform supports the sharing of a standardized data model that enables the capability to perform logistics support analysis. The PLM platform inside the product support (PS) domain would have conduits with the engineering, maintenance, training, and other system life cycle communities to support better logistics models and better supported systems (NSRP, 2019).

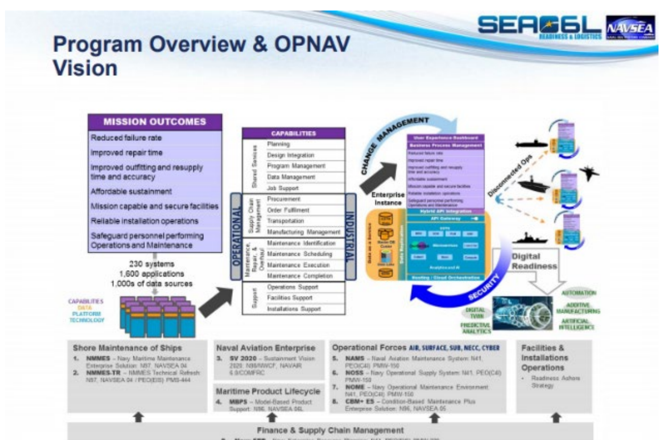


Figure 1. Logistics Digital Transformation Vision Overview (NSRP, 2019)

The current transformation occurring in the PS domain is also being pursued within the engineering domain with the exploration and implementation of model-based systems engineering (MBSE) concepts. Department of Defense (DoD) strategic documents have expressed the need that as systems become more complex, the DoD will require more robust engineering practices to develop weapon systems and maintain superiority over our enemies (Engineering, 2018). For many years, the DoD has relied on document-based, stove-piped engineering processes and is now looking to incorporate digital engineering practices to work more efficiently. The incorporation of digital engineering will require investment in new methods, processes, and tools in order to enable systems to become more lethal and affordable (Engineering, 2018). The Department of Navy (DoN) has embraced the goals set by the DoD Digital Engineering Strategy by developing its own set of high-level strategic documentation that discusses high-level implementation strategies and their alignment to the DoD documentation (DoN, 2020).

One of the alignment goals set in the DoD Digital Engineering Strategy and envisioned in the DoN Digital Systems Engineering Transformation Strategy is the formalization of the development, integration, and use of models. Using the system modeling language (SysML) and SysML tools, the capstone group built a conceptual system model development process based off the object-oriented systems engineering methodology (OOSEM). The OOSEM is a top-down, scenario-driven approach that leverages object-oriented concepts and other modeling techniques to support in the development of a more flexible and extensible system architecture that can accommodate the constant change in requirements or technologies (Friedenthal et al., 2012). The developed process encapsulates system modeling data within what is known in SysML as blocks, analogous to classes within the unified modeling language (UML).

The conceptual system modeling process was developed, and an example scenario was completed in which an organization has a need to develop and implement a model-based system engineering environment; henceforth named the Digital Engineering Environment (DEE), locally within the organization. The scenario walks through the development of the conceptual system model and pieces of the logical system model prior to a request for proposal (RFP) where vendors would bid on to develop a physical product based off the information presented to the vendor in the conceptual system model. The conceptual data model, shown in Figure 2, displays the type of models and artifacts that make up the system model and how they contribute to the development of the system of interest. The information and artifacts captured in the data model are developed within the system modeling process described in this capstone report.

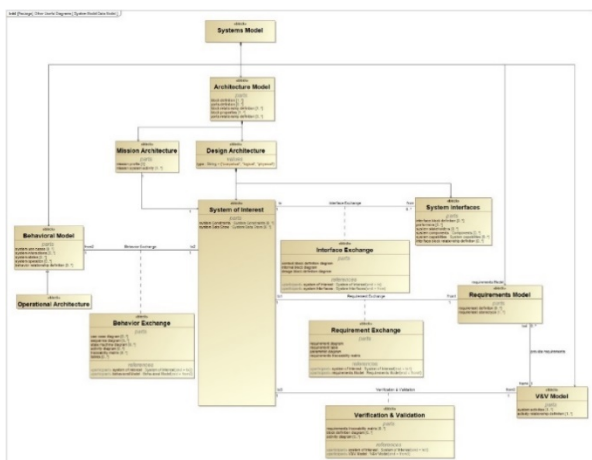


Figure 2. Proposed Conceptual Data Model for Developing a Systems Model

System model data collected over the design and development phases of a system must be capable of being consumed and of use to the PS domain to enable the reuse of system data for supportability analyses. The MBPS program overview presentation displayed the program's use of the S-Series specifications developed by the AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association (ASD/AIA). These specifications layout an extensible markup language (XML) schema with data classes useful for different types of PS efforts, including provisioning, maintenance task analysis (MTA), level of repair analysis (LORA), software support analysis, and other logistics support analyses. There is not a current mapping between the data elements within SysML to the UML data elements within the S-Series specification; however, the developers of the specifications have developed a data model, which can be consumed and useful to a model developed in a SysML toolset. As shown in Figure 3, element instances contain the useful PS data which, if contained within an isolated model, could be manually translated into XML and exported to the S-Series database for analysis use.

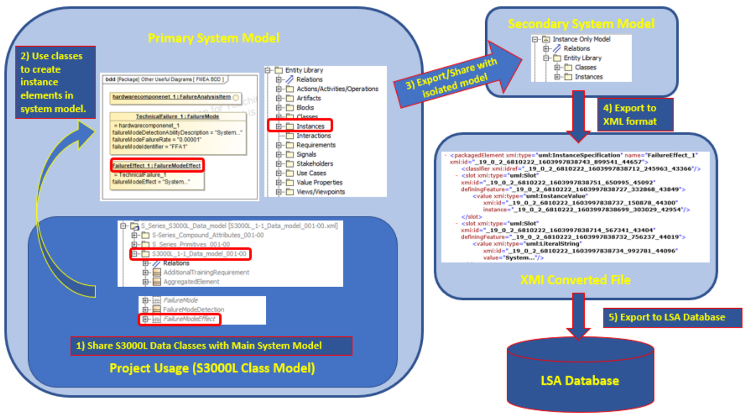


Figure 3. SysML Instances Translated into XML File Format for MBPS Consumption

Stakeholders were interested in verification of the system design which was supported with the presentation of system model data. Many stakeholders do not have experience in using system modeling tools but are familiar with many of the presentation formats within the model. Many system modeling tools have the capability of developing model-based documentation. Some of the presentation views within the developed model for the capstone's scenario were utilized to develop a model-based concept of operations (CONOPS). The CONOPS document template was downloaded from public online sources and configured using the velocity template language (VTL) to place model presentation artifacts into the CONOPS, automatically, upon a click of a button (Department of Veteran Affairs [VA], n.d.).

The model building process does explain the development of a conceptual data model but describes very little work on the development of a logical system model and does not approach the physical model development phase. More research is needed to understand the interfaces with other digital engineering tools and how related data can be used to further define certain aspects of the system model. The process completed a scenario in which useful products were developed for demonstration. To ensure its validity, verification and validation of the proposed process should occur using pilot projects to identify and fix any demonstrated gaps within the process. Future work should include the implementation of another scenario in which a fielded system wishes to undergo a system change. This scenario would require the system model to be updated and used to perform alternative analysis in both the engineering and PS domains.

The resulting scenario provided a collection of data points that represents different SOI viewpoints and that could be used within alternate domains to perform analyses. The conceptual system model in this instance would solely be used to demonstrate a problem and need to a design team or vendor. The instance of a problem would be derived from the technical capability audit (TCA) within the developed process whose following steps would be used to collect data and build presentation views. With the emergence of system of systems (SoS) modeling, it is theorized that existing and anticipated emerging gaps could also be a source of problems in which a TCA could be utilized to determine the necessary solution type (Mohammadi et al., 2014). Future work could explore the use of a TCA to identify future capability gaps as a second scenario to validate SysML models presented in this capstone. Using SysML and tailoring a process derived from the object-oriented system engineering methodology (OOSEM), enabled the encapsulation of system model data into a single SOI model element to communicate a design's architecture, behavior, requirements, and verification and validation activities. Review of the data developed during the simulation and the S3000L data model shows that there is a need for engineering data (Aerospace and Defense Industries Association of Europe and Aerospace Industries Association [ASD/AIA], 2014). The capstone presented a way to translate information from SysML into XML, but more work is needed to develop a data mapping to the S3000L XML data model that could lead to an automated conversion process.

Introduction and Background

Introduction

This project demonstrates a process that gives United States Navy (USN) organizations the capability to develop a conceptual system model, whose data can be used to initiate digital twin and digital thread capabilities. The process outlined in the appending pages is meant to be the foundation for creating the conceptual data model that would be created and matured over the life cycle of the system. This process utilizes the early steps of the object-oriented systems engineering methodology (OOSEM) approach, a model-based system design approach, as a guide in its design with the expectation that it will be used to assist Department of Navy (DoN) organizations in better defining and presenting conceptual system needs and requirements to design agents (Friedenthal et al., 2012). Process gaps within OOSEM were identified and tailored to better suit the needs of our stakeholders. For example, the project implements a data-driven approach to problem definition, something that is not included in OOSEM. To fulfill this capability, the technical capability audit (TCA) was added to the process. The TCA uses both quantitative and qualitative data from questionnaire or survey data to determine the type of problem the organization is facing (Mohammadi et al., 2014). Appended sections further expand upon this with the descriptions and applications of the technical capability audit (TCA) to perform problem analysis and parametric modeling for engineering analysis. At the conclusion of specified steps in the presented process, the modeler will have gathered enough data to enable the development of presentation artifacts. The systems modeling language (SysML) was utilized as the data model, while Cameo Enterprise Architecture (CEA) was used to produce SysML presentation artifacts. The produced artifacts were used as the process verification method and was performed using a generalized scenario, performing the outlined steps to create data points and artifacts that can be used to present to the system's stakeholders or to provide information to external systems in order to enable their own capabilities. The report will discuss the steps and artifacts developed through each step of the developed process. A discussion will follow that demonstrates potential uses for the data to support the development of acquisition documentation and the analysis of data communication with systems external to the systems engineering boundary.

Problem Overview

The Department of Defense (DoD) produced the DoD Digital Engineering Strategy to help spark and align a digital transformation in the engineering community. More recently, the DoN and Marine Corps delivered Digital Systems Engineering Transformation documentation that describes the goals for model-based systems engineering (MBSE) and lays a framework for MBSE implementation (DoN, 2020). Currently, MBSE is still immature relative to model-based product support and the enterprise technical reference framework (ETRF) and a fully matured enterprise capability may be some time off. In this scenario, it is assumed that the need for better, faster, and centralized tools and process in the system engineering community has been identified, and MBSE is the identified solution. With MBSE being as immature within the enterprise as it is, the DoN is still researching for more information on the MBSE subject and trying to identify how it will best be implemented alongside the product support digital transformation. There is not yet a formal standard set of processes, models, data and tools at the DoN enterprise level that align to all of the objectives in the Digital Engineering Strategy and local commands are beginning to develop their own local instances of MBSE environments. The lack of standardization of the processes, data formats and exchanges may lead to systems again becoming isolated and less efficient as their potential.

Background

As systems experience a never-ending increase in complexity, rapidly changing operational and threat environments, increased budget constraints, and more demanding schedules, the DoD needs more robust engineering practices. Current engineering processes are often document intensive and stove-piped. To meet their needs, the DoD is transforming its engineering practices to a digital engineering methodology utilizing model-based approaches, including MBSE (Engineering, 2018). MBSE is a subset of digital engineering and can be defined as the use of models to support the activities within systems engineering (SE) process, including requirements, architecture, design, verification, and validation (Giachetti, 2020). The implementation of MBSE has been theorized to enable new capabilities within the SE process (DoN, 2020). One of the primary objectives of implementing MBSE is to develop an integrated set of digitally integrated views that enables the capability of automating the engineering assessment of proposed designs. This automated capability would be able to identify risks and gaps through the simulation of operational scenarios. The digital environment would provide feedback data to enable the application of data-driven decision-making.

To maximize the effectiveness of MBSE, an organization must find a cohesive set of modeling tools and methods. The process supporting these activities is laid out in the implementation of OOSEM, applying SysML as the model syntax. The OOSEM is a top-down, scenario-driven approach that leverages object-oriented concepts and other modeling techniques to support in the development of a more flexible and extensible system architecture that can accommodate the constant change in requirements or technologies (Friedenthal et al., 2012). The activities within the OOSEM process reflect those of the fundamental SE process, including needs analyses, requirements analyses, architecture design, trade studies and analyses, and verification (Friedenthal et al., 2012). The primary output to the OOSEM process is a model of the system of interest (SOI). The collected data on the SOI is captured and encapsulated using a SysML block, an extension of the Unified Modeling Language (UML) class that includes allocated system elements describing different system views. This project explored a system’s architecture, behavior, requirements, and verification and validation (V&V) views. Each view contains a set of SysML diagrams, matrices, or tables to create a model of each system model view. These diagrams are presentation mechanisms to display different data sets of the system model to different stakeholders.

Digital transformation inside the DoN is not only an interest within the engineering domain, but within the entire enterprise. The DoN has a vision for digital transformation, and it has begun in the logistics IT domain with the implementation of the ETRF. The ETRF vision will provide a framework that will generate scalable, interoperable, flexible, and fluid technology solutions that will provide access to information and data at anytime, anywhere. One of the major capabilities of the ETRF is the implementation of an integrated platform as a service (PaaS) environment that will unify all logistics applications internal to the ETRF system and will deploy a set of application programming interfaces (API) to integrate with future and legacy systems. The vision of the ETRF will contain many logistics applications that will be managed by the PaaS. Applications within the ETRF will fall into one of the following four key mission areas: integrated readiness, supply chain management (SCM), maintenance, repair, and overhaul (MRO), or product life-cycle management (PLM; Accenture, 2019).

There are currently two major programs sponsored by the Office of the Chief of Naval Operations (OPNAV), Model Based Product Support (MBPS) and Navy MRO (NMRO), that are developing the applications to meet the objectives of these mission areas. These applications will be developed to deploy new methodologies, including model-based approaches, and replace legacy systems with new systems that utilize digital tools and processes to replace the old capability set. One of these programs is MBPS, which spans across all four of these mission areas and is of special importance to this project. MBPS is an initiative within the Naval Sea Systems Command (NAVSEA) with cooperation from the Program Executive Office (PEO) that will create and implement a digitally integrated environment focused on the support of Naval systems. The MBPS environment will support the production of many artifacts in the support of sustaining engineering, including reliability centered maintenance (RCM) artifacts, level of repair analysis (LORA), readiness at cost analysis, reliability block diagrams, fault tree analysis (FTA), and other product support documentation and analyses. An authoritative source of product support data, that will enable the supportability analyses listed above. The authoritative data structure will be established and MBPS and developed using industry standards to support the communication and exchange of data between systems internal and external to the MBPS environment (SEA06L, 2019). The integration of MBSE and MBPS is of great interest. It has been theorized that this integration could lead to systems that maximize availability, effectiveness, capability, and affordability (Kwon et al., 2018).

In order to perform cross-platform verification and analysis, data must be accessible by both environments through an authoritative data source. Currently, there are two identified potential authoritative data sources within the ETRF that are being sponsored for development. Within MBPS, there is the Navy Product Data Management (NPDM) that is being established as the authoritative data source for all system technical data once a system reaches the operation and sustainment phase of the system's life cycle. The ETRF will also be deploying the agile warfighter analytics readiness environment (AWARE) within NMRO. The AWARE is a data-as-a-service (DaaS) platform to manage and communicate maintenance data from data collected by ship-based NMRO applications to the AWARE. Any data needed by the applications will be stored and transferred through at least one of these data sources. For MBSE, this has been identified as a major integration point between SE and product support (PS) capabilities which, in the future, will communicate and supplement the capabilities of one another (Accenture, 2019).

Problem Statement

The USN has produced documentation describing the characteristics of a model-based engineering environment but has not yet realized a solution for a model-based engineering environment and how that environment would be implemented and integrated into the system of systems (SoS) enterprise digital transformation vision (DoN, 2020). A need has been identified by the SE community at the Naval Sea Systems Command, Port Hueneme Division (NSWC PHD) to implement a local model-based system engineering (MBSE) environment and to understand how the MBSE data set, capabilities and tools would integrate into the ETRF.

With the MBPS capability set being more mature than the MBSE capability set, this capstone looked to identify potential avenues of implementation that aligned to the high-level objectives within the DoD and DoN strategic documents. With the development of a standard modeling process, the standardization of data sets, presentation artifacts, tool sets, etc. will follow, enabling many of the MBSE capabilities. A standardized set of data of system model data will enable external boundary communication and the development of model-based documentation.

Project Objectives

This capstone team had two high-level objectives: develop a formal process using SE methodologies that would be capable of developing a conceptual system model and compile a final report that will explain the problem space, describe the solution space and how it solves identified issues, describe and explain the processes used, present the developed artifacts, and provide recommendations for future work or action.

The objective of the model is to provide a standard process for organizations to develop a conceptual system model that contains early system architecture, behavior, requirements, and verification and validation models. The conceptual model would be the starting point for a program's digital twin and thread that would mature along with the design to include data from the logical and physical levels of the design. The process and development of system model data enables the capability of producing model-based documentation that supports the development of programmatic documentation from templates. The report will demonstrate and explain the process of how the capstone team developed and produced a model-based concept of operations (CONOPS) from a Microsoft Word template found in the public internet domain.

To ensure the process satisfies the stakeholder objective and requirements, the capstone team applied the process to a development scenario to support the verification of the process. The model will be supplemented by a textual report that will further include explanations of the processes and recommendations for future work.

Project Scope

The scope of the capstone is set based on the scenario outlined in the Problem Overview section. Verifying the developed process with these scenarios will produce a set of artifacts that will be used to demonstrate to organizations how MBSE can be used. The documented process and developed artifacts are a part of the framework of this report, and the discussions that follow will be based off the development of the system model and the verification methods using the use case scenarios.

Section Summary

This section introduced the capstone, overall problem, background information, problem statement, scope, and objectives. This information is used in the understanding of the information and processes that will be discussed in the appended sections.

Having identified the need to utilize MBSE concepts to enhance the DoN's engineering capabilities, the capstone team documented a standard process. The process is used to support an organization's capability to develop conceptual system models. The process was developed using the object-oriented SE methodology as a guide as to what data is required for the development of the system model and the presentation artifacts were produced using Cameo Enterprise Architecture. To provide examples of artifacts to the stakeholders and this report, a fictitious scenario was applied. The appending sections will provide more detailed explanations for each phase of the process and the artifacts that are consumed and produced by each phase.

Problem Analysis

Important Definitions and Terms

Common definitions and terms are used throughout this report. These definitions were researched and established during the literature review. These terms are defined in this section to give the reader a general understanding of the topics to be discussed.

Model-Based Systems Engineering

The use of models to convey SE concepts and data either in place of or in conjunction with traditional textual methods has gained wide acceptance in recent years. This was introduced by INCOSE in 2007 as follows:

> Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life-cycle phases. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. (INCOSE Technical Operations, 2007)

Model-Based Product Support

There is no official definition of model-based product support (MBPS) in literature, but the collective sources support a general definition. Model-based product support is a broad term that essentially translates to model-based electronic tools and information systems that enable the support of logistics functions such as training, maintenance, operations, and sustainment. Model-Based Product Support is the cooperative initiative between NAVSEA and PEO that will provide multiple digital logistics capabilities to the DoN.

Architecture Framework

The Architecture Framework defines how an architecture will be created and subsequently utilized through a set of rules and practices. It is defined by the MITRE Corporation as follows:

> An architecture framework is an encapsulation of a minimum set of practices and requirements for artifacts that describe a system's architecture. Models are representations of how objects in a system fit structurally in and behave as part of the system. Views are a partial expression of the system from a perspective. A viewpoint is a set of representations (views and models) of an architecture that covers a stakeholder's issues. (MITRE, 2015)

Enterprise Technical Reference Framework

The push to consolidate existing systems into a new common logistics platform that leverages new technologies and innovations is necessary in order to adapt to the Navy's changing needs. The following two quotes describe this:

> The vision of Enterprise Technical Reference Framework (ETRF) is to enable and accelerate the overall objective of Navy Logistics IT. ETRF provides a digital logistics IT architecture that will generate scalable, interoperable, flexible and fluid technology solutions; maximizing access to information/data via applications anywhere, on any device at any time. (Accenture, 2019)

> The Enterprise Technical Reference Framework will leverage the Digital Transform Plan Services, Data, Technology, Security and Change Management strategies to provide a framework and roadmap to transform 1600+ current Applications and 5000+ data sources to a common unified logistics IT platform. (Accenture, 2019)

Digital Information Technology (IT) Transformation

The DoD digital IT transformation exists within the Joint Information Environment (JIE) framework that is comprised of a comprehensive Department-wide IT modernization that exists within the DoD Information Network (DoDIN). The JIE purpose is to "improve mission effectiveness, increase cybersecurity, improve interoperability, deliver capabilities faster, and realize IT efficiencies" (DoD, 2019). The DoD JIE framework is comprised of 10 Capability Objectives, as shown in Table 2.

Table 2. Alignment of DoD CIO Objectives to JIE Capability Objectives and Initiatives (DoD, 2019)

JIE Capability Objective	JIE Initiatives	DoD CIO Objectives
Modernize Network Infrastructure	Optical Transport Upgrades, MPLS Routers Buildout, ATM Switch and low speed TDM Circuit Elimination, Satellite Communications Gateway Consolidation and Modernization, IPv6 Implementation	• Modernize Warfighter C4 Infrastructure and Systems • Modernize DISN Transport Infrastructure • Modernize and Optimize DoD Component • Networks and Services
Enable Enterprise Network Operations	Establish global and regional operations centers, Establish the JIE Management Network, Converge IT Service Management (ITSM) solutions	• Modernize and Optimize DoD Component Networks and Services • Shift from Component-Centric to Enterprise-Wide Operations and Defense Model
Implement Regional Security	JRSS, JMS	• Modernize DISN Transport Infrastructure
Provide Mission Partner Environment (MPE)	Virtual Data Center, Applications and Services, MPE Transport, Mission Partner Gateways	• Strengthen Collaboration, International Partnerships, and Allied Interoperability
Optimize Data Center Infrastructure	Data Center Optimization Initiative (DCOI) and Application Rationalization Initiative	• Optimize DoD Data Centers
Implement Consistent Cybersecurity Protections	Enterprise Perimeter Protection Capabilities, Operate Securely in the Cloud, Endpoint Security, Data Center Security, Cyber Situational Awareness Analytic Capabilities (CSAAC)/ Big Data Platform (BDP), Identity, Credential, and Access Management (ICAM)	• Transform the DoD Cybersecurity Architecture to Increase Agility and Strengthen Resilience • Deliver a DoD Enterprise Cloud Environment to Leverage Commercial Innovation • Deploy an End-to-End ICAM Infrastructure
Enhance Enterprise Mobility	Purebred for Mobile, Defense Enterprise Mobility-Classified Consolidation, DoD Mobile Application Store, Pentagon Mobility	• Improve Information Sharing to Mobile Users
Standardize IT Commodity Management	Enterprise Software Agreements, Enterprise License Agreements, Enterprise Hardware Agreements, IT Asset Management, Windows 10 SHB Fourth Estate Network Optimization	• Improve IT Category Management • Transform the DoD Cybersecurity • Architecture to Increase Agility and Strengthen Resilience
Establish End-User Enterprise Services	Enterprise Collaboration and Productivity Services	• Optimize DoD Office Productivity and Collaboration Capabilities (ECAPS Capability Set 1) • Optimize DoD Voice & Video Capabilities (ECAPS Capability Sets 2 & 3)
Provide Hybrid Cloud Computing Environments	Cloud Services	• Deliver a DoD Enterprise Cloud Environment to Leverage Commercial Innovation • Optimize DoD Office Productivity and Collaboration Capabilities (ECAPS Capability Set 1) • Optimize DoD Voice & Video Capabilities (ECAPS Capability Sets 2 & 3)

Systems Modeling Language (SysML)

The Systems Modeling Language (SysML) is a general purpose MBSE language that uses “graphical modeling for specifying, analyzing, designing, and verifying complex systems that [include] hardware, software, information, personnel, procedures, and facility elements” (Object Management Group, n.d). SysML originated from the Unified Modeling Language (UML) 2 framework. Further, SysML “provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models” (Object Management Group, n.d).

Section Summary

This section discussed problem analysis to include stakeholders, definitions, and a literature review. Definitions were introduced to familiarize the reader with MBSE and MBPS and the environment they operate within. Policies such as the ETRF and digital IT transformation explain how DoD policies affect both modeling areas. A list of stakeholders was presented that explained their functional area, their relationship to this project, and how they are impacted by this project. The literature review familiarized the capstone group with modeling efforts within SE and product support. The literature review presented an overview of definitions and applications of MBSE and MBPS. Furthermore, the literature reviewed focused on DoD specific applications of modeling, including

- The USN’s legacy process being used at the time of this capstone.
- The capability gaps of the legacy processes.
- Future DoD-specific modeling trends in MBSE and MBPS.

Model Development

System Model Development Process

The system model development process, shown in Figure 4, was developed to establish a standard procedure in developing conceptual system models early in a system or project’s life cycle. The model development process was created using the OOSEM as a guide for the phases within the process. The process begins in the problem definition and analysis phase where the problem was defined with stakeholder concurrence and analysis to determine a recommended solution. A decision is then made based on the maturity of the solution to either integrate the existing solution set, if it is mature enough, or to develop a solution if one does not exist or is too immature. For this report is assumed that the decision has already been made that an immature solution will be pursued in the local implementation of the Digital Engineering Environment (DEE).

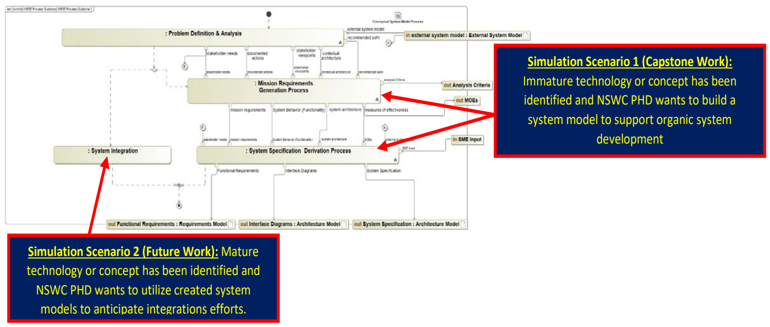


Figure 4. Overall System Model Development Process

During the model development process, the capstone team sought to meet the project objectives by utilizing the scrum framework with an iterative development approach. Team members were individually assigned diagrams through the sprint planning process. Completed diagrams were peer reviewed for content, flow, and formatting, and then were added into the master model. Periodic stakeholder reviews, including progress reviews, were conducted to gather feedback; feedback influenced model design and development to meet the stakeholder needs. Simulations of the model using a designed scenario were also performed iteratively throughout the development process to produce model data and artifacts.

Simulation Scenarios

The basis for the selection of the simulation scenarios and the corresponding activity diagrams were determined by the project objectives. Stakeholder analysis and the sponsor command objectives played a key role in the selection of the example scenarios to represent model function. The sponsor's prime objectives for in-service engineering played a key role in the selection of the following scenarios:

Addressing new business capabilities (Simulation Scenario): A new incoming business capability has been identified; or the command performs an internal audit which identifies a desired new capability. The capability set is immature, and there is not an existing system infrastructure that supports the capabilities. A system model is to be built from scratch to present conceptual information and high-level requirements of the desired solution. Post model development, the system model would be distributed to a development team for to be updated and refined as the system supporting the capability matures.

Addressing new capabilities to an existing system (Future Simulation Scenario): This scenario would focus on the addition of a capability set to an already existing system. A system model or system of systems (SoS) model exists and would be utilized to perform alternative analysis on the change prospects. Updates to the system model would happen iteratively as the change design matures and is implemented.

Activity diagrams were derived from these use cases. The pertinent activity diagrams were identified by determining the key aspects that affect the example scenarios. The activity diagrams that were modeled were:

- Problem Definition and Analysis
- Mission Requirement Generation Process
- System Requirements Generation Process
- System Integration

Object Oriented Systems Engineering Method

This capstone has utilized elements of the Object Oriented Systems Engineering Method (OOSEM) found within the practical guide to SysML. “[The] OOSEM is a top-down, scenario-driven process that uses SysML to support the analysis, specification, design, and verification of systems. The process leverages object-oriented concepts and other modeling techniques to help architect more flexible and extensible systems that can accommodate evolving technology and changing requirements” (Friedenthal et al., 2012). The OOSEM was created in 1998 and has been further refined by an International Council on Systems Engineering (INCOSE) OOSEM working group (Friedenthal et al., 2012). It is an INCOSE accepted SE management process. Most of the capstone artifacts have been captured using MBSE and SysML artifacts. These artifacts include stakeholder requirements, system requirements, problem space architecture, solution spaces architecture, use cases, and parametric diagrams. Due to the large nature of model-based artifacts, this capstone chose to employ elements of OOSEM due to its applicability in both SysML development and SysML enabled management. Figure 5 shows the OOSEM steps that helped this capstone team design a tailored process for developing a conceptual system model.

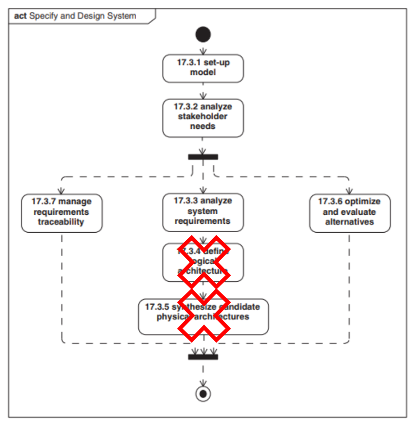


FIGURE 17.2

OOSEM *Specify* and *Design System* process. The action numbers refer to subsection where the action is described.

Figure 5. OOSEM Specify and Design Process (Friedenthal et al., 2012)

The SE steps shown in the figures are set-up model, analyze stakeholder needs, manage requirements traceability, analyze system requirements, optimize and evaluate alternatives, define logical architecture, and synthesize candidate physical architectures. This process was tailored to not include the optimize and evaluate alternatives, define logical architecture or synthesize physical architecture. These steps were removed as this capstone will not produce a full logical or physical system and would be up to the development team to refine the model to include the architecture definition. Instead, the focus will remain on developing a conceptual SysML model that describes the objectives laid out in the simulation scenario: The need of a MBSE environment that provides digital SE capabilities and can exchange meaningful data with other platforms within the digital transformation domain.

The model development utilized an iterative design process where incremental builds of the model were developed. These iterative builds incorporated a feedback loop to receive stakeholder input on the developed models. Stakeholder feedback has subsequently been incorporated into each iterative design of the model.

Problem Definition and Analysis

The problem definition and analysis phase, as shown in Figure 6, is meant to support the identification of the problem and need in a data-driven way, and to devise a solution that will help satisfy the needs. The process begins with a signal that triggers the first step in the process. The trigger can be scheduled or unscheduled, as in this process could be performed with a scheduled integrated product team (IPT) annually, every 6 months, etc., or it could be spontaneous, driven by innovation within the enterprise or based on direction provided by enterprise leadership.

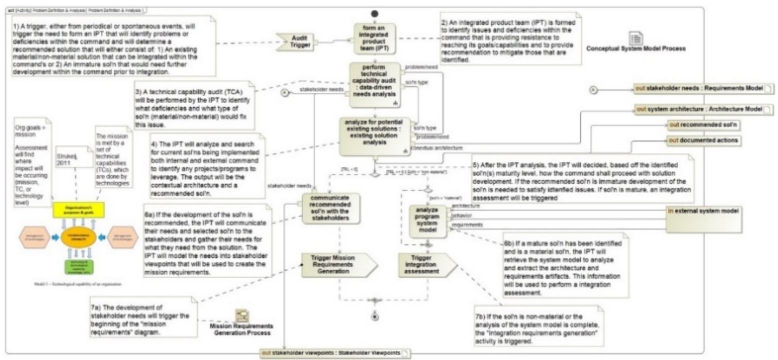


Figure 6. Problem Definition and Analysis Activity Diagram

Once the IPT is formed, their first responsibility would be to perform the technical capability audit (TCA). The TCA is the process of analyzing technical capabilities within an organization in a data-driven way to identify potential problems and solutions to those problems (Mohammadi et al., 2014, pp. 5–8). Technical capability in this context is defined as an organization's ability to utilize technologies in a way that is most useful to the organization's goals and mission. Technologies in this case refer to the machines and processes that the people of an organization utilize to perform their daily activities. Technology capabilities are influenced by technological innovation and changes in organizational goals or missions (Strukelj & Dolinsek, 2011).

The IPT develops a set of quantitative and qualitative indicators in which they can disburse to the workers of an organization to receive feedback. The indicators that form the TCA have four different aspects: hard, human, knowledge, and organizing and managing of technical capabilities. Hard aspects are the physical equipment, are the tools currently available to the workforce meet their needs. Human aspects relate to the skill set of the workforce and answers the question, "Does the workforce have the right skill set to perform this technical capability?" Knowledge aspects pertain to the understanding of the technological capability and is enough information known about it to make it a worthy investment. Lastly, organizing and managing of the technical capability is an aspect that focuses on how well an organization is structured, or funded, to develop new technical capabilities and the quality of the technical capability management process (Mohammadi et al., 2014, pp. 8–12). Feedback to the IPT from the workforce on the indicators can support the identification of problem areas where a solution is needed in order to satisfy the technical capability (Mohammadi et al., 2014, pp. 13–14).

For this example scenario, it is assumed that the TCA has already occurred and the problem has been identified to be a lack of hard aspects that is causing the greatest deficiency in achieving a MBSE technical capability at the organization. Upon completion of steps 3 and 4, as shown in Figure 6, the IPT should have completed the development of the stakeholder analysis, viewpoints and contextual architecture presentation views. An example of the stakeholder analysis is presented in Figure 7, while examples of viewpoints and contextual architecture are shown in the following section in Figures 12 and 13, respectively. For the purposes of this capstone, a formal stakeholder analysis was not performed and the Unified

Architecture Framework (UAF) provided guidance on stakeholder composition, concerns, purpose and presentation methods (OMG, 2020).

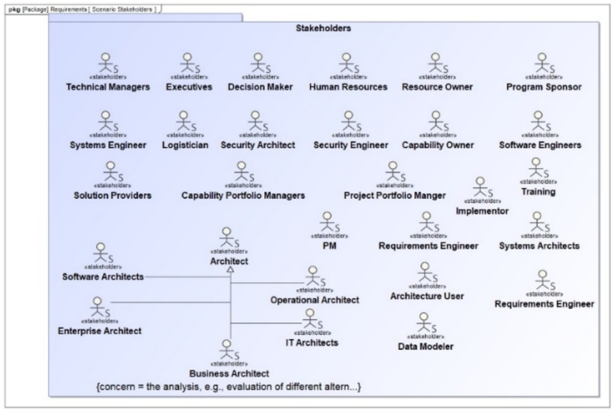


Figure 7. Analysis of the Stakeholders

Based on the maturity of the solution, the acquiring organization decided whether the system model developing process needs to begin at the beginning or a system model or artifacts exist, and the solution will be integrated into the model. For the capstone's scenario it is assumed that a new model describing the solution and the process will signal for the mission requirements generation process to begin. If the solution needs to be integrated into an already existing system model or SoS model, the model is consumed by the solution's model project and any updates and refinements are made as the solution matures.

Mission Requirements Generation

The mission requirements activity diagram in Figure 8 provides a set of top-level requirements and traceability to the other aspects of the system model to support the management of requirements throughout the completion of the process. This diagram will be explained by showing expanded views of each section to enhance readability and to summarize key actions occurring in each section.

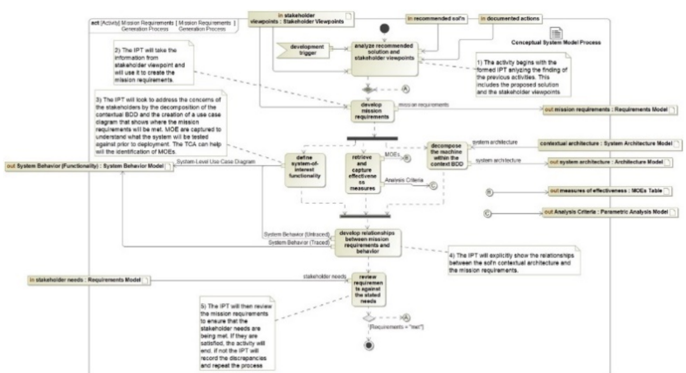


Figure 8. Mission Requirements Activity Diagram

The problem definition and analysis phase has several output object flows that are used as inputs to the mission requirement phase. The stakeholder viewpoints, recommended solution, and contextual architecture feed directly into various mission requirement blocks to further develop the system. The mission requirements diagram is the precursor to the system specification derivation process. As the precursor diagram, all outputs and generated artifacts are utilized in the system specification derivation process activity diagram.

The mission requirements generation process begins with a formed IPT analyzing the finding of the previous activities. The mission requirements phase initializes with the stakeholder viewpoints as well as the recommended solution from the problem definition and analysis phase. From the initialization, the IPT will enter a singular direction merge node, which allows for a repeat of the process should all requirements not be met. This merge node has no effect on the control flow of the process the IPT goes through from the initialization.

The control flow continues into the development of mission requirements. Mission requirements are built from the understanding of the problem and stakeholder needs that were established in the previous phase. From the development of mission requirements, the control flow then goes into a SysML fork where the IPT would perform three data collection tasks simultaneously. To exit the fork node the IPT must generate a block definition diagram (BDD) for system context, retrieve and capture measures of effectiveness, and decompose the machine within the context of the BDD.

The IPT will look to address the concerns of the stakeholders by the decomposition of the contextual BDD and the creation of the use case diagram that shows where the mission requirements will be met. The measures of effectiveness are captured to understand what the system of interest (SOI) will be tested against prior to deployment and implementation. The developed indicator from the TCA performed in the problem definition and analysis phase can be utilized to further strengthen the measures of effectiveness. From this block, the output of the BDD system context diagram is generated. This artifact is used to initialize the system specification diagram.

The last block within the fork requires the IPT to decompose the SOI within the context of the BDD. The object flow needed to complete this task is derived from the contextual architecture of the problem definition and analysis activity diagram.

With the satisfaction of the three proceeding taskings, the IPT control flow moves to join the control flows. The IPT will now be capable of defining the relationships between the solution contextual architecture and the mission requirements. As this development matures, the object flow output of a high-level system architecture transfers to the system specification derivation activity diagram.

The final logical control of the mission requirements activity diagram is to ensure that the stakeholders needs are being achieved. If gaps in requirements are identified, then the control flow allows for a repeat of the process flow for the IPT. The exit criteria for the mission requirements activity diagram is for the IPT to review the stakeholder requirements against the generated mission requirements If the stakeholder requirements are sufficiently satisfied the control flow exits the mission requirements activity diagram.

Mission requirements definition and refinement is an integral phase of the overall capability achievement of MBSE and/or MBPS within the digital engineering environment. The established object and control flows that this capstone project illustrates during the mission requirements activity diagram through the generated artifacts demonstrate the importance for an IPT to decompose and address the overall stakeholder need(s). The traceability aspect that OOSEM provides to the overall intent of the mission requirements diagram allows for further exploration of validation and verification that the system and component requirements satisfy the stakeholder requirements.

This part of the process was verified by the development of the input and output artifacts to ensure the required system model data was being produced, the following sections will discuss a selected number of these artifacts and will provide a short description pertaining to the artifacts importance to the overall presentation of the system model data.

Simulation Results

The process above describes an overall method for the second iteration of system model development. The process includes further refinement of the architecture facet of the system model, and it introduces the behavior and requirement viewpoints. In order to validate this method, scenario one was used as a use case and the system specification process was executed. The assumptions prior to moving into the process are that all required input artifacts have been completed from the previous activity diagrams. These input artifacts are displayed below as shown from the system context of the DEE and MBPS, where DEE is the SOI and MBPS is the identified external system.

Mission Requirements Generation Inputs

The first activity within the process requires the integrated product team (IPT) to revisit the information provided from the problem definition and analysis process. Other than the recommended solution, the IPT will be using the information provided in the stakeholder viewpoints as a guide to developing the different presentation views within the model. The stakeholder viewpoints represent different stakeholder perspectives and helps capture subsets of the model that are of interest to the stakeholder (Friedenthal et al., 2012). Shown in Figure 9 is the actual resources viewpoint. This viewpoint is of interest to a few different stakeholders, including the solution provider, business architect, human resources, and the systems engineer. Viewpoints capture stakeholder concerns and their preferred methods of presentation (OMG, 2020).

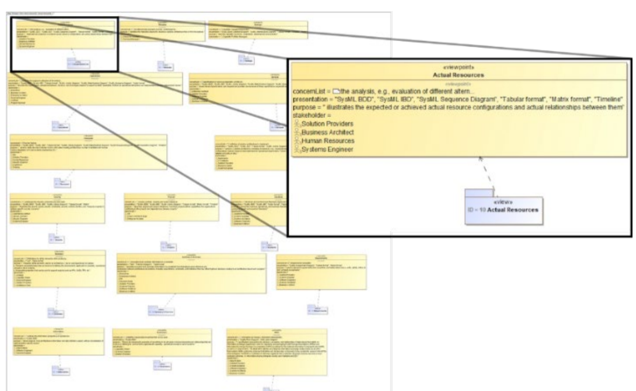


Figure 9. Stakeholder Viewpoint (Example)

Figure 10 displays another input from the problem definition and analysis phase that is used to help support the decomposition of the SOI. This artifact will define what is being decomposed, but the majority of the information needed to support the development would come from other programmatic artifacts, like a concept of operations (CONOPS), that would give the modeler a better idea of the necessary sub-systems or components needed to support the requirements for the SOI.

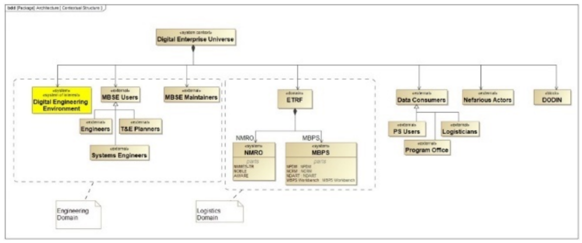


Figure 10. Contextual Architecture

Figure 11 represents an example of a set of stakeholder needs in diagram form. A diagram was chosen for this artifact, but a table is also an acceptable way for the same information to be displayed with the SysML syntax. The stakeholder requirements should always be the alignment mechanism during the development of systems and system models. SysML toolsets provide the platform for modelers to show stakeholders that their needs are being met and can provide traceable relationships to the modeled needs to ensure the designs are, in fact, meeting the modeled needs. An example of a requirement traceability matrix (RTM) is shown in Figure 12.

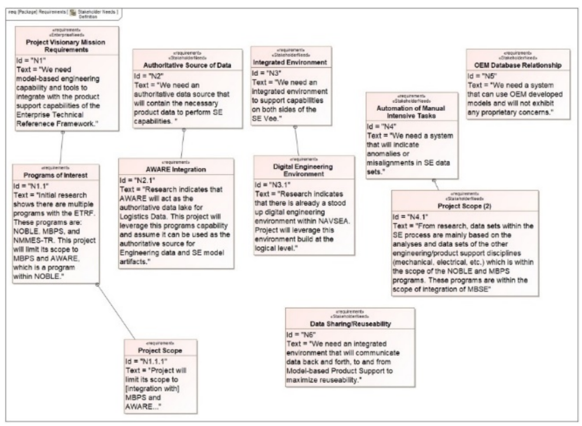


Figure 11. Stakeholder Needs Model

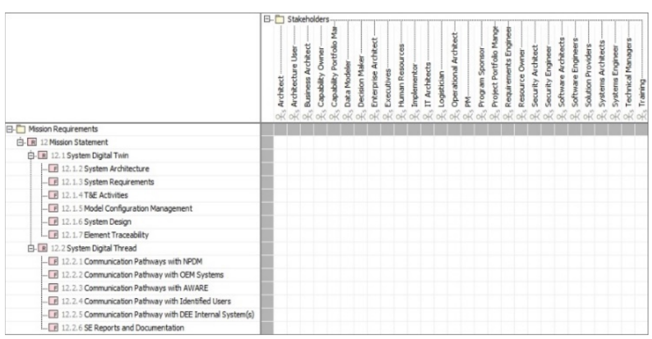


Figure 12. Example of a Requirement Traceability Matrix

Mission Requirements Generation Outputs

Measures of effectiveness (MOE) are captured in the model as shown in Figure 13. "[Measures of effectiveness] are mission-level performance requirements that reflect value to the customer and other stakeholders. They are derived from the stakeholder needs analysis that includes causal analysis and mission performance analysis" (Friedenthalet al., 2012). The MOEs help refine the black box behavior of the SOI by showing which properties and metrics are used to evaluate the system. For example, MOE 12 “required storage space” implies that the system must have a capability of storing data and that the size of the storage is important to the system final capability. The MOEs are also used in the mission requirements diagram to evaluate recommended system solutions.

#	Name	Applied Stereotype
1	bandwidth consumption	moe [Property]
2	development cost	moe [Property]
3	lifecycle sustainment cost	moe [Property]
4	past performance	moe [Property]
5	required storage space	moe [Property]
6	security factors	moe [Property]
7	software architecture	moe [Property]
8	software maturity	moe [Property]
9	staff profile	moe [Property]
10	staff turnover rate	moe [Property]
11	standard hardware compatibiliy	moe [Property]
12	system availability	moe [Property]
13	system redundency	moe [Property]
14	user capacity	moe [Property]
15	XML support	moe [Property]

Figure 13. Example MOE Table

Another function of the MOEs within the system model can be to create a criterion to which the program can base its decision-making. Shown in Figure 14 is a parametric diagram that provides an example of how parametric diagrams can be used in the design selection process. Contracting firms may submit bids to design the system laid out in Figure 14. The organization that sent out the request could use engineering analysis criteria in the parametric diagram, based off the modeled MOEs, to establish a plan for evaluating each submission. By placing a value on each MOE based on how well the contractor met that MOE, the evaluators will determine an overall score based on the selection criteria. As mission requirements have been developed within the model, the system modeler will look to begin the decomposition of the system architecture, based off the understanding of what is required of the system. The program or project is still in the very early stages in this scenario and there may be little information. Our simulation scenario from the overall process description is based off a set of known, but immature, concepts and capabilities. As shown in Figure 15, like-capabilities were grouped inside the capabilities boundary and assigned to the different capability areas, they could be called sub-systems, within our SOI. These capabilities would support the development of the top-level objectives, to create model and document artifacts that reflect the SOI.

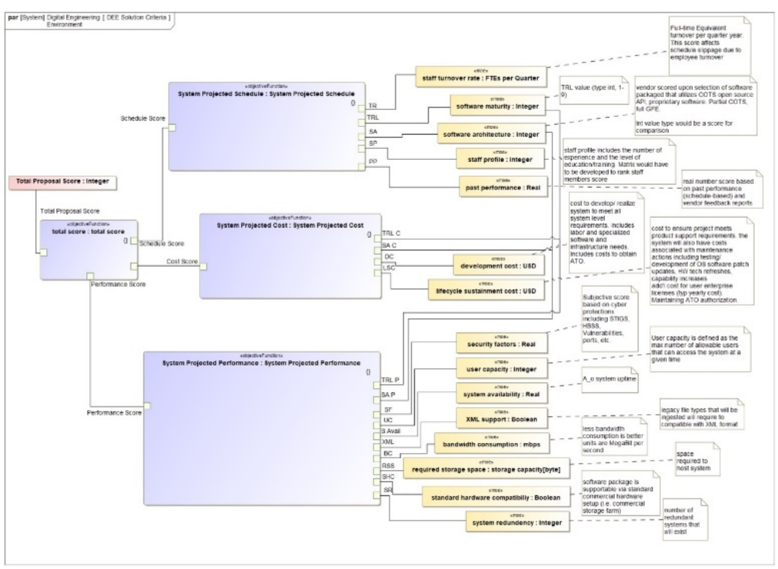


Figure 14. Engineering Analysis Criteria/Selection Criteria

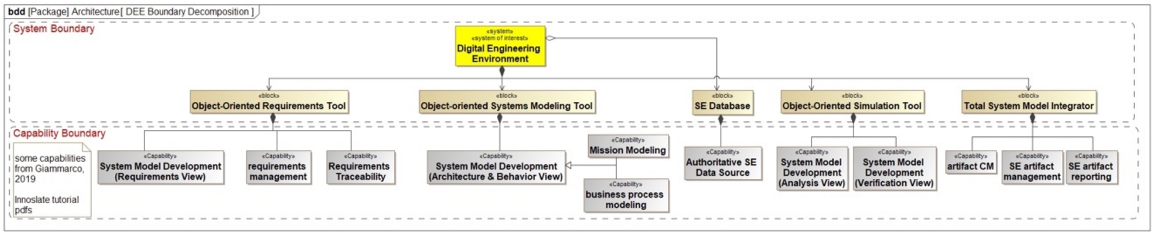


Figure 15. Boundary Decomposition

Figure 16 shows the final output and focus of the mission requirements process. The complete list of mission requirements is captured in the model and the proceeding processes use this diagram as an input at the start of the next process, system specification process.

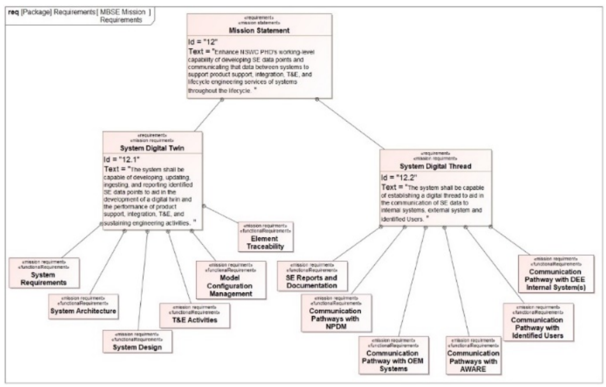


Figure 16. Mission Requirements: Requirements Model

With an understanding of the capabilities and requirements, the system modeler can begin brainstorming system use cases that will be later refined to describe behavior or be selected as a test case for system verification. Use case diagrams present the basic functionality of the system and its relation to performers or requirements. Figure 17 is the developed use case from the capstone's scenario simulation.

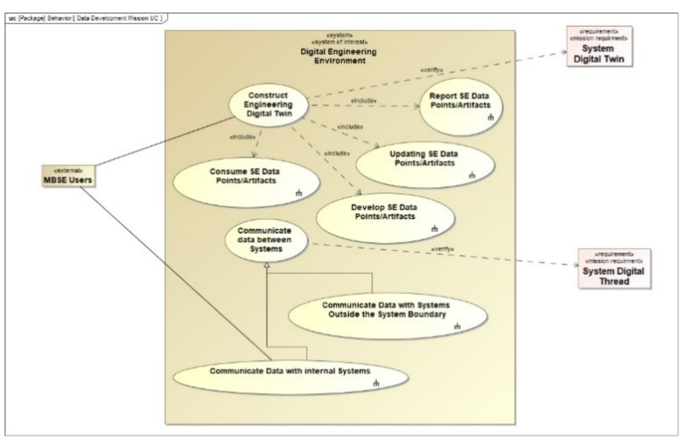


Figure 17. DEE Use Case Diagram

System Specification Derivation

The second activity that follows the mission requirements generation is the system specification derivation process. The purpose of the system specification diagram is to further mature the system model viewpoints, allowing for the further development of more mature requirements and a system specification. This activity is necessary to build an understanding of how the SOI will behave within the context of external and internal systems. Some constraints imposed on this activity flow down as inputs created during the mission requirements generation. These constraints include mission requirements and a block definition diagram (BDD) system context diagrams of the machine. The output of the activity is a system specification, an encapsulation of the SysML elements that are allocated to or share relationships with the SOI. With a clear definition of system behavior and function, a modeler and stakeholder can use the process to develop a list of functional requirements that describe what the SOI is required to do. However, this diagram does not specify how the SOI will perform its functions. This process occurs earlier in the life cycle in the conceptual system design phase. The machine specified in the diagram is the SOI for which the functional requirements are being generated. An overview of the activity is shown in Figure 18.

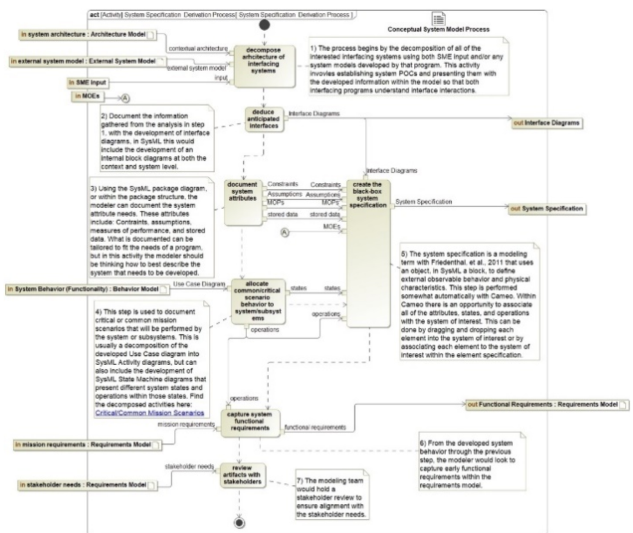


Figure 18. System Specification: Derivation Activity Diagram

The diagram inputs in Figure 18, shown on the border of the diagram are: mission requirements, system architecture, external system model, MOEs, subject matter expert (SME) input, system behavior (functionality), and stakeholder needs. Many of these artifacts were developed in the previous phases and will not be discussed further. Artifacts consumed in this process that were not developed in a previous phase will be discussed in the simulation results for the system specification derivation phase.

The first action of the process is to decompose the architecture of interfacing systems. The action focuses on defining touch points between the SOI and the external systems it will interact with. There are three inputs that facilitate this process: the architecture model, the external system model, and SME input. This first action involves searching for SMEs of external systems that can provide detailed interface diagrams and/or system models. The identified SMEs will also be presented with developed information for the SOI. In this manner, both groups will be able to identify potential points of integration and the types of data that will need to flow between the two. Once complete the next action is initiated, to deduce anticipated interfaces. In this action, the two groups will use the information found in the previous step and create an interface diagram within SysML. Internal block diagrams (IBD) of the external systems and SOI will assist in defining interfaces. Subsystems and subfunctions can help identify exact interface requirements between the systems. A model artifact is created, and the action outputs a developed interface diagram. Working through the rest of the diagram, the next actions support the development of the black box specification of the SOI. To accomplish this, system attribute needs are documented. This includes defining constraints, assumptions, measures of performance, measures of effectiveness and data requirements. By defining the attributes of the system, the black box specification can be refined to fit the constraints and needs of the system. In addition to system attributes, behavior models are created to show high level behavior based on system needs. This is accomplished by creating common mission scenarios for the SOI and designating critical/common behaviors or functions that system is expected to perform. These functions lead to the creation of behavior diagrams show interactions between subsystems previously identified in the IBD. Using all these inputs and constraints, the black box specification is developed. This can be captured as a BDD that lists model properties including constraints, parts or subsystems, properties or system functions, references, and value blocks tied model such as associated MOEs.

Lastly, the functional requirements are generated with the last two actions in the process in the functional requirements phase shown in Figure 18. The functional requirements are the main desired output artifacts of this process. and all other actions have led to its final production. This artifact is the focus of what the process is trying to create.

Using all the information from the previous steps, the functional requirements are drafted and tied to mission requirements. The mission requirement feed directly into this action to ensure that the functional requirements are derived and traced back to higher level mission requirements. A detailed list of functional requirements is generated and captured either in a requirements diagram or table. These requirements are then reviewed with stakeholder in order to receive concurrence on the final product. This review also ensures that the stakeholder needs are accurately addressed and traced to the functional requirements.

Simulation Results

The process above describes an overall method for developing the system specification and decomposing top-level requirements into functional system requirements. It is assumed that all required input artifacts have been completed from the previous activity diagrams prior to moving into the system specification process.

System Specification Process Inputs

Artifacts developed in the mission requirements generation phase and presented in the previous section are fed into the system specification process from the mission requirements generation. Mission requirements are used in the system specification process to refine and constrain system behavior and is ultimately traced directly to the functional requirements output. The system operational behavior is derived from the basic functionality expressed in the use case diagram and allocated to systems and sub-systems. As shown in Figure 17, functionality is traced to a mission requirement, enabling the support of system verification later. This analysis ensures that the system function requirements, which are generated from the behavior diagrams, are also traced back to a mission requirement.

The stakeholder needs in Figure 11 are compared against the developed functional requirements of the SOI. This is the last step in the diagram and is performed to ensure that the functional requirements align and address the previously created stakeholder needs. The mission requirements and stakeholder needs are reviewed with the stakeholder prior to finishing the process.

The BDD in Figure 19 shows the subsystems and properties of the overall external system MBPS. The MBPS system is decomposed into four subsystems: NPDM, NCRM, NDART, and MBPS workbench. Each subsystem contains parts, properties and data values. This detailed view of the external system assists in identifying potential integration points with the SOI. Figure 20 displays a free form diagram (FFD) of the six common/critical mission scenarios (functional behaviors) the black box is designed to perform. The FFD contextually allows for the presentation of various behaviors along all structured nested diagrams for exhibition. Each mission scenario has at least one decomposed diagram for further depth and relational exploration. For example, the scenario for communicate data with internal systems has three nested diagrams tied to its structure. Those diagrams are a sequencing diagram for the internal systems automatic updates, an interaction diagram for the internal systems manual update, and an interaction diagram for the internal systems save new data. One of these behavior diagrams, "updating se data points/artifacts" can be seen in Figure 21. Each functional behavior has a developed diagram as an artifact in the mode.

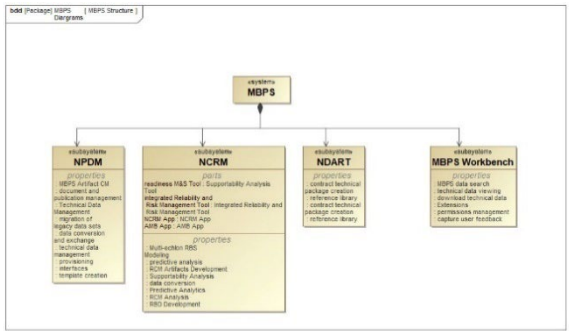


Figure 19. External System (MBPS) Model

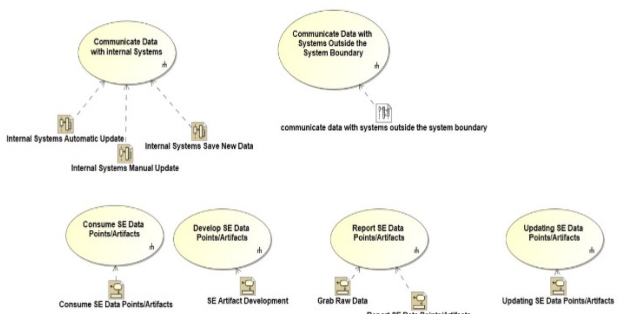


Figure 20. Mission Free Form Diagram: Critical/Common Mission Scenarios

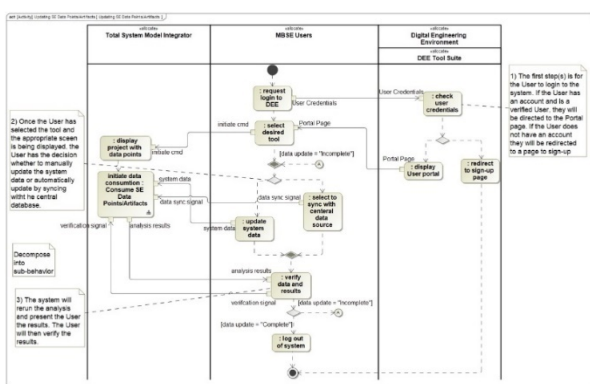


Figure 21. Behavior Diagram: Updating SE Data Points and Artifacts

Model Output Artifacts

The interface diagram is shown in Figure 22. This diagram describes various interfaces between the SOI and external systems. In this case it is showing the SOI (DEE) and how it interfaces with the three external systems: DoDIN, MBPS, and SE Database. The diagram also shows allocated subsystems where different elements, including classes and blocks, are passed.

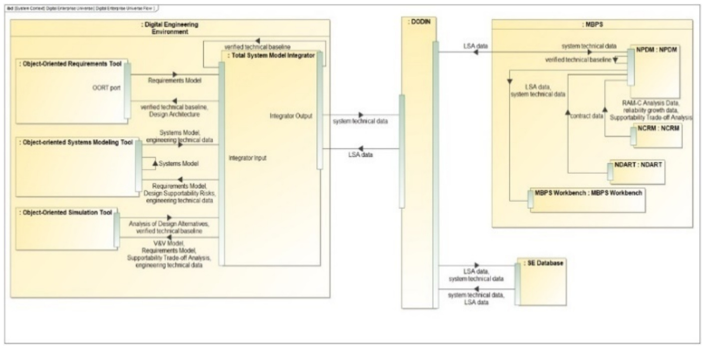


Figure 22. Interface Diagram.

The culmination of all the collected system data is shown in Figure 23 as the system specification. The system specification is an overview of the data elements contained within the SOI system model. The system specification displays the architecture information, allocated behavior, stored data elements, constraints, MOEs, MOPs, parametric information, and other related data items captured with the system model development process. The final artifact produced by this process is a list of functional requirements as shown in Figure 24. The functional requirements describe how the SOI needs to perform. When developed through the described process, these requirements can be directly traced back to mission requirements and are validated against stakeholder needs.

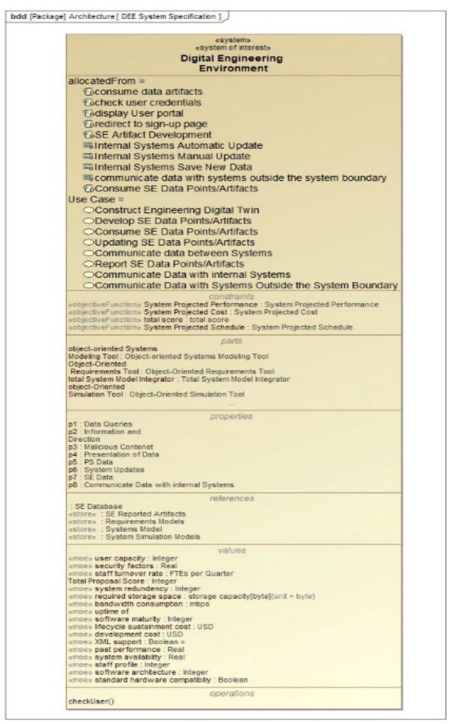


Figure 23. DEE System Specification

#	Name	Text	Satisfied By	Applied Stereotype
1	12 Mission Statement	Enhance NSWC PHD's working-level capability of developing SE data points and communicating that data between systems to support product support, integration, T&E, and lifecycle engineering services of systems throughout the lifecycle.		Requirement [Class] mission statement [Elemen
2	12.1 System Digital Twin	The system shall be capable of developing, updating, ingesting, and reporting identified SE data points to aid in the development of a digital twin and the performance of product support, integration, T&E, and sustaining engineering activities.		Requirement [Class] mission requirment [Eleme
3	12.1.2 System Architecture	The system shall be capable of developing a system architecture model to include tables and diagrams that can help describe a system's contextual, logical, and physical architecture.		mission requirment [Eleme functionalRequirement [Cl
4	12.1.3 System Requirements	The system shall be capable of developing a requirements model to include the development of requirements tables and diagrams.		mission requirment [Eleme functionalRequirement [Cl
5	12.1.4 T&E Activities	The system shall be capable of developing and storing models, diagrams, and/or documents that support a system's test & evaluation activities.		mission requirment [Eleme functionalRequirement [Cl
6	12.1.5 Model Configuration Management	The system shall be capable of performing configuration management of the models created within the system.		mission requirment [Eleme functionalRequirement [Cl
7	12.1.6 System Design	The system shall be capable of developing and storing system design models to include three-dimensional and two-dimensional computer aided models and drawings with any level of detail, from conceptual to commercial-level as described in MIL-STD-31000B.		mission requirment [Eleme functionalRequirement [Cl
8	12.1.7 Element Traceability	The system shall be capable of developing and storing relationships between requirements and all other facets of the system model to support the verification of system design.		mission requirment [Eleme functionalRequirement [Cl
9	12.2 System Digital Thread	The system shall be capable of establishing a digital thread to aid in the communication of SE data to internal systems, external system and identified Users.		Requirement [Class] mission requirment [Eleme
10	12.2.1 Communication Pathways with NPDM	The system shall be capable of exchange data between the Navy Product Data Management system of Model Based Product Support that will support the capabilities of all systems within Model Based Product Support.		mission requirment [Eleme functionalRequirement [Cl
11	12.2.2 Communication Pathway with OEM Systems	The system shall be capable of storing data incoming from orignal equipment manufacturer's to be integrated into a program's system model.		mission requirment [Eleme functionalRequirement [Cl
12	12.2.3 Communication Pathways with AWARE	The system shall be capable of communicating and sharing data with the AWARE to support AWARE's capabilities.		mission requirment [Eleme functionalRequirement [Cl
13	12.2.4 Communication Pathway with Identified Users	The system shall provide a graphical user interface where Users can interact with the system to perform all described functionality of the described system.		mission requirment [Eleme functionalRequirement [Cl
14	12.2.5 Communication Pathway with DEE Internal System(s)	The system shall be capable of sharing data between all of the system within the Digital Engineering Environment boundary that supports the capability of each system.		mission requirment [Eleme functionalRequirement [Cl
15	12.2.6 SE Reports and Documentation	The system shall be capable of developing dynamic reports that can import artifacts from a system model to be included in a template of the modeler's choosing.		mission requirment [Eleme functionalRequirement [Cl
16	N1 Project Visionary Mission Requirements	We need model-based engineering capability and tools to integrate with the product support capabilities of the Enterprise Technical Refereence Framework.		Requirement [Class] EnterpriseNeed [Class]
19	N2 Authoritative Source of Data	We need an authoritative data source that will contain the necessary product data to perform SE capabilities.		Requirement [Class] StakeholderNeed [Class]
21	N3 Integrated Environment	We need an integrated environment to support capabilities on both sides of the SE Vee.		Requirement [Class] StakeholderNeed [Class]
23	N4 Automation of Manual Intensive Tasks	We need a system that will indicate anomalies or misalignments in SE data sets.		Requirement [Class] StakeholderNeed [Class]
25	N5 OEM Database Relationship	We need a system that can use OEM developed models and will not exhibit any proprietary concerns.		Requirement [Class] StakeholderNeed [Class]
26	N6 Data Sharing/Reuseability	We need an integrated environment that will communicate data back and forth, to and from Model-based Product Support to maximize reuseability.		Requirement [Class] StakeholderNeed [Class]
27	N7 Digital Engineering Strategy Requirements			Requirement [Class]
28	N7.1 Strategic Goal 1	Formalize the Development, Integration, and Use of Model to Inform Enterprise and Program Decision Making.		Requirement [Class] StrategicObj [Class]
34	N7.2 Strategic Goal 2	Provide an Enduring, Authoritative Source of Truth.		Requirement [Class] StrategicObj [Class]
40	N7.3 Strategic Goal 3	Incorporate Technical Innovation to Improve the Engineering Practice		Requirement [Class] StrategicObj [Class]
46	N7.4 Strategic Goal 4	Establish a Support Infrastructure and Environements to Perform Activities, Collaborate, and Communicate Across Stakeholders		Requirement [Class] StrategicObj [Class]
51	N7.5 Strategic Goal 5	Transform the Culture and Workforce to Adopt and Support Digital Engineering Across the Lifecycle		Requirement [Class] StrategicObj [Class]

Figure 24. System Specification: Function Requirements

Model Summary

Three process diagrams were reviewed; each following actions are performed sequentially, which result in having documents/artifacts created that provide the necessary information to address an incoming capability. At the conclusion of these processes, the problem has been defined and analyzed, mission requirements are generated, and functional requirements are developed. All the artifacts provide a concrete strategy of what is needed to provide the command a strategy to address an incoming capability or what is known as scenario one. The stakeholders will be able to use these artifacts to clearly define a solution that details the necessary actions/steps to prepare the command for integrating a new capability.

Within each process are additional artifacts that help further document system architecture, expected behavior, parametric diagrams for analyzing the solution, and identifying interfaces between existing systems and incoming external systems. Together the models fully define the problem and an associate solution to that problem. After this point, the command will be able to start implementing the identified solution.

Section Summary

This section presented three process diagrams that describe the necessary actions to produce the required artifacts for developing the conceptual system model. The processes were explained through expanded diagrams and step by step instructions of walking through each process. Input and output artifacts were developed using a simulation scenario and summarized with provided descriptions that relate their usage within the diagram. After completing all three processes, the sponsoring command should have a clear understanding of the problem and a strategy ready for implementation to address that problem. As stated earlier, this project will not result in the creation of a physical system but will provide all information to allows for the creation of the solution.

Model Findings

Data Exchanges Between Domains

Findings on the MBPS program's capabilities shows that the program is implementing the AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association (ASD/AIA) S-Series standards to regulate the data necessary for their suite of capability. Shown in Figure 25, the logistics support analysis (LSA) data structure is the standard database and supports the other specifications (ASD/AIA, 2018).

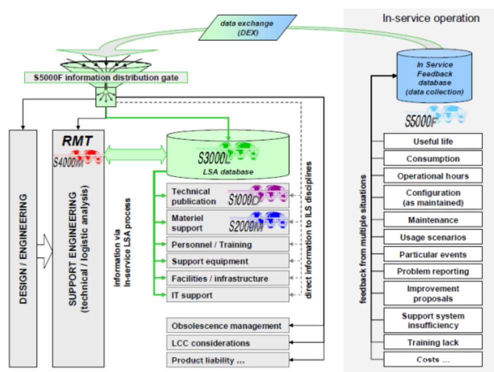


Figure 25. In-Service Data Analysis Process as an Example of S3000L Feedback. Source: ASD/AIA (2018).

The S3000L LSA database is built over the life cycle of the developed product and its development is supported by the import of engineering technical data. Figure 26 shows how the development of the database consumes and produces data for the development of the physical product. The LSA database is structured according to the S3000L extensible markup language (XML) schema presented in the standard. Therefore, any data exchanges between the database shall be supported by XML. Currently, some SysML tools support the importing and exporting of XML, but during the conversion some data, like SysML stereotypes, are lost or converted to its Unified Modeling Language (UML) equivalent (No Magic, Inc., n.d.). For example, shown in Figure 27, user capacity is stereotyped as a measure of effectiveness (MOE) within SysML. When converted to XML, the type is changed to a UML property of the Digital Engineering Environment (DEE) class, shown in Figure 28.

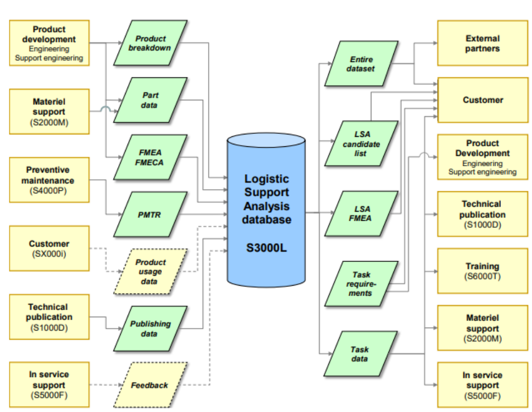


Figure 26. S3000L Data Exchanges. Source: ASD/AIA (2018).

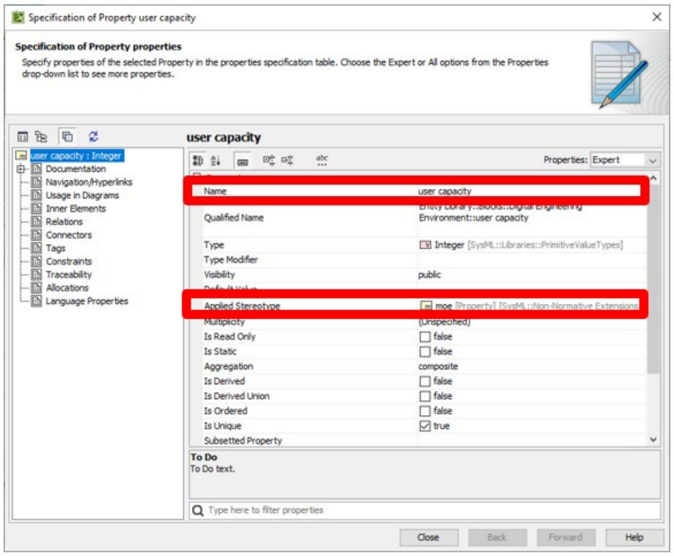


Figure 27. User Capacity MOE Specification in SysML

Figure 28. XML Data Table of User Capacity.

The S-Series specifications developed an importable XML file that contains the S-Series data model as UML classes. Instance elements, as shown in Figure 29, can be developed within a system model to create supporting data elements. Current XML exporting features only allow for a total model export. Due to this limitation, an isolated model containing the instances would be needed to ensure only required data is exchanged between systems. The creating of instances is currently a full manual process, which creates a lot of work if the system model is developed using processes that utilizes SysML and tool or process-specific stereotypes. The mapping of SysML-specific data types to the S-Series UML data model could support the creation of a translator that would drastically cut down the conversion time. Further work and research are needed to develop a data map that is able to automatically convert data from a SysML system model into elements capable of being consumed and useful within the PS domain.

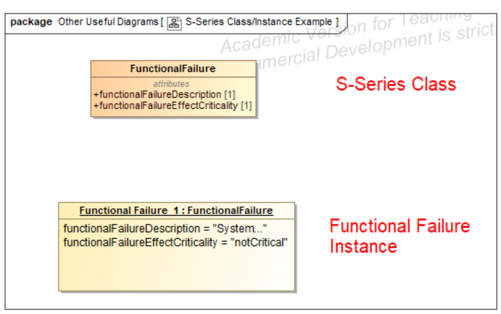


Figure 29. Example of Producing a Class Instance in a System Model with SysML.

Useful Engineering Artifact Creation

Artifacts Supporting Logistics Support Analysis (LSA) & S3000L

The LSA database interacts with the engineering community to gather engineering technical data to support the definition of the LSA database and performance of the system LSA (ASD/AIA, 2014). Shown in Figure 30, the engineering data set supports the performance of different reliability, availability, maintainability, and safety (RAM-S) analysis and reports. The data set is also stored in the database for future analysis iterations.

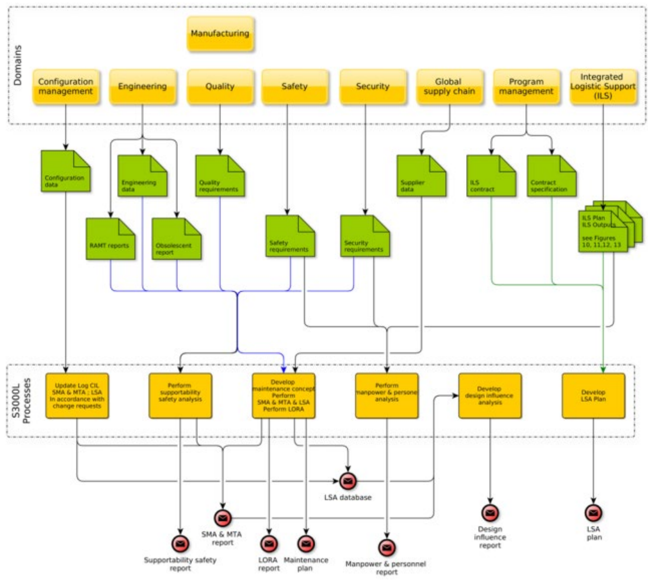


Figure 30. The Uses of Different Domain Data Sets for S3000L Processes. Source: ASD/AIA (2014).

A program’s system model is not going to contain the entirety of the required data sets. However, the data can be useful early in a program’s life cycle, when engineering drawings or three-dimensional models do not yet exist. For example, early level-of-repair analyzes are derived from the supportability failure modes and effects analysis (FMEA), which is derived from engineering inputs as shown in Figure 31 (ASD/AIA, 2014). When done correctly, a system model can be configured to output the elements required for these inputs, as shown in Figure 32 and Figure 33. Iterated over all the identified failure modes, a full FMEA can be developed in a SysML tool. Similar tables and diagrams can be created for other engineering analysis to be imported into the LSA database from the system modeling tool.

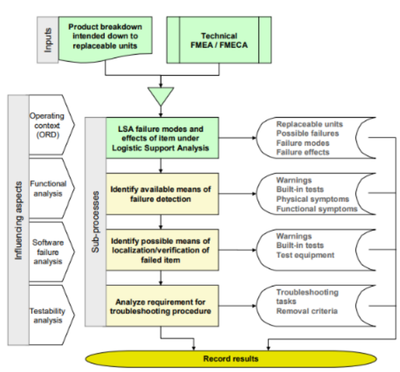


Figure 31. S3000L FMEA Development Process. Source: ASD/AIA (2014)

#	Name	Slot	Client Dependency	Dependency
1	hardwarecomponenet_1		Dependency[hardwarecomponenet_1 -> Total System Model I	Total System Model Integrator
2	FailureEffect_1	= TechnicalFailure_1 failureModeEffect = "System..."	Dependency[FailureEffect_1 -> TechnicalFailure_1]	TechnicalFailure_1 : FailureMode
3	TechnicalFailure_1	= hardwarecomponenet_1 failureModeDetectionAbilityDescription = "System..." failureModeFailureRate = "0.00001" failureModeIdentifier = "FFA1"	Dependency[TechnicalFailure_1 -> hardwarecomponenet_1]	hardwarecomponenet_1 : FailureAnalysisItem

Figure 32. Example of Functional Design FMEA within the System Modeling Tool.

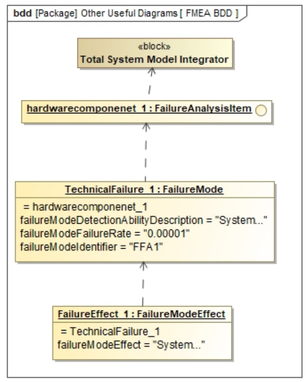


Figure 33. FMEA BDD within System Modeling Tool.

Model-Based Documentation Generation

Organizations will still require and benefit from creating documents throughout the SE process. A model-based documentation generation process can be utilized to extract model information and integrate it with current documentation templates to be supplemented with text, as shown in Figure 34. Currently, SysML tools allows for the automatic generation of reports based on an uploaded template. Once the template (*.docx file) has been configured with the correct dynamic code identifying where to find the correct model information, the user can generate reports based on that template. Shown in Figure 34, the capstone team developed a model-based document from a concept of operations (CONOPS) template using the velocity template language (VTL) to constrain which information is to be presented (Department of Veteran Affairs, n.d.). Using the stakeholder viewpoints developed early in the system model process, the modeler can present important stakeholder information in ways that is familiar and understood by the stakeholder without the need of understanding how to use and navigate through a new tool. For a command wanting to implement MBSE, it is recommended to build a library of VTL configured documents that enable the production of model-based documentation. To accomplish this, it is also recommended that a standard modeling format or a modeling style-guide be developed to enable the reuse of the model-based documents.

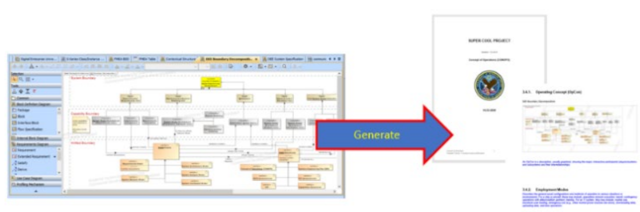


Figure 34. High-Level Concept of Generating Model-Based Documents from the System Model.

Section Summary

This section discussed pertinent findings related to the interactions between the MBSE framework and exterior environments. Utilization of the of the XML schema, as defined by the S-series specification, allows for an MBSE elements to be exported for use in alternate applications. Three instances of export use were discussed, beginning with the prospect of a direct interface between the model-based product support and digital engineering domains that can be structured to facilitate express data exchange. Second, the export of data and information from MBSE diagrams can be translated to a structure of artifacts that support S3000L LSA database entries. The creation of and/or modification to data elements would be enabled by XML data transfers. Lastly, the MBSE framework can be coupled with document templates to construct documentation utilized by traditional SE methods, such as the development of a CONOPS document using a predefined template.

Conclusions and Recommendations

A Summary of Project Objectives

This capstone object was to develop a formal process using SE methodologies to develop a conceptual system model and compile a report that explaining our development efforts, findings and conclusions from simulations and research, and recommendations for future work. This section summarizes the major findings that support the project objectives. It also includes insights that emerged and recommendations for future work.

Defining, Developing, and Importance of the System Model

The process proposed utilizes a tailored approached based on the object-oriented system engineering methodology (OOSEM) and the systems modeling language (SysML) to capture system modeling data into a system model. A proposed conceptual system data model is shown in Figure 35. The center of the system model is the SOI. The SOI of the system model acts as a piece of the digital twin, containing the architecture, behavior, requirements, and verification and validation models.

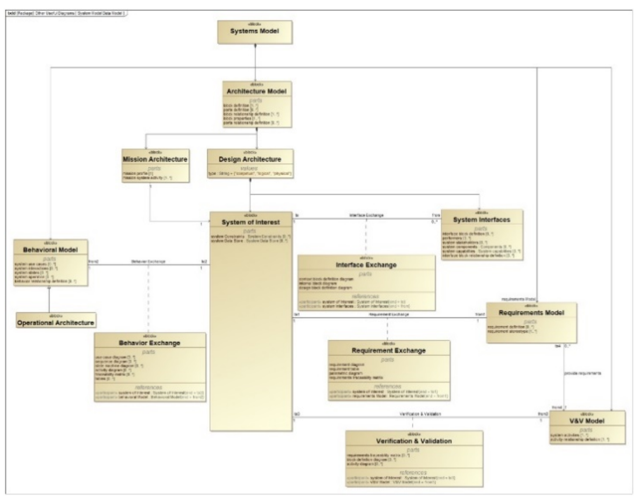


Figure 35. Proposed Conceptual System Data Model.

Over time, it would be expected that the attributes within the data model would remain constant but the level of detail of the presented information would change. For example, shown in Figure 36, activities of systems and subsystems are created at the conceptual level. Once more information about the desired capabilities of the SOI are known, the modeler can provide a logical definition to how the conceptual behavior is performed. In the selected scenario, the capability of one of the subsystems is the ability to communicate data developed within the environment to external databases. From the modeler’s understanding of the current conceptual system architecture, contextual system of systems (SoS) architecture and public information of system-to-system data exchanges a logical definition allocated to the system architecture can be formed. It would be up to the development team to further define these interactions at the physical level once the physical architecture is defined. As to the example, this would include the addition of computer coding that demonstrates how each interaction is performed. A block containing the coding information within SysML would be allocated to the signals displayed on the logical sequence diagram shown in Figure 36. Some SysML tools can auto generate a model from code developed outside of the tool, where inner model elements can then be related to different elements within the developed system model (Dassault Systems, n.d.).

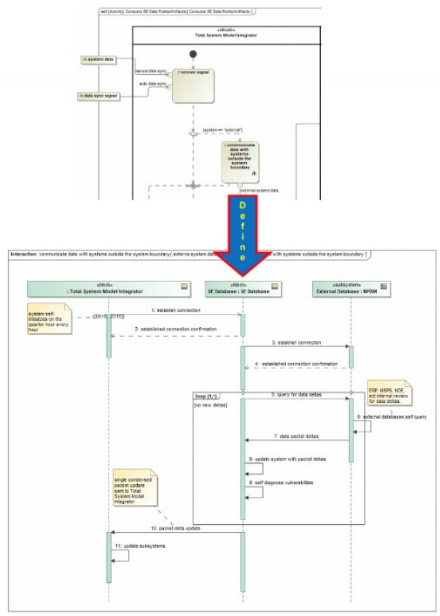


Figure 36. Transformation of a Conceptual Action into a Logical Sequence of Signals

Establishing relationships and traceability between elements within the systems model during design and development is critical for the reusability of the systems model throughout the rest of the system's life cycle (Friedenthal et al., 2012, pp. 349–352). System models developed using SysML can be used throughout system sustainment to support different changes to the system, including changes to design, mission, and maintenance procedures. For this to occur, a strong interoperability with the information technology (IT) systems in the PS domain is required.

Data captured within the system model has the capability of being transformed into a presentation graphic that could be shown to stakeholders to display the data in a way that is understandable. This capability of presenting model-critical data to decision-makers is critical to ensure the design meets expectations (DoD, 2018). Generating documents using models does not necessarily mean that the developed templates used within an organization are useless. Demonstrated in this capstone, SysML tools can utilize an organization's templates, as built, configure it to enable the document to collect model presentation artifacts, and embed them with the specified document area. Further developed could lead to auto generation of required programmatic documentation from the system model.

Conclusions

As systems are becoming more complex and more constrained, processes are going to have to become more streamlined. The MBSE stakeholders at the Naval Surface Warfare Center Port Hueneme Division (NSWC PHD) assigned the capstone group with the objectives to provide methods that would bring MBSE concepts to the command. From early research, it was determined that MBSE is early in its conceptualization with few processes being implemented across the Naval enterprise. The capstone provided a proposed workflow that was designed to be independent of a single system modeling tool and capable of developing a conceptual system model and a partial logical system model. The capstone used SysML to capture and present the modeling data, but the verbiage inside the workflow was presented in a way that another modeling language (UML, LML, etc.) could be used.

The stakeholders at NSWC PHD were also interested in learning about how an MBSE environment would integrate with another currently occurring digital transformation, the logistics IT (LOG IT) transformation. Data sharing is a major concern and an objective of the implementation of MBSE. With the current toolset and understanding of the systems within the LOG IT, out of the box data configurations would need to be translated in a suitable format in order to be usefully communicated across the domain. The MBPS program has established that their program would be setting up an LSA database based on the S3000L specification and an XML schema. Current importing and exporting capabilities in SysML limit the amount of data that can be converted and will convert all unmapped sources of data to its UML equivalent. The loss of data is not satisfactory, but information and artifacts useful to other domains could be created using instance elements within SysML and the UML classes that were developed by the S-Series specification authors. The data needed to be communicated can be exported to an isolated model, converted into an XML file, and consumed by the external MBPS system to develop analysis artifacts within its system.

Model generated documents can be utilized by programs to develop programmatic documentation from their model. A template of a CONOPS was discovered by the capstone team through the public domain, configured using VTL, and uploaded to the selected SysML tool to generate a report with the developed system model artifacts from the process simulation. Any template can be configured and uploaded if it is a supported format and could be a very useful tool to present system model data to different stakeholders.

Recommendations

It has been identified that the artifacts and findings developed from this capstone are not as mature as they could be. The developed process had completed a single simulation developed for this capstone to present potential outputs, but more research and implementation are needed to verify and validate this existing process. The process's implementation in pilot programs can help identify any unaccounted-for gaps and allowing for updates.

The process also does not consider the data developed during more detailed design efforts, including a majority of the logical and the physical architecture. The introduction of computer aided drawings (CAD), computer aided manufacturing (CAM), computer aided software engineering (CASE), finite element analysis (FEA), and other computer aided simulation artifacts could help support further definition of the system model but further research on this implementation is needed.

The conversion of instances supported in SysML to XML were mostly manual, and since the XML data format is in place to be the format of choice for existing systems, it would be of interest to look for ways to automate the data conversion and transmission. The process outlined in this capstone for conversion can support this automated process. Development of a standard system data model completed with data mappings to the S3000L XML data structure is the logical next step to automating the process. It is theorized then plug-in software or middleware could be developed that supports and automates XML conversion.

With the increase in interest of studying system of systems (SoS), SoS engineering, and SoS modeling, researching the effect of SoS concepts have on the development of a system model could be of interest to many stakeholders. Capability gaps could be produced from emerging capabilities within the SoS, signaling a need for a solution and the start to the capstone's developed process. This fact was not considered, but its effect and further iterations of the process should include research into how the implementation of SoS modeling could affect the process. The system model development process did consider that building a new system is not always the best choice and some solutions require updates or refreshes to existing systems, but the process is currently incomplete and lacks simulation results. Further development of the process to include system changes and refreshes is recommended for future project work

References

Accenture. (2019). *Enterprise technical reference framework* [Unpublished manuscript].

AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association. (2018). *International guide for the use of the S-Series Integrated Logistics Support (ILS) specifications*. http://www.sx000i.org/docs/SX000i_Issue_1_2.pdf

AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association. (2014). *International procedure specification for Logistics Support Analysis LSA*. http://www.s3000l.org/docs/S3000L-Issue%201.1.pdf

Dassault Systems. (n.d.). *Code engineering*. Retrieved October 25, 2020, from https://docs.nomagic.com/display/MD190/Code+Engineering

Department of Veteran Affairs. (n.d.). *Concept of operations (CONOPS).* Retrieved October 5, 2020, from https://www.voa.va.gov/documentview.aspx?documentid=46#:~:text=The%20Concept%20of%20Operations%20or,will%20be%20employed%20and%20supported.&text=It%20is%20used%20as%20a,project%20planning%20and%20decision%20making

DoD. (2009). *Department of Defense standard practice technical data packages* (MIL-STD-31000). http://everyspec.com/MIL-STD/MIL-STD-10000-and-Up/MIL-STD-31000A_45888/

DoD. (2018, June). *Department of Defense digital engineering strategy*. https://fas.org/man/eprint/digeng-2018.pdf

DoD. (2019). *DoD digital modernization strategy: DoD information resource management strategic plan FY 19–23*. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf

DoN. (2020). *United States Navy & Marine Corps digital systems engineering transformation strategy.* Office of the Deputy Assistant Secretary of the Navy Research, Development, Test and Evaluation. https://go.usa.gov/xfQpx

Friedenthal, S., Moore, A., & Steiner, R. (2012). *A practical guide to SysML the systems modeling language* (2nd ed.). Morgan Kaufmann.

Giachetti, R. (2020, May 15). *SEBok: Guide to the systems engineering body of knowledge*. https://www.sebokwiki.org/wiki/Digital_Engineering

INCOSE Technical Operations. (2007). *Systems engineering vision 2020, version 2.03* (INCOSE-TP-2004-004-02).

McDaniel, S. (2019, September 30). What is data-as-a-service (DaaS)? *Talend Real-Time Open Source Data Integration Software*. https://www.talend.com/resources/what-is-data-as-a-service/

MITRE. (2015, April 10). *Architectural frameworks, models, and views*. https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/architectural-frameworks-models-and-views

Mohammadi, M., Elyasi, M., & Mohseni Kiasari. M. (2014). Developing a model for technological capability assessment case of automotive parts manufacturing in Iran. *International Journal of Innovation and Technology Management*, *11*(2), 1–19. https://doi.org/10.1142/S021987701450014X

No Magic, Inc. (n.d.). *Cameo XSD import: Adding XML data models to your MagicDraw model repository.* Retrieved October 25, 2020, from https://www.nomagic.com/files/brochures/Cameo_XSD_Import_Plugin_brochure.pdf

Object Management Group. (n.d.). *What is SysML?* OMG SysML. Retrieved April 23, 2020, from https://www.omgsysml.org/what-is-sysml.htm

Object Management Group. (2013, August). *Unified profile for DoDAF and MODAF (UPDM) version 2.1.* UPDM. https://sparxsystems.com/downloads/pdf/datasheets/updm2_technology.pdf

Object Management Group. (2020, April). *Unified architecture framework (UAF) domain metamodel*. https://www.omg.org/spec/UAF/1.1/DMM/PDF

Page, K., Kwon, S., & Weinstein, K. (2018). Integrating a model-based systems engineering and model-based product support approach for affordable system sustainment. *INCOSE International Symposium, 28*(1), 1412–1419. https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2018.00557.x

SEA06L. (2019, July 18). *Model based product support (MBPS) overview* [Presentation]. In-service Navy PLM Project Kick-off. https://www.nsrp.org/wp-content/uploads/2019/08/05-MBPS_Overivew_June-2019-Updated_v5.pdf

Strukelj, P., & Dolinsek, S. (2011). 3-Level modeling of organization's technological capability. *Proceeding of PICMET '11: Technology Management in the Energy Smart World (PICMET)*, 1–10. https://ieeexplore.ieee.org/abstract/document/6017617

PANEL 19. MANAGING RISK IN SOFTWARE DEVELOPMENT AND ACQUISITION

Thursday, May 13, 2021	
9:15 a.m. – 10:30 a.m.	**Chair: Lieutenant General David Bassett, USA,** Director, Defense Contract Management Agency ***A Complexity-based Approach to the Measurement of Technical Risk in Defense Acquisition*** Antonio Pugliese, Cornell University Roshanak Nilchiani, Stevens Institute of Technology ***Addressing Software-Based, Platform Interoperability Risks by using Distressed Debt Financial Strategies*** Ann Gallenson, Naval Postgraduate School Scot Miller, Naval Postgraduate School Susan Higgins, Naval Postgraduate School ***Making the Kessel Run: Re-Insourcing Software Development in the U.S. Air Force*** Bryan Hudgens, Naval Postgraduate School Nathan Taylor, U.S. Air Force Robert Hollister, U.S. Air Force Kathryn Aten, Naval Postgraduate School Jenny Around, U.S. Air Force

Lieutenant General David Bassett, USA—is the director of the Defense Contract Management Agency, headquartered at Fort Lee, Virginia. As the director, he leads a Department of Defense agency consisting of more than 12,000 civilians and military personnel who manage more than 300,000 contracts, performed at more than 10,000 locations worldwide, with a total value in excess of $5 trillion.

Bassett assumed leadership of DCMA on June 4, 2020. He came to the agency after serving as Program Executive Officer for Command, Control and Communications-Tactical (PEO C3T) since January 2018, where he was responsible for the development, acquisition, fielding and support of the Army's tactical network, a critical modernization priority.

Bassett was commissioned into the Signal Corps in 1988 through ROTC concurrent with a Bachelor of Science in Electrical Engineering from the University of Virginia. As a junior officer, he served in Germany in tactical positions with the 2nd Armored Cavalry Regiment and 123rd Signal Battalion, 3rd Infantry Division.

Following the Signal Officer's Advanced Course and completion of a Master of Science in Computer Science through the University of Virginia, Bassett was assigned to the U.S. European Command Staff, where he served as the Requirements Analysis and Interoperability Action Officer, J6.

He transferred to the Army Acquisition Corps in 1999 and was assigned to Fort Monmouth, New Jersey, as Operations Officer, Communications and Electronics Command Software Engineering Center. Bassett went on to manage software development efforts for the Army's Future

Combat Systems program. He then served on the Joint Staff as the Ground Maneuver Analyst, Capabilities and Acquisition Division, J8.

From July 2009 to May 2012, Bassett served as the Army's Project Manager for Tactical Vehicles within the Program Executive Office for Combat Support & Combat Service Support (PEO CS&CSS). He then managed the Joint Program Office, Joint Light Tactical Vehicles (JLTV), through the Engineering and Manufacturing Development award.

In September 2013, Bassett was appointed Program Executive Officer, Ground Combat Systems, where he managed the portfolio of the Army's combat vehicle fleet including major modernization efforts to Abrams, Bradley, Stryker and self-propelled howitzer programs while also initiating the Army's Armored-Multi Purpose Vehicle program. Previous he served as Deputy Program Executive Officer for CS&CSS.

Bassett is a graduate of the Army Command and General Staff College at Fort Leavenworth, Kansas, and a distinguished graduate of the Industrial College of the Armed Forces in Washington, D.C.

Structural Complexity Analysis to Evaluate Technical Risk in Defense Acquisition

Dr. Antonio Pugliese—Cornell University [pugliese@cornell.edu]

Dr. Roshanak Nilchiani—Stevens Institute of Technology [rnilchia@stevens.edu

Maximilian Vierlboeck—Stevens Institute of Technology [mvierlbo@stevens.edu]

Abstract

The study of "-ilities" in systems engineering has been fundamentally connected to the evaluation of system complexity in recent years. Complexity has been inherent in all defense acquisition programs where technology and human organizations interface. Complexity can be inherent in design of a defense system/System-of-systems, at the organizational layers of defense systems, and in the environment, every now and then imposing its unpredictability or non-linearity to an acquisition program. Increased knowledge and understanding of defense systems complexity can shed light on some various unknown and emergent behavior of such systems, as well as guiding us to better solution sets when facing major decisions or challenges.

The goal of our research is to identify, formulate, and model complexity in technical segment of defense acquisition programs, as the increased level of complexity contributes to increased fragility and potential failure of the system. In another word, complexity measure is an indirect measure of risk in complex systems. The future direction of our research aims at replacing a large portion of subject matter experts' opinions on technical systems risk assessment, with actual complex risk measures and therefore improve the decision-making process more objective.

Introduction

Defense acquisition programs are essential and fundamental to the goals of the United States in terms of defense and peace-keeping activities. The 2016 report on Performance of the Defense Acquisition System states that long-time issues such as large cost growth, heavy changes in requirements, and responsiveness in initiating new programs, which have been addressed in years of research in acquisition management, are now under control (Kendall, 2016). The same report warns future leaders to not neglect system "-ilities" when evaluating a system, claiming that well-engineered systems are more often effective. Reliability, availability, and maintainability are prerequisites to the system performing its function (Kendall, 2016).

The study of "-ilities" in systems engineering has been fundamentally connected to the evaluation of system complexity in recent years (Enos et al., 2019; Fischi et al., 2017; Pugliese et al., 2018; Pugliese & Nilchiani, 2017; Salado & Nilchiani, 2013). Complexity has been inherent to defense acquisition programs where technology and human organizations interface. Complexity can be inherent to design of a defense system/system-of-systems, at the organizational layers of defense systems, and in the environment, occasionally imposing its unpredictability or non-linearity to an acquisition program. System "-ilities" such as flexibility, reliability, modularity, etc. are most successful when they are embedded in large-scale programs where a fundamental understanding of the complex structure and behavior of such systems exists. Therefore, it is necessary and urgent to better understand, model, measure, and formulate such defense programs considering their complex behavior. Increased knowledge and understanding of defense systems complexity can shed light on various unknown and emergent behavior of such systems, as well as guide us to better solution sets when facing major decisions or challenges.

The goal of our research is to identify, formulate, and model complexity in technical segments of defense acquisition programs, as the heightened level of complexity contributes to increased fragility and potential failure of the system. In other words, complexity measure is an indirect measure of risk in complex systems. The future direction of our research aims at replacing a large portion of subject matter experts' opinions on technical systems risk assessment with actual complex risk measures and therefore improve the decision-making process by enabling it to be more objective.

In software systems, complexity can be defined as "a measure of the resources expended by a system while interacting with a piece of software to perform a given task" (Basili, 1980). From this general definition many can be derived depending on the choice of the specific system interacting with the software under study (Mens, 2016). If the interacting system is a computer, we are looking at theoretical complexity, which can be of two types: algorithmic complexity, if the focus is on the time and storage space required to execute the computation, or computational complexity, if the focus is on the complexity of the problem at hand, regardless of the algorithm used to solve it. Efficient algorithms will have an algorithmic complexity that is close to the computational complexity of the problem at hand (Mens, 2016). If the interacting system is the user of the software system, then the corresponding complexity is complexity of use, usually referred to as a common system characteristic: usability (Mens, 2016). If the interacting system is a software developer, the type of complexity is structural complexity (Darcy et al., 2005).

Software structural complexity focuses on the software architecture, defined as the organization of the components of the software and how they relate to each other. A structural complexity analysis is performed by looking at the source code of the software under study, and is therefore dependent on the programming language and on a specific implementation of the solution. Depending on the level of granularity at which the software is analyzed, this static analysis, as it is also known among computer scientists, can consider as atomic units of the system the modules or files, inner constructs, such as classes and functions, or single instructions. A finer level of granularity can lead to a more detailed understanding of the dependencies, but requires the software to be completed before this analysis can be carried out.

Literature Review and State of the Research

When looking at software architecture (SA) in its general form and where the architectural aspects originated from, the history shows that the first approaches that are now all combined in SA can be traced back all the way to the early 1970s. Especially over the last 30 years, software architecture emerged as an important field for both research and practice (Shahin et al., 2014). On a general level, SA can be defined as the representation and definition of software and the software system. Such a representation includes descriptive elements which cover the relationships between elements and sub-elements (Angelov et al., 2009; Avci et al., 2020; Garlan & Shaw, 1993).

Early on in the 1960s and 1970s, research emerged that addressed data and data structures, which lead to an accentuation of certain structural elements above the level of the software code itself. This accentuation led to an abstraction and organizational understanding, and as a result, software architecture emerged in the following decades (Garlan & Shaw, 1993). The first appearances and mentions of SA can be found in the publication of Parnas in 1972. In this work, the author described the concept behind the module decomposition structure. Specifically, Parnas describes criteria that can be used to decompose the structure of systems into modules. Throughout the 1970s, Parnas published various other papers that outlined additional aspects of structures, and over time, the field of SA progressed and more nuances were added to differentiate between various forms of structures (Bass et al., 2012).

From the aforementioned time till around 1990, architecture in scientific fields was mostly related to systems (Kruchten et al., 2006). Yet, SA as a separate discipline in research and science emerged in the 1990s (Kruchten et al., 2006; Perry & Wolf, 2000) and has been flourishing since then, also including empirical research approaches (Qureshi et al., 2013). The first book about SA was also published during these beginning times in 1994 (Witt et al., 1994).

Because of the pace increase, numerous approaches were developed in the 1990s in academia but also by companies, such as Lockheed Martin and IBM. Kruchten et al. (2006) lists various approaches that resulted from these efforts: Software Architecture Analysis Method (Kazman et al., 1994), the 4+1 view (Kruchten, 1995), Siemens's four views (Soni et al., 1995), and numerous other patterns that address the design of SA (Buschmann et al., 1996) as well as Architecture Description Languages (ADLs; Shaw & Clements, 2006).

Building upon the momentum, more companies started to participate in SA and its methodologies since the beginning of the third millennium. Two notable approaches for general architecture were standardized to unify certain efforts: RM-ODP (ISO/IEC, 1995; Linington, 1995; Putman, 2000) and IEEE 1471 (IEEE, 2000). Overall, a lot of pre-made platforms and architectures ready to use have been developed and are today available. Open-source software adds to this abundance. It is thus safe to say that SA has reached what Shaw and Clements (2006) describe as "popularization." Therefore, new trends and explorations also must be considered since they are a natural continuation of the described state.

Looking at the last 5 years, a few trends in SA emerge that are currently being pursued. The first of these trends is cloud and service related and addresses the question how SA is connected to such fields and how it can be utilized (Amal et al., 2018; Bahsoon et al., 2017; Hästbacka et al., 2019; Malavolta & Capilla, 2017). Second, a focus on intelligent architecture can be seen, which introduces topics such as machine learning into the field of SA and enables phenomena such as emergent architectures that only appear during runtime and are not pre-managed or set (Woods, 2016). This trend also increases the reliance of SA on data and algorithms, which will require rethinking of previously mentioned approaches, such as the 4+1 View, which did not originally include any views for data or underlying information (Kruchten, 1995; Woods, 2016). Third, also related to the previous one, the use of SA in agile environments has become more and more important and has thus moved into the focus of research as well (Dingsøyr et al., 2018; Venters et al., 2018). Agile and SA propose different viewpoints with the former advocating for flexible as well as iterative implementation of changes and the latter standing for fundamental decisions that might even be deferred until they can be made in the most informed manner if they are not already defined up front (Dingsøyr et al., 2018; Hasselbring, 2018). Hence, the integration of architecture into agile environments has been seen as a trend as well (Dingsøyr et al., 2018). Lastly, a focus on sustainability also in relation to longevity and scalability can be seen. Since scalability can be an issue with integrated databases due to their high coherence (Hasselbring, 2002), the applicability and longevity of SAs can become problematic if they are tightly vertically integrated. Thus, approaches such as Microservices (Francesco et al., 2017; Newman, 2015; Taibi et al., 2017) and other solutions to these problems (Capilla et al., 2017), which then also address sustainability (Cabot et al., 2019; Venters et al., 2018), are being pursued.

Lastly, for the research at hand, a categorization approach and characterization within SA is critical to allow for a methodological analysis. Thus, the most frequently used and applied structures were researched and are described hereinafter. On an overarching level, structures in SA can be seen as threefold (Bass et al., 2012): decomposition structure, use structure, and class structure. Each of these three categories can again be subdivided into more nuanced categories, but such detailed subdivisions can be strongly dependent on the case of application. Thus, for the work at hand, three of the subcategories of the module structure shall be outlined as they are directly related to the research presented as depicted in Figure 1: decomposition structure, use structure, and class structure.

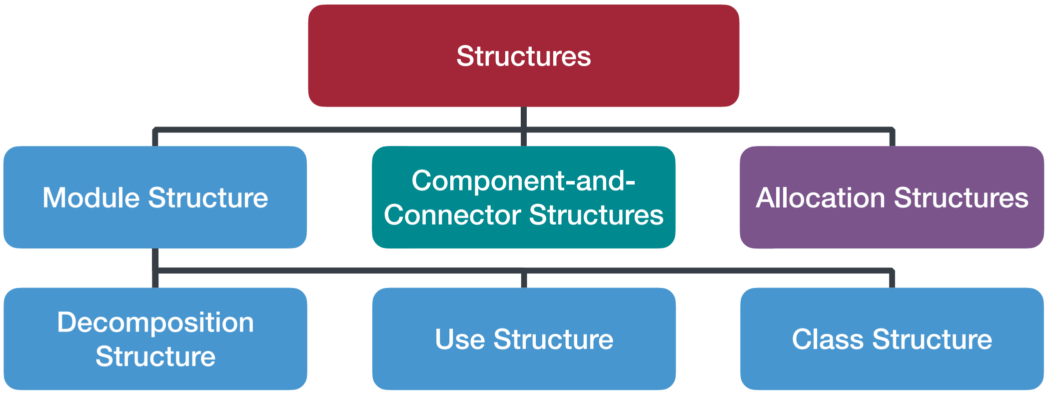


Figure 1. Classification of Relevant Architectural Structures for Software Systems

In this paper we analyze the source code of an open-source Python library, Snorkel. This is a static analysis that focuses on the module structure. In particular, the codebase is parsed to generate a class structure, which includes details about modules, classes, and methods. A series of relationships between these entities allow us to define a particular case of a use structure, which will be used as the basis of the static analysis.

Methodology

This paper presents a static analysis of the source code of a software package developed using the Python 3 programming language. The source code is parsed using the Abstract Syntax Tree (AST) module in the Python Standard Library. This module is based on the parser used in the native Python compiler and is continuously updated with any grammar change in the language. This parsing process leads to the creation of a graph where functions and classes are nodes and inheritance and functional calls are edges.

The resulting graph is known as a *module dependency graph* and has been a subject of a number of graph-theoretical research efforts (MacCormack et al., 2006). The module dependency graph is a particular case of a use structure. In this research, the module dependency graph will be analyzed with a series of complexity metrics based on the eigenvalues of various representations of the graph (Pugliese & Nilchiani, 2019). These metrics are based on other metrics, such as graph energy (Gutman, 2001) and natural connectivity (Jun et al., 2010).

The module dependency graph is built using an ad hoc model of Python objects and interdependencies. This version introduces function-level granularity, from file-level of the previous one, and is based on the Python AST module instead of simply parsing the code. The graph is built using the following rules:

- A file that imports code from another file is dependent on that file.
- A class that inherits from another class is dependent on that class.
- A function that calls another function is dependent on that function.
- A file that contains a class is dependent on that class.
- A file that contains a function is dependent on that function.
- A class that contains a function is dependent on that function.

Figure 2 shows the types of dependencies among the elements of the graph.

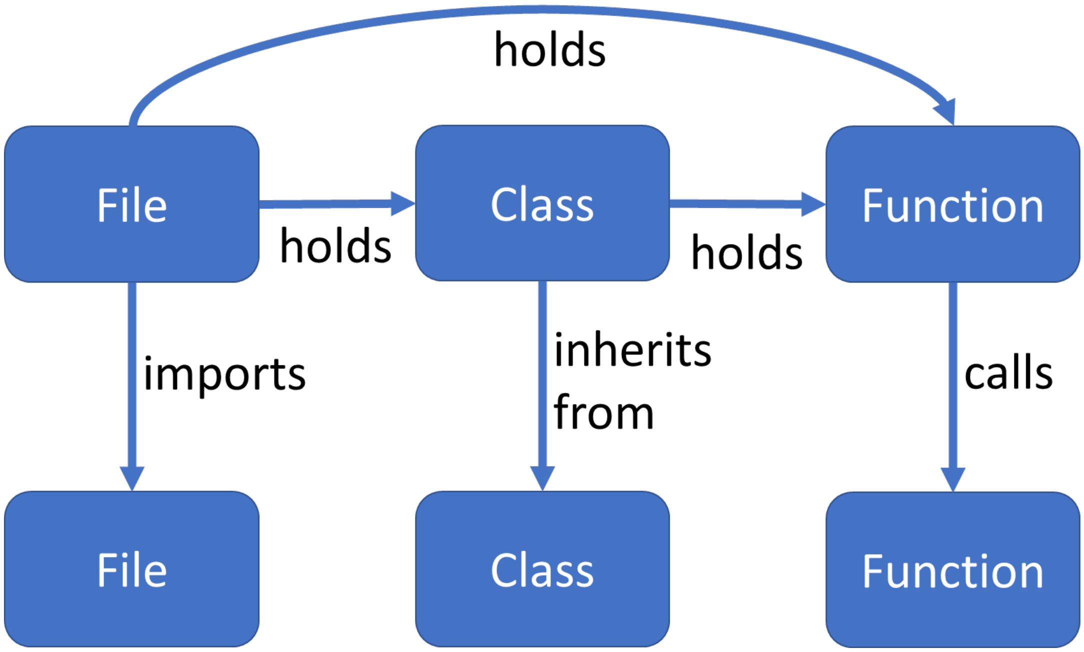


Figure 2. Types of Dependencies Among Graph Elements

The analysis of the module dependency graph is carried out using a set of spectral complexity metrics developed by our research group and represented using the following formula:

$$C(S) = f\left(\gamma \sum_{i=1}^{n} g\left(\lambda_i(M) - \frac{tr(M)}{n}\right)\right)$$

where $f_1(x) = x, g_1(y) = |y|, f_2(x) = \ln x\,, g_2(y) = e^y$ are the possible values for the functions f and g, the coefficient γ can be $\gamma_1 = 1, \gamma_2 = n^{-1}$, and the matrix representation of the graph can be either $M_1 = A, M_2 = L, M_3 = \mathcal{L}$, which have been defined in our previous publication (Nilchiani & Pugliese, 2016).

Table 1 shows the metrics that can be derived from this formula through combinations of the described parameters. Two sets of functions, two values for the coefficient γ, and three matrices yield 12 possible metrics. Throughout this paper, the metrics are referred to using acronyms: graph energy (GE), Laplacian graph energy (LGE), normalized Laplacian graph energy (NLGE), natural connectivity (NC), Laplacian natural connectivity (LNC), normalized Laplacian natural connectivity (NLNC). Where the acronym has a trailing n, such as in (GEn), the factor $\gamma = 1/n$.

Table 1. Twelve Examples of Spectral Structural Complexity Metrics

	Adjacency Matrix	**Laplacian Matrix**	**Normalized Laplacian Matrix**
$\boldsymbol{\gamma = 1}$	$GE = \sum_{i=1}^{n} \lvert\lambda_i\rvert$	$LGE = \sum_{i=1}^{n} \left\lvert\mu_i - \frac{2m}{n}\right\rvert$	$NLGE = \sum_{i=1}^{n} \lvert\nu_i - 1\rvert$
	$NC = \ln\left(\sum_{i=1}^{n} e^{\lambda_i}\right)$	$LNC = \ln\left(\sum_{i=1}^{n} e^{\mu_i - \frac{2m}{n}}\right)$	$NLNC = \ln\left(\sum_{i=1}^{n} e^{\nu_i - 1}\right)$
$\boldsymbol{\gamma = \frac{1}{n}}$	$GEn = \frac{1}{n}\sum_{i=1}^{n} \lvert\lambda_i\rvert$	$LGEn = \frac{1}{n}\sum_{i=1}^{n} \left\lvert\mu_i - \frac{2m}{n}\right\rvert$	$NLGEn = \frac{1}{n}\sum_{i=1}^{n} \lvert\nu_i - 1\rvert$
	$NCn = \ln\left(\frac{1}{n}\sum^{n} e^{\lambda_i}\right)$	$LNCn = \ln\left(\frac{1}{n}\sum^{n} e^{\mu_i - \frac{2m}{n}}\right)$	$NLGEn = \ln\left(\frac{1}{n}\sum^{n} e^{\nu_i - 1}\right)$

Results

This section presents the results of analysis on the module dependency graph for the Snorkel project published on GitHub. The project was selected due to its relatively small size of ~2,600 commits and less than 300MB of code as of March 2021, which allows us to run our analytical programs on a laptop. The number of contributors (50), the history of commits, and the prevalence of Python code were other attributes that affected this choice. Future and optimized versions of the code will aim at analyzing larger codebases.

The evolution of the graph at indicated time stamps is depicted in Figure 3. In these plots, the nodes are colored according to their type: file (blue), library (black), class (red), and function/method (green). These images suggest how even a relatively small project, such as Snorkel, can become eminently complex to manage and architect.

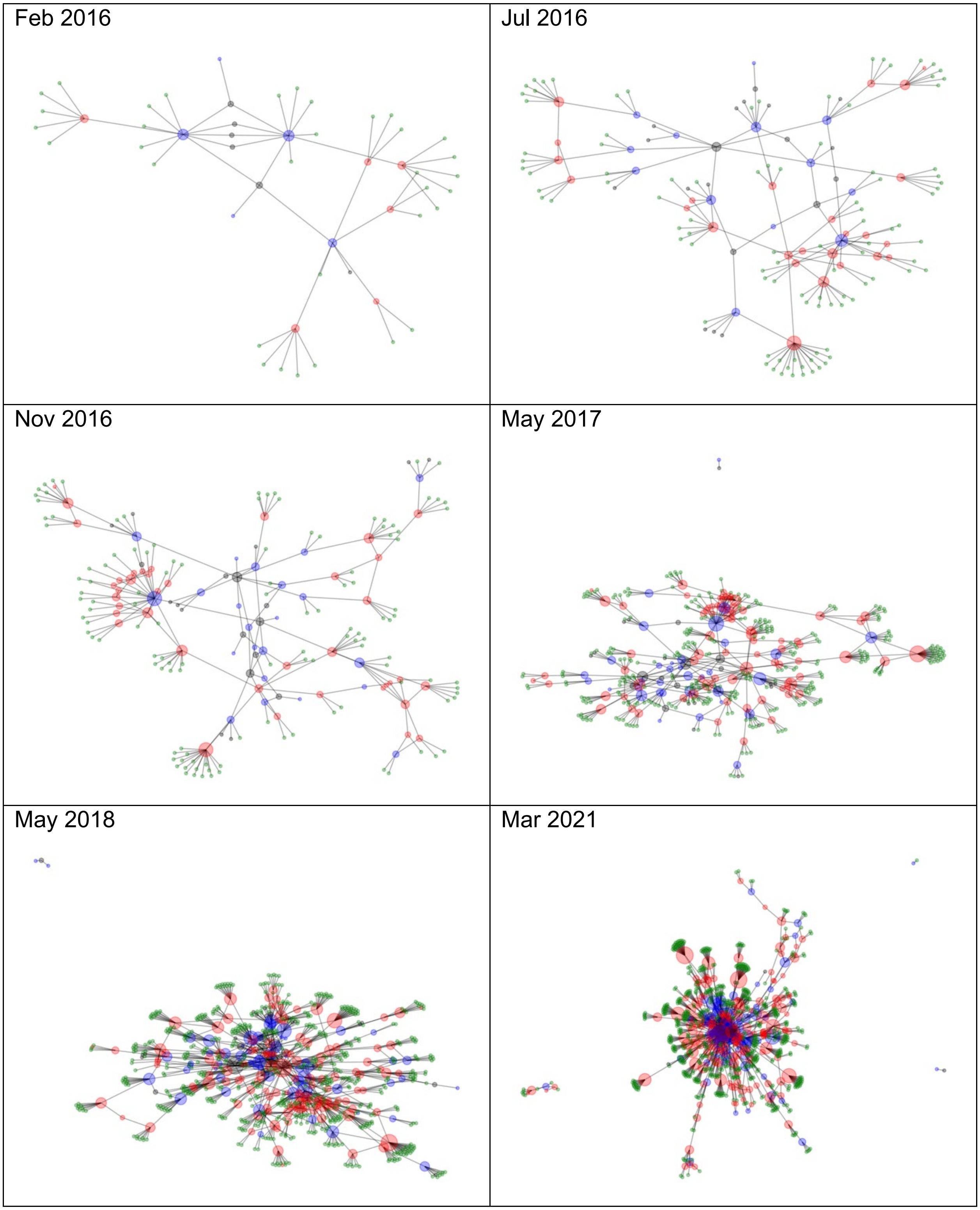


Figure 3. Evolution of the Module Dependency Graph at Select Points in Time for the Snorkel Project.
Snapshots are taken at intervals of approximately 530 commits.

Linear Correlation Analysis

A linear correlation analysis of the metrics is described hereinafter. Using the Pearson correlation coefficient (r), it is possible to see if any of the metrics evaluated for the dependency graph are linearly co-dependent. These dependencies can provide insights regarding characteristics of the Snorkel code base.

As shown in Figure 4, the following group of metrics show $r > .99$ in all pairwise comparisons: GE, LGE, NLGE, n, m.

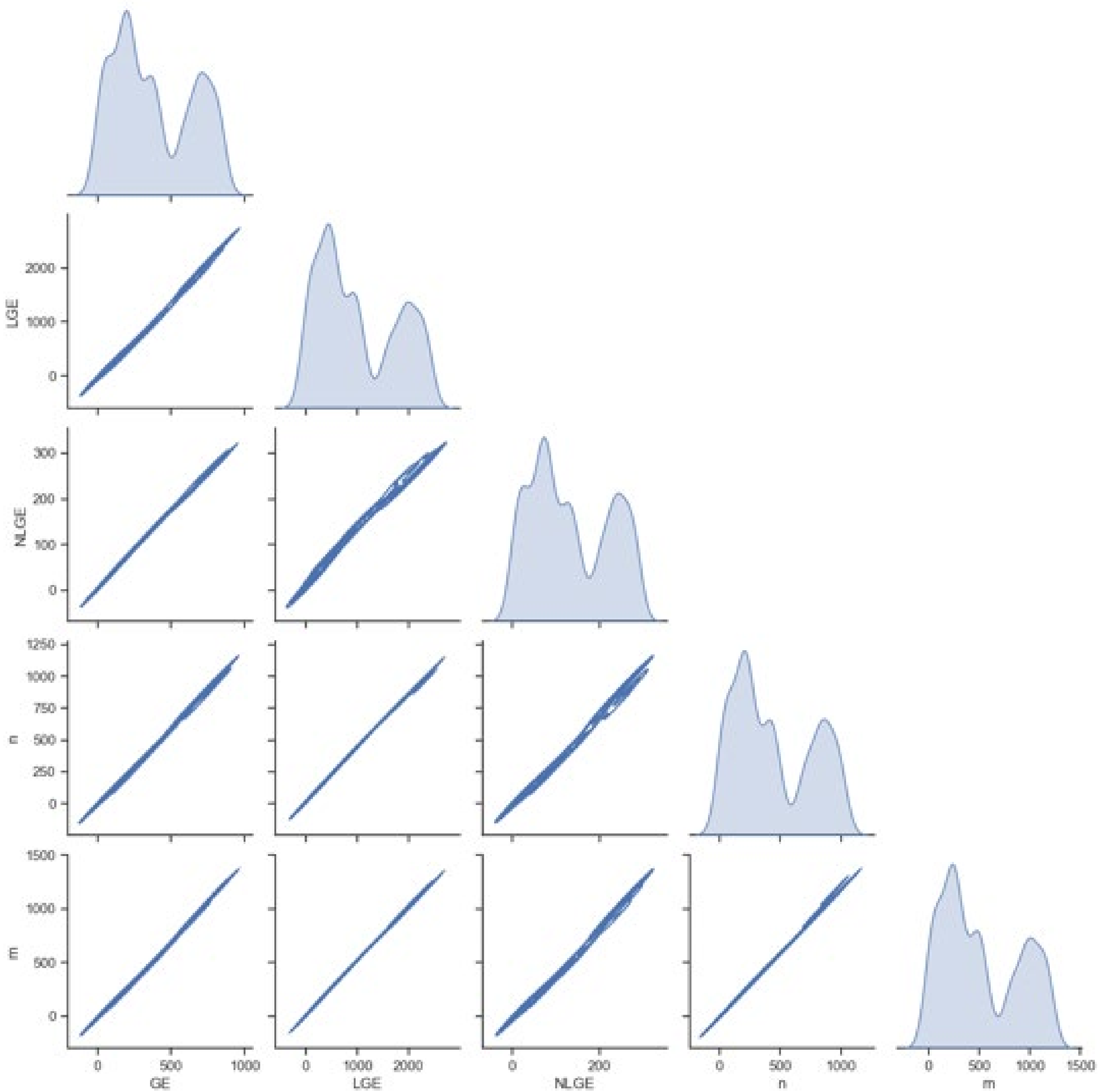


Figure 4. Comparison of GE, LGE, NLGE, Number of Nodes, and Number of Edges

The linearity between number of nodes (n) and number of edges (m) can be seen as a symptom of localized development. The addition of a module to the source code is followed by the connection of this module to one or more others. If for each additional module a low number of connections are made, it means that the module is only being used in that specific part of the code. While a percentage of additions are justifiably of this type, most modules might also be reused in other locations and therefore should create more additional connections. A long-lasting linear relationship between n and m suggests a need for refactoring.

The linear relationship between GE and LGE is common in graphs with a close to uniform distribution of node degrees. In star graphs, GE would grow superlinearly with the number of nodes while LGE's behavior would converge to linear. The dissimilarity between the current dependency graphs and graphs with highly skewed distribution of node degrees is also seen in NLGE, which would be zero for star graphs.

Figure 5 shows a linear relationship ($r > .99$) in three pairwise comparisons between LNC, LNCn, and the maximum node degree. A linearity between LNC and LNCn is a characteristic of star graphs and wheel graphs. For graphs with more uniform degree distribution, the value of LNCn plateaus quickly with the number of nodes, while LNC's growth slows down more gently. This result is in contrast with the insights found in Figure 4, and adds a new research question regarding the relationship between these metrics and fundamental graph characteristics.

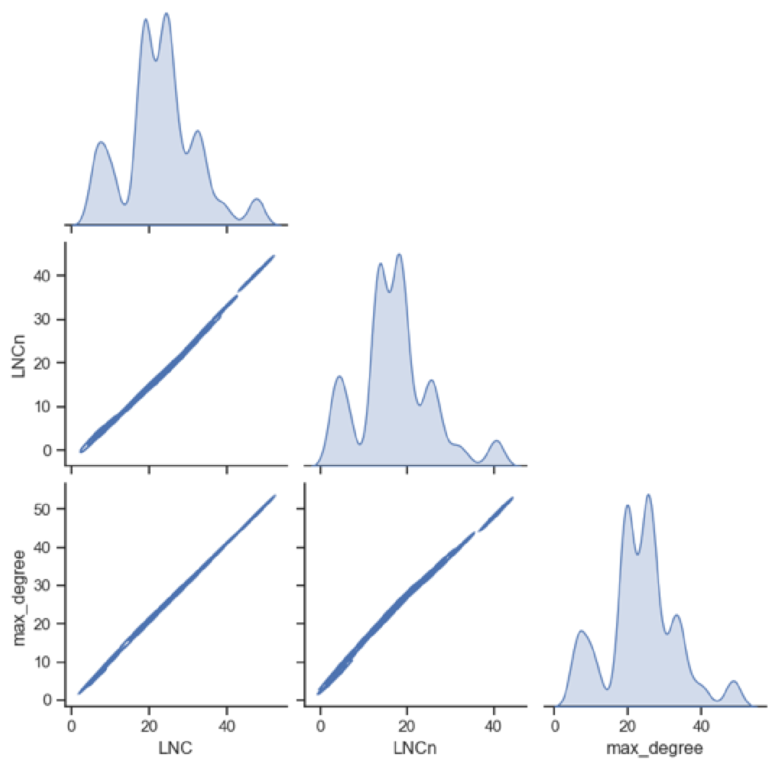


Figure 5. Comparison of LNC, LNCn, and Maximum Node Degree

The linear relationships of LNC and LNCn with the maximum node degree of the graph indicate that these metrics are connected to the size of the largest hub in the graph. This linearity is also found in star graphs, while in complete graphs, where there are no hubs by definition, and each node is equivalent to all the others, LNC would grow with a descending rate, and LNCn would plateau asymptotically towards 1.

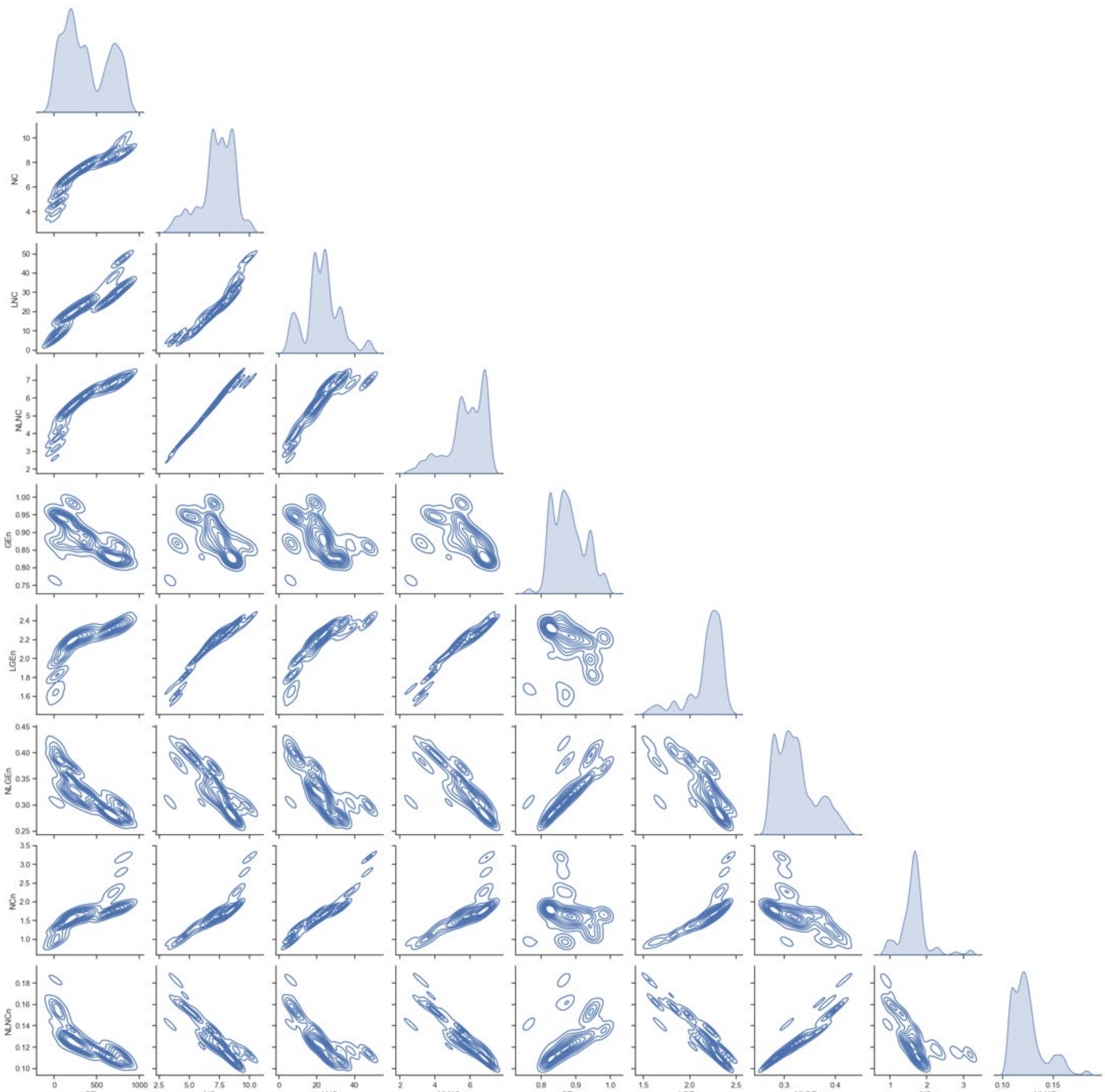


Figure 6. Comparison of Uncorrelated Metrics

Figure 6 shows the pairwise comparisons of all the metrics that do not present a clear linear correlation in the Snorkel code base. Some of these relationships are planned to be analyzed in subsequent research efforts, but an effort in narrowing the pool of metrics and towards a more purposeful metric design will be necessary to measure meaningful characteristics of software architectures.

Trends Over Time

The linear correlation analysis allows the connection of different metrics, in an effort to characterize the topology of the dependency graph. The actual development and creation of the codebase over the 5-year period can be analyzed by plotting some of these metrics over time. The evolution of the dependency graph presented in Figure 3 is depicted by the values of four of the metrics shown in Figure 7: GE, NC, GEn, and NCn.

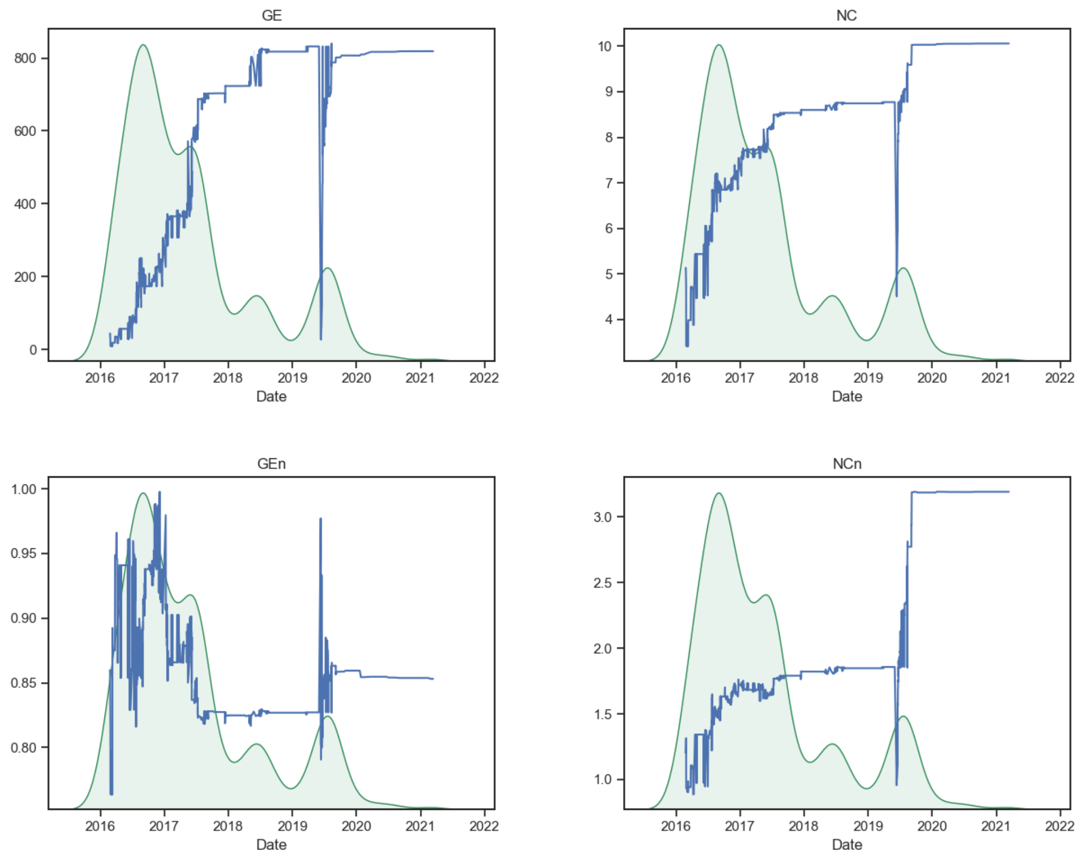


Figure 7. Trends for GE, NC, GEn, and NCn Over 6 Years of Project Development

Figure 7 presents a series of time plots for this select subset of metrics. For each metric, the green shaded area represents the frequency of commits in the project at a specific point in time. This frequency is not connected to the values on the y-axis. The plots show that the development of the project was very active in 2016 and 2017, with a smaller spike of activity in 2019, when, according to the commits, the project underwent a small overhaul, with frequent additions and removals of code. This allows us to better contextualize the changes in each metric and see how they react when the codebase is changed.

Graph energy (GE) quickly rises during the initial development, and fluctuates significantly during the overhaul, only to settle at essentially the same level afterwards. Natural connectivity (NC), on the other hand, rises also after the overhaul, suggesting that the changes made to the codebase in 2019 increased the cohesion of the whole project, without unnecessarily increasing coupling.

The comparison between GE and GEn shows the effect of the normalization factor $\gamma = \frac{1}{n}$, which was introduced to allow a comparison of graphs of different size (number of nodes). In this case, this normalization affects GEn to the point that the metric only seems to capture the frequency of the commits, and not the growth of the graph (as expected). This behavior is not the case when this normalization is applied to NC, as NCn still seems to be affected by the graph growth.

Conclusion

This paper presented a methodology to study the behavior of complex software systems in terms of their structural complexity with a focus on the modifiability of the code base. This approach is based on the parsing of the code and the creation of a dependency graph, a particular case of architectural structure that focuses on the dependency between software modules and the various ways they can call each other.

The dependency graph has been analyzed through the evaluation of a series of spectral metrics, which have shed light on some characteristics of the graph and given insights on the quality of the development effort. It is important to note that this approach forgoes the analysis of the actual lines of code and the dynamic effects that they will have at runtime and is therefore to be considered limited in scope and applicability.

In parallel to this analysis being carried out, the behavior of each metric is also being discovered, thus bootstrapping their applicability to the metrics. Behind the scenes, the metrics have been applied to conventional graphs, but the use case of a real software project is necessary to gauge the limitations of this approach.

Future research will continue the effort of connecting these and other metrics to important attributes of software code bases. Improvements to our own software tools will allow for analysis of projects with larger repositories, and with a longer development time frame, where the effects of technical debt might be more pronounced. Additional improvements are also planned for the visual representation of modifiability in software systems.

References

Amal, A., Sliman, L., Kmimech, M., Bhiri, M. T., & Raddaoui, B. (2018, September 26–28). *Towards a formal verification approach for cloud software architecture* [Paper presentation]. New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 17th International Conference on New Trends in Intelligent Software Methodology, Tools and Techniques (SoMeT18), Granada, Spain.

Angelov, S., Grefen, P. W. P. J., & Greefhorst, D. (2009). A classification of software reference architectures: Analyzing their success and effectiveness. *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*, 141–150. https://doi.org/10.1109/WICSA.2009.5290800

Avci, C., Tekinerdogan, B., & Athanasiadis, I. N. (2020). Software architectures for big data: A systematic literature review. *Big Data Analytics*, *5*(1), 5. https://doi.org/10.1186/s41044-020-00045-1

Bahsoon, R., Ali, N., Heisel, M., Maxim, B., & Mistrik, I. (2017). Software architecture for cloud and big data. In I. Mistrik, R. Bahsoon, N. Ali, M. Heisel, & B. Maxim (Eds.), *Software architecture for big data and the cloud*. Morgan Kaufmann.

Basili, V. R. (1980). Qualitative software complexity models: A summary. *Tutorial on models and methods for software management and engineering.*

Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.). Addison-Wesley.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-oriented software architecture*. John Wiley & Sons.

Cabot, J., Capilla, R., Carrillo, C., Muccini, H., & Penzenstadler, B. (2019). Measuring systems and architectures: A sustainability perspective. *IEEE Software*, *36*(3), 98–100. https://doi.org/10.1109/MS.2019.2897833

Capilla, R., Nakagawa, E. Y., Zdun, U., & Carrillo, C. (2017). Toward architecture knowledge sustainability: Extending system longevity. *IEEE Software*, *34*(2), 108–111. https://doi.org/10.1109/MS.2017.54

Darcy, D. P., Kemerer, C. F., Slaughter, S. A., & Tomayko, J. E. (2005). The structural complexity of software an experimental test. *IEEE Transactions on Software Engineering*, *31*(11), 982–995. https://doi.org/10.1109/TSE.2005.130

Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2018). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, *23*(1), 490–520. https://doi.org/10.1007/s10664-017-9524-2

Enos, J. R., Farr, J. V., & Nilchiani, R. R. (2019). Identifying and quantifying criticalilities in the acquisition of DoD system. *Defense Acquisition Research Journal*, *26*(1), 18–43. https://doi.org/10.22594/dau.18-799.26.01

Fischi, J., Nilchiani, R., & Wade, J. (2017). Dynamic complexity measures for use in complexity-based system design. *IEEE Systems Journal*, *11*(4), 2018–2027. https://doi.org/10.1109/JSYST.2015.2468601

Francesco, P. D., Malavolta, I., & Lago, P. (2017, April 3–7). *Research on architecting microservices: Trends, focus, and potential for industrial adoption* [Paper presentation]. 2017 IEEE International Conference on Software Architecture.

Garlan, D., & Shaw, M. (1993). An introduction to software architecture. *Advances in software engineering and knowledge engineering* (pp. 1–39).

Gutman, I. (2001). *The energy of a graph: Old and new results.* Berlin, Heidelberg.

Hasselbring, W. (2002). Web data integration for e-commerce applications. *IEEE MultiMedia*, *9*(1), 16–25. https://doi.org/10.1109/93.978351

Hasselbring, W. (2018). Software architecture: Past, present, future. In V. Gruhn & R. Striemer (Eds.), *The essence of software engineering* (pp. 169–184). Springer Open.

Hästbacka, D., Halme, J., Larrañaga, M., More, R., Mesiä, H., Björkbom, M., ... Hoikka, H. (2019, October 27–30). *Dynamic and flexible data acquisition and data analytics system software architecture* [Paper presentation]. 2019 IEEE SENSORS.

IEEE. (2000). Recommended practice for architectural description of software-intensive systems. In *IEEE 1471:2000*.

ISO/IEC. (1995). Reference model of open distributed processing (RM-ODP). In *ISO/IEC 10746:1995*.

Jun, W., Barahona, M., Yue-Jin, T., & Hong-Zhong, D. (2010). Natural connectivity of complex networks. *Chinese Physics Letters*, *27*(7), 078902. https://doi.org/10.1088/0256-307x/27/7/078902

Kazman, R., Bass, L., Webb, M., & Abowd, G. (1994). *SAAM: A method for analyzing the properties of software architectures* [Paper presentation]. Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italy.

Kendall, F. (2016). *Performance of the defense acquisition system*. DoD.

Kruchten, P. (1995). The 4+1 view model of architecture. *IEEE Software*, *12*, 45–50.

Kruchten, P., Obbink, H., & Stafford, J. (2006). The past, present, and future for software architecture. *Software, IEEE*, *23*, 22–30. https://doi.org/10.1109/MS.2006.59

Linington, P. F. (1995). RM-ODP: The architecture. In K. Raymond & L. Armstrong (Eds.), *Open distributed processing: Experiences with distributed environments. Proceedings of the third IFIP TC 6/WG 6.1 international conference on open distributed processing, 1994* (pp. 15–33). Springer U.S.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science*, *52*(7), 1015–1030. https://doi.org/10.1287/mnsc.1060.0552

Malavolta, I., & Capilla, R. (2017, April 5–7). *Current research topics and trends in the software architecture community: ICSA 2017 workshops summary* [Paper presentation]. 2017 IEEE International Conference on Software Architecture Workshops.

Mens, T. (2016). *Research trends in structural software complexity*. https://arxiv.org/abs/1608.01533

Newman, S. (2015). *Building microservices*. O'Reilly.

Nilchiani, R. R., & Pugliese, A. (2016). *A complex systems perspective of risk mitigation and modeling in development and acquisition programs*. Stevens Institute of Technology.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Commun. ACM, 15*(12), 1053–1058. https://doi.org/10.1145/361598.361623

Perry, D., & Wolf, A. (2000). Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, *17*. https://doi.org/10.1145/141874.141884

Pugliese, A., Enos, J., & Nilchiani, R. (2018). *Acquisition and development programs through the lens of system complexity*. Naval Postgraduate School.

Pugliese, A., & Nilchiani, R. (2017). *A systems complexity-based assessment of risk in acquisition and development programs*.

Pugliese, A., & Nilchiani, R. (2019). Developing spectral structural complexity metrics. *IEEE Systems Journal*, *13*(4), 3619–3626. https://doi.org10.1109/JSYST.2019.2912368

Putman, J. (2000). *Architecting with RM-ODP*. Prentice Hall.

Qureshi, N., Usman, M., & Ikram, N. (2013). *Evidence in software architecture, a systematic literature review*. Paper presented at the Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering, Porto de Galinhas, Brazil.

Salado, A., & Nilchiani, R. (2013). *Using requirements-induced complexity to anticipate development and integration problems: Analysis of past missions* [Paper presentation]. AIAA SPACE 2013 Conference and Exposition.

Shahin, M., Liang, P., & Babar, M. A. (2014). A systematic review of software architecture visualization techniques. *Journal of Systems and Software*, *94*, 161–185. https://doi.org/10.1016/j.jss.2014.03.071

Shaw, M., & Clements, P. (2006). The golden age of software architecture. *IEEE Software*, *23*(2), 31–39.

Soni, D., Nord, R., & Hofmeister, C. (1995). Software architecture in industrial applications. *17th International Conference on Software Engineering*, 196–196.

Taibi, D., Lenarduzzi, V., Pahl, C., & Janes, A. (2017). *Microservices in agile software development: A workshop-based study into issues, advantages, and disadvantages* [Paper presentation]. Proceedings of the XP2017 Scientific Workshops, Cologne, Germany.

Venters, C. C., Capilla, R., Betz, S., Penzenstadler, B., Crick, T., Crouch, S., ... Carrillo, C. (2018). Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software*, *138*, 174–188. https://doi.org/10.1016/j.jss.2017.12.026

Witt, B., Baker, T., & Merritt, E. (1994). *Software architecture and design: Principles, models, and methods*. Van Nostrand Reinhold.

Woods, E. (2016). Software architecture in a changing world. *IEEE Software*, *33*(6), 94–97. https://doi.org/10.1109/MS.2016.149

Addressing Software-Based, Platform Interoperability Risks in Defense Systems by Using Distressed Debt Financial Strategies: A Technical Debt Mitigation Concept

Ann Gallenson—is a Lecturer with the Center for Executive Education and the Naval Postgraduate School. She facilitates Navy leaders and their executive teams in devising strategic initiatives to meet changing mission requirements. Her research interests center around intrapreneurship and how new technologies and practices create opportunities for organizational change and transformation. [acgallen@nps.edu]

CAPT Scot Miller—is a Faculty Associate Researcher at the Naval Postgraduate School. He began his active-duty career as a P-3C pilot. He is a proud plank owner of the U.S. Navy's Information Professional Community. [scot.miller@nps.edu]

Susan Higgins—is a Faculty Associate Researcher in Information Sciences at the Naval Postgraduate School, where she has been an advisor to senior military and defense leaders on innovation, culture change, and learning for 20 years. As an active-duty naval officer for the previous 20 years, she helped to transform the Navy's use of technologies in space systems, information sharing, learning, and communications. [shiggins@nps.edu]

Abstract

This concept paper explores an innovative approach to detecting and managing software vulnerabilities in cyber-physical defense systems. Software-based vulnerabilities that hinder or preclude the maintainability and evolvability of combat systems are a pernicious form of technical debt that threaten all cyber-physical systems. The risks associated with technical debt across increasingly interdependent DoD cyber-physical systems will accelerate if left unchecked. Without changes in acquisition and maintenance practices, we can foresee cascading, potentially catastrophic cross-system failures. To illustrate the risk and possible solutions, we focus on the software embedded in combat systems that are subject to ongoing modernization efforts that extend their applicability to evolving operations. Our research revealed that software vulnerabilities in critical combat systems can threaten the reliability and readiness of those systems. These vulnerabilities provide an opportunity for the defense acquisition communities to create a new capability within their organizations, an Acquisition Technical Debt Team (ATDT) to help detect, manage, and mitigate technical debt. We explore risk classification by including interoperability into risk evaluation schemas. We then apply common distressed debt management models to suggest when and how the ATDT might help manage and mitigate technical debt to help rehabilitate an ailing system.

Introduction

Over the past 60 years the Navy's critical combat, command, control, and communications systems have evolved into software-dependent technological ecosystems that combine weapon, communication, and detection systems. These cyber-physical systems rely on the tight coupling of computational and physical processes to create responsive, timely, accurate, and lethal deterrence capabilities. This coupling creates management challenges throughout a system's life cycle. Physical defense systems are designed over multiple years, maintained for decades, and periodically re-engineered to meet evolving operational requirements. Modern software systems rely on adaptive development cycles to meet evolving cybersecurity, technical, operational, and interoperability changes. Legacy software in critical, networked defense systems can pose an increasing risk to the systems and the warfighters who rely on them. Modernization requirements for both the hardware and software often introduce further misalignments and vulnerabilities. Software incompatibilities, diminished systems interoperability, and direct cyberattacks are a few of the risks that system maintainers must mitigate to ensure combat system readiness. These software vulnerabilities, commonly known as technical debt, arise throughout the life cycle of the system and are mostly due to trade-offs made to meet time, cost, and capability requirements.

Technical debt is a software engineering term that characterizes the design or implementation trade-offs taken to meet short-term business and development requirements that create barriers to future changes, including the maintainability and evolvability of the system; it can make these future changes more costly or impossible (Krutchen, 2019, p. 5). The technical debt landscape depicts the software elements that contribute to evolution and maintenance challenges (see Figure 1). Combat systems interoperability requirements can occur at any time in the software's life cycle and primarily impact the software's internal, invisible, elements. Furthermore, networked warfare systems may be susceptible to software vulnerabilities that occur in one combat component but permeate across the network.

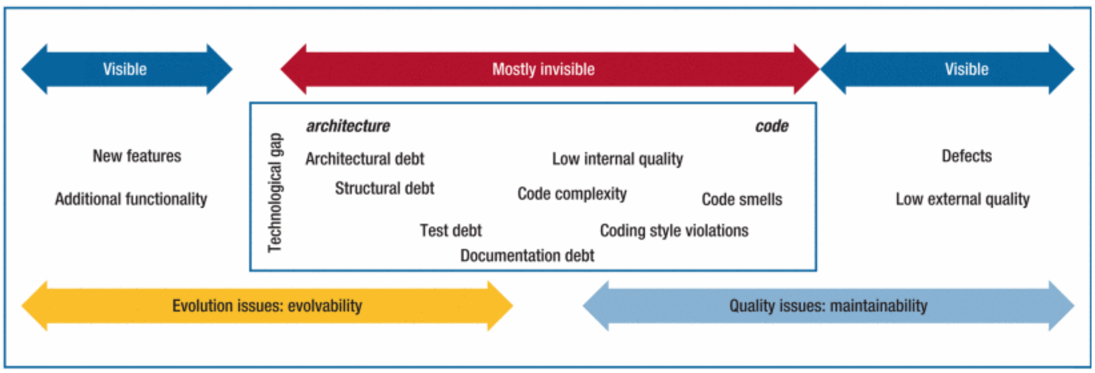


Figure 1: The technical debt landscape depicts the visible and invisible software elements that contribute to evolution and quality challenges.
(Kruchten, 2012, p. 19)

The interdependence of cyber-physical combat systems and their need to function and adapt to changing environmental conditions and operational requirements led us to wonder if the methods used for dealing with technical debt in the private sector were adequate for managing technical debt for weapon systems. Our research surveyed current practices in technical debt detection, classification, and management in the private and public sectors through a literature review. We augmented this overview with discussions with software scientists, combat system engineers, and military leaders regarding current practices and challenges of managing technical debt in combat systems. The research questions were: 1) What methods are being used to detect, assess, monitor, and mitigate technical debt, and realign/re-introduce refactored code into existing systems; and 2) What program management practices can best meet the time and technical demands of maintaining platform reliability while addressing complex, embedded software reliability issues?

This paper is organized into the following sections: 1) a brief review of the impact of poor- quality software and technical debt on the U.S. and world economies; 2) a literature review of technical debt practices and concerns; 3) an overview of the challenges of maintaining weapon systems while managing technical debt; 4) exploration of a technical debt program management using distressed debt practices; and 5) conclusion and next steps.

Technical Debt Impact

The Consortium for Information and Software Quality (CISQ) reported that in 2018 the cost of poor software quality (CPSQ) in the United States was $2.26 trillion and the future (deferred) estimated cost of technical debt was $580 billion. They estimate that $635 billion was spent on legacy systems. The top five global IT failures in the news in 2018 were Wells Fargo Bank, PSA Airlines, Uber Technologies, TSB Bank, and Welsh NHS IT (Krasner, 2019). The CISQ report for 2020, in the midst of a global pandemic, reported the CPSQ in the United States was $2.08 trillion and the future (deferred) estimated cost of technical debt was $1.31 trillion residing in severe defects that need to be addressed in the future. They estimate that $520 billion was spent on legacy systems. The top five global IT failures in the news in 2020 were Home Depot, Iowa Caucus Smartphone app, NASA/Boeing Starliner, Heathrow Airport disruption, and the Google Plus Security glitch. The majority of the costs were aggregated from publicly available source material on the cost of poor software. However, most IT and software organizations do not currently collect cost of software quality data, making these figures an underestimate of the actual costs (Krasner, 2021).

The CISQ 2020 CPSQ report estimated that worldwide cybercrime will cost companies an estimated $6 trillion annually by 2021. They conclude that software vulnerabilities across the IT industries in the United States are creating a fertile environment for cybercrime. The three cost contributors to this exploitable environment are $260 billion spent on unsuccessful projects ($177.5 billion in 2018), $520 billion on legacy system problems ($635 billion in 2018), and $1.56 trillion spent on software failures in operational systems. Given that there could be overlap between legacy problems and operational failures, the report reduces legacy system failures by 50% and concludes with a total estimate of $2.08 trillion in U.S. CPSQ for 2020 (Krasner, 2021, p. 24).

Technical Debt Practices, and Concerns

The results of our research revealed that IT professionals and academics across the public and private sectors are actively working to identify, assess, mitigate, and manage software technical debt. Technical debt vulnerabilities can be introduced at any time in the software life cycle and across the technical debt landscape (see Figure 1). A meta-study of 94 technical debt research studies conducted from 1992 through 2013 identified 10 technical debt types, eight technical debt management activities, and 29 tools for technical debt management (Li, 2015).

The Object Management Group (OMG), an international technology standards consortium, has developed an automated technical debt measurement (ATDM) tool to measure source code reliability, security, performance efficiency, and maintainability and then create a time estimate for remediations of the found weaknesses to help guide corrective actions (OMG, 2017). Early efforts that use machine learning methods to detect technical debt through natural language processing (Rantala, 2020) and forecasting for technical debt evolution through the use of linear Regularization models and non-linear Random Forest regression (Tsoukalas, 2020) are producing encouraging results.

Federal IT professionals report that technical debt directly impacts their mission by limiting the speed with which new functionality can be delivered. The complexity of the code base makes it more difficult and time consuming to modernize and increases the probability that additional technical debt will be injected into software (Curtis, 2018). There is a growing concern that modernizing or developing software capabilities by using machine learning for prediction or other functions, though cost effective, introduces more complicated, hidden, and rapidly accumulating technical debt vulnerabilities into the system. Software engineers at Google, Inc., warn that machine learning integrated into a system's design can result in specific risk factors that include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and system-level anti-patterns (Sculley, 2015).

For the Department of Defense (DoD), technical debt is complicated by a spectrum of requirements including specialized combat systems, extreme and changing environmental conditions, changing adversarial requirements, changing operation plans, longevity expectations, spending requirements, contractual agreements, and legal constraints. Military missions often require an array of commercial off-the-shelf (COTS) components for software and hardware modification. COTS requirements add another layer of complexity to technical debt due to system requirements differing from COTS capabilities. COTS and system mismatches include functional, performance, interoperability, configuration problems across versions, documentation, COTS evolution limits, interface differences, and COTS longevity maps. COTS requirements add additional technical debt detection and sustainment needs (Yang, 2018).

Technical debt can result in vulnerabilities that can be exploited through a cyberattack. Weapon systems are cyber-physical systems of systems that are networked and rely on software integration and interoperability to perform critical functions. As the systems become more software interdependent and complex, the probability of cyberattacks increases. As weapon systems become more lethal, the need for sophisticated and trustworthy cybersecurity systems becomes more imperative (Chaplain, 2018).

Managing Combat Systems and Technical Debt

Our discussions with U.S. Navy leaders, software engineers, weapon systems engineers, and combat system maintainers revealed that they have been managing and working around technical debt issues for decades. Due to the specialized nature of weapons and weapon systems, many of their programs were early adopters of cyber-physical integration and have been in the forefront of developing these capabilities. Continual hardware and software modernization has resulted in complicated middleware processes to ensure interoperability across multiple systems. The need for increased capabilities without modernizing legacy code can create internal complexities and increase the probability of accruing additional technical debt. Sophisticated development and testing capabilities have been devised to maintain reliability and adaptability. Software architecture is a key component to interoperability, usability, and adaptability. Technical debt assessment needs to be broken down by components and interoperability patterns.

Systems routinely undergo risk assessments and are scheduled for maintenance based on a standard risk scale that measures the severity against the likelihood of the risk. Technical debt may accrue because the assessed risk falls below the critical mark. Maintainers need a way to communicate the need for technical debt mitigation to program managers before functional or interoperability risks become critical.

Maintenance and management of complex, integrated, and interoperable weapon systems suffer from technical and organizational constraints. Dealing with complex system interoperability issues across different generations of weapon systems is complicated, time consuming, and expensive. The acquisition requirements for meeting scope, cost, and time parameters are often in conflict with the realities of managing and mitigating legacy system technical debt.

In summary, cyber-physical combat systems represent a special case of technical debt that falls outside of many public and private sector requirements. Their customized builds and configurations, the complexity of the systems, the age of some of their components, their interoperability requirements, and the criticality of their functions make them more vulnerable to the risks of technical debt. These characteristics also make combat systems good candidates for the creation of a new acquisition capability designed to assist in reducing the defense specific risks associated with technical debt. Processes and practices acquired through this endeavor could save the Navy time and money by minimizing the current and future costs of technical debt while leveraging expertise within the Navy enterprise.

Recommendation: Create a Technical Debt Program Management Capability Using Distressed Debt Practices

Our recommendation argues that mitigating and managing combat systems' technical debt risks during a system's maintenance phase will require specialized and innovative teams within the acquisition community. The goal of an acquisition-based team is to assist combat systems managers and maintainers with the complexities of managing and mitigating technical debt across the component portfolio of their systems. By creating a specialized team that initially focuses on one combat system, program managers and combat systems experts can create and refine sustainable processes for technical debt assessment, reporting, management, mitigation, and reintegration. Ultimately the processes can grow into an acquisition capability that could be applied to systems across the Navy. A technical debt capability in acquisition could help mitigate technical debt at early stages and throughout the life cycle of cyber-physical combat systems.

The U.S. bankruptcy codes and distressed debt management practices present a starting point for developing assessment and mitigation processes. The codes and practices have evolved to evaluate the "type" of debt-related stresses a company is under and methods to best rehabilitate the company or liquidate their assets. The basic goal is to realize that there is value in the larger cyber-physical system that can be rehabilitated if the ailments due to technical debt are mitigated. Creating mitigation strategies based on both the severity of the risk and the interoperability of the system is akin to treating the whole system rather than a single symptom. The initial focus on combat systems includes an inherent portfolio approach that combines integrated and interoperable systems whose dependencies may exhibit technical debt through association. The following is a conceptual framework whose intent is to serve as a starting point for future exploration, creation, and refinement.

Create an Acquisition Technical Debt Team

Create a cross-organizational Acquisition Technical Debt Team (ATDT) with expertise in acquisition processes and an initial focus on a pilot combat system that has been dealing with technical debt due to the interoperability of legacy systems. The team should include members of the combat system's engineering and managerial staff, program managers who have experience with the system, and external experts familiar with the technological and operational needs for the system. The purpose of the team is to develop ways of assessing the health of the combat systems and identifying the processes needed to manage and resolve technical debt issues. The team could initially include external experts versed in assessing, managing, and liquidating distressed debt companies.

Assess and Manage System-Wide Risks Associated with Technical Debt

Given the integration and interoperability requirements of combat systems, we have created a risk cube that places interoperability along the z-axis (see Figure 2). The incorporation of interoperability into the assessment of technical debt risk will help determine the associated risk to external components and associated vulnerabilities that extend across the greater system. The goal of this approach is to create a classification schema, similar to the U.S.

bankruptcy codes, to helps the combat system's resident team determine when to report the risk to the program manager and when to request a transfer of the problem to the ATDT.

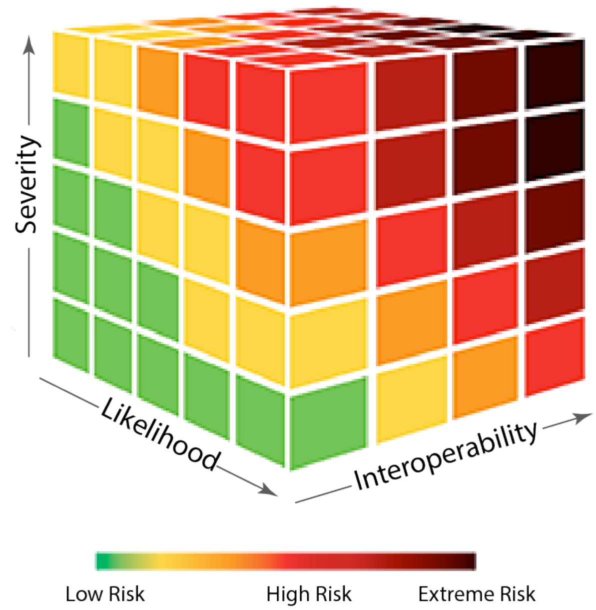


Figure 2: The Inclusion of Interoperability in a System's Risk Assessment Charts the Degree that Technical Debt or Component Failure Effects the Greater System

In practice, the risk cube would be generated using data from the combat system, its repair history, and relevant vendor or industry models. Multiple signals at the ATDT level over a short period of time, operational shifts, fielding of new capabilities for that combat system, etc., could place the system at a level where the ATDT should take over for the program manager until the risks are mitigated.

In Distressed Debt Management (DDM), a distressed business performing "below investment grade" is determined to be either undervalued or about to default on their loans. A company may become distressed because of excessive debt, but other factors that can threaten the company are declining profitability, loss of competitive position, or changing business climate. There are three management strategies that depend on the type of debt the company is dealing with and the time horizon for recovery or liquidation. The three management strategies are as follows (Jain, 2011):

1. **Active Control**—the business process is taken over by the management team with an exit timeline of 2–3 years and a profit target of 15–25% per year.
2. **Active Non-Control**—the management is actively involved with existing management but does not take over the business with an exit strategy of 1–2 years and a profit target of 12–20% per year.
3. **Passive**—the management company invests in the existing company to create financial buffers with an exit strategy of 0.5–1 year, and a profit target of 12–15% per year.

These management strategies can be extrapolated to the type and degree of involvement the ATDT would pursue to manage and mitigate the risks from technical debt on a combat system and the systems reliant on it. The time horizons and profitability parameters could be used as guides for ATDT involvement and follow-on savings and/or estimated extended use of the combat system.

Using the categorizations from the interoperability risk cube we can classify the type of involvement of the ATDT as follows:

A. **No ATDT Involvement—**if the severity and the likelihood of a vulnerability are high but interoperability is low (HHL), then there is no need to elevate the problem to the ATDT. The risk should be dealt with by the combat systems managers and engineers.

B. **ATDT Passive Involvement—**if the severity, likelihood, and interoperability of a vulnerability are moderate (MMM), the problem should be elevated to the program manager for that combat system, who would monitor the progress and might ask for guidance from the ATDT specialists.

C. **ADTD Active Non-Control—**if the severity is low, the likelihood is moderate, and the interoperability of a vulnerability is high (LMH), the program manager and ATDT should be notified.

D. **ADTD Active/Non-Control** or **Active Control—**if the severity is high, the likelihood is moderate, and the interoperability of the vulnerability is high (HMH) the ATDT might want to monitor the risk, or if there have been several risks reported, they could take over the management and mitigation of the technical debt of the system from the program manager and return the authority when interoperability risks have been mitigated and components replaced or refactored.

E. **ADTD Active Control—**if the severity, the likelihood, and the interoperability of the vulnerability are high (HHH), the ATDT would take over the program management of the system and work in tight coordination with the combat system's management and engineering team. Once the debt has been mitigated and the system has been restabilized (or retired), the management would be returned to the original program management team.

Conclusion and Next Steps

The effects of technical debt on software-based systems are creating a global concern and costing trillions of dollars in recovery and mitigation. The vulnerabilities inherent to complex systems and those reliant on legacy architectures are creating a fertile ground for cyber criminals and other adversaries. The DoD weapon systems are complex cyber-physical systems. These systems have been subject to ongoing requirements to modernize and modify both their physical and software systems to meet changing environmental factors and operational requirements. These modification pressures result in technical debt that their managers and maintainers have been contending with for decades. The risks associated with technical debt across increasingly interdependent DoD cyber-physical systems will accelerate if left unchecked. Without changes in acquisition and maintenance practices we can foresee cascading, potentially catastrophic cross-system failures. We suggest that the acquisition enterprise help manage and mitigate these risks by creating cross-functional teams dedicated to classifying and managing technical debt. We have used distressed debt management practices from the financial sector to seed an acquisition-based conversation.

Suggested next steps:

- Select a combat system that is already engaged in working through technical debt issues to learn existing technical debt evaluation processes and create a risk cube from their data.

- Create a cross-functional team familiar with the combat system and the technical issues that are contributing to vulnerability concerns.
- Evaluate the interoperability risk cube and the technical debt management strategies suggested here to learn the relevance of these models to the combat system's needs.
- Continue this research to refine these concepts with the acquisition community.

References

Curtis, W. (2018, February 23). *Solving the "technical debt" problem*. FCW. https://fcw.com/articles/2018/02/23/comment-technical-debt-curtis.aspx

Chaplain, C. (2018). *Weapon systems cybersecurity: DoD just beginning to grapple with scale of vulnerabilities* (GAO-19-128).

Jain, S. (2011). Investing in distressed debt. *UBS Alternative Investments, June*, *15*.

Krasner, H. (2019). *The cost of poor quality software in the U.S.: A 2018 report* [Technical report]. Consortium for IT Software Quality.

Krasner, H. (2021). *The cost of poor quality software in the U.S.: A 2020 report* [Technical report]. Consortium for IT Software Quality.

Kruchten, P., Nord, R., & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *IEEE Software*, *29*(6), 18–21.

Kruchten, P., Nord, R., & Ozkaya, I. (2019). *Managing technical debt: Reducing friction in software development*. Addison-Wesley Professional.

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, *101*, 193–220.

Object Management Group. (2017, December). *Automated technical debt measure specification* (Version 1.0) [Computer software specifications]. Consortium for IT Software Quality. https://www.omg.org/spec/ATDM

Rantala, L. (2020, October). Towards better technical debt detection with NLP and machine learning methods. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (pp. 242–245). IEEE.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems* (pp. 2503–2511).

Tsoukalas, D., Kehagias, D., Siavvas, M., & Chatzigeorgiou, A. (2020). Technical debt forecasting: An empirical study on open-source repositories. *Journal of Systems and Software, 170,* 110777.

Yang, Y., Wade, J., Alelyani, T., & Stanton, P. (2018). *RT 193: Framework for analyzing versioning and technical debt*. Stevens Institute of Technology.

Making the Kessel Run: Re-Insourcing Software Development in the U.S. Air Force

Captains Jennifer Aroune, Robert Hollister, and **Nathan Taylor**—graduated from the Naval Postgraduate School in Monterey, CA, earning Master of Business Administration degrees with a concentration in Enterprise Sourcing. Aroune and Taylor serve as contracting officers in the Air Force Life Cycle Management Center at Wright-Patterson Air Force Base, OH. Hollister serves as a contracting officer in Air Combat Command's Acquisition Management and Integration Center at Langley Air Force Base, VA.

Bryan Hudgens, Lt Col (Ret.), USAF—is a Senior Lecturer in Logistics and Operations Management at the Naval Postgraduate School, Monterey, CA. Hudgens's early professional experience spans the field of government contracting, from supporting front-lines contracting in the Persian Gulf, to serving as a contracting officer on major weapon system contracts for the United States and its allies. Hudgens's teaching, research, and consulting focus on interorganizational relationships, within the end-to-end supply chain and among first responders.

Kathryn Aten—is an expert in organization science focusing on the adoption and use of emerging and information technologies. She draws on her expertise using surveys, behavior observation, focus groups, and individual interviews to help Navy leaders reduce technology acquisition and transition costs, design innovative systems for manning, and develop resilient warfighters and systems. Her research and publications investigate how individuals and organizations adopt and transition technologies as well as how technology implementations influence work and communication. Aten teaches strategic management and change, strategic management of information technology, research design, and qualitative and applied research methods.

Abstract

The U.S. Air Force has traditionally acquired software through the military acquisition process. This acquisition process is used broadly for diverse types of purchases and requires considerable time. Concerned that this approach is not ideally suited for the acquisition of rapidly evolving, innovative software, the U.S Air Force re-assessed its previous make or buy decision regarding software and stood up Kessel Run. Kessel Run is an innovative, in-house software development organization, intended to assess the viability of "re-insourcing" software development. This research reports on a case analysis of Kessel Run. We explore why and how the Air Force developed Kessel Run and offer insights into its operation. Our results suggest that, while cost-reduction goals certainly facilitated Kessel Run's origin and cost reduction has remained a benefit, the primary driver and benefit of re-insourcing Kessel Run is capability development. This finding seems to support a resource-based logic to insourcing. Our results also show that the leaders of Kessel Run succeeded in establishing the organization by relying on commercial start-up concepts, which allowed the organization to attract talent, develop capabilities, and meet customer needs far faster than typical acquisition approaches. We identify lessons learned and implications for other re-insourcing opportunities.

Background

Software is a critical component of defense weapon systems, but "some of the costliest failures in military procurement have been blamed on software … [and] … billions of dollars continue to be spent on software projects that are way over budget and behind schedule" (Erwin, 2018). Software development, when successful, may take years in the traditional procurement process, from requirement inception to software deployment in the operational Air Force (see, e.g., Wallace, 2018). Software development delayed the F/A-22 aircraft (National Research Council, 2007) and the U.S. Air Force (USAF) canceled the Expeditionary Combat Support System with more than $1 billion of sunk obligated funding, when it determined another billion dollars would not salvage the program (Kanaracus, 2012). While the rest of the Air Force and the DoD are still outsourcing most of their innovative acquisitions, including software development—contractors are responsible for the bulk of U.S. defense software development (Jones, 2002)—the Air Force's Kessel Run, an internal software development organization, has decreased the time to field software substantially from years to weeks.

Because the DoD's operation and support costs have not commensurately declined with the ongoing personnel reduction since the Cold War, the DoD resorted to outsourcing or buying functions that are not considered core competencies (Office of Management and Budget [OMB], 1983). Software development is one such outsourced function, traditionally occurring through legally binding instruments governed by the Federal Acquisition Regulation (FAR), Air Force Installation Contracting Center (AFICC) guidance, and other regulatory bodies (Air Force Installation Contracting Center, n.d.; Federal Acquisition Regulation [FAR], n.d.). Three common methods of procuring custom software exist: the procurement of custom software as a commodity using typical contracting approaches, the procurement of software as a service (SaaS) using typical contracting approaches, and the procurement of custom software utilizing Other Transaction Authorities (OTAs; OTA Brief, 2016). While all three methods rely heavily on industry efforts, the former two options are subject to the FAR as opposed to OTAs (OTA Brief, 2016).

With the addition of Kessel Run, a modern approach has altered the software development process and enables the Air Force to self-develop software. Kessel Run has significantly reduced software development lead time by applying agile software development. Agile development is an "extreme" form of programming, used for "complex, volatile" requirements that include "erratic" changes (Highsmith, 2002). By re-insourcing software development and applying agile software approaches, the Air Force has reduced software development cycle times in some cases from years to weeks; however, this is not an easy transition because insourcing requires the Air Force to revamp training for active-duty members and defense civilians in coding and cyber defense core competencies that are unfamiliar to the current workforce (Williams, 2018).

The Air Force has traditionally relied on outsourcing software development but has taken a new direction with Kessel Run. This case study explores Kessel Run, and seeks to understand its insourcing approach and how that approach benefits its customers, when compared with the traditional outsourcing acquisition approach.

Background and Literature Review

Make or Buy Decisions

The decision to make or buy supplies or services within an organization can be complex, but it is one of the most important decisions to make; it is fundamentally a decision on how an organization will manage its supply chain, and thus, it impacts many other decisions and processes (Henriksen et al., 2012). Organizations engaging in strategic make or buy decisions must identify their core competencies (Prahalad & Hamel, 1990). By examining any competitive advantages that arise from those competencies, organizations can litmus test their continued relevance to the organization (Leonard-Barton, 1992). If competencies remain relevant to competitive advantage, the organization should retain these competencies in-house ("make" them). If irrelevant to competitive advantage, the organization should consider the competency with the bulk of other tasks that may potentially be outsourced ("buy"). Then, the organization must conduct transaction cost analysis to better discern what tasks should be kept in-house and what may be outsourced (Williamson, 1996). To this basic framework, Quinn and Hilmer (1994) add that organizations must also consider their vulnerabilities if markets fail in providing outsourced products and services, and how best to mitigate those vulnerabilities.

Core Competencies and Competitive Advantage

At its essence, a core competence embodies the strength of an organization and is typically something the organization has dedicated itself to perfecting. It is "a harmonized combination of multiple resources and skills that distinguish a firm [1][or an organization] in the marketplace" (2013, p. 117). In industry, a core competence must meet three criteria (military analogs are straightforward):

- It must provide access to a wide variety of markets
- It must make a significant contribution to the perceived consumer benefits
- And it must be difficult for competitors to imitate (Prahalad & Hamel, 1990).

An organization's competence must meet all three criteria, and the organization must continuously develop and invest in the competence, to maintain its benefits.

Being integral aspects to their strategies, firms do not intentionally outsource core competencies (Hudgens, 2008). Although in some cases circumstances may change and a core competency is either unnecessary or harmful, core competencies must be performed organically. Leonard-Barton (1992) defines harmful core competencies as "core rigidities" (p. 188) and explores how previously beneficial competencies can hinder the development of new or innovative capabilities. An organization would be well served to abandon, or limit the influence of, any competencies that have become problematic as they create problems (Leonard-Barton, 1992). When an organization is able to grow and maintain core competencies, competitive advantages arise that give it an edge over its competition (Prahalad & Hamel, 1990).

Transaction Costs

Transactions costs are the costs incurred by an organization in the process of conducting business (Chen, 2019). More specifically, transaction costs refer to costs associated with searching, communicating, and bargaining activities (Klein, 2013). Firms and organizations, including the Air Force, need to consider transaction costs when making decisions on how they conduct business.

The Nobel economist Ronald Coase first conceptualized the idea of transaction costs in his article "The Nature of the Firm" published in 1937 (Klein, 2013). While Coase's purpose for his article was to define the term *firm* that, at the time, was widely used but apparently ill-defined, he succeeded in doing more. As Coase explained, "the main reasons why it is profitable to establish a firm would seem to be that there is a cost of using the price mechanism" (1937, p. 390). The price mechanism to which Coase refers is the "invisible hand" specter that guides market transactions so that resources may flow to where they may be most efficiently utilized. Coase gives his abstract cost reference body by providing examples. The first such example is the cost of determining germane market pricing that, while able to be minimized through specialization (e.g., third-party pricing databases), cannot be avoided (Coase, 1937). Negotiating contracts for each transaction also represents a cost that firms can streamline although not eliminate completely (Coase, 1937). Coase does not explore every possible

---

[1] Some core ideas come from literature on business, management, and industry, and thus refer to firms; the ideas typically apply directly to other organizations, particularly the military contexts in this paper. This is especially true given the discussion focuses on whether to insource or outsource a function, which would involve the military (here, the Air Force) interacting with industry as part of its proposed supply chain for any outsourced goods or services.

transaction cost, but he explains that executing transactions come at a cost, and firms may reduce these transactions (Coase, 1937).

Oliver Williamson (1981) expanded on Coase's exploration of transaction costs in "The Economics of Organization: The Transaction Cost Approach," where he described these costs as the economic counterpart to friction. While one may look to moving mechanical parts of a machine and the friction experienced where parts meet to judge its efficiency, Williamson (1981) explains that economic friction may arise in how cooperatively, or uncooperatively, parties interact. Williamson (1981) opined that "transaction cost analysis is an interdisciplinary approach to the study of organizations that joins economics, organization theory, and … contract law." Transactional costs are more than just a reason for the existence of firms; they drive the way firms make decisions.

Williamson (1996) stated, "Whether a firm makes or buys—that is, produces for its own needs or procures a good or service from an outside supplier—turns largely on the transaction costs of managing the transaction in the firm, as compared with mediating the transaction through the market" (p. 25). As discussed, these transaction costs represent the costs required to conduct business with outside vendors or to maintain the support infrastructure to fulfill the requirement organically. If filled organically, transaction costs can include training personnel, maintaining equipment, or similar costs (Coase, 1937). If outsourced, costs may be incurred for monitoring vendor performance or conducting lengthy negotiations (Coase, 1937). An organization must include this assessment of transaction costs in any make or buy decision.

Government Policy on the Make or Buy Decision

Government and DoD outsourcing policy is rooted in United States Code. Further regulations and policies are implemented through national defense authorization acts passed by Congress and through Office of Management and Budget Circular A-76 (OMB, 1983; RAND, 1997). OMB Circular A-76 addresses the make or buy decision, authorizing insourcing in four distinct circumstances: there are no satisfactory commercial sources available, matters related to national defense, patient care, and lower cost (OMB Circular, 1999). The OMB (1999) clarifies that "the general policy of the Government [is] to rely on commercial sources to supply the products and services the Government needs" (p. 1). The OMB also states that "it is the policy of the United States Government to: achieve economy and enhance productivity, retain Governmental functions in-house, and rely on the commercial sector" (OMB Circular, 1999, pp. 1–2).

The DoD relies on the national defense insourcing language to exclude 58% of 640,000 positions conducting commercial activities from outsourcing (RAND, 1997). The same report cites several studies that identify savings, both from insourcing and outsourcing activities, ranging from 20%–35%, although inconsistencies related to actual versus projected savings may exist (RAND, 1997). DoD-specific policy is driven by two DoD Directives: *4100.15 Commercial Activities Program* (DoD, 1985a) and DoD Instruction 4100.33 *Commercial Activities Program Procedures* (DoD, 1985b; RAND, 1997). Directive 4100.15 (DoD, 1985a) essentially parrots OMB Circular policy and assigns specific responsibilities to government positions while Directive 4100.33 (DoD, 1985b) describes the procedures for determining whether government personnel or commercial sources should satisfy needs. More recently, DoD Directive 1100.4 *Guidance for Manpower Management* (DoD, 2005) detailed policy that "assigned missions shall be accomplished using the least costly mix of personnel (military, civilian, and contract) consistent with military requirements and other needs of the Department" (p. 3).

Policy indicates that commercial activities not subject to OMB's four general exceptions should be outsourced to commercial activities. However, DoD Directive 1100.4 goes further and mandates that personnel address military needs with the least costly mix of labor. While the policy does note that the "least costly" approach must be consistent with military requirements, which presumably would include requirements driving a need to keep core functions in house, this directive seems to encourage outsourcing for cost reduction versus attempting to focus on core competencies.

Air Force Policy Directive 38-1 (Department of the Air Force, 2019) echoes DoD Directive 1100.4 by also requiring the least costly mix of personnel to meet military and Air Force needs. Supporting this "least costly" argument, the *Government-Wide Category Management Guidance Document* (Office of the Secretary of Defense, 2015) contains no reference to core competencies or competitive advantages while returning 91 instances of "cost" and 48 instances of "saving." Through the policy documents, directives, and the current focus on category management in government acquisition, it would appear that government make or buy decisions are primarily driven by cost reduction objectives instead of core competency objectives.

Research Approach

This study seeks to understand why and how the Air Force stood up the Kessel Run organization to internally develop (insource/make) software in lieu of the traditional method of procurement by contracting (outsource/buy). This case study focuses on an explanatory analysis (Yin, 2018) of Kessel Run to provide background and rationale for its beginning. An explanatory case study is a study whose purpose is to explain how or why some condition came to be (e.g., how or why some sequence of events occurred or did not occur; Yin, 2018). Studies asking "why" and "how" questions typically seek more explanatory results, which supports the use of the case study, history, or experiment as the preferred method of research (Yin, 2018).

The case study is designed around qualitative analysis of Kessel Run as an organization. The qualitative analysis extracts data in three forms: 1) information provided directly by the Kessel Run staff and leadership, including information posted on U.S. Air Force official websites, 2) information collected from interviews from Kessel Run members, and 3) information posted publicly by those outside of the organization. Because Kessel Run is a new organization, our four interview included participants whose experience in the organization ranged from 7 months to 2 and a half years.

Kessel Run: The Organization

The Air Force wanted their new organization to be fast and agile. They chose the name, Kessel Run, from the movie *Star Wars* and specifically the scene where Han Solo is showing his starship, the *Millenium Falcon*, to Obi-Wan Kanobi. Han tells Obi-Wan that his starship is so fast that it did "the Kessel Run in less than 12 parsecs" (Kelman, 2019).

Although Kessel Run stood up as an independent organization in May 2018, its inception started back in August 2017 and was initiated by the Targeting & Geospatial Intelligence (GEOINT) (T&G) Modernization Program and the [recently terminated] Air Operations Center (AOC) 10.2 program (Kessel Run, n.d.). According to the Kessel Run Acquisition and Contracting Playbook (2019), Kessel Run was originally developed to fill a specific goal which was to:

> Deliver Air Operations Center (AOC) 10.2 Dynamic Targeting Mission Thread, including modern platform and automating associated 3rd party systems, alongside AOC 10.1 at 609th (AFCENT) in <12 months; initial delivery within 90 days of letting contracts...nominally six (6) applications in initial phase. (Kessel Run, n.d.)

Kessel Run quickly expanded to T&G's and AOC's entire portfolio, which was a direct reflection of the value they provide. Now, they are a detachment within the Air Force Life Cycle Management Center (AFLCMC/HBH; 2019).

According to the Kessel Run Acquisition and Contract Playbook (2019), the mission of Kessel Run is to "continuously deliver war-winning software our Airmen love." Additionally, the vision of Kessel Run is to "Build a software company that can sense and respond to conflict in any domain, anytime, anywhere" (2019). Upon reading the mission and vision, it is apparent the mission reads similar to civilian software development agencies, like Google, which highlight the importance of employee satisfaction and its correlation to productivity and creativity (Forbes Technology Council, 2018). The vision is closer to the typical military vision, which incorporates the wartime ("conflict") mission and rapid ("anytime") deployment.

Kessel Run practices "Lean Product Development," "Extreme Programing," and "User Centered Design" (Kessel Run, n.d.). Being lean allows them to, "validate our assumptions and mitigate risk at every turn"; extreme programing allows them to, "always feel confident to go fast, forever"; and their user-centered approach ensures that they "are always delivering value to our users" (Kessel Run, n.d.). The development process follows an 11-step process that includes predictable steps such as "Value Stream Mapping," "Product Scoping," and "Testable," but also such non-military steps as the "Vader Sprint" (a risk-mitigation step) and the final step, "Joyful," which focuses ensuring users adopt the product joyfully and plans for the sunset of the software in its eventual legacy state (Sanders, n.d.).

Is Kessel Run's process working? It reports software development results that include: 1) average time from concept to operations of approximately 4.5 months, 2) reduction of lead time from 5 years to 3.5 days, 3) the ability to push continuous authority to operate to the secure network in less than one hour, and 4) an observed production deployment frequency (capabilities to operations) of 42 capabilities per month, among other improvements and accomplishments (Sanders, n.d.). During our study, we analyzed one project that was completed—from concept inception to operation—in 88 days, or just under 3 months.

Analysis

We wanted to understand why and how the Air Force stood up Kessel Run, so we explored its origin, its purpose, and possible make or buy decision motives, as well as factors that explained Kessel Run's organization and culture.

Kessel Run Origin: A New Hope

Kessel Run rose from a previous failure. When asked, "Why was Kessel Run created?" three of four respondents mentioned AOC 10.2. One respondent reported, "Kessel Run was built out of the ashes of a $500 million effort to modernize the AOC Weapon System, originally called AOC 10.2. After spending 10 years and $500 million and delivering absolutely nothing, Congress cancelled the program" (Participant A). Another interview response was "The failed 10.2 program [AOC 10.2] opened the door to prove out a different way of software development and delivery" (Participant C). Third, "AOC 10.2 failed to deliver any working software in 10 years.… Rather than continue the old way of doing things, we decided to take the bull by the horns and build software in partnership with industry. That way, the government owns the code and can modify the code as necessary without contracting actions" (Participant D). AOC 10.2 was a project the Air Force contracted for with Northrop Grumman to upgrade the Air Operations Center network (Insinna, 2017). After not receiving a finished product in ten years and spending over $500 million, Congress decided to quit funding the program and the Air Force cancelled the project to seek an alternative solution (Innis, 2017). Flowing from this failure, the USAF launched AOC Pathfinder which would include “industry best practices” and “incorporate an agile software development technique called DevOps” (Insinna, 2017). Per the interview responses, one respondent explained that Kessel Run was formally known as AOC Pathfinder. Furthermore, one respondent clarified, “the AOC Pathfinder, now known as Kessel Run” (Participant A).

Why Make the Kessel Run? Software Development Spending

Given, Kessel Run arose from a failed $500 million investment, we expected costs would be a strong motivation to re-insource software development. We first performed a spend analysis which revealed that costs of traditionally procured software development have increased tremendously in recent years. We used the Computer Program and Software Development North American Industry Classification System (NAICS) code 541511 because we concluded that it was most aligned with Kessel Run’s mission. Furthermore, we used NAICS code 541511 because the alternative Product and Service Code (PSC) classification system was more confusing and did not provide a clear PSC for software development. The spend analysis was conducted with Federal Procurement Data System (FPDS) and Air Force Business Intelligence (AFBIT) Competency Cell data (“Air Force Business Intelligence Lite - Profile | Tableau Public,” n.d.; “Federal Procurement Data System—Next Generation,” n.d.). While FPDS contains data for all sectors of Government contracting, the data pulled from FPDS only included the Contracting Office Agency identification code 5700, which represents Air Force contracting offices. Both FPDS and AFBIT data were separately sorted in chronological order and adjusted for inflation. We converted the raw FPDS and AFBIT data into 2017 and 2018 dollars, respectively, by factoring in the appropriate Consumer Price Index (CPI) figure from the Bureau of Labor Statistics (U.S. Bureau of Labor Statistics, n.d.). Furthermore, we plotted each data on a graph where the Y-axis shows the obligation or spend amount and the X-axis represents years pertinent to the data. A trendline and equation were generated to define the slope that provides the average change in obligation or spend amount and the best-fit R-squared value for each sample. Figure 1 represents FPDS data measured by obligated dollars from FY2011 to FY2017 and concluded with an average increase of approximately $90 million per year with an R-squared value of 0.7745.

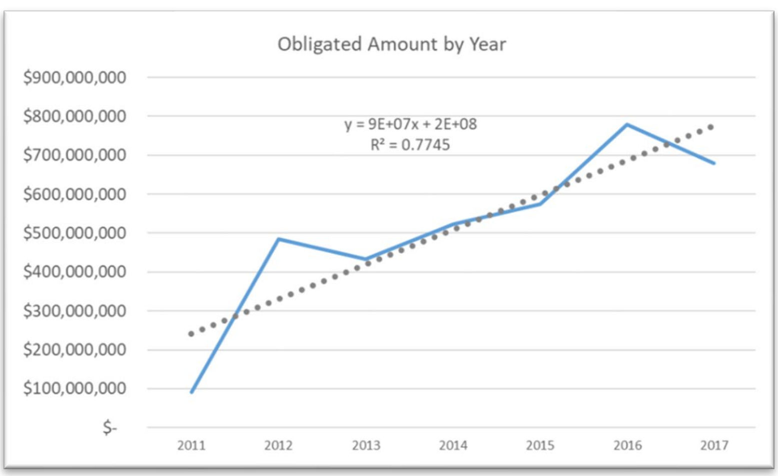


Figure 1: FPDS Data (USAF Data from Federal Database) Spend Analysis of NAICS 541511 from FY11–17 with a Trendline

Figure 2 represents AFBIT data measured by spend amount from FY2014 to FY2018 and concluded with an average increase of approximately $100 million per year with an R-squared value of 0.9392.

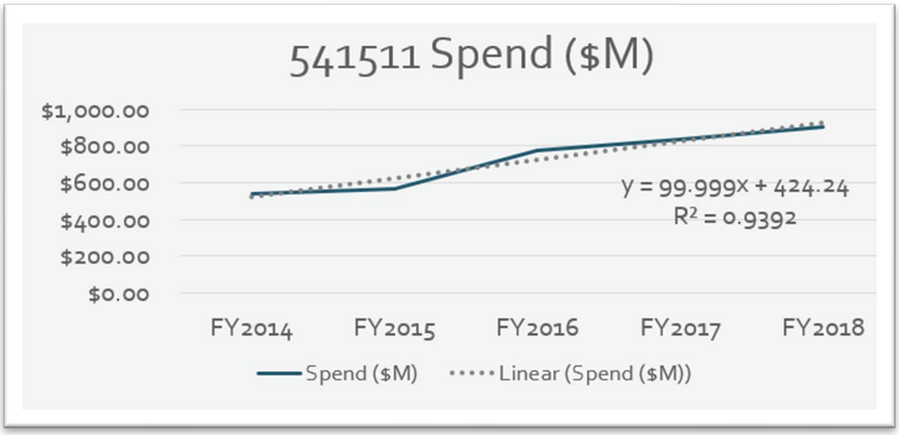


Figure 2: AFBIT Data (USAF Data from USAF Database) Spend Analysis of NAICS 541511 from FY14–18 with a trendline

The second spend analysis with AFBIT data verifies the Air Force's aggressive increase in average dollars per year spent on software development under NAICS code 541511. While both sets of data have relatively high best-fit R-squared values, which indicates that the trendlines are more representative of the data than not, we noted that the available data for the overlapping years between the two samples are inconsistent. This discrepancy calls the sources of the data into question and provides the appearance of data manipulation before we extracted the data. That said, these are the official data sets that document government spending for the federal government and the Air Force respectively, and thus represent the best sources of data available. Additionally, both sets of data presented an indisputable, substantial increase in Air Force computer programming and software development spending over the years at an average $90 million to $100 million per year. The spend analysis yields compelling results that could explain the adoption of more agile software development methods, such as the creation of Kessel Run.

Why Make the Kessel Run? Organizational Purpose

Federal policy states that commercially available requirements should be contracted out and cost comparisons should be conducted to the maximum extent practicable to determine the best organization for fulfilling a requirement (OMB, 1983). As our spend analysis shows, software development costs are increasing generally, and combining that general trend with the specific sting of the failed $500 million AOC 10.2 effort might justify a pilot test to re-insource the capability. However, our study suggested a competing motive.

According to Kessel Run's official website, the organization's purpose and mission is "Continuously delivering war-winning software our airmen love" (Kessel Run, n.d.). Additionally, none of our interview participants mentioned costs as a primary motivation. When asked "What is the purpose of Kessel Run?" interview respondents offered similar responses for the organization's purpose: changing the way the Air Force delivers software by continuous delivery and delivering software the warfighters love. According to one interview response, "Kessel Run is modernizing the Air Operations Center Weapon System through user-centered design, lean start up management and other industry best practices. Our mission is to build and deliver software that warfighters LOVE [emphasis in original response]" (Participant A). Another interview response claimed to purpose was "to change the way the Air Force and DoD delivers software" (Participant B). Furthermore, the third interview response agreed the organizational purpose was "to continuously deliver war winning software that the warfighter loves" (Participant

C). Finally, the last interview respondent reported, "To build the capacity to sense and respond to a changing threat environment with software. While we build products that warfighters love, we could throw those products out the window and still have the ability to deliver combat capability. That is the true power of Kessel Run" (Participant D).

Making the Kessel Run: Building Organic Capability

Interestingly, developing an organic capability might have been more coincidental than intentional (Kelman, 2019). Program Executive Officer, Digital, Steve Wert was more concerned with improving agility no matter the mix of government and contractor blend (Kelman, 2019). He elaborated, "We were primarily looking to demonstrate that modern commercial practice could be successfully applied at scale with the DoD" (Kelman, 2019). That re-insourcing became the preferred approach appears to be a happy consequence. Owning the work and intellectual property and knowing software development practices is instrumental in Kessel Run's effectiveness. As one of our respondents reports, "Having in-house expertise ensures that the Government owns the code baseline and the product development.… The other element is that when we do contract with companies to provide service-based support, we have the in-house technical ability to assess whether a company is performing appropriately" (Participant A).

External observers seem to agree. Kelman (2019), for example claimed Kessel Run leverages "both the use of agile and an increasing role for organic capacity in software development—two issues that are not so related but are being pursued together under the moniker 'Kessel Run'" (Kelman, 2019). Industry adopted these processes about a decade ago (Pomerleau, 2019), and our data suggests the DoD is well aware its software development practices are outdated and unable to effectively enable the mission. The Defense Acquisition University Powerful Example Library (2019) notes, "Its [Kessel Run's] non-standard approach is designed to do two things, speed up the acquisition process and turn the Air Force into a software company that happens to fly planes."

The agility and organic capability appear to be important to Kessel Run and represent a completely different approach to how the DoD normally develops software. Cost does not seem to be a major point of concern when it comes to agility; however, interview respondents were frankly concerned that the slow delivery of software through traditional approaches which hinders the warfighter from obtaining relevant software. An interview respondent explained, "The information we're planning off of has changed and may no longer be valid or important" (Participant B). Another respondent answered, "Traditional acquisitions lends itself to a waterfall approach, i.e., requirements, contract award, development, test, security, compliance, and then fielding. This sequence of events takes more than 10 years on average. This means that the warfighter goes 10+ years without the software they need" (Participant D). The fact that the warfighter is not supported appropriately because of acquisition bureaucracies and inefficiencies is extremely alarming and the USAF has recognized this deficiency.

Making the Kessel Run: It Takes a Different Approach

The Air Force seemed to realize that achieving Kessel Run's goals of speed and agility would require cultural changes, essentially "building a software lab in Boston that was modeled after successful Silicon Valley companies" ("Air Force teaming with Pivotal to rapidly deploy software worldwide," 2019). According to one participant, "Kessel Run utilizes industry and Silicon Valley best practices like lean start up management, user-centered design, and DevSecOps" (Participant A). One respondent emphasized that *The Lean Startup* "was essentially our Bible [*sic*]" (Participant D). Participants referred to, far more than to government instructions, a list of references that would be comfortable on the shelf (or in the digital library) of any Silicon Valley startup: *Start Up Way, Development and Software Operations (DevOps) Research and Assessment (DORA)—State of DevOps, Accelerate: Building and Scaling High*

Performing Technology Organizations, Sense and Respond, How Google Works, Lean Enterprise, The Lean Startup, Extreme Programming, and *Design Thinking*. One participant summarized the startup mentality: “We were very scrappy” (Participant D).

A different culture meant adopting different management practices. One of the interview participants noted:

> things like psychological safety are valued so the most optimal decisions can be made. The traditional military hierarchy that enabled Platoon leaders to navigate the jungles of Vietnam don’t [sic] work for a modern software organization. The highest ranked person doesn’t always have the context or knowledge to make or inform the decision; our flat management structure at the product team level allows decisions to be made by the right people. (Participant 4)

Psychological safety appears to be very important to the organization and its success. We asked for a list of potential interview participants, including an explanation for why they could provide key insights; a key informant responded, “At Kessel Run, we don’t concern ourselves with rank. We put the best people in the roles, whether they are an E-5 or a 2Lt (Second Lieutenant)” (participant, email to author, October 1, 2019). One outside observer collaborated this observation, “it is not every 36-year Air Force old-timer who is willing to listen to kids in hoodies, and even to allow them to name a pet project after a favorite Star Wars trope. (Kudos to the Air Force as well for not squelching this name.)” (Kelman, 2019). This enables Kessel Run’s innovation and ability to obtain useful information from any source without dealing with the same bureaucracy witnessed in the rest of the USAF and DoD.

To build this new start up, Kessel Run was challenged to interface with and develop methods to attract the “New Workforce,” which: 1) was born between 1980 and 2000, 2) prefers cities or large towns, and 3) would trade other benefits for better workspace (Newell, 2018). Furtado explains, “We’re battling industry, especially in Boston, for top end tech talent. … You can’t ask people to hey come and take a $30,000 pay cut, also it’s going to be a bad environment and you’re not going to be happy here” (Pomerleau, 2019). Consequently, Kessel Run operates out of a “brightly lit We Work office” in Boston (Ward, 2019). Alongside startup companies, “T-shirt-and-jeans-wearing airmen milling around its fully-stocked kitchenette” might be indistinguishable, other than “[sounding] a bit different from their similarly dressed office mates” (Newell, 2018).

Admittedly concerned about competing with industry for hiring and retaining talent, the Air Force first invested in developing its own coding talent, recruiting Airmen from around the Air Force and putting them through 6 months of agile coding training (Kelman, 2019).

Interview participants reported training Airmen in software development provided several benefits over outsourcing software development:

- Owning the software and intellectual property
- Reducing the contractor’s competing goals
- Government open-source model enables sharing between teams
- Efficient tactical strategic execution with Government code repository
- Ability to assess contractor performance (Participants A, B, and D)

Not only did the USAF adopt industry’s practices to attract top tech talent, Kessel Run also took a page out of industry’s coding practices by employing continuous delivery methods (Johnson, 2019). The waterfall approach of traditional acquisitions takes years. Continuous authority to operate (ATO) is a best practice adopted from industry which replaces the process of testing all of the code at the end of the project and makes developing software quicker

(Pomerleau, 2019). Essentially, Continuous ATO allows Kessel Run to constantly be working instead of waiting for approvals which can take time.

According to an interview response, “There are numerous companies who continuously deliver software. Even though we’re the government, why couldn’t we get to a state where we were also able to do it. Companies, such as Amazon, Google, and Pivotal, have provided numerous lessons learned that we can use to grow” (Participant D). Using continuous user feedback loops, continuous delivery means software is delivered in weeks and is reiterated to make it better throughout its use. It is not a final end-product that takes years to deploy without effective feedback mechanisms (Johnson, 2019). Colonel Oti, the Detachment Commander for Kessel Run, explained, “Continuous testing of increments of software is crucial to speeding up deployment, and prevents the do-loop of software not being tested till the end, problems discovered, and a cycle of fixes and re-tests. It is never perfect the first time around, but changes can be made based on feedback from initial use” (Kelman, 2018; Kessel Run, n.d.).

The continuous delivery method is aligned with DevOps, an approach which initially produces a “minimally viable product” (Rosenberg, 2019). According to Chaillan, the co-director of the DoD Enterprise Development, Cyber Security, and Software Operations (DevSecOps) Initiative, “When you add cybersecurity experts to this process, working alongside both the developers and the users/operators from the beginning to ensure the code isn’t easily hacked, DevOps becomes DevSecOps” (Rosenberg, 2019).

Is It Fast Enough? Is That Good Enough? The Metrics

Our interview participants all noted four Software Delivery Performance Metrics from Accelerate:

- Deployment Frequency
- Lead Time
- Mean Time to Restore
- Change Failure Rate

Kessel Run’s metrics clearly focus on time, and it is interesting that Kessel Run does not include cost as a significant metric. We had initially expected cost might be an important metric, based on our spend analysis results, but that is not the case. The organization may be performing cost efficiently, such that that cost is not a significant concern to them, or they might be leveraging improved practices to reduce costs; regardless, cost is not a top metric Kessel Run uses to run its organization.

Kessel Run does, however, include two additional core metrics. The first is called “The One Metric That Matters (OMTM).” This metric answers the question posed by the chief of staff of the Air Force, “What combat capability are you delivering?” One of the interview respondents defines it as, “‘the ‘north star,’ which determines whether a product is bringing user value” (Participant A). Finally, one respondent expressed interest in Kessel Run’s personnel turnover rates because “I want to make Kessel Run the place where people want to come and work” (Participant D). Given a need to develop and retain talent as part of this new competency, this metric makes sense.

Outrunning the Imperial Starships: A Few Obstacles

Perhaps unsurprisingly, the interview respondents primarily keyed in on the obstacle of gaining senior leadership’s trust. Interestingly, the challenge most interviewees disclosed was communication among all of the stakeholders.

The obstacle most mentioned in the interviews was gaining USAF acquisition leaders’ trust in taking a new approach in DoD software development. As stated by one respondent,

> The hardest part was gaining senior leaders trust on a completely different approach. … Not doing requirements-based development was a radical change for them. We had to convince leadership to allow us to talk to our users, cloud enabled IT can provide mission assurance, test and security can be incorporated into the development process, and we should use commercial IaaS [Infrastructure as a Service] and PaaS [Platform as a Service] instead of trying to build our own. Also, had to be honest with them that our culture and infrastructure suck so bad it's impending success for innovation, culture sucks because we have way too many gates before we can get to the user (lack of trust) and engineers are just watching people work and infrastructure because we can't develop software and attract top talent in our crumbling infrastructure at Hanscom AFB [Air Force Base] (both networks and facilities) [*sic*]. (Participant C)

Another respondent shared,

> Fear of the unknown. Most of the senior leaders in AF acquisitions have been in their profession a long time. As people grow in their careers, they generally become more risk averse because there is no incentive to take risks. Taking risks is actually a disincentive. A shift from being completely requirements driven to a combination of what I call big "R" requirements (as laid out by ACC) and small "r" requirements (gathered through user-centered design) was met with much skepticism. Accepting that we don't know everything upfront was quite the mindset shift. We learn more as we iterate. We had to educate senior leadership about how not knowing everything upfront and then building things in small increments buys down the level of risk you incur. In addition to fear of the unknown, we had a culture that didn't optimize for doing the best thing for the user. Recognizing that culture had a lot of room for improvement allowed us to make the necessary changes to be more user-focused, attract top talent, and make decisions based on ideas and context, not rank. (Participant D)

As previously mentioned, Kessel Run is taking a new approach to software development, both USAF- and DoD-wide. While it may seem like the obstacle Kessel Run faced is nothing special, it is a considerable stumbling stone when trying to be agile. As another interview expressed, "it makes it difficult sometimes to get things done" (Participant A).

They Made the Kessel Run? So What?

In addition to clear wins in terms of development speed, when asked "How has Kessel Run impacted the Air Force's software development capabilities?" many of the respondents noted that Kessel Run pioneered a different way to deliver software to the Air Force and former Kessel Run members launched more software factories in the Air Force, including Section 31, BESPIN, LevelUp, and SpaceCamp (Participants C and D). One respondent answered, "The largest impact is the conversation Kessel Run has started" (Participant B). By leading the way for change in Air Force software development, Kessel Run has started the conversation of operating differently and adopting industry's best practices and enabled the potential for more radical changes in the military.

Conclusions

We set out to understand why and how the Air Force established Kessel Run, and how its insourced approach benefits its customers when compared with the traditional outsourcing acquisition approach.

One of the research questions we wanted to answer in this case study was, "Why did the Air Force decide to internally develop (insource) software using the Kessel Run organization in lieu of the traditional method of contracting for software (outsource)?" Initially, we believed that rising software development costs drove the desire to create Kessel Run. Viewed through the lens of the make or buy decision, it appeared that the decision to insource made sense from a cost perspective. Based on spend data analysis, the cost of contracting for software development has increased dramatically over the last 5 years. As the failed AOC 10.2 project demonstrated, the cost of capabilities the Air Force sought to procure far outpaced the actual value of the end result. In an era of ever-tightening budgets and fiscal hawks seeking to cut costs, any budget item that shows increasing costs gets put in the crosshairs—sighted up for elimination. The spend data analysis supports the idea that software procurement costs are rising, and we anticipated interview responses to identify costs as a reason for establishing Kessel Run.

Despite the costly and doomed-to-fail AOC 10.2 program giving Kessel Run the opportunity to exist, and the general increasing trend in costs for software development, our analysis discovered that cost metrics were not a primary focus of the organization. Indeed, cost savings appear to be more of a welcome but unintended side effect of Kessel Run's success. Based on our analysis of official communication, interviews with Kessel Run members, and external sources, Kessel Run was created to continuously deliver war-winning software for the warfighter at an unprecedented pace (Kessel Run Acquisition and Contracting Playbook, 2019). Delivering capable and responsive capabilities to their end user is the motivator for Kessel Run personnel. Kessel Run, it appears, was created to re-insource and develop capabilities.

The second research question was, "How did the Air Force develop the Kessel Run organization?" Traditionally, military bureaucracy is slow to act and slow to change. In this organization, steeped in hierarchical structures and procedurally based actions, new initiatives take time to develop. Federal policy appears to prioritize commercial activity contracting versus competing with the private sector as well as cost reduction versus a core competency focus. However, Kessel Run sought to return software development capacity back to the military. Seemingly somewhat out of line with federal policy, the organization prioritized competency development with no direct goal of reducing costs.

Based on analysis of Kessel Run's published literature, personnel interviews, and information from outside of the organization, Kessel Run circumvented the traditional bureaucratic organizational structure by learning and imitating industry best practices. Founding members leaned heavily on startup literature such as *The Lean Startup* to establish the organization's identity. Organizing and behaving like a startup company instead of a traditional military organization enables Kessel Run to attract talent, swiftly respond to customer needs, and deliver capabilities at a rate unmatched in traditional government procurement.

Recommendations

Why does Kessel Run matter? Because the end-user and the Air Force mission matter. Kessel Run has adopted a proven civilian method to develop, administer, and maintain software in a very short amount of time and at a cost much lower than the traditional method of procurement.

The most impactful aspect of Kessel Run's success is the organization's delivery, administration, and maintenance of effective software for the warfighter. The organization's ability to overcoming the shortfall of the AOC 10.2 program through continuous capability development is an example of the support the warfighter deserves. Kessel Run is built around the Air Force's mission and a dedication to the warfighter. It is apparent that they pride themselves on their ability to rapidly deliver innovative state-of-the-art software in any domain at any time.

The second take-away is that Kessel Run has shown that alternatives to the traditional procurement cycle can be effective and can work within the Air Force. Kessel Run was able to circumvent the typical military bureaucracy by adopting the practices and methods of new-age startup tech companies as well as large established tech companies, like Google. Removing or reducing red tape and embracing an agile, innovative structure can deliver positive products, reduce timelines, and save money. Kessel Run is proof that deviating from the standard operating mode of military software development is possible and can benefit both direct end users and the Air Force as a whole.

Kessel Run has proven that a nontraditional organization can benefit the Air Force. However, that does not mean that a Kessel Run–type organization is the right response to all procurement shortcomings. Based on our findings, however, the Air Force should explore areas where vulnerabilities might exist if the market fails outsourced needs (Quinn & Hilmer, 1994), and that failure would represent unacceptable risk to mission execution and determine if insourcing the capability is a viable alternative. Although a startup-structured organization may not be the best way to insource in all situations, asking the question of whether something should be insourced will force the Air Force to identify areas of outsourced risk to mitigate.

Our review of federal policy revealed that policy prioritizes commercial contracting, except in specific circumstances, and prioritizes cost minimization. With that being said, Kessel Run appears to do the exact opposite; the organization has insourced previously outsourced capabilities and prioritized metrics that do not track cost. The benefits of focusing on developing software competency through insourcing has led to better, timely-delivered products. Further research should be done to review other federal government contracting efforts related to outsourcing capabilities to determine if the efforts are in line with federal policy or if they could benefit from deviating from policy as Kessel Run has done.

References

Air Force Business Intelligence Tool Lite - Profile | Tableau Public. (n.d.). Retrieved June 10, 2019, from https://public.tableau.com/profile/afbit#!/

Air Force Installation Contracting Center. (n.d.). Retrieved June 10, 2019, from https://www.wpafb.af.mil/aficc/

Air Force teaming with Pivotal to rapidly deploy software worldwide. (2019, April 19). WTOP. https://wtop.com/sponsored-content/2019/04/air-force-teaming-with-pivotal-to-rapidly-deploy-software-worldwide/

Archives Portal Category. (n.d.). Retrieved June 10, 2019, from https://www.fpds.gov/fpdsng_cms/index.php/en/archives-9.html

Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management*, *17*(1), 203–227. https://doi.org/10.1177/014920639101700108

Barney, J. (2002). *Gaining and sustaining competitive advantage*. Prentice Hall.

Bowen, H. K., & Spear, S. (1999). Decoding the DNA of the Toyota production system. *Harvard Business Review*, 96–106. https://www.leancompetency.org/wp-content/uploads/2015/12/Decoding-DNA-Spear-Bowen.pdf

Cerasoli, A., Janousek, R., & Mills, B. (n.d.). Decision 2025. *Economist*. Retrieved October 7, 2019, from https://www.economist.com/sites/default/files/creighton_ws.pdf

Chen, J. (2019, March 31). *Transaction costs*. https://www.investopedia.com/terms/t/transactioncosts.asp

Coase, R. (1937). The nature of the firm. *Economica, 4*(16), new series, 386–405. https://doi.org/10.2307/2626876

Congressional Research Service. (2017). Defense primer: DoD contractors. *In Focus*, *7*(5700), 2.

Davenport, C. (2019, January 10). Despite flaws, Air Force accepts Boeing's long-delayed and troubled tanker. *Washington Post*. https://www.washingtonpost.com/business/2019/01/10/despite-flaws-air-force-accepts-boeings-long-delayed-troubled-tanker/

Defense Acquisition University Powerful Example Library. (2019, February 27). *Powerful example: Kessel Run—Smuggling innovation into the DoD*. https://www.dau.edu/powerful-examples/Blog/Powerful-Example-Kessel-Run

Department of the Air Force. (2019, July 2). *Manpower and organization* (Air Force Policy Directive (AFPD) 38-1). https://static.e-publishing.af.mil/production/1/af_a1/publication/afpd38-1/afpd38-1.pdf
DoD. (2005, February 12). *The defense acquisition system* (DODD 1100.4). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodd/110004p.pdf
DoD. (1985a, March 10). *Commercial activities program* (DODD 4100.15). https://biotech.law.lsu.edu/blaw/dodd/corres/pdf2/d410015p.pdf
DoD. (1985b, September 9). *Commercial activities program procedures* (DODD 4100.33). https://biotech.law.lsu.edu/blaw/dodd/corres/pdf2/d410015p.pdf
Ehlinger, S. (2018, May 15). *Air Force looks to rapidly develop software with Project Kessel Run*. https://www.fedscoop.com/project-kessel-run-air-force-software-pivotal-inc/
Erwin, S. (2018, April 10). *Pentagon advisory panel: DoD could take a page from SpaceX on software development*. SpaceNews. https://spacenews.com/pentagon-advisory-panel-dod-could-take-a-page-from-spacex-on-software-development/
Federal Acquisition Regulation. (n.d.). Retrieved June 10, 2019, from http://farsite.hill.af.mil/
Federal Procurement Data System—Next Generation. (n.d.). Retrieved October 30, 2019, from https://www.fpds.gov/fpdsng_cms/index.php/en/
Feinberg, E., Benki, J. R., Berry, R., & Sylvester, J. (2018). *ForeSee experience index: Retail CX insights*. https://learn.foresee.com/hubfs/FXI_CX_Insights_Report.pdf
Flyvbjerg, B. (2006). Five misunderstandings about case-study research. *Qualitative Inquiry*, *12*(2), 219–245. http://doi.org/10.1177/1077800405284363
Forbes Technology Council. (2018, February 8). 13 reasons Google deserves its "best company culture" award. *Forbes*.https://www.forbes.com/sites/forbestechcouncil/2018/02/08/13-reasons-google-deserves-its-best-company-culture-award/#409489263482
Forecast International's Aerospace Portal. (n.d.). *Top-100 U.S. defense contractors FY 2018*. Retrieved October 17, 2019, from http://www.fi-aeroweb.com/Top-100-Defense-Contractors.html
George, A. L., & Bennett, A. (2005). *Case studies and theory development in the social sciences*. MIT Press.
House Armed Services Committee (HASC) Panel. (2010, March 23). *HASC panel on defense acquisition reform: Findings and recommendations.* GPO.
Henriksen, B., Rolstadås, A., & O'Sullivan, D. (2012). Make or buy? *Manufacturing Outsourcing*, 2012, 95–100. https://doi.org/10.1007/978-1-4471-2954-7_11
Highsmith, J. (2002, October). *What is agile software development?* 4–9. http://www.stsc.hill.af.mil
Hudgens, B. (2008). Strategic purchasing. In *Management of defense acquisition projects* (pp. 241–257). American Institute of Aeronautics and Astronautics.
Johnson, N. B. (2019, August 30). *The Air Force is implementing DevOps, here's what it means for airmen.* GovLoop. https://www.govloop.com/the-air-force-is-implementing-devops-heres-what-it-means-for-airmen/
Jones, C. (2002). Defense software development in evolution. *CrossTalk*. Presented at the The Journal of Defense Software Engineering. https://pdfs.semanticscholar.org/7dc1/db33fff828edd7d456e0f9f30dbc4055f553.pdf
Kanaracus, C. (2012, November 14). *Air Force scraps massive ERP project after racking up $1 billion in costs*. CIO. https://www.cio.com/article/2390341/air-force-scraps-massive-erp-project-after-racking-up--1-billion-in-costs.html
Kelman, S. (2018, June 18). *Agile at DoD—Going beyond sticky notes and standing meetings*. FCW. https://fcw.com/blogs/lectern/2018/06/kelman-dod-agile.aspx
Kelman, S. (2019a, February 12). *Why Kessel Run is such a big deal*. FCW. https://fcw.com/blogs/lectern/2019/02/kelman-kessel-run-usaf-big-deal.aspx
Kelman, S. (2019b, July 2). *A closer look at the Air Force's in-house software development.* https://fcw.com/blogs/lectern/2019/07/kelman-usaf-software-development-in-house.aspx
Kessel Run. (n.d.). Retrieved October 30, 2019, from https://kesselrun.af.mil/mission/
Klein, P. (2013). Transaction cost theory. In *Encyclopedia of Management Theory*. SAGE Publications. http://dx.doi.org/10.4135/9781452276090.n299
Leonard-Barton, D. (1992). Core capabilities and core rigidities: A paradox in managing new product development. *Strategic Management Journal*, *13*(S1), 111–125. https://doi.org/10.1002/smj.4250131009
Levy, J. S. (2008). Case studies: Types, designs, and logics of inference. *Conflict Management and Peace Science*, *25*(1), 1–18. https://doi.org/10.1080/07388940701860318
Ma, D. (2007). The business model of software-as-a-service. *IEEE International Conference on Services Computing (SCC 2007), 701–702.* https://doi.org/10.1109/SCC.2007.118

Manfred, T. (2013, February 26). Tom Brady signed a contract for $30 million less than he's worth—Here's why. *Business Insider.* https://www.businessinsider.com/tom-brady-less-money-contract-2013-2
National Research Council. (2007). *Software for dependable systems: Sufficient evidence?* https://doi.org/10.17226/11923
Newell, B. (2018, May 15). *Kessel Run lab hits hyperdrive.* Department of Defense. https://www.defense.gov/Newsroom/News/Article/Article/1522214/kessel-run-lab-hits-hyperdrive/
Oliver, C. (1997). Sustainable competitive advantage: Combining institutional and resource-based views. *Strategic Management Journal*, *18*(9), 697–713. https://doi.org/10.1002/(sici)1097-0266(199710)18:9<697::aid-smj909>3.0.co;2-c
Office of Management and Budget. (1983). *Performance of commercial activities* (OMB Circular A-76) https://www.whitehouse.gov/sites/whitehouse.gov/files/omb/circulars/A76/a076.pdf
Office of the Secretary of Defense. (2015, October 15). *Government-wide category management guidance.*
Other Transaction Authority Brief. (2016, April 6). https://www.transform.af.mil/Portals/18/documents/OSA/OTA_Brief_Ver%206Apr2016.pdf
Pomerleau, M. (2019, January 14). *How the Air Force's new software team is proving its worth.* C4ISRNET. https://www.c4isrnet.com/it-networks/2019/01/14/how-the-air-forces-new-software-team-is-proving-its-worth/
Prahalad, C. K., & Hamel, G. (1990). The core competence of the corporation. *Harvard Business Review*, 79–91. http://web.b.ebscohost.com.libproxy.nps.edu/ehost/pdfviewer/pdfviewer?vid=1&sid=650a974a-5d61-41e6-8646-f1f94802da8f@sessionmgr101
Quinn, J. B., & Hilmer, F. G. (1994). Strategic sourcing. *Sloan Management Review*, *35*(4), 43–55. http://libproxy.nps.edu/login?url=https://search-proquest-com.libproxy.nps.edu/docview/224960020?accountid=12702
Rosenberg, B. (2019, June 20). *Fail fast, not twice: DoD's push for agile software development.* Breaking Defense. https://breakingdefense.com/2019/06/fail-fast-not-twice-dods-push-for-agile-software-development/
Schilling, M. A. (2013). *Strategic management of technological innovation* (International Ed.). McGraw-Hill Education.
Tellis, W. M. (1997, July 1). *Introduction to case study*. 3(2). https://nsuworks.nova.edu/cgi/viewcontent.cgi?article=2024&context=tqr/
U.S. Bureau of Labor Statistics. (n.d.). *Consumer Price Index*. Retrieved October 30, 2019, from https://www.bls.gov/cpi/
Ward, D. (2019). *Vignette 5—Kessel Run: The future of defense acquisitions is #AgileAF*. 3.
WatchBlog. (2019, May 28). *Federal government contracting for Fiscal Year 2018.* https://blog.gao.gov/2019/05/28/federal-government-contracting-for-fiscal-year-2018-infographic/
Wheeler, W. (2014, October 7). *Refusing to misunderstand the defense acquisition problem*. POGO. https://www.pogo.org/investigation/2014/10/refusing-to-misunderstand-defense-acquisition-problem/
Williams, L. C. (2018, October 4). *Air Force may require agile for all acquisitions*. Washington Technology. https://washingtontechnology.com/articles/2018/10/04/usaf-kessel-run-standard.aspx
Williamson, O. (1981). The economics of organization: The transaction cost approach. *American Journal of Sociology*, *87*(3), 548–577. http://www.jstor.org.libproxy.nps.edu/stable/2778934
Williamson, O. E. (1996). *The mechanisms of governance.* https://ebookcentral.proquest.com
Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.
Zukin, S., & DiMaggio, P. (Eds.). (1990). *Structures of capital: The social organization of the economy*. CUP Archive.

PANEL 20. A GLOBAL PERSPECTIVE ON DEFENSE ACQUISITION

Thursday, May 13, 2021	
10:45 a.m. – 12:00 p.m.	**Chair: Rear Admiral Francis D. Morley, USN,** Director, Navy International Programs Office ***How to Measure Value from Defence Spending? The Malaysian Case Study*** Kogila Balakrishnan, WMG University of Warwick ***Rapid Innovation with Chinese Characteristics: National Defense Science and Technology Innovation Rapid Response Teams and the Military-Civil Fusion Innovation Ecosystem*** Emily de La Bruyere, Horizon Advisory, Association for the New Century Nathan Picarsic, Horizon Advisory, Association for the New Century

Rear Admiral Francis D. Morley, USN— Rear Adm. Francis D. Morley is a native of Phoenix, Arizona. He earned a Bachelor of Science in Physics and a commission as an ensign from the Naval Reserve Officer Training Corps at San Diego State University. He is a graduate of the U.S. Naval Test Pilot School and holds a Master of Science in Aviation Systems from the University of Tennessee. He is a graduate of the Air Command and Staff College, Joint Forces Staff College, Defense Systems Management College, George Washington University National Security Studies Program and Harvard's Kennedy School of Government National and International Security Program.

Morley served operationally as an F/A-18 pilot in Strike Fighter Squadron (VFA) 192 deployed with USS Independence (CV 62), VFA-83 deployed with USS George Washington (CVN 73) and command of VFA-87 deployed with USS Theodore Roosevelt (CVN 71). He also served as assistant navigator deployed with USS Enterprise (CVN 65). During these tours, he participated in operations Southern Watch, Desert Fox, Noble Eagle, Joint Guardian, Enduring Freedom and Iraqi Freedom.

His shore tours include service as a test pilot in Air Test and Evaluation Squadron (VX) 23, including being an F/A-18 E/F engineering, manufacturing and development test pilot where he was the fifth pilot to fly the super hornet and first to land aboard ship; Chief, Air and Missile Defense (J3) U.S. Pacific Command; deputy program manager (PMA-265) where he directed the final year of development and fleet introduction of the EA-18G Growler; program manager, F/A-18 and EA-18G Program Office (PMA-265); and most recently, as vice commander of Naval Air Systems Command headquartered in Patuxent River, Maryland.

In September 2016, he assumed duty as director, Navy International Programs.

Morley has been recognized as the Commander, Naval Air Force Atlantic ship handler of the year and the Department of the Navy program manager of the year. He has more than 3,500 flight hours and 750 carrier arrested landings. He has flown more than 35 different types of aircraft, including the F/A-18A-F, EA-18G, AV-8B, F-14, F-15, F-16 and MiG-29.

How to Measure Value From Defence Spending? The Malaysian Case Study

Kogila Balakrishnan—is the Director for Client and Business Development (East Asia) at WMG, University of Warwick; Adjunct Professor at the Malaysian National Defence University; and the Former Under Secretary of the Department for Defence Industry, Ministry of Defence, Malaysia. He is the author of *Technology Offsets in International Defence Procurement* (Routledge, 2018). [Kogila.balakrishnan@warwick.ac.uk]

Abstract

More than ever, nations and their citizens demand clear evidence as to the benefits of defence spending and any subsequent value created. The opportunity cost of defence expenditure against other sectors is constantly queried, and the question of how to measure value remains highly contentious. Similarly, it is challenging to find consensus amongst traditional economists, policy-makers, and other stakeholders on how to measure value in the context of defence, where most outcomes are intangible. Hence, this article attempts to offer solutions using the case study of Malaysia, a maritime nation in Southeast Asia. The study uses a hypo-deductive approach underpinning qualitative research methodology. Primary data sources include open-ended and semi-structured interviews to produce a thematic discussion, as well as secondary resources such as journal articles, government reports, and online sources. The author argues that it is hard to appraise defence value, as measurements are case-specific, and even successful attempts cannot be generalised. Rather, the paper will use a novel "Triple-Defence Value Framework" to argue that value can be measured by dividing the role of defence into a primary level: for protection and safety; a secondary level: for socio-economic prosperity; and a tertiary level: for soft power projection. The paper concludes by using the framework to measure the value derived from the Malaysian defence sector.

Key words: defence value, defence spending, defence economics

Setting the Scene

Is Adam Smith's (1776) statement that "defence is more important than opulence" still relevant today? There is continuous debate justifying defence spending, which is often seen as unnecessary and providing poor return on opportunity costs compared to other sectors. In the United Kingdom (UK), for example, taxpayers continue to question the government's highly controversial decision to maintain, let alone replace, the Trident nuclear submarine fleet. Nonetheless, we have witnessed an ongoing and significant rise in global defence spending since the Second World War. In 2018, global defence spending totalled approximately U.S. dollars (USD) $1.8 trillion (Stockholm International Peace Research Institute, 2020). Globally, the United States of America (USA) remains the largest defence spender at USD $648.8 billion (3.2% of GDP),[1] followed by China at USD $250 billion (1.9% of GDP; Stockholm International Peace Research Institute, 2020). Figure 1 shows the world's top defence spenders in 2018 (Stockholm International Peace Research Institute, 2020). Traditional economists argue that defence is one of the few areas requiring "the expense of sovereign and commonwealth," justifying state intervention in the economy. Hitch and McKean (1960), in their seminal piece, *The Economics of Defense in the Nuclear Age,* also highlighted the deliberate choice that has to be made between a nation changing its national budget and reshaping its armed forces as long as that change appears to gain more than its costs (Hitch & McKean, 1960).

[1] Since September 2001, Congress has apportioned an additional discretionary $2 trillion as emergency requirements for Overseas Contingency Operations/Global War on Terrorism (OCO/GWOT).

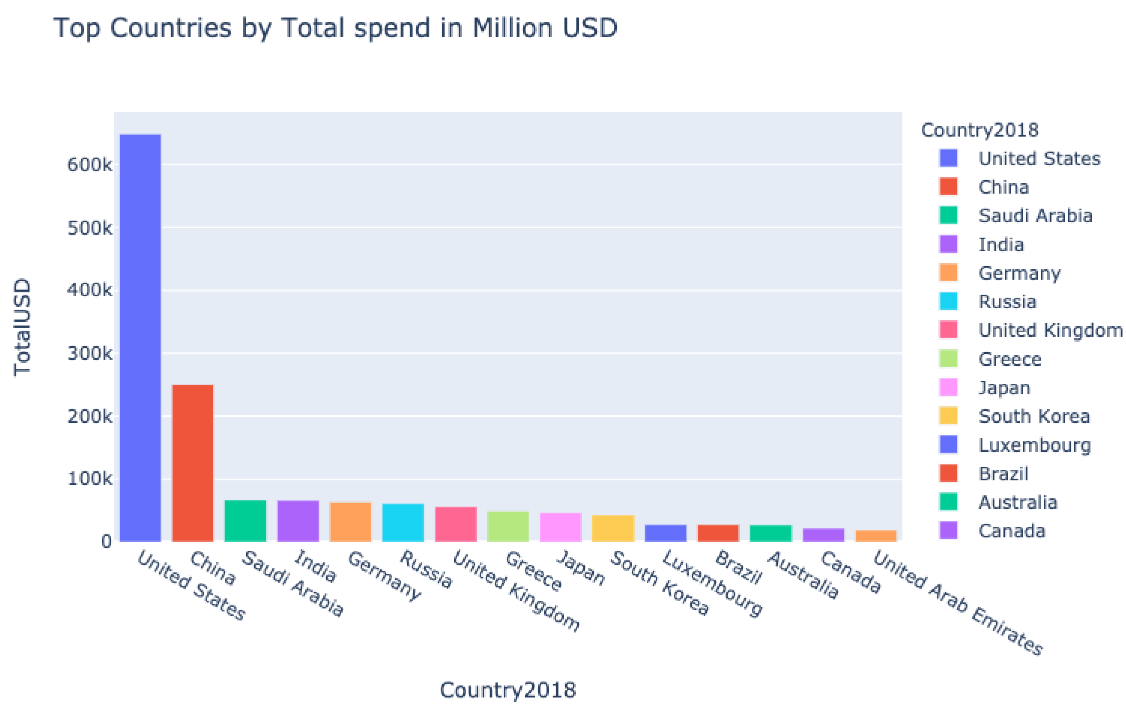


Figure 1. Top Countries by Total Spend in Million USD (Stockholm International Peace Research Institute, 2018)

The attractiveness of an assessment of the value derived from defence expenditure is self-explanatory but difficult to define. There is constant debate over the nuances encompassing value and the importance of measuring the input, output, and impact derived. Some have also questioned the difference between value-creation and value-extraction (Mazzucato, 2018). The topic of value becomes even more complex when seen in the context of defence expenditure. An oft-mentioned term in defence is the concept of “value for money” (VFM), which is used to justify defence procurement activity and relies on evidence-based claims utilising in-depth evaluation and current research findings.

Most studies have addressed this topic from an economic perspective, discussing the subject from a principal–agent model or cost–benefit analysis, often struggling to integrate the hard evidence of value derived from defence expenditures. Practitioners in the field have attempted to address this topic from a more pragmatic angle, but nonetheless apply several economic principles to explain the concept of value and the rationale behind measuring value from defence expenditure. Therefore, the author has applied the proposed theoretical framework to the real-world example of Malaysia. This choice was due to the author’s familiarity with the Malaysian defence context and, subsequently, greater ease in obtaining data, which can be extremely difficult in the defence sector.

The framework has considered input from various sources including academic papers, government reports, and news articles, supported by interviews with experts in the field, policy-makers, commercial stakeholders, and military personnel. The author was indirectly involved in providing input and guidance to senior stakeholders when the government developed its 2020 Defence White Paper (DWP; Malaysia Ministry of Defence, 2020). This article is divided into three sections. The first section critically analyses the meaning of value, theories that relate to value, and the concept and complexity of measuring value in the defence context, and proposes a theoretical framework on measuring value. The second section briefly outlines Malaysia's DWP and evaluates why measuring value is important for the Malaysian defence sector. The third section evaluates how the Malaysian defence sector is delivering value to taxpayers. This section also critically discusses the challenges faced by Malaysia in delivering value and raises some intriguing questions on how Malaysia can more effectively boost defence value through defence spending.

Measuring Value in the Defence Sector

What is Value?

The concept of value received less attention in the field of economics. The theory of value originated in the 17th century when Francis Quesnay, a French economist, formulated the first systematic theory of value, displaying how value was being created and circulated in the economy. The 18th century economics associated value to land and agricultural productivity and farmers as creating value. The 18th and 19th century classical economists, including Adam Smith, David Ricardo, and Karl Marx, measured value by labour invested in an activity and the successful adoption of new technologies (Malaysia Ministry of Defence, 2020). However, the 20th century neoclassical economists defined value as a function of demand and supply and saw value as utility, or perceived property.

What is value? How and from whose perspective is value being measured, and against what? Value can be direct, numerically-measured (quantifiable), or indirect, such as intrinsic value. Value can be labelled as shared value, VFM, or added-value, and even sometimes measured by one's conduct and behaviour. The theory of value examines the subject from several perspectives, including ethics, politics, economics, and philosophy. From a utilitarian perspective, the greatest value lies in that which provides the greatest utility to the greatest number, whilst in a business context, value for a shareholder can reside in profit maximisation, as opposed to other stakeholders who may find greater value in innovation. Overarchingly, it is often argued that any definition of value and how it is measured is prescriptive.

At the core of economic thinking, value is defined as price set by supply and demand. Hence, when an activity results in a payment, this is seen as value. Value can also be seen as an intermediary process or "flow," such as adding value to a manufacturing supply chain or creating new knowledge bases (Porter, 1998). Then there are further riddles as to how these outputs are produced, how they are shared across society, whether production is useful, and what is done with subsequent earnings.

Value Derived by Governments

The public sector is often castigated by some commentators as being "unproductive" in delivering value, as government expenditure is funded by taxing the productive parts of the economy. However, these critics do not fully recognise public sector contributions in value-delivery. The simplistic view posits government as an acute example of an unproductive sector. Others argue that government expenditure in delivering value should be seen as a process, or intermediary, that creates the conditions for a peaceful state, connected through infrastructure, police, national defence, and the rule of law that allows for production and services, all of which contribute to a nation's prosperity. Intrinsic value generated through government intervention cannot be analysed in the same way as other sectors of the economy. Even traditional economists have recognised the important role of government in providing sufficient investment into sectors such as the military, judiciary, and other essential public services, as they provide the basic institutional and physical infrastructure necessary for economic growth. The public sector then pursues identified goals, manages the inputs, and convinces taxpayers of the value delivered through spending (United Kingdom HM Treasury, 2017).

Valuing the Impact of Defence and Armed Forces

Valuing the impact of defence is complex, often sparking a debate over the benefits and tangible outcomes derived from defence to a nation. Defence is often considered an opportunity cost dilemma, with some suggesting it has limited direct economic benefits. Therefore, the challenge is in demonstrating the value of defence and effectively communicating this to civil society. This value may be intangible: How do you measure risk and protection? How can you convincingly prove a negative that defence spending results in threat deterrence? Can we argue that defence is essential in delivering the requisite protection and safety to conduct business and create order? In defence, measuring direct and indirect value is tedious and often becomes subjective. What is being measured? Is it the costs and benefits, inputs and outputs, outcomes or impacts? Defence has to articulate how activities improve the quality of life for every citizen, or it will fail to adequately justify defence expenditure or prove the existence of viable returns to taxpayers (Paul et al., 2015). A common dilemma arises when considering the optimal size of national armed forces and defence budget allocations as a proportion of gross domestic product (GDP). Figure 2 shows the proportion of defence budget against GDP for the top 20 countries with the highest defence spending in 2019. For example, military spending in the Kingdom of Saudi Arabia (KSA) represents 8.8% of its GDP, one of the highest in the world (Stockholm International Peace Research Institute, 2020). Despite the fact that the KSA remains embroiled in an ongoing war in Yemen, is this spending rational relative to the KSA's overall national income? Disproportionate defence spending at the expense of other sectors can diminish overall value. In order to be treated as critical, the defence sector and the armed forces must invest in the value-creation process in order to demonstrate that their services are essential in both war and peacetime.

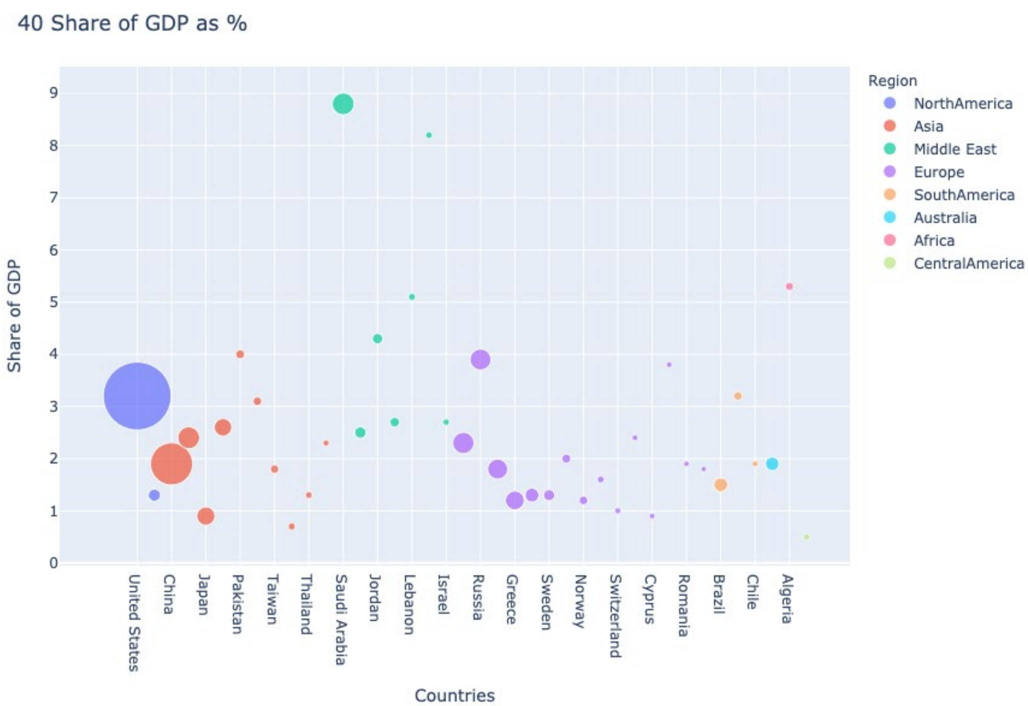


Figure 2. Proportion of Defence Budget Against GDP for the Top 20 Countries (Stockholm International Peace Research Institute, 2019)

Several defence economists have questioned the efficiency of national defence represented by the armed forces and domestic defence, and whether they provide value for money (Benoit, 1973; Hartley, 2012; Melese et al., 2015). Defence inputs were identified as personnel, equipment, and supporting infrastructure, as part of a defence spending policy. Defence output includes measures such as the protection of citizens and their assets, businesses, economic infrastructure, national institutions, and natural resources, but also as deterrence, warfighting capability, disaster and humanitarian relief, as well as economic contributions to employment and possibly even exports. By contributing to international peace and stability, defence enables globalised trade, exchange, and foreign direct investment, thus contributing to national prosperity through endeavours such as the safeguarding of shipping against piracy. Prosperity impacts from defence spending include innovation and spin-offs, human capital investments, local economic impact, and long-term influences on local labour markets.

There are several government reports that have tried to incorporate value into how defence policies can impact society. The 2017 UK Defence Industrial Policy (DIP) referred to defence delivering “wider economic, international value and national security objectives” (Williamson, 2017). The report highlights how defence procurement strengthens productivity (especially at the local level), boosts exports, and contributes to national prosperity on a sustainable basis (UK Defence Industrial Strategy, 2017). The Philippe Dunne (2018) report titled *Growing the Contribution of Defence to UK Prosperity* was the outcome of an enquiry by the National Audit Office (NAO) on the outcome and impact of defence spending to British citizens. The report attempted to validate the national contribution of defence and the armed forces and measured the contribution of defence to economic growth, national life, people, ideas and innovation, and location, five core factors used to justify UK defence spending (Dunne, 2018).

Again, the 2018 UK Ministry of Defence document titled *Mobilising, Modernising, and Transforming Defence* identified three key themes. It scrutinised defence value through broad national security objectives: to protect our people; to project our global influence; and to promote our prosperity (3P; United Kingdom Ministry of Defence, 2018). This concept of value was also discussed at a January 2020 workshop titled “Prosperity and Value: What is Defence’s Triple Bottom Line?” organised by the International Institute for Strategic Studies (IISS) in conjunction with the Development, Concepts and Doctrine Centre (DCDC) at the Defence Academy of the United Kingdom (IISS, 2020). The workshop represented various stakeholders yet managed only to scratch the surface in terms of consensus on a definitive qualitative measure for the outcome and impact of defence spending as tied to UK prosperity (Wylie, 2017).

Technological innovation and spin-offs to the civil sector have always been cited as a major contribution from defence spending. Many major innovations from the past were spin-offs through investment in defence research and development (R&D). The internet was originally developed for secure military communication. ATM machines, from which most of us receive cash, rely on the Global Positioning System (GPS) for theft protection—originally developed for U.S. military communication. Flat screen televisions, radar, laptops, tablet computers, and touch-screen mobile phones use liquid crystal displays, another invention made possible by past military R&D. More recent examples of technology crossing over from military to civil use are autonomous cars, the voice assistant SIRI, and thermal imaging. The defence sector has also significantly invested in education and training. If effectively utilised, this investment into human capital should be translated into knowledge and capabilities that increase a country's productive capacity and aid overall potential growth.

In the context of this paper, the TDV framework as seen in Figure 3 is used to measure value derived from defence spending. TDV is a suggested policy framework that could be applied by policy-makers and practitioners to measure the value outcome from defence spending. The framework was developed using a variety of sources ranging from secondary literature and government reports to interviews with stakeholders and workshops attended by the author. The data gathered to measure value in this context were mostly qualitative, as it can be difficult to obtain quantitative data in a defence context. The model is mainly aimed at measuring the outcomes of defence spending, though it can be argued that specific processes in defence can also be counted as producing value. For example, the act of defence diplomacy is a process that is highly valuable as it requires complex negotiation and networking to thwart war. Unfortunately, such processes do not get captured when measuring value in defence.

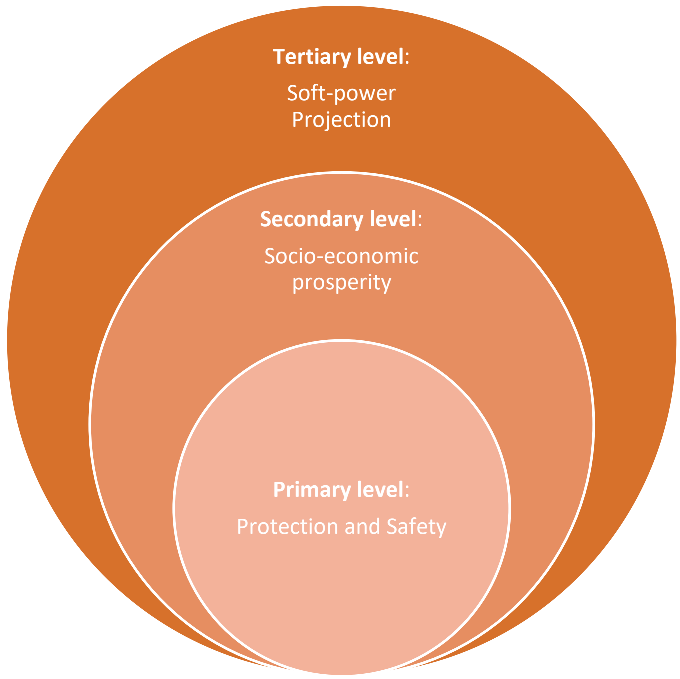


Figure 3. Triple-Defence Value Framework (TDV)

In the context of this paper, we measure the outcome from defence activities as opposed to the processes in defence. The primary role of the armed forces and the Ministry of Defence is to protect citizens at home and abroad, protect critical infrastructure, prevent conflict, secure national borders and overseas territories, and ultimately deliver peace. In order to achieve these roles, the armed forces require the ability to access and anticipate future security threats, including direct military action and terrorism or cyber-threats, and the subsequent ability to plan around specific capabilities required to face identified threats. These capabilities would include equipment, training, information, concept and doctrine, personnel, infrastructure, organisation, and logistics. The defence budget is then used either to procure the required capability, fund indigenous production through the allocated R&D budget, or to enter into collaborative partnerships. In the context of the TDV framework, primary value could be derived as the level of readiness of units, their warfighting capability, and their subsequent deterrent effect. However, empirical measurement of these primary value indicators poses a significant challenge as data is either not available or confidential.

The secondary level of value in this framework is national economic benefit from defence spending and benefits to society. The economic benefit of defence spending is derived from factors such as the contribution to technological innovations, exports, employment, intellectual property rights (IPRs), foreign direct investments, industrial partnerships and offsets, regional development through building regional clusters and technology parks, and a number of small and medium size industries (SMEs) supported through defence activities. Societal benefit is measured, first, by costs to society if attacked by enemies due to loss of infrastructure, jobs, ability to conduct business, and an overall reduction in GDP. In economic terms, this is also known as protection adjusted life years (PALYS). Second, the volume and extent of participation of the armed forces in peacekeeping missions, engagement in disaster relief, border control, sea patrol, rescue missions, and humanitarian aid, as well as conducting crisis management such as during a pandemic or incidents caused by climate change. The defence sector's other ancillary societal value also includes its ceremonial roles, supporting veterans and their families, contributing to armed forces pensions, and adding value through civil–military integration during crisis and reconstruction phases.

The tertiary level of value is defined as soft power projection. Value in this context is defined as capacity-building measured through training and education programmes (including their quality and relevance); the numbers of trained military personal; and promotion of universities, strategy-oriented think-tanks, and other defence-related institutions, both domestically and internationally, that can help establish credibility. Soft power is also used to promote the "rules-based international order" and may be measured through numbers of bilateral and multilateral defence co-operation agreements and alliances, joint-military exercises, participation at defence exhibitions, air-shows, and international defence conferences or fora.

Measuring Defence Value: The Case of Malaysia

The Defence White Paper

The Malaysian government published its first Defence White Paper (DWP) in January 2020 (Parameswaran, 2019). The Malaysian Ministry of Defence (MINDEF) highlighted three very ambitious reasons for drafting the DWP (Cheng, 2020): first, to foster a transparent leadership process through change in how the government delivers value to the public (Abbas, 2019); and second, to inform the Malaysian public on the role of the armed forces, but also to promote Pertahanan Menyeluruh (HANRUH; Hamzah, 2016)—or total defence,[2] a concept introduced to express the collective responsibility of government and civil society in managing security and handling crises. The third and most relevant reason is specifically to inform the Malaysian public on the value derived from defence spending.

The DWP was drafted based on consultation and dialogue (Balakrishnan, 2019) amongst various governmental agencies, extensive meetings with defence and security circles, overseas governments,[3] and members of the public (Chwee & Hamzah, 2019). Formulation was spearheaded by the Malaysian Institute for Defence and Security (MIDAS) together with the Strategic and Policy Planning Division of MINDEF, a technical team from the Armed Forces, academics, and the defence industry sector. The final version of the DWP was tabled on December 2, 2019 (New Straits Times, 2019). The DWP explains the ministry's vision, strategy, and implementation of defence policy, and has a timeline of 10 years (2020–2030; Malaysia Ministry of Defence, 2020). The policy document is aligned to the National Security Policy (NSP),[4] National Foreign Policy (NFP; Ministry of Foreign Affairs Malaysia, 2019), and the National

---

[2] The concept of total defence or comprehensive defence was used by Singapore (1984), modelled after countries like Sweden, Denmark, and Austria. The six pillars of total defence include military, civilian, economy, social, psychological and digital defence. For further details, see *What is Total Defence?*, https://www.mindef.gov.sg/oms/imindef/mindef_websites/topics/totaldefence/about.html.

[3] During the interview with Vice Admiral Dato' Ganesh Navaratnam, president at the National Centre for Defence Studies (PUSPAHANAS), he stated that the DWP was closely modelled after the 2016 Australian Defence White Paper. Malaysia had also consulted other overseas countries such as the UK, New Zealand, France and Germany on their experiences of formulating the DWP.

[4] The National Security Policy (NSP) was formulated under the National Security Council Act 2016 (Act 776). Refer to National Security Policy, 2017, https://www.pmo.gov.my/2019/07/national-security-policy/

Defence Policy (NDP).[5] Nonetheless, the DWP has faced criticism in that the content is not significantly different from the 2010 Defence Policy Paper.[6] Others argue that the DWP has "loopholes" and that it has failed to address the real challenges facing Malaysia, and some say that the paper has failed to demonstrate empirical value derived from Malaysia's defence spending (Rodzi, 2019). However, there seems to be strong support from stakeholders in adhering to the DWP and proposed plans.

How does the Malaysian government define value? Is this different from general value terminology? There is no hard evidence regarding its usage of the concept of "value," except the term "value for money" in procurement policies. The Ministry of Finance's treasury circular refers to the Outcome Based Budget (OBB) introduced since the 10th Malaysia Plan (2010–2016), used to measure outcomes from government spending. Each Malaysia Plan is a 5-year comprehensive economic development blueprint prepared by the Economic Planning Unit within the Prime Minister's Department (Portal Rasmi, 2020). Unfortunately, the results are yet to be available in the public domain. Hence, the value derived from Malaysia's defence spending based on the DWP in this paper is measured using the TDV model.

Defence Context

It is contentious whether defence spending has added-value and contributes to Malaysia's prosperity. It is often argued that Malaysia's internal security counters more immediate threats, and hence is more critical than defence. However, defence is still seen as an integral component of Malaysia's comprehensive security environment. The 2010 Malaysian defence policy specified that the primary role of defence is to protect and defend the nation's strategic interest, with sovereignty, territorial integrity, and economic wellbeing as core factors.[7] Malaysia is a parliamentary democracy with a multi-ethnic population of around 32 million, located strategically within sea lanes between the South China Sea, Sulu Sea, the Andaman Sea, and the Straits of Malacca, through which 30% of global seaborne trade passes (Malaysia Ministry of Defence, 2020). As a small littoral state with limited defence capability, Malaysia projects a defensive posture whilst actively promoting defence diplomacy as the first line of defence through bilateral and multilateral negotiations at ASEAN, regional, and global levels (Noor & Qistina, 2017). However, the DWP recognises there are imminent, immediate, and future threats facing the country and that a substantive level of deterrence capability is essential. Hence, a credible defence force is crucial to protect East and West Malaysia's land, sea, and air borders.

Defence Strategy

According to the DWP, the Malaysian defence strategy is grounded on three pillars: concentric defence, comprehensive defence, and credible partnership (Malaysia Ministry of Defence, 2020). The first pillar refers to the concept of concentric deterrence that divides strategic interests into base, extended, and front-line areas (Noor & Qistina, 2017). The second pillar refers to the role of the "whole of government" and "whole of society" and emphasises the importance of internal cohesion in building defence resilience. The third pillar refers to building, strengthening, and widening collaboration and external relations through bilateral and multilateral platforms. However, the document is vague on strategic priorities, how these strategies will be operationalised, or the targeted resources required for this purpose.

---

[5] Noor Hisham (director general, Malaysian Institute of Defence and Security, MIDAS, Kuala Lumpur), in discussion with the author, February 4, 2020.

[6] The National Defence Policy (NDP) is a classified document produced in 1971 and reviewed several times in 1979, 1981, 1986, 1993, and 2006 prior to the DWP. An open version of the NDAP was published in 2010.

[7] Ahmad Nadzri (deputy secretary general [Policy], Ministry of Defence, Malaysia), in discussion with the author, January 15, 2020.

Defence Spending and Budget Allocation

The question remains whether Malaysia has targeted sufficient and effective investment towards its defence budget. Figure 4 illustrates Malaysia's defence spending over the past 10 years with a total figure of USD $3.827 billion (1% of GDP) in 2019. Average defence spending (highest in 2003 at around 2.3% of GDP) has been constantly declining to around 1–1.5% of GDP since 2014 (Subramaniam et al., 2018). This figure is still higher than many other ASEAN countries, but lower than Singapore and Indonesia. This downward spiral is mainly attributed to overall government budgetary constraints and the country's economic turmoil, but will now be further exacerbated by the financial impact of COVID-19. This negative trend has certainly been a concern considering the Malaysian Armed Forces' (MAF) lagging defence capabilities and the country's inability to address some of the critical security challenges both facing Malaysia and looming in the region (Hosoyo, 2020).

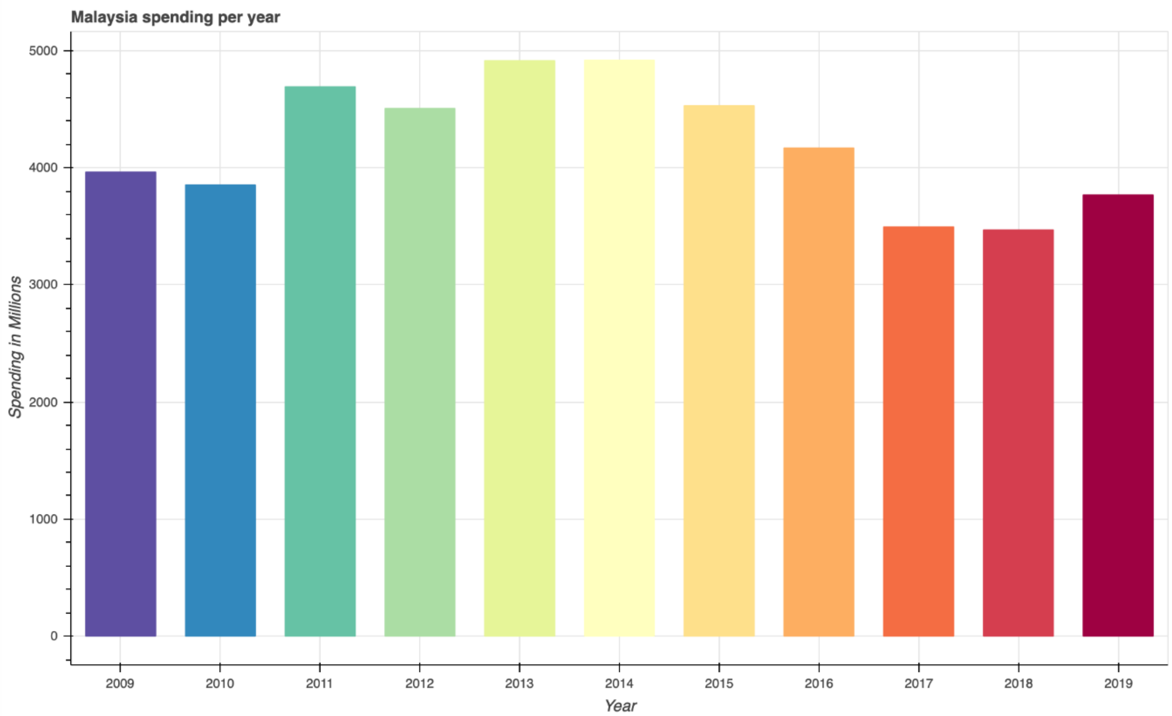


Figure 4. Malaysia's Defence Spending (2009–2019) in USD (Stockholm International Peace Research Institute, 2020)

Figure 5 illustrates the bulk of Malaysia's defence budget, of which 77.5% is allocated to operating expenditure (OE; Subramaniam et al., 2018). The development expenditure (DE) of 22.5% is allocated to the procurement of equipment and services, infrastructure development, R&D, and other procurement-related costs (Economic Planning Unit, 2018). In the DWP, there is mention of long-term funding streams and efforts to reduce the existing funding gap in the MAF. However, the budget lacks clarity on projected future allocation, priority of expenditure as per services and sectors, and sources of funding. The white paper would have been more effective had it shown greater clarity in identifying top priorities. Further, as defence management is currently under the purview of the federal government, there must be greater co-operation between the federal and state levels in order to realise the concept of "whole of government." Nonetheless, each state has devolved powers to run administrative functions and manage revenue and budgets. Going forward, the lack of transparency on levels of future defence spending may have an adverse effect on commitments to procuring and maintaining military capability, especially in the post COVID-19 era, unless the government continues to see the importance of defence as contributing to national security.

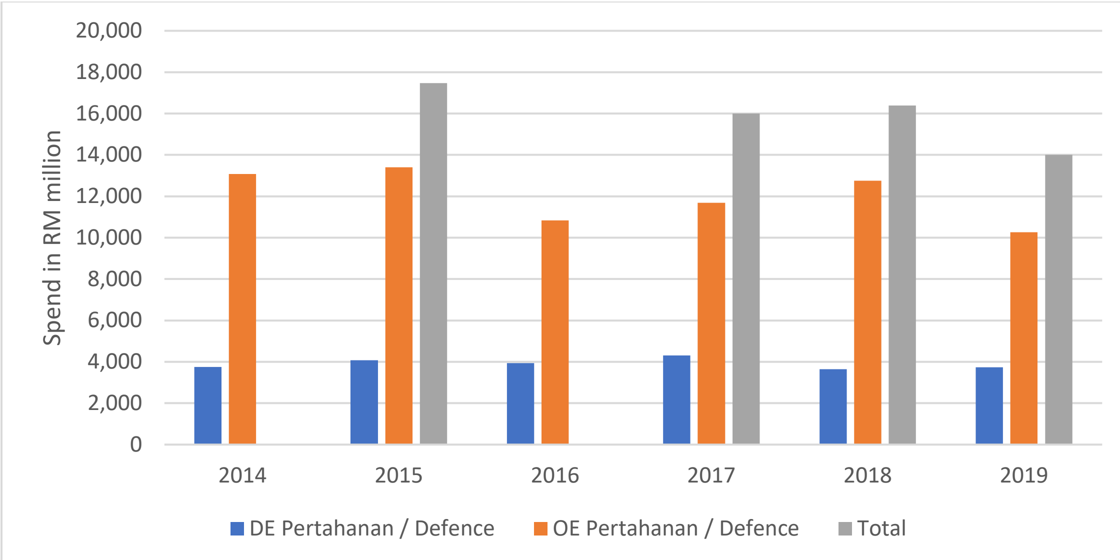


Figure 5. Malaysia: Defence Expenditure by Development (DE) and Operational Expenditure (OE; 2014–2019) (Ministry of Finance, Malaysia, 2020, https://www1.treasury.gov.my/index.php/fiskal-ekonomi/data-ekonomi.html)

Strength of the Armed Forces

Malaysia possesses a reasonable body of active military manpower, retaining the fourth largest military establishment within ASEAN. Figure 6 shows a detailed breakdown for the three services of the MAF between 2017 and 2019.

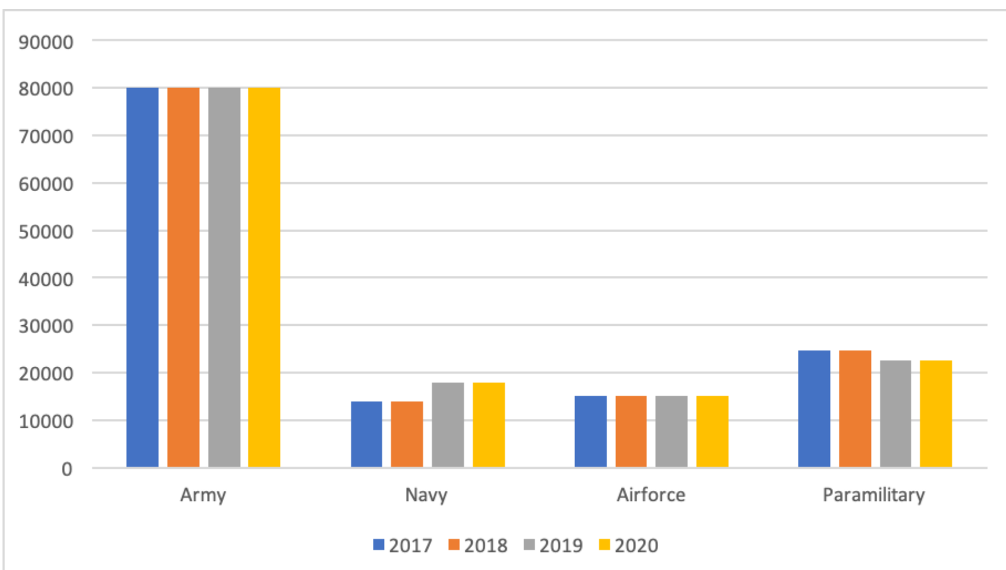


Figure 6. Malaysia: Size of the MAF by Services and Para-Military (2017–2020) (IISS, 2020)

The ceiling for military personnel has slightly increased from 109,000 in 2017 to 113,000 in 2020, mainly due to the intake of some 4,000 navy recruits (IISS, 2020). Figure 7 illustrates Malaysia's armed forces per capita, at 14 military personnel for every 1,000 people, and compares this to several other ASEAN countries. The army is still considered the backbone of Malaysia's defence and the core of civil–military integration, as well as providing support to the police during internal disorders and crisis situations, which are equally important. However, the strength of the navy has become increasingly critical in the past 5 years, as Malaysia views itself as a maritime nation becoming increasingly vulnerable to challenges to its territorial waters and resources, increasingly due to escalated tensions between major powers in the South China Sea. There is no mandatory conscription service, though National Service for youth aged 18 was introduced for a short while from 2009–2017. Still, since its abolition, there has been interest in whether another structured programme should be developed to engage youth into defence.[8] Despite a multi-ethnic population, Malays form 95% of the MAF and the civil service, with senior MOD officials claiming that it is hard to attract and recruit non-Malays into government positions.[9] The greater involvement of youth and non-Malays in the defence force may be vital in realising diversity and inclusion involving the whole of society in order to best deliver value through defence in Malaysia.

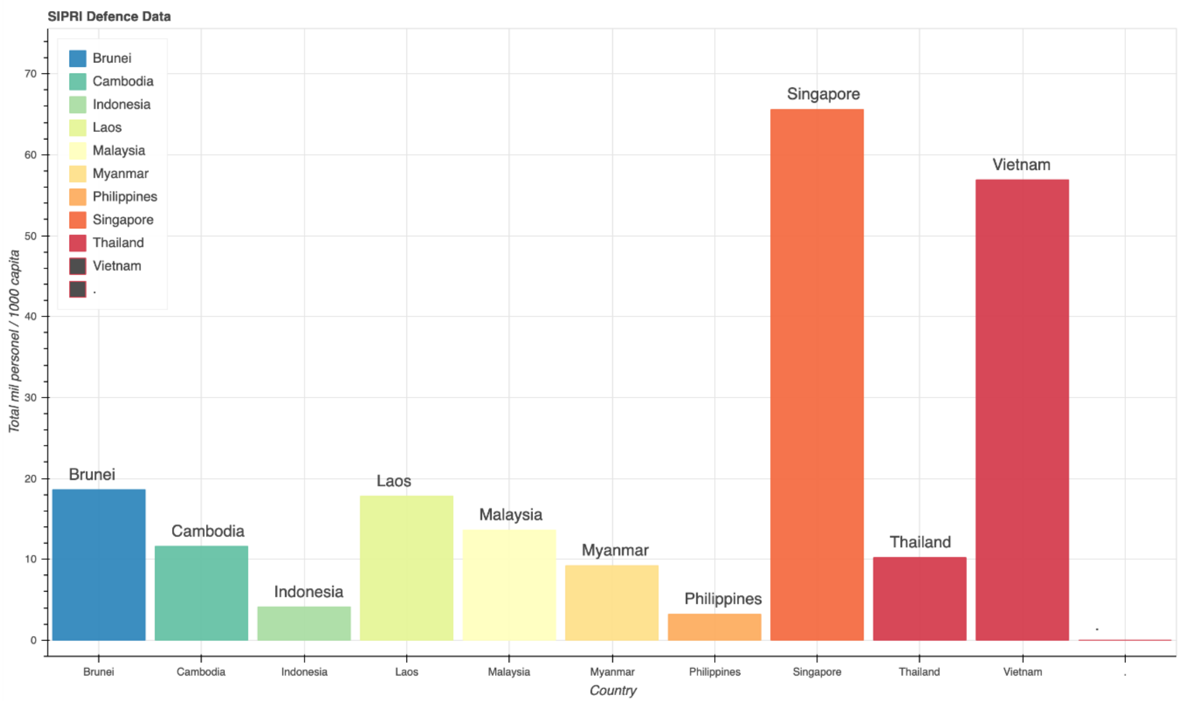


Figure 7. ASEAN: Total Number of Military Personnel per Capita of 1,000 People (IISS, 2020)

Security Challenges

Unlike some of its other neighbours in Southeast Asia (SEA), Malaysia has thus far enjoyed a continuous benevolent security environment with only minor flash points.[10] Hence, some argue that the prosperity, stability, and peace-dividend being enjoyed by Malaysians is due to the government's continuous investment in diplomacy, international engagement, and defence to create a strong and robust internal and external security environment (Noor, 2019). It is also

[8] The National Service (PLKN) was introduced in 2003 for Malaysian youth 18 years of age to encourage friendship between Malaysian youth of different races and ethnic groups to address serious concerns of the country becoming increasingly divided along racial lines. PLKN was halted in 2018 and abolished in August 2018.

[9] Malay bumiputra and other indigenous people of Malaysia are also called the "son of the soils."

[10] Malaysia went through two emergencies, one from 1948–1960, and a short-lived confrontation with Indonesia.

argued that the threats and challenges facing Malaysia are built on common interests and issues within ASEAN, the wider region, and globally (Laksmana, 2018). However, some contend that this is also attributable to Malaysia's "soft approach" in dealing with its adversaries, where it refuses to be openly confrontational. There are various perspectives on the external threats facing Malaysia, considering both its strategically valuable land and sea borders, as well as the fact that East and West Malaysia are divided by the South China Sea (SEA). This paper discusses Malaysia's major external security challenges and issues in the context of the DWP and several other sources (Malaysia Ministry of Defence, 2020).

First, the largest impact on Malaysia and SEA remains the major power rivalry between China and the United States. U.S.–China interaction, their positioning, and their competition for influence and dominance in the region pose the greatest strategic and diplomatic challenges (Blasko, 2015; Chang, 2015). The issues range from flexing of presence, violation of international maritime law, and encroachment into the South China Sea and countries' exclusive economic zones (EEZs; Storey & Yi, 2016). The issue is exacerbated by the presence of middle powers such as the United Kingdom, Japan, and Australia, and the pressure to take sides through the formation of alliances, or regional caucuses. Another potential source of external security challenge comes from Malaysia's common issues and disputes with its neighbouring ASEAN countries. This especially pertains to land and border issues, such as human, drug, and animal trafficking (often through cartels), poaching, and other effects from conflicts within neighbouring states (spill-over), such as an influx of refugees (Malaysia Ministry of Defence, 2020). It is estimated that Malaysia loses approximately RM 366 million in tax revenue from cigarette smuggling (Sebastian, 2020), RM 6 billion from illegal fishing, and around 980 thousand tonnes of sea-based produce annually (Malaysia Ministry of Defence, 2020). Another major issue is the long-term poor health and economic disruption due to haze from illegal logging and deforestation, especially in Borneo. These threats are not exhaustive, and Malaysia's rich natural resources, diversified economy, and strategic geopolitical location will continue to raise security threats.

Procurement Strategy

Malaysia is a net procurer, importing almost 95% of defence equipment and services. Pre-Cold War, Malaysia mostly purchased Western products, primarily from the United Kingdom, Europe, and the United States. However, since the 1990s, as per Figure 8, Malaysia's procurement trend has hugely diversified to include purchases from countries including Russia, Poland, Turkey, Brazil, and of late, South Korea and China. Within ASEAN, Malaysia has also procured equipment from Indonesia. Government-to-government processes and restricted tenders have been a more popular option than open tender in defence procurement decisions. There is an emphasis on VFM and seeking the lowest procurement cost, though this may not be the case in practise where technical specification and offsets can also be the deciding factor. Offsets or industrial collaborative programmes (ICP) are mandatory as part of international defence procurement above a specific threshold. Since 2009, several large contracts, such as the 8-by-8 armoured vehicles and the littoral-combat ship (LCS), were awarded to local prime contractors, with the expectation of a faster rate of technological absorption and a scaling up of indigenous capability.

However, Malaysia's defence procurement environment still struggles due to a lack of knowledge in threat assessment, which leads to a subsequently poor understanding of military requirements. Other procurement issues include acknowledging the wider whole-life-cost of a capability and compatibility during systems integration processes. Another consideration is the need to balance between total dependence on a single source and the procurement strategy of purchasing a small number of platforms and systems from a diverse range of contractors. Integrating all of these systems to make them work could be logistically challenging, hugely expensive, and inefficient.

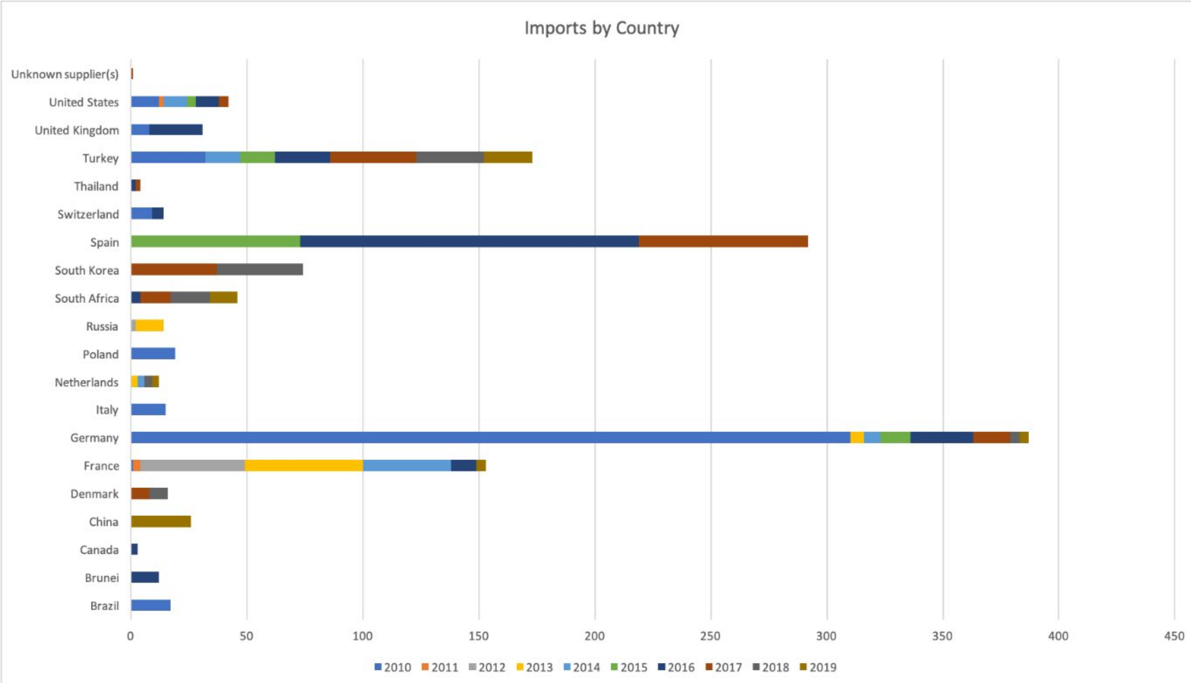


Figure 8. Malaysia: Import of Defence Equipment by Country (2010–2019) (Stockholm International Peace Research Institute, 2019)

How has Defence Demonstrated Value in the Malaysian Context?

Delivering Value at the Primary Level: Protection and Safety

The DWP states that the primary role of the MAF is to defend the nation's interests and protect its people against any form of external aggression. Articles 74 and 77 of the 1972 Armed Forces Act establishes the roles and functions of the three services—army, navy, and air force (Malaysia Ministry of Defence, 2020). Malaysia's primary defence value and the ability to deter aggression from foreign nations is measured through the state of readiness of units and platforms (Young, 2019). The concept of readiness itself is highly contentious. For the purpose of this discussion, readiness is defined as the ability to engage at a short notice in active operations. All three services of the armed forces follow a readiness matrix. The inspector general's office is the caretaker of the performance management system, which has been developed primarily using the balanced score card method. By nature, it is hard to obtain detailed measurement indicators that are currently being used to measure the state of readiness, as nothing is available in the public domain.[11] Such information is still treated as highly confidential by the MAF. However, Table 1 demonstrates an approximate picture of basic measurement indicators for the three services. The relevant information was sourced through interviews with senior military officials at MINDEF and from the wider defence industry.

Table 1. Malaysian Armed Forces Readiness

ARMY	**NAVY**	**AIR FORCE**

[11] Dato' Ganesh Navaratnam (vice admiral, president of National Centre for Defence Studies [PUSPAHANAS]). Interviewed by Kogila Balakrishnan, January 20, 2019.

Situational force scoring matrix is used. - One collective training and at least one live-fire field exercise per year. -Brigade HQ to undergo infield training exercises (FTX) 3 to 4 times a year -Command Post Exercise (CPX) to demonstrate that they can conduct command and control.	-Navy strategy map provides the mechanism for ensuring readiness on a daily, weekly, monthly and annual basis (DiRaja, 2013). -This is monitored by the operational commanders as well as the navy inspector general's office.	-Flying hours for helicopters, transport and fighter aircraft at minimum of 10 hours per month to maintain qualified pilots who are operational ready -Quarterly test provides additional qualification -Additional biannual qualification (not compulsory), mainly long distance flying and physical assessments - Annual checks on overall operational and physical fitness of the pilot

The armed forces organise war games and simulations on the probability of attack or aggression from a foreign state or non-state actors. The objective is mainly to use mathematical modelling and scenario-planning to measure the level of threat, identify gaps, and ensure that the military has sufficient capability to deter aggression. The armed forces aspire to have well-trained personnel who can understand and provide accurate analysis to senior military commanders, who can feed this information to the National Security Council within the prime minister's office. Despite current efforts, the question remains whether the army, navy, and air force are sufficiently trained and equipped to be operationally ready.

There are several key challenges facing the MAF in achieving readiness. Firstly, there are difficulties in mobilising the whole of the MAF and civil service to be capable of conducting joint operations. The military is still stove-piped in its defence capability planning, procurement practices, and budgetary planning, with a "top down approach" within each service. There is also a strong sense of boundary control and inter-service rivalry, which remains difficult to dissolve. Mistrust still exists between the MAF, civil servants in MINDEF, and other agencies (Anandhan & Inderjit, 2014). This contributes to a lack of understanding on how to develop a real threat assessment within the MAF, leading to poor prioritisation when determining the capabilities required for war-fighting.[12] Further, capability requirements constantly change due to the competing costs of maintaining existing equipment versus purchasing new equipment.[13] Figure 9 illustrates the fighting capability across the three services, with MINDEF investing heavily in traditional platforms such as ships, armoured vehicles, and aircraft systems, as opposed to air defence systems, sensors, or missiles.

[12] Azhar Mohamad (lt colonel [ret.], director of business development, BAE Systems, ex-commanding officer, Air Force College Malaysia). Interviewed by Kogila Balakrishnan, February 25, 2020.
[13] Tan Sri Zulkifli Zainal Abidin (former chief of MAF). Interviewed by Kogila Balakrishnan, January 20, 2020.

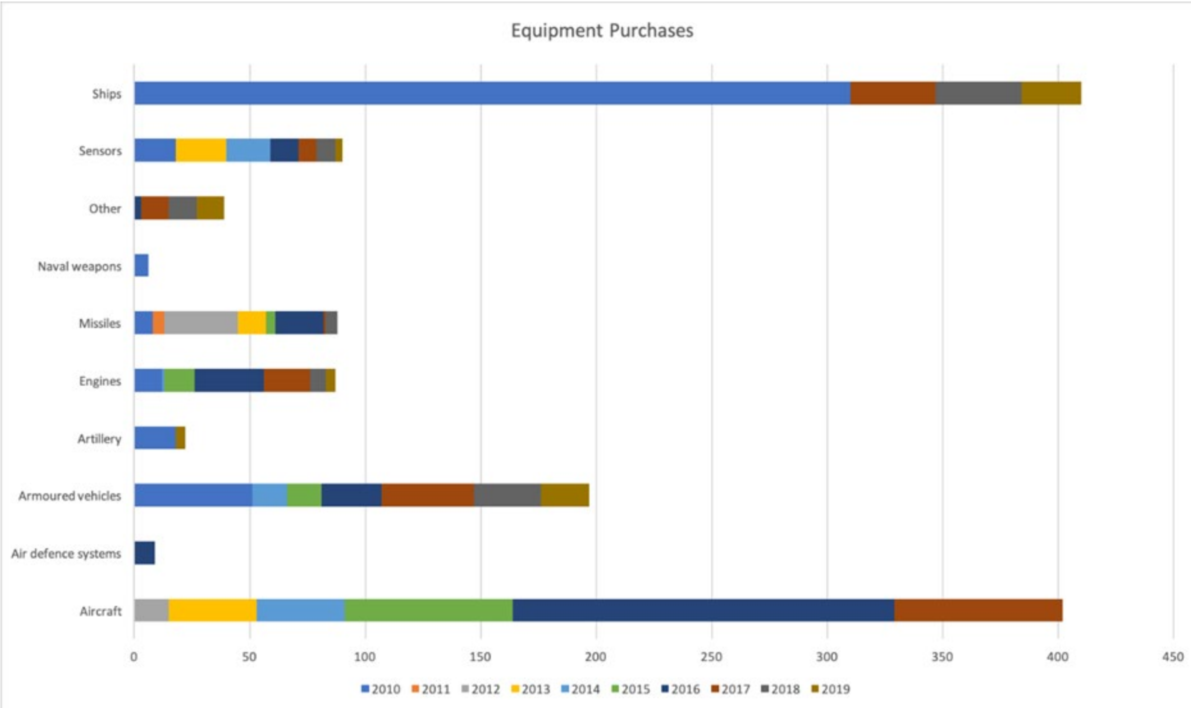


Figure 9. MAF Fighting Capability of the Army, Navy, and Air Force (USD) (Stockholm International Peace Research Institute, 2019)

These challenges are exacerbated by buying an assortment of these packages from a variety of sources, leading to a subsequently high cost of maintenance and logistical support. The last MINDEF budget sanctioned for this purpose 5 years ago was insufficient and did not factor in the escalating cost of maintenance, repair, and overhaul (MRO) activities. For example, the air force remains unable to conduct modern air operations with the existing overly diverse range of aircraft. Therefore, the government must undertake a severe restructuring of the air force with a view to reducing the variety of frames in their inventory.

The DWP has identified a long shopping list of platforms and systems to be acquired by the MAF to close this capability gap. These items are focused on emerging technologies such as artificial intelligence; cyber capabilities; robotics; Network Centric Operations (NCO); and Command, Control, Communication, Computers, intelligence, surveillance, targeting, acquisition, and reconnaissance (C4ISTAR) (Malaysia Ministry of Defence, 2020). However, meeting these requirements is costly and requires significant investment, especially to upgrade or buy new platforms and sub-systems. This leads to the current situation, where the Malaysian government must consider if it can fulfil its military capabilities with only 1% of GDP allocated to defence spending (Daim & Harun, 2019). Malaysia's recent defence spending pattern seems to contradict the previously mentioned aspirations, with major budget reductions followed by an overall reduction in defence spending. Hence, hard decisions must be made to balance investments between operational training, equipment, systems, and other areas.

Delivering Value at the Secondary Level: Socio-Economic Prosperity

It is harder to measure value from defence at the secondary level. The DWP mentions the role of the defence industry as double-pronged: a platform to support the MAF at the frontline of defence, but also as a medium that contributes to highly skilled employment, economic redistribution through regional cluster development, opportunities for diversification through the enhancement of supply chains, and SME development, growth, export, and innovation

(Balakrishnan, 2010). It is also argued that the offsets policy is useful for defence industry growth, economic diversification, and foreign direct investment. The offset value for 50 programmes and 250 offset projects amounted to RM 32 billion between the periods of 2015–2020. It is estimated that around 3,000 jobs were generated; 1,000 engineers were trained for high skilled work; and 500 local companies benefitted through new business opportunities (*Defence Industry Division*, Malaysia Ministry of Defence, 2020).

Figure 10 provides a detailed breakdown (2015–2020) in terms of offsets distribution, where 51% of the offsets credits were allocated to local content or industrial development, 35% towards technology development, 9% to investments in marketing and branding, 3% to capability development and MRO, and 2% to education and training (*Defence Industry Division*, Malaysia Ministry of Defence, 2020).

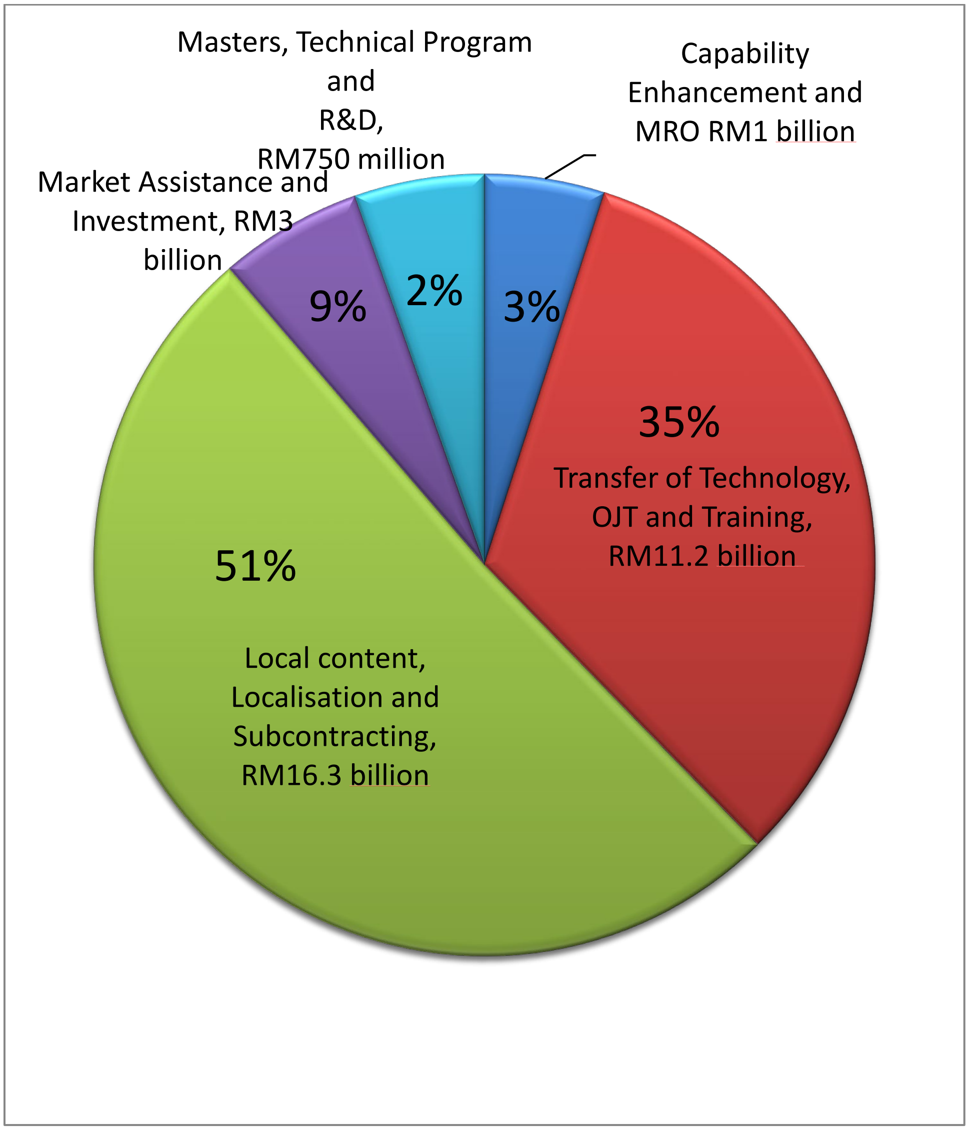


Figure 10. Malaysia: Distribution of Offsets Credit Value (2015–2020) (*Defence Industry Division,* Malaysia Ministry of Defence, 2020)

Since 2005, MINDEF has made a concerted effort to increase defence R&D funding. Table 2 shows the R&D budget between the 8th and 11th Malaysia Plans (2005–2020) for the Science

& Technology Research Institute for Defence (STRIDE), a government research organisation under MINDEF. Although there is a significant increase from RM 2.2 million in 2005 to RM 92.2 million under the 11th Malaysia Plan (2016–2020), this is still a meagre sum compared to the total defence R&D budget of countries with a strong innovation culture, such as Singapore, South Korea, and Israel (Malaysia Ministry of Defence, 2020). That said, the total R&D figure does not capture the R&D budget of the defence industry and universities undertaking defence or dual-use research in Malaysia. Currently, STRIDE, National Defence University Malaysia (NDUM), and a few other universities and research organisations are involved in defence R&D. Considering the negligible size of government defence R&D investment, it is unsurprising that there is little hard evidence of patents, licenses, key technology spin-outs, or start-ups, let alone publications of international standing. Overall, defence R&D investments have not yielded substantive benefits.

Table 2. Malaysia's Defence Spending (*Science, Technology, Research Institute for Defence (STRIDE),* Ministry of Defence, 2020)

Malaysia Plan	**Development Budget for STRIDE**
8th Malaysia Plan 2005 – 2010	RM2.2 million
9th Malaysia Plan 2006 – 2010	RM17.5 million
10th Malaysia Plan 2011 – 2015	RM10 million
11th Malaysia Plan 2016 – 2020	RM92.2 million

Malaysia has yet to penetrate the defence export market but has made headway with dual-use exports. Offsets have been used to enter the international defence supply chain for large OEMs, with companies such as Composite Technology Research Malaysia (CTRM), Contraves Advanced Defence (CAD), and Sapura Defence exporting parts and components (Malaysia Ministry of Defence, 2020). The defence sector has also been a catalyst to the development of several industrial clusters around Malaysia, including a maritime cluster in Lumut, an aerospace cluster in Subang, and an automotive cluster in Pekan. These clusters contribute mainly towards dual-use industrial activities that assist in creating wider socio-economic impact and long-term multipliers. However, the sustainability of these clusters is highly contentious. Participation from commercial entities, the level of civil-defence integration, assessing the real value of collaborative activities, and the level of integration within these clusters all pose challenges. Several SMEs were outsourced work through defence contracts and offsets agreements.

Hence, there is the question of whether defence spending allocations towards R&D, offsets, and sustaining a defence industry are commensurate with the expected outcomes. Despite arguments against the economic value of defence to the Malaysian economy, Malaysia has been successful in using defence spending for dual-use activities and in capturing commercial business and technology. The defence industrial sector that began with a strong footing in the 1980s and 1990s, and seemed to be on an upward trajectory, has slipped in the past 10 years due to a lack of vision and focus, poor leadership, and mismanagement of funds within the defence environment (Balakrishnan, 2008). The defence industrial sector has had difficulties in adapting to the changing defence ecosystem, struggling to embrace the adoption of new innovative ideas in developing the sector (Balakrishnan, 2008). Overdependence on the government as the single source of contracts, corrupt procurement practises, a lack of appetite for investment in innovation, and low technological absorption capability have combined to create a gloomy outlook for this sector (Hughes, 2011).[14] Despite huge investments into the defence industry sector, R&D, as well as offsets, there is an overall lack of compelling evidence that there have been substantive contributions to the economy from defence (Bitzinger, 2017). Despite more than 20 years of investment into offsets, Malaysia's defence industry sector has not progressed and is still hugely dependent on foreign suppliers. Furthermore, recent controversial decisions around the trading of commodities such as palm oil for defence equipment may further erode the chances of rebuilding the defence industrial base via government-to-government technology transfer agreements, offsets, and industrial collaboration programmes (Grevatt, 2019).

As Malaysia has a relatively small defence industry, stretched in terms of budget but hugely concentrated in high-technology sectors including electronics and semiconductors, the economy will see significant benefits if the defence industry focuses its efforts and investments into dual-use technology fields such as autonomous vehicles, systems integration, data science, cyber security, radars, space technology, and battery technology (Lele, 2013). Defence technology should be strategically positioned not just for deterrence, but also for contribution to overall national industrial policy focused on dual-use technology. There is a need to engage larger defence companies (primes) with SMEs to tap into commercial technologies and encourage open innovation, as most technological innovation today is a spin-off from smaller SMEs and start-ups. The Defence Industry Blueprint is said to address some of these shortfalls and encourage the development of strategic drivers to focus on how the defence industry could be more effective, but this is yet to be seen (Balakrishnan & Matthews, 2009). There is also mention that 10–15% of all nominal offsets value from industry collaboration will be allocated for R&D purposes (Malaysia Ministry of Defence, 2020).

However, there are many unanswered questions concerning the implementation of allocated funds and whether there will be similar commitments from the industry in the form of matching grants to access the funding. Further issues lie in accountability, disbursement criteria, as well as selection of projects. Perhaps now is the time to develop a separate technology-based entity to support human capability development in emerging technology areas, such as C4ISTAR and NCO, by reaching out to non-traditional defence suppliers, small- and medium-sized enterprises and academia, as well as traditional defence suppliers, in order to develop new capabilities for Malaysia's defence sector (United Kingdom - Defence Science and Technology Laboratory, 2018).

As current trends demonstrate that most of the emerging technology innovation originates from the commercial sector, defence must strive to gain access to such technologies as robotics, autonomous vehicles/vessels, cyberspace, artificial intelligence, and space. The offsets funding allocated to defence R&D should be governed appropriately and used for defence-specific technologies that cannot be offered by the commercial sectors, such as radar, missile technologies, weapons technologies, and ammunition. The R&D allocation could be more effectively utilised by introducing more attractive incentives, such as the patent box scheme

[14] Chris Hughes (2011), in his paper, discussed three major challenges for Japan: budget to meet the long-term military demand, procurement management and international collaboration as solutions for Japan's revival of its defence industrial base, and "techno nationalism."

whereby companies receive tax credits in return for money spent on R&D (United Kingdom Government, 2020).

Furthermore, the civil offsets programme requires restructuring in order to maximise technology transfer from civil prime companies. This is necessitated by the fact that emerging technologies in civil sectors reside with tech giants such as Google, Facebook, Apple, Microsoft, and Uber. The government should also set up an organisation (United Kingdom - Innovate UK, 2020) that funds and capitalises on innovation to support Malaysia's defence prosperity and value (United Kingdom Ministry of Defence, 2019). This should be supported by the building of an innovation network consisting of government, private sector, academia, and industry to fund and supply emerging defence industry and SMEs with both human and physical capital, as well as developing innovative ideas that can be translated into products and services (United Kingdom Ministry of Defence, 2020).

The Malaysian defence sector also performs secondary roles in delivering valuable societal services such as carrying out humanitarian and disaster relief, search and rescue, and non-combatant evacuation operations as part of the United Nations (UN) charter and protection of critical national infrastructure. Malaysia is part of the ASEAN Humanitarian Aid and Disaster Relief (HDRF) military readiness group. In 2010, Malaysia wanted to demonstrate leadership as part of UNHRD and set up a World Food Programme (WFP) logistics centre at Subang airport, acting as a centre for humanitarian relief items within 48 hours of crisis. The armed forces also collaborate with the Federation of Red Cross, Red Crescent Societies (IFRC), and the International Council of Red Cross (ICRC) to strengthen international humanitarian law. For example, the military was called to support Rohingya refugees by lending the A400M aircraft to transport medical personnel to Bangladesh in 2017 (Malaysia Ministry of Defence, 2020). The MAF have been involved in 40 peacekeeping missions, the first in the Republic of Congo (1960–1964), and most recently in Yemen (2019) until now. Other active missions include Darfur and South Sudan since 2007, Liberia since 2003, West Sahara since 1991, Philippines since 2004, Lebanon since 2007, and Bangladesh since 2017 (United Kingdom - Innovate UK, 2020). The COVID-19 pandemic also highlighted how the military supports the police force and medical professionals during a national crisis through patrols, use of military hospitals and doctors, as well as military equipment for airlift and fast transportation.

The immediate value question is whether Malaysian taxpayers should fund humanitarian aid and peacekeeping missions in distant places. What value does this create for the citizens? One argument is that these activities demonstrate that Malaysia is a mature and reliable nation capable of contributing to collective world peace and international order.

Delivering Value at the Tertiary Level: Soft Power Projection

Despite being a small maritime nation, Malaysia aspires to be a SEA lynchpin for the wider region (Subramaniam et al., 2018). Strategically, it remains firmly supportive of the international world order and democratic values and objectives whilst retaining a deep historical suspicion of Chinese objectives and intentions in the region. Malaysia attempts to exert its values through bilateral and multilateral engagement, joint-training exercises, and by sending military and civilian MOD officials for training overseas. Malaysia has also built traditional partnerships with non-ASEAN countries in the region, such as with India, Pakistan, and Japan, focused on joint exercises and training.

As part of defence diplomacy, Malaysia has established defence bilateral co-operation with individual ASEAN countries to enhance primary capability and save money through the sharing of capabilities. Malaysia's history of defence bilateral co-operation began with Thailand (1965) and Indonesia (1972) through the Government Border Control (GBC) agreement. Since then, it has signed a dozen other bilateral defence MOUs with Brunei (1992), Philippines (1994), Vietnam (2008), Cambodia (2018) and Laos PDR (2019). It is in the interest of the armed forces to build strong defence links and negotiate for the procurement of common platforms and systems, which can then lead to better interoperability.

The FPDA, or the Five Power Defence Arrangement (1971), is the only apolitical, multilateral platform formed predominantly for joint-military exercises amongst the United Kingdom, Australia, New Zealand, Malaysia, and Singapore. The FPDA is said to contribute to Malaysia's primary defence value, as the platform is used for integrated air defence capability, the sharing of resources and intelligence, as well as maintaining existing assets.

At the global level, the Malaysian Armed Forces have also built defence co-operation with the United States, China, Australia, France, and the United Kingdom, mainly focused on combined military exercises, courses, training, exchange of visits and intelligence discussions.[15] Platforms such as the ASEAN Defence Ministers Meeting (ADMM) and ADMM Plus are used to communicate the importance of enhancing national security and expressing how each nation views the others' power projection on overlapping conflicts.[16] Malaysian military personnel mostly attend courses at local universities, with some also attending courses and undertaking degree programmes using government and foreign-funded scholarships to prestigious universities in the United States, United Kingdom, Australia, and France.[17] Military personnel are seen as a high value human resource—especially in the technology sector—assuming that the average military career is less than 10 years in length. However, if it is significantly longer than that, the value of retiring military personnel to the civil sector may be reduced.

Conclusion

This paper set out to measure the value derived from defence spending. Historically, this topic has proven to be challenging and problematic, but this does not imply that the armed forces and the defence sector do not have to prove that value is derived from defence spending and make efforts to improve the overall impact of defence. Unfortunately, few countries have tried to measure this. To do so, we constructed the TDV framework. The TDV framework, divided into primary, secondary, and tertiary levels, can be applied to analyse any country's defence investment. Extrapolations using this framework should be modified by each nation according to its context. One has learned through its application that it becomes increasingly harder to demonstrate value as we move down the value-chain from the primary to the tertiary level. The paper argues that at the primary level, the most critical value measurement is associated with the level of protection and safety that is provided to a nation and its citizens by the armed forces and defence sector. This is determined by the ability of the armed forces to deter aggression, precipitated by ensuring readiness and building sufficient capability. At the secondary level, value is measured through socio-economic prosperity, and finally, at the tertiary level, value arises through soft-power projection to demonstrate the criticality of a nation in its geopolitical and strategic positioning.

Covering Malaysia, the paper applied the TDV framework to analyse the value derived from its defence spending. The TDV was a useful evaluation tool to understand Malaysia's current value from defence spending at all levels. However, insufficient public data made objective analysis and determination of what and how to measure value challenging. The DWP is the first public

[15] Hisham Noor (director for the Malaysian Institute for Defence and Security, Ministry of Defence, Malaysia). Interviewed by Kogila Balakrishnan, May 4, 2020.

[16] The ADMM includes ASEAN plus Australia, New Zealand, Japan, India, Republic of Korea, and China.

[17] The most popular overseas universities and colleges include those such as Sandhurst, Berkshire, Royal College of Defence and Security (RCDS), UK Defence Academy, Naval Postgraduate School United States, U.S. Military Academy (West Point), Australian Defence Force Training Centre (ADFA) and Australian Defence College.

document that has discussed the value derived from defence spending to the Malaysian public, despite claims that the content differs little from previously classified defence policy documents.

In the case of Malaysia, the primary role of defence was measured by the state of readiness of the MAF. There were several challenges beyond the author's limitations in obtaining sufficient data, such as operational activity irregularity in the systematic and consistent collection of such data, as well as tenuous political will in investing resources to attain higher levels of readiness.

The outcome from the secondary role of defence concerning socio-economic prosperity resulting from defence spending has also been minimal. After almost 50 years of government investment into defence industrialisation, offsets and R&D, results have been modest. There is a lack of appetite for enhancing innovative capability, addressing competitiveness and penetrating the global market within the defence industry sector. Positive outcomes include spin-offs from defence spending to developing commercial supply chains and skills development in high-technology sectors. Further, outcomes from HRDF activities were positive where the MAF has successfully supported critical missions abroad.

Finally, outcomes from the tertiary role of defence regarding soft power projection has led to an increased status for the Malaysian defence sector in defence diplomacy, such that this successful projection as a friendly nation means it is often invited by major powers to broker peace talks in the region. In recent years, Malaysia has also increased its level of joint exercises with neighbouring ASEAN countries and major powers in the region and globally. Malaysia has been very successful in keeping alive the dynamics and contributing actively to multilateral platforms such as ADMM, ADMM Plus and FPDA.

The valuation framework can be very fuzzy, and their respective explanations are subjective—they require careful scrutiny and are not a "one-size-fits-all" framework. Further, the framework is culturally and socially dependent on what each country would like to identify as value derived from defence spending. We still have problems appraising value when it comes to buying defence or deterrence. Hence, this framework is not perfect and represents a general template that may provide a useful starting point when initiating value measurements for the defence sector. As previously mentioned, it is hard to measure value in public service delivery, let alone defence. These indicators provide guidance that should be used selectively and modified to the strategic context of each country.

There are several issues that the defence sector should consider when attempting to improve value measurement in the future. First, it is important to critically evaluate existing policy and processes that are in place to capture and manage data. Data could be highly sensitive and will require careful handling and classification. It is also necessary to have at least a basic platform with the right architecture to capture data on the value of defence spending. Success in this area would be seen through the ability to publish yearly reports on value derived from defence spending based on systematic analysis. However, the overall process of realising the importance of communicating value from defence spending must be driven by strong leadership that believes such efforts are vital. At the same time, there must be a bottom-up process that works to minimise the implementation gap that hinders defence value maximisation.

The TDV framework could be used as a preliminary model by other countries in Southeast Asia, the region, and globally to examine defence value, understand the strengths and weaknesses that arise from the value evaluation process, and use the outcomes to develop or improve on existing defence policy and implementation. The defence community, especially politicians, policy-makers and the armed forces must recognise the importance in demonstrating value from defence spending, especially in a world where defence budget allocations are increasingly contested. The TDV framework is a work in progress and can be a useful public policy tool used by the defence community to drive the idea of demonstrating value through defence spending.

References

Abbas, M. (2019, April 10). *Defence white paper, why*. Malaysian Defence. https://www.malaysiandefence.com/defence-white-paper-why/

Anandhan, S., & Inderjit, S. (2014). Capabilities based planning for force development: Issues and challenges for the Malaysian armed forces. *ZULFAQAR International Journal of Politics, Defence & Security*, *1*, 1–12.

Balakrishnan, K. (2008). Defence industrialisation in Malaysia: Development, challenges and the revolution in military affairs. *Security Challenges*, *4*(41), 135–155.

Balakrishnan, K. (2010). Evaluating the role of offsets in creating a sustainable defence industrial base: The case of Malaysia. *The Journal of Defence and Security*, 1–28.

Balakrishnan, K. (2019). Malaysian defence industry: Issues, challenges and prospects. *Perwira Dialogue.*

Balakrishnan, K., & Matthews, R. (2009). Malaysia's defence industry and the role of offsets. *Journal of Defence Peace and Economics*, *20*(2), 341–358.

Benoit , E. (1973). *Defence and economic growth in developing countries.* Lexington Books.

Bitzinger, R. (2017). *Arming Asia, techno-nationalism and its impact on local defence industry.* Routledge.

Blasko, D. J. (2015). The 2015 Chinese defence white paper on strategy in perspective. *China Brief*, *15*(12).

Chang, F. K. (2015, May 27). *Strategic intentions: China's military strategy white paper*. Foreign Policy Research Institute. https://www.fpri.org/2015/05/strategic-intentions-chinas-military-strategy-white-paper

Cheng, K. C. (2020, February 1). Institute of Malaysian & International Studies (IKMAS), National University of Malaysia (UKM).

Chwee, K. C., & Hamzah, B. A. (2019, December 6). The making of defence white paper. *New Straits Times*. https://www.nst.com.my/opinion/columnists/2019/12/545531/making-defence-white-paper

Daim, N., & Harun, H. N. (2019, December 2). Mat Sabu to propose that military spending to be set at 1% of GDP. *New Straits Times*. https://www.nst.com.my/news/nation/2019/12/544073/mat-sabu-propose-military-spending-be-set-1pc-gdp

Dunne, P. M. (2018, July). *Growing the contribution of defence to UK prosperity.* Ministry of Defence. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/723679/20180709_MOD_Philip_Dunne_Review_FOR_WEB_PUB.pdf

Economic Planning Unit. (2018). *Development planning*. Economic Planning Unit, Prime Minister's Department. https://www.epu.gov.my/en

Grevatt, J. (2019, April 17). *Malaysia looks to exchange palm oil for materiel*. Janes Defence Industry. https://www.janes.com/defence-news/news-detail/malaysia-looks-to-exchange-palm-oil-for-materiel

Hamzah, B. A. (2016, November 8). Embrace total defence concept. *New Straits Times*. https://www.nst.com.my/news/2016/11/186804/embrace-total-defence-concept

Hartley, K. (2012). Conflict and defence output: An economic perspective. *Revue d'Economie Politique*, *122*(2), 171–195.

Hitch, C. J., & McKean, R. N. (1960). *The economics of defense in the nuclear age.* Oxford University Press.

Hosoyo, Y. (2020, February 6). *Exploring Abe's role on the chessboard with professor Yuichi Hosoya*. Center for Strategic and International Studies. https://www.csis.org/analysis/exploring-abes-role-chessboard-professor-yuichi-hosoya

Hughes, C. W. (2011). The slow death of Japanese techno-nationalism: Emerging lessons for China's defence production. *The Journal of Strategic Studies*, *34*(3), 451–479.

IISS. (2020). *Military balance 2020*. International Institute for Strategic Studies. https://www.iiss.org/press/2020/military-balance-2020

IISS. (2020). *Prosperity and value: What is defence's triple bottom line?* Chief of Defence Staff's Strategy Forum, Royal Air Force Club, London. International Institute for Strategy and International Studies.

Laksmana, E. A. (2018). Why is South East Asia rearming? An empirical assessment. In R. Dassani & S. W. Harold (Eds.), *U.S foreign policy in Asia, perspectives for the future* (pp. 106–137). RAND Corporation.

Lele, A. (2013). *Asia's space race: Rhetoric or reality.* Springer.

Malaysia Ministry of Defence. (2020). *Defence industry division.* Defence Industry Division.

Malaysia Ministry of Defence. (2020). *Defence white paper: A secure, sovereign and prosperous Malaysia.* Malaysia National Library.

Malaysia Ministry of Defence. (2020). *Science, Technology, Research Institute for Defence (STRIDE).* Science, Technology, Research Institute for Defence (STRIDE).

Mazzucato, M. (2018). *The value of everything: Making and taking in the global economy.* Penguin Books.

Melese, F., Richter, A., & Solomon, B. (2015). *Military cost–benefit analysis.* Routledge.

Ministry of Foreign Affairs Malaysia. (2019, June). *Foreign policy framework of the new Malaysia: Change in continuity.* https://www.pmo.gov.my/wp-content/uploads/2019/09/Foreign-Policy-Framework-for-New-Malaysia-min.pdf

New Straits Times. (2019, December 2). *Malaysia's first defence white paper tabled in Parliament*. https://www.nst.com.my/news/nation/2019/12/543948/malaysias-first-defence-white-paper-tabled-parliament

Noor, E. (2019, November 7). *Foreign and security policy in new Malaysia*. Lowly Institute. https://www.lowyinstitute.org/publications/foreign-and-security-policy-new-malaysia

Noor, E., & Qistina, T. N. (2017). Great power rivalries, domestic politics and Malaysian foreign policy. *Journal of Asian Security*, *13*(3), 173–192.

Parameswaran, P. (2019, February 22). What's behind Malaysia's new defence white paper talk? *The Diplomat*. https://thediplomat.com/2019/02/whats-behind-malaysias-new-defense-white-paper-talk/

Paul, C., Yeats, J., Clarke, C. P., Matthews, M., & Skrabala, L. (2015). *Assessing and evaluating Department of Defence efforts to inform, influence and persuade: Handbook for practitioners.* RAND Corporation. https://www.rand.org/content/dam/rand/pubs/research_reports/RR800/RR809z2/RAND_RR809z2.pdf

Portal Rasmi. (2020, June 22). *Malaysia. Jabatan perdana menteri, unit perancangan ekonomi*. https://www.epu.gov.my/ms/profil-jabatan/organisasi/bahagian-dan-unit/bajet-pembangunan

Porter, M. E. (1998). *Competitive advantage: Creating and sustaining superior performance.* Free Press.

Rodzi, N. H. (2019, December 2). Malaysia unveils first defence white paper; critics find 10-year plan lacking. *The Straits Times*. https://www.straitstimes.com/asia/se-asia/malaysia-unveils-first-defence-white-paper-critics-find-10-year-plan-lacking

Sebastian, M. A. (2020, April 24). Better security on land: How about the sea? *New Straits Times*. https://www.nst.com.my/amp/opinion/columnists/2020/04/587030/better-security-land-how-about-sea

Smith, A. (1776). *An inquiry into the nature and causes of the wealth of nations* (Vol. 5). W. Stratton.

Stockholm International Peace Research Institute. (2020, May 2). *SIPRI arms transfer database*. https://www.sipri.org/databases/armstransfers

Storey, I., & Yi, L. C. (2016). *The South China Sea dispute: Navigating diplomatic and strategic tensions.* ISEAS - Yusof Ishak Institute.

Subramaniam, A., Sulaiman, A., & Loong, W. L. (2018). Defence spending in an era of uncertainty and budgetary constraint, defence science & technical bulletin. *Science & Technology Research Institute for Defence*, *11*(2), 332–335.

United Kingdom - Defence Science and Technology Laboratory. (2018, March 29). *DSTL's Serapis framework*. https://www.gov.uk/government/publications/dstls-serapis-framework

United Kingdom - Innovate UK. (2020, February 4). *Innovate UK*. https://innovateuk.blog.gov.uk/

United Kingdom Government. (2020, May 7). *Corporation tax: Use the patent box to reduce your corporation tax on profits*. https://www.gov.uk/guidance/corporation-tax-the-patent-box

United Kingdom HM Treasury. (2017). *Public value framework and supplementary guidance*. https://www.gov.uk/government/publications/public-value-framework-and-supplementary-guidance

United Kingdom Department for Business, Energy & Industrial Strategy. (2017). *The UK's industrial strategy*. United Kingdom Ministry of Defence. Https://www.gov.uk/government/topical-events/the-uks-industrial-strategy

United Kingdom Ministry of Defence. (2018). *Mobilising, modernising and transforming defence: A report on the modernising defence programme.* https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/931705/ModernisingDefenceProgramme_report_2018_FINAL.pdf

United Kingdom Ministry of Defence. (2019). *Advantage through innovation: The defence innovation initiative.* https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/553429/MOD_SB_Innovation_Initiative_Brochure_v21_web.pdf

United Kingdom Ministry of Defence. (2020, February 4). *Defence and security accelerator (DASA).* https://www.gov.uk/government/organisations/defence-and-security-accelerator/about

Williamson, G. M. (2017). *Industry for defence and a prosperous Britain*. Ministry of Defence. https://www.gov.uk/government/publications/growing-the-contribution-of-defence-to-uk-prosperity-a-report-for-the-secretary-of-state-for-defence-by-philip-dunne-mp

Wylie, R. (2017). Defence procurement, innovation and value for money. In K. Burgess, & P. Antill (Eds.), *Emerging strategies in defence acquisition and military procurement* (pp. 167–183). IGI Global.

Young, T.-D. (2019). What are governments in Central and Eastern Europe not buying with their defence budgets? The readiness clue. *The RUSI Journal*, *164*(2), 36–55.

Rapid Innovation with Chinese Characteristics: National Defense Science and Technology Innovation Rapid Response Teams and the Military-Civil Fusion Innovation Ecosystem

Emily de La Bruyère—is a Co-Founder of Horizon Advisory, a geopolitical consultancy, as well as a Senior Fellow at the Foundation for Defense of Democracies (FDD). Her work focuses on China's standardization ambitions, military–civil fusion strategy, and platform geopolitics, as well as their implications for global security and the economic order. She holds a Bachelor of Arts summa cum laude from Princeton University and a Master of Arts summa cum laude from Sciences Po, Paris, where she was the Michel David-Weill fellow. (ebde@alumni.princeton.edu)

Abstract

To respond to China's military–civil fusion (MCF) strategy, the United States needs a prioritization framework. The United States must determine what technologies to protect and capabilities to develop. These determinations must be informed by U.S. goals, accounting for relative strengths and weaknesses. The determinations must also be informed by the adversary: how China operates, to what ends, and with what resources.

This paper leverages the technological demands of China's National Defense Science and Technology Innovation Rapid Response Teams in order to begin to address those questions—and to provide an example of the sort of data sets that might be used to answer them more comprehensively moving forward. Beijing's MCF innovation ecosystem clearly prioritizes information technology, broadly. More specifically, entities charged with fusing commercial and military innovation appear to prioritize autonomous systems (e.g., UAVs, UUVs), sensing and network technologies to dock into and connect them, and information aggregation and analysis platforms. Advanced algorithms and software do not feature prominently in the surveyed data set. These findings can inform U.S. Department of Defense (DoD) acquisition. Defensively, Beijing's priorities and commercial dependencies should shape the DoD's efforts to protect. Offensively, the insight this data provides into Chinese capabilities can assist U.S. efforts to identify and exploit weaknesses.

Introduction

We no longer require National Defense Strategies to signal that the United States and China are locked in a great power competition. And Beijing's military–civil fusion (军民融合; MCF) strategy—once a niche, technical concept reserved for fora like this one—has become a widely recognized component of today's great power contest (de La Bruyere & Picarsic, 2019). Discussion of MCF appears everywhere from U.S. policy documents to mainstream media coverage (e.g., Swanson, 2020). With this growing awareness of MCF comes growing awareness of the strategic asymmetries it portends, namely Beijing's ability to weaponize its integration into the international commercial ecosystem both to obtain resources and project power.

This diagnosis of the United States–China competition, and the role of MCF within it, is long overdue. Yet however accurate, broad diagnosis neither ensures an appropriate strategic response to China's challenge nor resolves the asymmetries of Beijing's competitive approach. MCF entails the integration of military and civilian resources, actors, and positioning for the sake of comprehensive, and international, power. The strategy is designed to take advantage of the twin trends of information technology and globalization. With innovation flowing relatively freely across national borders, Beijing siphons advanced research and development from abroad in order to bolster its own military and economic apparatus (de La Bruyere, 2020). In a world dependent on multinational supply chains, Beijing develops positions of relative control over key nodes within them—and then converts that control into coercive leverage (Bradsher, 2010). As international systems come to depend on information and its flow, Beijing works not only to collect superior information but also to build and to control emerging global information infrastructures (e.g., telecommunications, digital payment systems, surveillance networks). Beijing does so across military and commercial domains, for fused military and commercial advantage (de La Bruyere & Picarsic, 2020).

There is no easy response to China's approach. The U.S. commercial sector innovates, including in dual-use domains. That innovation circulates relatively freely at home and abroad. Washington can neither stop such circulation nor move domestic innovation into the isolated confines of government operations. Nor can Washington re-shore, or allied-shore, all supply chains over which China might exert coercive leverage—or match Beijing point for point in all emerging infrastructures everywhere.

Rather, the United States must prioritize. In military as well as in commercial and civilian competition, Washington will have to develop a prioritization framework—of what technologies to protect, what resources and manufacturing capabilities to develop, and through what infrastructures to compete. That framework should be derived from U.S. goals, factoring in U.S. strengths and weaknesses, including relative standing vis-à-vis China. The framework must also take into account the adversary's framework—how China operates, to what ends, and with what resources—including

- those technologies and applications that Beijing prioritizes for the emerging great power competition;
- those technologies for which Beijing relies on the civilian sector; and
- those actors within the Chinese system that play the most critical roles in the process.

These are large questions but not impossible ones. Reliable open-source indicators exist to benchmark comparative analysis and map out answers. This paper leverages one data set in particular: technological demands of China's National Defense Science and Technology Innovation Rapid Response Teams since 2018. This data set by no means reflects the entirety of the MCF innovation ecosystem. Neither it nor the analysis presented in this paper is exhaustive. Rather, they are offered as examples of the sort of open-source data, and analysis of it, that can fuel the requisite understanding of Beijing's systems and priorities.

The Rapid Response Team data set suggests that the MCF innovation ecosystem places particular emphasis on information-related technologies and applications. Among notable examples, those include autonomous vehicles—for air, ground, and maritime deployment—as well as the sensing and network capabilities to be docked into them. They also, and perhaps most significantly, include networks or platforms through which collected information can be integrated and analyzed. Those networks and platforms, and the power that they promise, apply to military as well as to civilian domains: MCF fuses both civilian and military inputs to develop outputs that are applied in traditional military domains. This much is already well documented in U.S. analysis. The MCF innovation effort, as reflected in this sample, also uses fused civilian and military inputs to develop outputs that are used to project power in civilian domains. These civilian domains those conventionally seen as cooperative rather than competitive. In other words, not only are military and civilian inputs to be fused, but military and civilian outputs are as well.

National Defense Science and Technology Innovation Rapid Response Teams

In 2018, the Science and Technology Committee of China's Central Military Commission established the first National Defense Science and Technology Innovation Rapid Response

Team, known as the Rapid Response Team. That team was based in Shenzhen (Office of the Military–Civil Fusion Development Committee of the Shenzhen Municipal Committee of the Communist Party of China, 2018).[1] Since, additional teams in Chongqing and Dalian have been established.

These Rapid Response Teams might be thought of as the Chinese equivalents of the Defense Innovation Unit (DIU). As the Office of the Military-Civil Fusion Development Committee of the Shenzhen Municipal Committee of the CCP puts it, the Rapid Response teams "link advanced commercial technologies and products to national defense capabilities." Their main task is to "pay close attention to advanced commercial technologies, concepts, and models; actively discover and quickly respond to commercial technologies and products with military application potential; and build a bridge between the frontiers of military and commercial innovation" (Office of the Military–Civil Fusion Development Committee of the Shenzhen Municipal Committee of the Communist Party of China, 2018). To that end, the Rapid Response Teams

- study advanced commercial technologies, monitoring new developments for applications to national security, including through monthly field research;
- solicit fast, innovative commercial solutions based on national defense needs, generally with a 6-to-12-month timeline for delivery; and
- use military-oriented demonstrations of technologies and their applications to build a communication platform for the military, innovative companies, innovative teams, and investment vehicles (Office of the Military–Civil Fusion Development Committee of the Shenzhen Municipal Committee of the Communist Party of China, 2018).

Since 2018, the Rapid Response Teams have issued requests for at least 57 technological solutions (see Appendix) and have organized at least six technology competitions or challenges (see Table 3). The technology solution requests tend to have timelines ranging from 6 to 12 months for delivery, with the exception of three directly related to COVID-19 response, which call for 5- to 30-day turnaround. In some but not all cases, the technology solution requests state explicitly that the projects should "give priority to domestic materials, equipment and systems."[2]

This paper uses the set of requests and challenges as a proxy for assessing those technologies and applications around which China's commercial-based MCF innovation efforts revolve and for identifying associated actors. Of course, the Rapid Response Teams and their projects do not reflect the entirety of China's national defense science and technology project or the MCF apparatus. By any metric—funding, employees, projects—these constitute a relatively small program. However, they do exist at a ripe intersection of MCF and innovation. The insight they provide might not be exhaustive, but it is unusually valuable.

Rapid Innovation with Chinese Characteristics

The National Defense Science and Technology Innovation Rapid Response Teams' technology requests and challenges reveal a decided focus on information-related technologies, namely autonomy and sensing. Nine of the Rapid Response Teams' 57 known technology requests relate directly to unmanned aerial vehicles (UAVs), and another three relate to

---

[1] Shenzhen is considered China's high-tech hub (see, for example, Borak & Xue, 2021).

[2] See, for example, the shipborne 3D printing intelligent rapid repair manufacturing technology and the high voltage, high current pulse inductor coil assembly projects.

unmanned underwater vehicles (UUVs).[3] Those are matched by a similar number of requests for sensing and network capabilities that could be docked into them. Maritime examples include video-based maritime intelligent target recognition technology, shallow surface target detection technology, software-defined multifunctional sonar, environmental perception adaptive underwater acoustic mobile ad hoc network technology, and underwater distributed optical fiber sensing technology. Advanced algorithms and software do not feature prominently in this data set. That may be a function of information availability (e.g., stricter confidentiality protocols around their development) or time frame (i.e., they may require more lead time and therefore not fall under the purview of rapid innovation). The gap could also reflect a difference in prioritization (see Tables 1 and 2).

Table 1. Rapid Response Teams Requests and Challenges, by Technology Focus

Technology	Project Count
Autonomy/Guidance	17
Sensing/Networks	17
Manufacturing	14
Power	8
Medical	3
Explosives	2
Operations and Maintenance	1
Flight	1

Table 2. Rapid Response Teams Requests and Challenges, by Domain Focus

Domain	Project Count
Maritime	16
Air	14
Energy	7
Advanced Materials	6
Ground	6
Information Space	6
COVID-19	4
Electronics and Circuits	4

Across the board, regardless of the focus domain or technology, the data set reveals an emphasis on information collection, integration, and operationalization. This emphasis is evident in the host of technology requests focused on autonomy and sensing (e.g., video-based maritime intelligent target recognition technology). The emphasis is even more evident in the Rapid Response Teams' technology challenges.

[3] In what might reflect an asymmetry of technological approach, the technological requirements for one of these projects, the "hand-thrown micro aircraft," requested in January 2020, note that the communication module should "later provide the connection function with the 5G communication network."

Table 3. Rapid Response Team Challenges: 2018–2020

Date	Name	Additional Organizing Entities	Description
Dec 2018	Exoskeleton quick sprint project	-	Execution period of 3 years (2018.12 ~ 2021.12); carried out in three 12-month stages, including two application sprints (6 months) and one technical sprint.
Apr 2019	Massive Optical Remote Sensing Data Intelligent Processing Algorithm Competition	Harbin Institute of Technology (Shenzhen); Aerospace Dongfanghong Satellite Co.; Changchun Institute of Optics, Fine Mechanics and Physics (Chinese Academy of Sciences); Shenzhen Pusheng Intelligent Technology Co.; Zhuhai Orbit Aerospace Technology Co.; China Yuan Jinxin (Dalian) Information Technology Co.	Promote the development of massive optical remote sensing satellite data processing technology, aim at in-orbit applications, and support the realization of massive remote sensing data processing and application capabilities.
Dec 2019	Ground Target Detection and Recognition Technology in Complex Environments Challenge	National Defense Science and Technology Innovation Special Zone Expert Group, National Space Science Center of the Chinese Academy of Sciences, Institute of Automation of the Chinese Academy of Sciences.	Focus on detection and recognition needs of ground targets in complex urban scenes through algorithm performance comparison tests based on real-environment video data sets; the goal is to enhance the intelligent recognition and real-time warning capabilities of ground threat targets in the perimeter of key areas.
Jan 2020	"Ground Sentinel" Challenge	National Defense Technology Innovation Special Zone Expert Group	Focus on the needs of security prevention and control in the core area of the city and the detection and identification of ground threats, as well as to promote and lead the innovative development of the AI security field; the goal is to select new solutions for intelligent image/video analysis in natural surveillance scenarios and promote rapid development of smart security technology.
Jan 2020	"Unlimited Smart Communication-2020" Urban Environment Broadband Communication Technology Challenge	National Defense Technology Innovation Special Zone Expert Group, China Academy of Ordnance Science, PLA Army Research Institute, Chongqing High-Tech Zone Feima Innovation Research Institute, Chongqing University of Technology, China Ordnance Industry Testing Research Institute, Chongqing Science and Technology Bureau	Based on large-scale activities in the urban environment, explore the multi-scenario, multi-node dynamic communication group in the urban environment based on the complex geographic environment, electromagnetic environment, and dynamic movement of people in the city.
Jan 2020	Accurate Aerial Docking Technology of "Range Go" UAV Challenge	National Defense Technology Innovation Zone Expert Group, China Aerospace Science and Industry Corporation, The Third Institute of Aerospace Science and Technology of China, China Electronics Technology Group Corporation Electronic Science Research Institute, Shanghai Jiaotong University, University of Electronic Science and Technology, Xidian University, Yuanwang Think Tank, Shanxi Datong Aviation Sports School	This challenge focuses on the drone aerial docking process, including precise flight control of drones against aerodynamic interference, precise identification and tracking of docking points, the efficient approach strategy for air pre-docking, and collision avoidance and obstacle avoidance technology.

Of the six known challenges held since December 2018, four revolve around information collection, processing, and analysis, and a fifth on aerial docking of UAVs (see Table 3). A host of other entities cooperate with the Rapid Response Teams in organizing those challenges, suggesting a breadth of data types, collection methods, and applications—as well as of players—involved in the MCF information innovation process. For example, the organizing entities for the January 2020 Unlimited Smart Communication Challenge span military, industrial, academic, and government units. They include the China Academy of Ordnance Science and China Ordnance Industry Testing Research Institute, both of which exist under China North Industries Group Corporation (NORINCO), one of China's centrally state-owned defense industry conglomerates; the People's Liberation Army's (PLA's) Army Research Institute; the Chongqing High-Tech Zone Feima Innovation Research Institute (重庆高新区飞马创新研究院), a research institute dedicated to the development of dual-use technology and housed within a MCF-focused industry zone (Chongqing University Graduate Employment Information Network, n.d.); as well as the Chongqing University of Technology and the Chongqing Science and Technology Bureau (National Defense Science and Technology Innovation Rapid Response Team [Shenzhen], 2020).

That 2020 Unlimited Smart Communication Challenge also points to another core aspect of the MCF information project reflected in the Rapid Response Team data set: These requests and competitions do not simply entail converting civilian inputs to military use cases. They also suggest an effort to use civilian outputs to project military-relevant power. The 2020 Unlimited Smart Communication Challenge revolved around using flexible broadband communication and dynamic networking capabilities in an urban environment. Participating teams connected competition equipment—distributed in buildings, across city blocks, and in underground passages—to a test access platform; their scores were determined by video and voice transmission capability as well as average transmission delay. (National Defense Science and Technology Innovation Rapid Response Team [Shenzhen], 2020). This contest took place in a civilian environment, leveraging commercial technology, but was organized by military entities under the mandate of national defense innovation.

The Ground Sentinel Challenge, also held in 2020, tells a similar story. The Rapid Response Team's description of the competition describes it as "focusing on the needs of security prevention and control in the core area of the city and the detection and identification of ground threats." Accordingly, the challenge revolved around intelligent video and image analysis solutions in "natural surveillance scenarios" to "promote the rapid development of smart security technology" for urban areas. Goals included unconstrained face recognition, suspicious object recognition, and dangerous behavior recognition. In determining the participating entities, priority was given to applicants who had past histories of supporting national ministry and provincial security systems (National Defense Science and Technology Innovation Rapid Response Team, 2020). This contest, too, took place in a civilian environment, focused on civilian subjects, leveraging commercial technology.

This blurred line between national defense innovation and civilian-focused applications is by no means surprising. It underlines a critical nuance of China's MCF strategy. MCF is not only about fusing military and civilian inputs for the sake of Beijing's power projection—using commercial UAV technologies in military operations, for example. The MCF project also fuses military and civilian *outputs*, whether industrial positioning or technological applications, in order to bolster China's power projection. In this data set, China's smart city technology, information collection, and control stands out as a ripe example. The point applies more broadly, especially across emerging infrastructures, networks, and platforms (de La Bruyere, 2020b).

The Rapid Response Teams' technological requests also point to the other direction in which MCF can work: transfer of resources from the military to the civilian. Four of the requests issued in 2020 focus on technologies to be used in COVID-19 response, ranging from personal protective equipment to UAVs for operation in areas affected by the epidemic (see Table 4). Perhaps the most interesting element of these requests is their timeline: The vast majority of other technological requests call for delivery within a 12-month window. The shortest time frame for non–COVID-19 relevant technological requests, appearing only on three occasions, is 6 months. However, three of the four COVID-19 relevant requests call for delivery in a matter of days, including as few as 5. These requests provide an example of Defense Production Act–like Chinese acquisition. They also suggest the timing with which such rapid acquisition might take place.

Table 4. COVID-19 Relevant Technology Requests

Date	Request	Timeline (months)
1/30/20	Research on the prediction of the spread of harmful microorganisms	6
2/8/20	Emergency operation UAV in epidemic area	>1 (5 days)
2/11/20	A batch disinfection system for personal protective equipment that can be quickly deployed	>1 (30 days)
2/11/20	Protective clothing that can be re-sterilized and used	>1 (20 days)

Conclusion: Implications for U.S. Acquisition

An appropriate response to the asymmetric threat of China's MCF strategy will require a prioritization framework. The United States, and the U.S. acquisition system, will have to determine what technologies to protect, what resources and manufacturing capabilities to develop, and through what infrastructures to compete. Those determinations must be informed by U.S. goals, factoring in U.S. strengths and weaknesses—what is effective and ineffective, efficient and inefficient in the U.S. system.

The determinations must also be informed by the adversary—by how China operates, to what ends, and with what resources. The United States must understand

- those technologies and applications that Beijing prioritizes for the great power competition;
- those technologies and applications for which Beijing relies on the civilian sector; and
- those actors within the Chinese system that play the most critical roles in the process.

This paper leverages the technological demands of China's National Defense Science and Technology Innovation Rapid Response Teams in order to begin to answer those questions—and to provide an example of the sort of data sets, and analysis, that might be used to answer them more comprehensively moving forward. This single data set and this paper's findings are only one piece of the overall puzzle. Still, direct implications emerge for an adversary-informed prioritization framework.

- Beijing's MCF innovation ecosystem clearly prioritizes information technology, broadly.
- More specifically, entities charged with fusing commercial and military cutting-edge technology appear to prioritize autonomous systems (e.g., UAVs, UUVs), sensing and network technologies to dock into and connect them, and information aggregation and

analysis platforms more specifically. Advanced algorithms and software do not feature prominently in this data set. That may be a function of information availability (e.g., stricter confidentiality protocols around their development) or time frame (i.e., they may require more lead time and, therefore, not fall under the purview of rapid innovation). The gap could also reflect a difference in prioritization.

- The MCF innovation ecosystem relies on participation and support from a range of government, industrial, and academic players, both military and civilian—local governments as well as military organs; state-owned defense industry conglomerates (e.g., NORINCO) as well as private, commercial companies (e.g., Shenzhen Pusheng Intelligent Technology Co., a subsidiary of Guangzhou Hengchuang Intelligent Technology Co.); state and military research institutes (e.g., the Chinese Academy of Sciences) as well as universities.

These findings can directly inform DoD acquisition processes. Defensively, the DoD might take particular efforts to protect those technological areas that Beijing both focuses on and leverages the commercial sector in developing. Defensive measures can include screening of commercial DoD partners (e.g., Small Business Innovation Research [SBIR] applicants). Defensive measures can also include more systemic efforts, like the Trusted Capital Program. The DoD might also use the taxonomy of players contributing to MCF innovation efforts to inform monitoring of, and restrictions on, Chinese entities, including expansion of the companies identified under Section 1237 of the 1999 National Defense Authorization Act and corresponding actions. More offensively, these findings—and the more granular details of the technologies being requested and developed—might assist U.S. efforts to identify, and exploit, weaknesses of the Chinese defense apparatus.

Today's great power competition has been described as a technological race. Before we start running, let's figure out where the finish line is. And how to stop our adversary from drafting off of us.

Appendix. Rapid Response Team Technology Requests, 2018–2020

Date	Request	Timeline (months)
4/11/18	Conformal antenna and circuit integration rapid manufacturing technology	12
4/11/18	High-pressure hydraulic control terminal control technology	12
4/11/18	Video-based maritime intelligent target recognition technology	12
4/11/18	High voltage, strong pulse current flexible coaxial cable	12
4/11/18	High-precision, high-reliability, low-cost servo drive controller	12
4/11/18	Intelligent human–computer interaction module	12
4/11/18	Micro system heat sink	12
7/24/18	Shipborne 3D printing "first aid kit" intelligent rapid repair manufacturing technology	-
6/3/19	Deepwater heavy-duty manipulator	12
6/3/19	Deep water, high power density, high reliability, brushless propulsion motor	12
10/15/19	Low-cost small air-to-surface missile technology	-
11/12/19	Low-cost small aircraft that can be deployed at high altitudes	-
11/12/19	High-altitude flying technology research	-
11/12/19	Vertical takeoff and landing aircraft dense and rapid deployment system	-
12/23/19	Self-protecting solder paste for high-reliability three-dimensional packaging	-

12/23/19	0.6-inch high resolution silicon-based OLED microdisplay	-
12/23/19	Strong special-shaped tempered glass	-
12/23/19	Servo for high-overload low-cost missiles	-
12/23/19	Research on low-cost, high-temperature, infrared and low-radiation technology for vehicles	-
12/23/19	High rigidity and impact resistance precision reducer	-
1/7/20	Long-distance fishing net detection technology	-
1/7/20	Wide temperature semiconductor laser research	-
1/7/20	High voltage, high current pulse inductor coil assembly	6
1/7/20	High-reliability DC brushless special motor with hollow cup	-
1/7/20	Anti-strong magnetic field and high overload DC brushless special motor	-
1/13/20	Integrated communication system and terminal for indoor distributed communication and positioning based on illumination light	12
1/13/20	Hand-thrown micro aircraft	-
1/21/20	Shallow surface target CT system	12
1/21/20	Research on shallow surface target detection technology	6
1/30/20	Research on the prediction of the spread of harmful microorganisms	6
2/8/20	Emergency operation UAV in epidemic area	>1 (5 days)
2/11/20	A batch disinfection system for personal protective equipment that can be quickly deployed	>1 (30 days)
2/11/20	Protective clothing that can be re-sterilized and used	>1 (20 days)
2/24/20	Software-defined multifunctional sonar	12
2/24/20	Research on the preparation technology of ceramic-based material metal coating microstructure	6
3/9/20	“Low, slow, small” drone detection technology in urban complex environment	12
3/9/20	“Low, slow, small” drone agile disposal technology in complex urban environment	12
3/9/20	“Low, slow, small” UAV link takeover technology in urban complex environment	12
3/12/20	Development and application of on-site fatigue damage testing technology under extreme conditions	-
3/12/20	High-speed UUV near-field track electromagnetic passive measurement system	12
3/12/20	New aircraft anti-bird strike coating	12
3/20/20	Formation collaborative navigation technology of underwater man-machine hybrid system	12
3/20/20	Environmental perception adaptive underwater acoustic mobile ad hoc network technology	12
3/20/20	Multi-domain and cross-media integrated communication and networking technology	12
3/20/20	Intelligent detection and recognition technology for typical targets of divers	12
3/20/20	Smart diving mask integration optimization technology	-
3/26/20	UAV rapid obstacle avoidance technology	8
3/26/20	UAV rapid target recognition and tracking technology	12
3/26/20	Enhanced low-light scope	12

3/26/20	Personnel identification technology in dark environment	12
4/16/20	Configurable RF chip	12
4/16/20	Modular robot	12
4/26/20	Underwater special connector	12
4/26/20	Lightweight high-performance electromagnetic shielding paint	9
4/30/20	Underwater distributed optical fiber sensing technology	12
4/30/20	High-performance four-quadrant laser receiver components	12
4/30/20	High-performance high-voltage DC/DC power supply module	10

Reference

Borak, M., & Xue, Y. (2021, April 4). How Shenzhen, the hi-tech hub of China, became the drone capital of the world. *South China Morning Post.* https://www.scmp.com/tech/big-tech/article/3128004/how-shenzhen-hi-tech-hub-china-became-drone-capital-world

Bradsher, K. (2010, September 22). Amid tension, China blocks vital exports to Japan. *The New York Times*. https://www.nytimes.com/2010/09/23/business/global/23rare.html

Chongqing University Graduate Employment Information Network. (n.d.). *重庆高新区飞马创新研究院简介* [Introduction of Chongqing high-tech zone Feima Innovation Research Institute]. Retrieved from http://job.cqu.edu.cn/company/view/id/527203

de La Bruyere, E. (2020a, April 12). The new metrics for building geopolitical power in a new world. *The National Interest*. https://nationalinterest.org/feature/new-metrics-building-geopolitical-power-new-world-143147?page=0%2C1

de La Bruyere, E. (2020b, April 28). Beijing's innovation strategy: Threat-informed acquisition for an era of great power competition. *Proceedings of the 17th Annual Acquisition Research Symposium*. https://dair.nps.edu/bitstream/123456789/4251/1/SYM-AM-20-091.pdf

de La Bruyere, E., & Picarsic, N. (2019, May). *Military–civil fusion: China's approach to R&D, implications for peacetime competition, and crafting a U.S. strategy* [Paper presentation]. Naval Postgraduate School. Defense Acquisition Research Symposium.

de La Bruyere, E., & Picarsic, N. (2020, June 22). How to beat China's military–civil fusion. *The American Interest*. https://www.the-american-interest.com/2020/06/22/how-to-beat-chinas-military-civil-fusion/

National Defense Science and Technology Innovation Rapid Response Team. (2020, January 6). *"地面哨兵"挑战赛通知* [Notice of "ground sentinel" challenge].

National Defense Science and Technology Innovation Rapid Response Team (Shenzhen). (2020, January 16). *"无限智通-2020"城市环境宽带通信技术挑战赛通知* [Notice of "unlimited smart communication-2020" urban environment broadband communication technology challenge].

Office of the Military-Civil Fusion Development Committee of the Shenzhen Municipal Committee of the Communist Party of China. (2018, April 11). *快响小组简介* [Introduction to the rapid response team].

Swanson, A. (2020, November 12). Trump bars investment in Chinese firms with military ties. *The New York Times*. https://www.nytimes.com/2020/11/12/business/economy/trump-china-investment-ban.html

PANEL 21. PRINCIPLES FOR MANAGING SOFTWARE ACQUISITION

Thursday, May 13, 2021	
10:45 a.m. – 12:00 p.m.	**Chair: Robert Stoddard,** Principal Researcher, Software Engineering Institute, Carnegie Mellon University ***Speed Limits for Software Acquisition*** David Tate, Institute for Defense Analyses John Bailey, Institute for Defense Analyses ***Using Value Engineering to Propel Cyber-Physical Systems Acquisition*** Nickolas Guertin, Carnegie Mellon University Alfred Schenker, Carnegie Mellon University ***Integration of Production Management into Software Development and Acquisition Processes*** Caitlin Kenney, University of Maryland

Robert Stoddard—currently serves as a Senior Member of the Technical Staff within the Carnegie Mellon University federally-funded research and development center known as the Software Engineering Institute (SEI). Robert is responsible for research, Air Force Logistic Center client consulting and the development and delivery of advanced training on cost estimation and software predictive analytics, specifically with regards to CMMI High Maturity process performance baselines and models. Robert continues to promote the integrated use of Six Sigma and CMMI as exemplified in a 2008 published book entitled “CMMI and Six Sigma: Partners in Process Improvement” by Addison Wesley, and is an invited speaker to a wide variety of annual conferences and symposiums held by the SEI and the American Society for Quality (ASQ). Robert remains active with both the IEEE Reliability Society and the ASQ Software Division. Robert holds certifications with the ASQ on five topics including Six Sigma Black Belt, Reliability Engineering, Quality Engineering, Software Quality Engineering and Quality Audits. Robert became a Motorola-certified Six Sigma Black Belt in 1993 and Motorola-certified Six Sigma Master Black Belt in 2003. Robert earned a B.S in Finance and Accounting from the University of Maine, an M.S. in Systems Management from the University of Southern California, and has completed PhD course work in reliability engineering and engineering management. Prior to joining the SEI in 2005, Robert served for seven years as a Motorola Director of Quality and Distinguished Member of Technical Staff and for 14 years as a Texas Instruments Site Software Quality Manager.

Factors Limiting the Speed of Software Acquisition

David Tate—joined the research staff of the Institute for Defense Analyses' (IDA) Cost Analysis and Research Division in 2000. Prior to that, he was an Assistant Professor of Industrial Engineering at the University of Pittsburgh, and the Senior Operations Research Analyst (Telecom) for Decision-Science Applications, Inc. At IDA, he has worked on a wide variety of resource analysis and quantitative modeling projects related to national security. These include an independent cost estimate of Future Combat Systems development costs, investigation of apparent inequities in Veterans' Disability Benefit adjudications, and modeling and optimization of resource-constrained acquisition portfolios. Tate holds bachelor's degrees in philosophy and mathematical sciences from the Johns Hopkins University, and MS and PhD degrees in operations research from Cornell University. [dtate@ida.org]

John Bailey—has worked for both the former Computer Science Division and the Cost Analysis and Research Division of IDA since the 1980s. Prior to that, he conducted software programming research for General Electric and co-founded Software Metrics, Inc. While completing his PhD in software reuse methodology, he helped Rational establish its Ada development environment, which was later acquired by IBM. In addition to his software technology research, Bailey has recently been adapting computer science best practices to livestock farming in northern Virginia. [jbailey@ida.org]

Abstract

The time required to complete a software development or upgrade, like any other project, depends on the content of the project, how it is managed, and its preexisting conditions. The factors limiting the speed of software acquisition and enhancement fall into these principal categories, in rough order of importance:

1. Required functionality—what you need the software to do (and not do)
2. Architecture—the organizing structure of the software and its operating environment
3. Technology maturity—to what extent the intended design uses novel solutions
4. Resources—the people, skills, funds, data, and infrastructure needed to do the work
5. Testing strategy—acquiring the information to fix defects early in development
6. Contract structure—the alignment of contractor incentives with DoD satisfaction
7. Change management—the processes for trading off performance, schedule, cost, and sustainability

This paper considers this taxonomy and examines how each category affects the pace of development.

Introduction

Improving the agility of defense acquisition is a high priority goal for both the Office of the Secretary of Defense (OSD) and the military departments. Improving the speed at which the Department of Defense (DoD) can develop, deploy, and update software-enabled capabilities would enable more general acquisition agility, given modern defense systems' critical dependence on software. This point was emphasized in both a 2018 Defense Science Board report and in the 2019 Defense Innovation Board (DIB) Software Acquisition and Practices (SWAP) study.

There is now widespread consensus that software development is the pacing activity not only in traditional information systems acquisition, but in all major defense systems acquisition. As early as 20 years ago, the Naval Postgraduate School was already teaching students that "software is now acknowledged as sitting on the critical path of most major weapon systems and is widely regarded as the highest-risk element in an acquisition" (Nissen, 2019). Other recent research has confirmed this insight (Tate, 2016). If we want our acquisition enterprise to deliver capabilities more quickly and upgrade them more frequently, we need to speed up defense software development.

Given this universal interest in accelerating software acquisition and sustainment, it is important to understand the fundamental factors that limit how quickly software can be developed, deployed, and upgraded. These factors fall into seven principal categories; in rough order of importance, they are:

1. Required functionality—what you need the software to do (and not do)
2. Architecture—the organizing structure of the software and its operating environment
3. Technology maturity—to what extent the intended design uses novel solutions
4. Resources—the people, skills, funds, data, and infrastructure needed to do the work
5. Testing strategy—acquiring the information to fix defects early in development
6. Contract structure—the alignment of contractor incentives with DoD satisfaction
7. Change management—the processes for trading off performance, schedule, cost, and sustainability

This paper considers each category in this taxonomy in turn and examines how each affects the pace of both initial development and future upgrades.

Required Functionality

The time to develop and field a software-enabled capability is primarily determined by what the software needs to do and in what context—the *content* of the software. This seems obvious and has been common knowledge with regard to the cost of systems for decades, but it is still common for schedules to be imposed on software development efforts independent of their content.

Although content is often thought of in terms of requirements, it is important to note that, historically, not all mandatory performance attributes have been managed in the same way. In particular, "negative requirements" that describe outcomes the system should avoid are often handled outside the primary requirements management process once development begins. They are thus at risk of being neglected as the design evolves or of being waived or relaxed when programs face time pressure. Negative requirements include cybersecurity specifications, safety standards, reliability/availability/maintainability (RAM) thresholds, interoperability requirements, and other aspects of system suitability.

Current software development best practices (notably agile and DevOps) specify successive releases of a software product to implement an increasingly complete solution. The first mission-capable release of the software is called the minimum viable capability release (MVCR). DoD Instruction 5000.87 "Operation of the Software Acquisition Pathway" says defining the MVCR is the responsibility of the PM and the sponsor and is the earliest release that will meet the "initial warfighting capabilities to enhance mission outcomes" (DoD, 2020). It further says the MVCR "must be deployed to an operational environment within 1 year after the date on which funds are first obligated to acquire or develop new software capability including appropriate operational test." As will be discussed under Architecture, this may be a tall order for major defense systems that must be structured to withstand decades of sustainment and enhancement. We may learn whether large, long-lived systems can be granted an exception to this rule.

The minimum time to operationally deploy the first version of a new software-enabled system is therefore driven by the content of the MVCR. What counts as "initial warfighting capability" is a stakeholder decision that involves trade-offs between lead time and capability—discussed further in the "Change Management" section. Minimum standards for the negative requirements must also be included in the definition of the MVCR. Note that a system that is not secure or cannot work with real data cannot be the MVCR—it is not yet useful for any actual operational mission.

The MVCR is the first build that can be released to operations, but earlier builds are also important. In particular, the minimum viable product (MVP) plays a key role in development. DoDI 5000.87 defines the MVP as "an early version of the software to deliver or field basic capabilities to users to evaluate and provide feedback on. Insights from MVPs help shape scope, requirements, and design" (DoD, 2020). It is possible that the MVP may also serve as the MVCR, although an iterative development would likely first release an MVP to achieve consensus among users, sponsor, and developers to ensure that the product is on track to supply the expected capabilities. As will be discussed under Architecture, it may not always be sensible to enhance an MVP to obtain an MVCR. An MVP may be a prototype for demonstration or user involvement purposes only and might ignore important design decisions about security, maintainability, or other qualities that are essential aspects of the underlying architecture of a system. If this is the case, project cost and schedule estimates should account for any technical debt incurred by the choice of an MVP that is not a subset of the MVCR.

For any software increment, there is a theoretically ideal staffing profile that would achieve the most efficient compromise between maximizing work rate and incurring unnecessary technical debt. Trying to go faster than this ideal schedule often leads to significant redesign, rework, and delay. Going slower than the ideal schedule (e.g., due to resource constraints or overly conservative limits on technical debt) also leads to unnecessary delay and inefficient use of labor resources.

The length of the ideal schedule depends on how tightly coupled the software subsystems (modules) are. Software designs that do not explicitly enforce independence of subtasks[1] can lead to systems that are not modular enough to support full parallelization of development efforts, resulting in longer development schedules for future increments.

It should be noted that agile/lean methods (where feasible) are (at best) modestly faster than traditional waterfall development for a software increment with fixed requirements. The time savings realized by agile development come mostly from *not* implementing functions that turn out not to be critical—the savings are due to work avoided, not higher productivity. Thus, the effort to complete the MVP and MVCR will be roughly the same as for a traditional waterfall development approach to that much code.

[1] Architectural enforcement of subtask independence is sometimes referred to as "separation of concerns" in the computer science literature.

Architecture

As noted previously, software and hardware architectures for platforms and systems strongly influence how rapidly and effectively new capabilities can be added. Therefore, the selection of an architecture for a new defense system has important consequences throughout the acquisition process. For major defense systems, achieving an MVCR that is structured to support decades of ongoing agile improvements can require years of initial architecture, design, and development effort.

Every software-enabled system requires some architecture and design prior to the beginning of coding. The amount of effort needed for an architecture depends on a number of factors:

- How important is it for the system to be upgradeable in the future?
- How fast/cheap/frequent/extensive do those upgrades need to be?
- What is the intended lifespan of the platform or system and its upgrades?
- How portable/reusable does the software need to be?
- What other systems will the new system need to interoperate with?
- What are the cybersecurity needs of the system?
- How much is the cyber threat environment expected to change over time?
- How close to the cutting edge of current technical capabilities is the system intended to operate?

The answers to these questions help determine the appropriate software architecture for the new system, which (combined with hardware architecture) in turn is a determinant of the required cost and schedule to implement the MVCR. Systems that must be easy and cheap to modify over a long-life cycle require more architecture effort (and thus more content in the MVCR and a longer initial development increment). In particular, the more you want to use agile/lean/DevSecOps methods for ongoing parallel system improvement of large systems, or to complete upgrades among multiple vendors, the more you need a modular architecture.

Thus, the architecture is part of the MVCR content. An early increment (such as the MVP) with an architecture that does not support MVCR (and future) requirements is just a prototype but may be useful to demonstrate intended functions and elicit user feedback. In particular, systems with significant cybersecurity concerns must address those concerns in the MVCR architecture and design requirements. Adding security later does not work. For some mission needs, a quick-and-dirty architecture that supports current (but not future) requirements might be appropriate, providing some needed capability quickly at the expense of future sustainment and upgrades. This choice to field an MVCR that has limited ability to support future upgrades should be a conscious decision at design time, with its implications understood by stakeholders.

Similarly, when adding new capabilities to a legacy system whose architecture was not originally designed to support ongoing upgrades, stakeholders need to decide whether it is more important to implement specific new capabilities quickly or to make the system more rapidly upgradeable in the future. In some cases, it may be preferable to accept cost and delay today to re-architect and re-write the legacy code to enable a longer and more agile future upgrade path.

Technology Maturity

Whenever system designs involve technologies that are less than fully mature, the software and hardware design processes contain an element of experimentation and exploration. This can lead to a redesign that changes the nature of the software required or one that shifts functions originally envisioned as being performed by hardware into software. This adds both new development and regression testing burdens to the software project and lies on the critical path for the program. As a result, technologies with a current Technology Readiness Level (TRL) lower than 6 (as defined in Title 10 U.S. Code, § 2366b(a)(3)(D)) can contribute to software delays.

At the same time, critical software technologies tend to be overlooked in Independent Technology Readiness Assessments. Past examples where this has led to major program delays include sensor fusion algorithms, ad hoc network management, and automated traction control for off-road vehicles.[2]

The DoD has recently stated in multiple venues that several new software technologies, including machine learning, artificial intelligence, and autonomous systems, will be key enablers of future U.S. military capabilities. These are immature technologies that impose novel burdens throughout the acquisition life cycle, including learning how to specify requirements that are testable, how to validate and maintain training data sets for machine learning, and how to assess the effectiveness and suitability of machine cognition and human-machine teaming concepts (Tate & Sparrow, 2018).

It is important to remember that, for TRLs greater than 4, technology maturity is always measured relative to a specific set of requirements and intended operational environments.[3] A technology can be mature (i.e., TRL 6 or higher) for certain operational uses or environments, yet immature for others, as demonstrated by the traction control example. This is just as true of software technologies as it is for lasers, engines, or sensors. Failure to recognize that a given software technology is critical to success, or that it has not yet been successfully demonstrated in the intended operational role and environment, can lead to costly delays in development down the road. (For an annotated list of TRL definitions, see DoD, n.d.)

Resources

Software development is about sufficient skilled labor, appropriate infrastructure, and (increasingly) adequate data. As a result, the most important potential resource issues that software projects can face are a shortage of appropriately skilled personnel (either through insufficient funding to purchase their labor or lack of access to the right talent), or insufficient access to platforms or data sets needed to implement the desired capabilities. These challenges are time-phased—you need to simultaneously have enough of each of these things at the right times to support the ideal schedule of the project. Valid input data is often needed early in development. For machine learning,

---

[2] Traction control algorithms used in commercial vehicles assume that the vehicle is driving on a hard surface. They do not perform well in off-road situations where the loss of traction is due to surface failure (e.g., mud or sand) rather than being due to low friction with the road.

[3] TRL 4 requires "Component and/or breadboard validation in a laboratory environment," which is independent of requirements or intended operational environments. TRL 5 requires "Component and/or breadboard validation in a relevant environment," which explicitly compares the environments where the technology has already been demonstrated against some intended environment.

obtaining a sufficient quantity of accurately labeled training data can be a bottleneck in capability development.

An adequate supply of appropriately skilled developers is not a given. The supply of cleared software professionals is not keeping up with the demand for national security software (Tate, 2017). As a result, programs often have trouble hiring and retaining the required quality and quantity of software talent. Staffing shortfalls cause delays, and project staff turnover and loss of institutional knowledge cause additional delays. Mergers and acquisitions, common in defense industries, sometimes exacerbate this problem. Either not having enough people or not having sufficiently talented people as part of an experienced team can cause program delays.

Similarly, funding instability or mismatches between planned spending profiles and actual workloads can lead to inefficient use of labor resources and subsequent schedule delays. This is particularly true for projects with significant experimentation and discovery during the development of the MVCR, as the duration and cost of the experimentation cannot be predicted with accuracy. Such projects should ideally be identified in advance and provided with significant levels of contingency funding, but this is politically difficult in many cases.

For many software-enabled defense systems, the direct costs of software development are a relatively small fraction of total program costs. However, as noted previously, software delays tend to be on the critical path, resulting in delays that add cost at a roughly constant burn rate. For major programs, this represents not merely a delay in achieving the desired capabilities, but also a large opportunity cost, wasting funds that could have been used elsewhere.

It is important to note that the majority of software development effort (and cost)—often as much as 80% of total effort—occurs *after* the system has initially been fielded. Software deployment generates an ongoing future workload that lasts as long as the system is in use. The resources needed to perform future upgrades and integration are generally the same resources needed for initial development. This means that future enhancements compete with new systems for people, as well as for dollars.

In addition to people, many advanced software capabilities also require specialized infrastructure and data. These can include supercomputing resources, labeled training data sets, modeling and simulation support, and test ranges with specialized instrumentation. Shortfalls in any of these areas can also lead to significant fielding delays.

Testing Strategy

Although it is theoretically possible to do "too much testing" in a software-intensive development program, this is nearly unknown in practice. Finding and fixing defects can be the most expensive identifiable software cost driver, and high defect levels are the primary reason for software schedule and cost overruns (Jones & Bonsignour, 2012). Software development success is strongly correlated with a commitment to continuous, rigorous testing from day one, identifying defects early—while they are still relatively easy to isolate and correct.

While it is hard to do too much testing, it is easy to do testing wrong. In order to realize the benefits of continuous testing and early defect elimination, the test strategy for the program must ensure that the right information is collected at the right times and fed back into the development process efficiently. Unaddressed defects have cascading effects—they interfere with development of additional capabilities, mask other defects, and complicate the required fixes. Past research has found that the average cost to remedy a defect grows by an order of magnitude between the design phase and the implementation phase and grows by another order of magnitude between implementation and post-deployment maintenance (Kan, 2003). Bear in mind here that "defects" can include flaws in requirements, inaccurate documentation, adherence to inappropriate architectural specifications or standards, or other issues that are not necessarily "bugs" in the code.

The most common (and costly) testing error by far is not testing early enough. For example, the Warfighter Information Network–Tactical (WIN-T) underwent an initial operational test and evaluation (IOT&E) in May 2012, with planned full-rate production (FRP) scheduled for September 2012. IOT&E revealed numerous problems that should have been caught much earlier in testing, including poor network stability, poor performance, and poor reliability. FRP was deferred until after a follow-on operational test and evaluation (FOT&E) could be conducted. That FOT&E revealed continuing problems, leading to another schedule slip. The system finally entered FRP in June 2015.

When testing is done right, test activities make the project cheaper and faster than it otherwise would have been. Delays "due to testing" are almost always actually delays due to *not* testing—that is, extra time required to fix defects that were discovered late in development (when testing is expensive) or after fielding (when testing is very expensive) because of insufficient early testing. Defects discovered after fielding are often due to incorrect or misunderstood requirements. Finding ways to incentivize contractors (and program offices) to perform adequate testing is challenging.

For new systems, or for modifications of legacy systems that are intended to transition to agile development for future increments, it is important to develop the associated agile test strategies and automated testing infrastructure during MVCR development. This should involve all stakeholders and be done in parallel with (and be informed by) MVCR design and implementation.

Contract Structure

Typically, a small percentage of the overall profit on a major defense system comes from software development. At the same time, the future upgrade and maintenance contracts for a software-intensive system can provide an open-ended source of future revenue. This is analogous to losing money on each laser printer sold, but making big profits on the toner cartridges. This gives contractors a strong incentive to erect barriers to sustainment competition that all too often also function as barriers to efficient upgrades and modernization.

The DoD would prefer that all software be modular, reusable, and open to enhancement by third-party vendors. It is very difficult to write a contract that requires, or even incentivizes, the prime contractor to make that happen. Modular design makes it easy for competitors to break the prime's monopoly on sustainment; reusable code makes it possible for competitors to benefit from the prime's work and potentially to gain access to their proprietary technologies. For these reasons, contractors prefer to restrict government data rights as much as they can. As Van Atta (2017) notes, "There is a vast legacy of defense systems, amounting to billions of dollars in sustainment costs, for which the necessary [intellectual property] data and rights for organic depot or competitive sustainment were not acquired."

The services are sometimes complicit here—for example, by waiving statutory requirements for modular open systems architecture (MOSA) on the grounds that requiring MOSA would delay initial fielding and cost too much. Even if this were true, it is a false economy—the delays and costs of fielding subsequent upgraded capabilities for a major system will easily outweigh the original savings. As an example, the AWACS Block 40/45 upgrade program, intended to migrate the hardware and software architectures and applications on the E-3 AWACS aircraft from legacy proprietary systems to new open architecture hardware and software, has now been struggling for nearly 20 years to match the existing operational capabilities of the legacy Block 30/35 aircraft.

Admittedly, it is hard to write requirements for modularity, openness, and data rights that are verifiable at the time of system delivery and that actually make it likely that future upgrades will be easier and cheaper. This is similar to the problem of writing software quality requirements that are enforceable. Although quality requirements can sometimes be enforced through user participation in agile development teams, during development it is hard for users and other stakeholders to tell whether the choices being made will actually facilitate future upgrades as desired. Furthermore, current law prohibits making access to intellectual property rights a condition of contract award (although access to data rights can be an evaluation factor).

The DoD often struggles with aligning contractor incentives with service interests, so that the contractor profits more only when the service gets more of what it really needs (including agility). There is no free lunch—firms will require more compensation up front for initial system development if they cannot count on monopoly profits during sustainment. If the services are serious about wanting to speed up deployment of future capabilities, they must accept this and plan for it by developing contracting strategies that will allow them to realize the benefits of open system architectures.

Change Management

There is a widening disconnect in the defense acquisition world between the people who control system requirements and the people who actually develop and field systems. This is true of all requirements, but especially true of software requirements (and system requirements that end up being implemented in software). In theory, the process looks something like this:

1. Characterize the future fight and define mission needs.
2. Identify capability gaps and alternative ways to mitigate them.
3. Analyze the alternatives and select a preferred alternative.
4. Set threshold requirements.
5. Develop and field a system that meets those requirements.

In practice, there are problems with how this is realized. Translating future warfighting concepts into specific mission needs is hard, and translating mission needs into threshold performance requirements for individual systems is even harder. Thresholds are often set at aspirational levels, rather than at genuine threshold levels below which there would be no military utility. Changes during development concerning what to implement in hardware versus what to implement in software can lead to changes in system architecture that break other parts of the development or hinder future maintainability. Seemingly minor changes in the intended use of machine learning subsystems can require retraining from scratch or even development of entirely new training data sets.

As a result, some programs are unexecutable as envisioned. Faced with unexecutable programs, program managers must make trades. The realities of annual funding make it easy to accept future costs and operational shortfalls (e.g., due to poor reliability or architecture violations) in order to save money and minimize delays in initial deployment. Similarly, deferring important testing is a common way to save time and money in the short run but at the expense of greater delays and costs later.

Worse yet, program managers generally do not have the authority to relax key requirements. They must appeal to a variety of senior stakeholders and achieve a consensus that relaxing a requirement is the best way forward for the program. This need for consensus also adds friction and delay to the process. Furthermore, once a program has funding authority, the services seldom step back and reconsider whether the system being developed is still the preferred alternative, given what they now know about costs and capabilities.

Agile software development methods depend on empowering a coalition of developers and users to decide what functionality is most useful on an iterative and ongoing basis. As noted previously, the services have historically been unwilling to confer that level of decision authority to development teams or to ensure embedded stakeholder support (including actual users) in program offices. This is especially true for systems that are not "pure" software systems—it is much easier to devolve authority for user interface design and database transactions than it is to devolve authority to change the operational specifications of weapon systems. Nevertheless, to realize the benefits of agile development, programs will need to have significantly more authority to prioritize and trade requirements, to enforce architectures, and to deny external change requests than is typically delegated to them.

Summary

Software development takes time. As with any other kind of project, how much time it takes depends on the content of the project, how it is managed, and the preexisting conditions:

- How much software is needed?
- How complicated or novel is the software to be developed?
- What is the software expected to do (today and tomorrow)?
- What resources are available for the work?
- What are the contractor's incentives?
- What change management authorities does the development team have?
- Is this an initial development, or an upgrade to an existing system?
- If it's an initial development,
  - How easy to upgrade does it need to be?
  - What is the minimum viable capability release?
  - Have testable MVCR requirements been clearly specified?
  - Does the test strategy support early discovery and correction of defects?
- If it's an upgrade,
  - How well does the legacy architecture support insertion of new capability?
  - What data rights does the government own?
  - What institutional knowledge from the original development still exists?
  - Are agile development processes and tools in place?

For programs to field new software-enabled capabilities quickly, someone must have spent the time and money in the past to create an environment that supports rapid capability insertion. Ideally, this environment would include a modular (and preferably open) software architecture, adequate data rights, platforms with excess space and power available, and an industrial base that can provide enough people with the right skills, curated input and training data, and developmental test infrastructure (including modeling and simulation resources where appropriate). For agile development, this environment must also include localized change management authority within the developer/stakeholder team.

Putting these enabling environmental features into place will often require accepting delays and up-front expense, as well as reduced capability in the initial increments of those platforms. Unless service leadership accepts this reality and empowers new system developers to preserve these features even when faced with cost overruns, schedule delays, and demands for greater capability up front, they will not be implemented. Absent this kind of empowerment, software capability insertion will continue to be as slow, expensive, and unreliable as it is today.

References

10 U.S. Code. § 2366b(a)(3)(D). Major defense acquisition programs: Certification required before Milestone B approval.

Defense Innovation Board. (2019). *Software is never done: Refactoring the acquisition code for competitive advantage*. https://media.defense.gov/2019/Mar/26/2002105909/-1/-1/0/SWAP.REPORT_MAIN.BODY.3.21.19.PDF

Defense Science Board. (2018). *Design and acquisition of software for defense systems*. https://apps.dtic.mil/dtic/tr/fulltext/u2/1048883.pdf

DoD. (n.d.). *Technology readiness levels and their definitions* (DoD Deskbook 5000.2-R; Appendix 6).

DoD. (2020, October). *Operation of the software acquisition pathway* (DoD Instruction 5000.87). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500087p.PDF

Jones, C., & Bonsignour, O. (2012). *The economics of software quality*. Addison-Wesley.

Kan, S. H. (2003). *Metrics and models in software quality engineering*. Addison-Wesley.

Nissen, M. E. (1997). *JSOW alpha contracting case study*. https://fas.org/man/dod-101/sys/smart/docs/jsowcase.htm

Tate, D. M. (2016). *Acquisition cycle time: Defining the problem* (IDA Document NS D-5762). IDA. https://www.ida.org/research-and-publications/publications/all/a/ac/acquisition-cycle-time-defining-the-problem-revised

Tate, D. M. (2017). *Software productivity trends and issues* (IDA Document NS D-8367). IDA. https://www.ida.org/research-and-publications/publications/all/s/so/software-productivity-trends-and-issues-conference-paper

Tate, D. M., & Sparrow, D. A. (2018). *Acquisition challenges of autonomous systems* (IDA Document NS D-8982). IDA. https://www.ida.org/research-and-publications/publications/all/a/ac/acquisition-challenges-of-autonomous-systems-conference-paper

Van Atta, R., Kneece, R., Lippitz, M., & Patterson, C. (2017). *Department of Defense access to intellectual property for weapon systems sustainment* (IDA Paper P-8266). IDA.

Using Value Engineering to Propel Cyber-Physical Systems Acquisition

Alfred R. Schenker—has worked in the Software Engineering Institute's (SEI) Software Solutions Division for more than 20 years. He works to improve software acquisition and product development practices throughout the armed services and other organizations. He has actively worked in software process, architecture, model-based systems engineering, and metrics. Before joining the SEI, Schenker spent more than 20 years in industry as an active contributor in all phases of product development. Schenker is also an inventor and has obtained patents for a pressure switch (used in automotive airbag applications) and for a manufacturing process to seal gas inside a vessel. [ars@sei.cmu.edu]

Nickolas H. Guertin—is a Senior Software Systems Engineer at Carnegie Mellon University's Software Engineering Institute. He received a BS in engineering from the University of Washington and an MBA from Bryant University and is a registered Professional Engineer. His combined military/civilian career was in submarine operations, ship construction, weapon and sensor development, and defense acquisition improvement. He provides engineering and transformational leadership to programs in the system acquisition and engineering communities. He performs software systems engineering, improving competition, modular open systems, as well as prototyping and experimentation leadership in government and academia for software and cyber-physical systems. [nhguertin@sei.cmu.edu]

Abstract

The Department of Defense's approach to building and deploying software-intensive systems is constantly under revision. In parallel, the tools and methods to model and test architecture representations of candidate products have also evolved. We investigate the adaptation of value engineering (VE) methods into the acquisition of software-intensive weapon systems where the candidate product architecture can be modeled and used to guide implementations throughout the lifecycle. Aligning model-based engineering with VE will accelerate innovation in the development process (De Graaf et al., 2019). When used with a process performance baseline, this method can establish a comparison framework for cost-benefit analyses of alternative approaches.

VE is used to evaluate different approaches or processes to improve the end product cost (including future, unaccounted for cost overruns), performance, or quality. We believe that VE can be applied to the development of software-intensive weapons systems as well. To date, the cost to perform system modeling and virtual integration in early product development stages has been traded off against showing early results. The benefits of virtual integration are significant, especially in the long-term value of being able to evaluate system upgrades and changes both while in development and after the system transitions to sustainment.

Keywords: Value Engineering, Architecture Centric, Virtual Integration, Model Based Engineering, Model Based Systems Engineering, Software Engineering, Hardware Engineering, Systems Engineering, Functional Decomposition, Design Patterns, Integration, Modularity, Acquisition, Contractual Incentive, System of Systems, Cyber-Physical, Real-Time, Safety-Critical, Cyber-Secure, Maintainable, Automated Testing

Introduction

The intent of this paper is to stimulate out-of-the-box thinking about what we can do to improve the status quo of acquiring cyber-physical systems (CPS). We will explore how to incorporate VE principles as a means to stimulate new ways of developing and sustaining the embedded computing resources of a complex CPS. VE is a structured, multidisciplinary approach to improve contract performance (International Council on Systems Engineering [INCOSE], 2015). In short, a traditional VE approach would have the U.S. government and the Department of Defense (DoD) provide additional funds to a contractor as an investment in their product development process. The investment is recouped over the life of the contract in one of several ways (e.g., reduced manufacturing cost, higher quality, etc).

When we use the term *VE principles*, we mean

1. to use VE as a means to motivate changes to the workflow of organizations that develop computing platforms for CPS
2. to reward innovations that reduce the acquisition risk of embedded computing systems in DoD CPS
3. that we need to reward innovations based on a new way to count value that identifies and mitigates "showstopper" defects early in the lifecycle as a leading indicator of traditional VE values (e.g., cost, schedule)

The basic structure of this paper is as follows:

- We begin by characterizing the status quo for the acquisition of these types of embedded computing resources, which seems to need improvement.
- We then make the point that if we continue to do things the same way, we should expect the same (or similar) results.
- We introduce Architecture Centric Virtual Integration (ACVI), a proven means for identifying issues in embedded computing resources as a way to try a different approach.
- We emphasize the need to reward the "right kinds of behavior" to try to get the ACVI process integrated into the contractor's workflow—to reward honest effort as opposed to checking the box.
- We describe how important it is for the contractor to establish a measurement program focused on early identification (and resolution) of showstopper issues.
- We construct a notional means of financing the costs associated with the additional modeling effort (the value).
- We point out that we do not know enough about the contractor's operations to be prescriptive, but we have general guidance.
- We describe the need to create an environment that allows the contractors to fail without penalty (current status quo) yet rewards the right behavior.
- We conclude by providing some examples of how contracts could be written to provide the right kinds of incentives and some ideas for how contractors could integrate ACVI into their workflow.
- Finally, we make some recommendations for acquirers and contractors.

The Status Quo

In the development of software-intensive systems, more specifically CPS employing embedded computing resources, we are not aware of the construct of VE ever having been used. One limitation imposed by a VE approach is the inherent lack of accuracy in software cost estimation on which to make VE trades. A key element of VE is to recoup an investment, and it is hard for a contractor to commit to a reduction in cost if they can't predict the cost with confidence.

Other formal approaches, such as those found in the Society of American Value Engineers (SAVE) or the International Council of Systems Engineers (INCOSE), have a standard set of activities (SAVE International, n.d.) We will not attempt to map those process steps precisely to this study but to reflect on the strategy of capturing value in making product and process improvements as a mechanism for managing the lifecycle of a software-intensive system (Defense Innovation Board, 2019). These are VE principles, not the actual wording of the VE contract language.

Unlike hardware systems, software systems are produced as they are designed and are continuously updated and changed over time. Even products that do not require functional change have to be updated to manage vulnerabilities in the products that make up the overall software system or through managing technology obsolescence. As such, the context of managing the classic hardware lifecycle cost does not precisely map to software-intensive systems. What is worth considering are the major functional elements of VE (e.g., establishing a baseline, developing a plan for the process improvement, and monitoring the process to identify the opportunity for incentive as they apply to embedded computing systems).

Model Based Engineering (MBE) and digital analysis are well-developed practices for physical products. For example, decades-old technology exists for modeling and analyzing physical properties (thermal, mass, structural, dynamic, electrical) and electro-magnetic interference. These models and analyses have demonstrated their value many times over and, to some extent, these techniques are widely accessible and used by all. However, rigorous equivalent application of MBE for cyber-physical/embedded systems with integrated computing has not achieved widespread adoption. Abstracting and modeling inherently abstract designs like embedded computing systems can lead to over-application of functional or system decomposition. In order to advance the state of the practice for modeling and analysis of embedded computing systems, there must be more process rigor employed when developing and using models and analyses. Models of the system components that interact with each other can virtually verify integration issues and intra-/inter-product interoperability properties. To establish this practice will require both new requirements and new incentives to drive innovative, alternative behaviors.

One author was recently a judge for a middle/high school regional science fair for the category of engineering. Several students used a combination of physical modeling and additive manufacturing to fabricate components for their entries or to do virtual prototypes to model their designs before full implementation.

Let's think about the introduction of microwave cooking technology. When first introduced, homeowners did not fully know how to use the microwave nor what it would be best applied to do, but the potential was clear. To take the union of VE and MBE all the way through the development cycle, there is a need to develop methods and techniques that take advantage of this potential. To use an architecture model as a pivotal piece of design that is established and nurtured throughout the lifecycle of a product has its uses, but it is not going to apply to everything all at once, at least not at first (Clements et al., 2012). Professionals who have seen development fads come and go are reasonably skeptical at the beginning. Such was also the case with microwave cooking technology. When microwaves were first introduced in the 1970s, there were people that thought that placing a microwave radiation emitter in their kitchen would cause harm. Juxtaposed to those concerns were the staunch advocates that thought it could be used for everything. Fifty years later, nearly everyone has a microwave oven in the home; we know how to use it, and we are comfortable with it. Who could have predicted then that most of America today would end up using a pre-sealed pouch to make popcorn? Equally important to adopting a new approach is that we also learn when not to use a new tool. Many of us have subsequently found, for example, that a microwave is not optimal for reheating pizza.

Effective use of tools—such as MBE, architecture analysis, and digital twins—will reduce acquisition risk and are not minor adjustments to workflows but are fundamental changes to what we do and how we work.

Changing the Results

Over the past 20-plus years, few would argue that the track record of the U.S. government's acquisition of CPS has met expectations. For example, a recent Government Accountability Office (GAO) report on the F-35 stated,

> In 2020, the F-35 program resolved 33 of the deficiencies it had identified in developmental and operational testing but it continues to find more. Of the 872 open deficiencies, the program characterizes 11 as category 1 (critical in nature and could jeopardize safety, security, or another requirement), and 861 as category 2 (could impede or constrain successful mission accomplishment). This represents two more open category 1 deficiencies than we reported in May 2020. According to DOT&E officials, additional new discoveries are due to quality problems with the F-35 software, resulting in a high rate of deficiency discoveries during operational testing and by pilots in the field. According to program officials, seven of these open category 1 deficiencies will be resolved prior to the completion of operational testing. Four will not be addressed until the third quarter of 2021. (GAO, 2021, p. 16)

The report went on to say,

> In recent years, program officials did not identify nearly a quarter of all defects until they were already delivered to test aircraft. *Ideally, according to the program office, the contractor would identify defects in the software lab or before the software is fielded to the developmental test aircraft* [emphasis added]. However, a November 2020 analysis conducted by a third-party consulting firm on behalf of the program office found that between December 2017 and September 2020, 656 software defects (or 23 percent of all software defects) were identified after the software was delivered to the test aircraft.

As illustrated above, the heart of the problem of acquiring CPS are issues identified late in the program lifecycle (GAO, 2019). These issues arise when physical devices are provided for integration, and the embedded computing systems do not behave as expected (Chilenski & Kerstetter, 2015). In addition, as a result of the late changes needed to make the system work, efforts to attain certification (safety, airworthiness, security, etc.) can be delayed. These delays increase the overall program schedule, which most often is tied to program budget. The objective is to avoid being on the long list of acquisition programs that have significantly exceeded schedules and budgets or have even failed to deliver, such as the Future Combat System, Ground Combat Vehicle, Advanced Amphibious Assault Vehicle, and Comanche. Other services have had their impact felt in the dramatic reduction of the number of items produced, like the F-22 and DDG-1000 (Rodriquez, 2014). The impact of delays, overruns, and cancellations hurt the taxpayer with higher costs to maintain the national defense.

Figure 1 is a graphic that illustrates the issue that this paper seeks to address. The x- and y-axes represent the schedule and cost budgets of the program. The blue area represents the expected capability of the platform as it evolves. In CPS acquisitions, it is typically the case that we get to the end of the schedule having spent all the money, but the system does not work as expected. Additional funding and schedule are needed to complete the platform. One leading cause for this is that the program is funded assuming "normal" integration issues will be found. As the program progresses and gets to the point where the program should be finished, unexpected issues cause delays, and the acquirer must expend more funds to finish the program. The program may also choose to sacrifice planned capability or attempt to do a balance of both an increase cost and a reduction in delivered capability. The value we seek to capture (e.g., the reduction or possible elimination of unforeseen future costs) is illustrated with the bullseye. The extra funding is a direct hit to the DoD budget and impacts other programs, as this additional funding is unplanned. The extra schedule is a direct hit to the warfighter, as the delay impacts when the platform can actually be deployed.

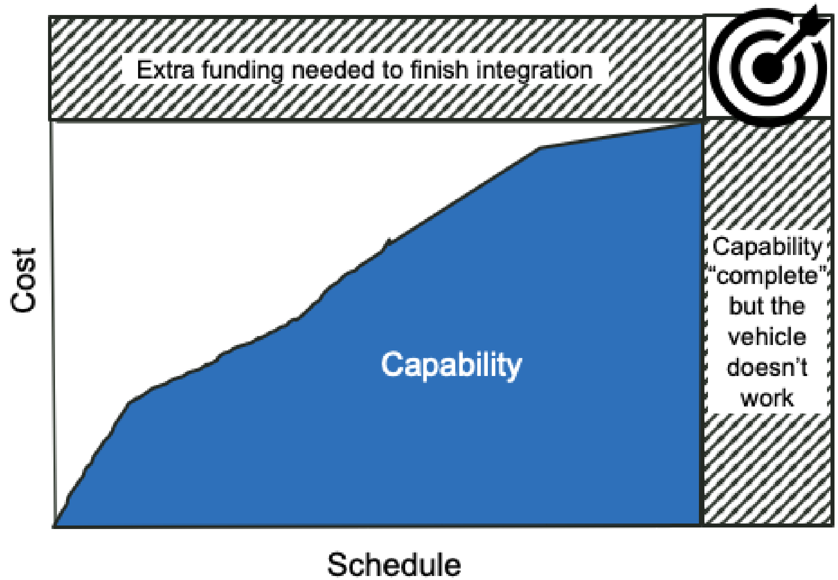


Figure 1. Identifying the Opportunity, the Value Bullseye

However, it does not have to be that way. The accepted colloquial definition for a risk is "an issue that has not yet happened." Integration issues are going to happen, so at this point they can be characterized as risks that are highly likely to occur, more specifically as embedded computing system integration risks. Software system integration issues are the largest single factor on recent new aircraft systems (Office of the Under Secretary of Defense for Research and Engineering, 2018). Also, 50% of the system development effort goes to software rework, and 80% of the software issues in system development are found in integration (Hansson et al., 2018). This means that status quo approaches do not effectively find these issues, leaving programs exposed to these risks if they do not address them overtly and early in the lifecycle. These embedded computing system integration risks apply to the acquisition of all CPS programs. The acquirer can choose how to mitigate these risks. One way that has potential is the Architecture Centric Virtual Integration Process (ACVIP).

ACVIP: A Different Approach

ACVIP focuses on analysis of embedded software computing systems, using architecture artifacts instead of a fully implemented design. ACVIP provides methods and tools to address product development and integration challenges where computing execution runtime sensitivity, safety, and cybersecurity are critical. A major source of system development schedule growth in these systems is the increasing complexity of interactions between the application software, the embedded computer system, and interfaces with physical systems. ACVIP provides a *virtual* integration environment that enables the detection of defects not typically found until much later. This is frequently accomplished by employing the Architecture Analysis and Design Language (AADL), continuous verification throughout the entire development lifecycle, and a single model that supports the analysis domains (e.g., safety, security, resources) that impact architecture and integration.

ACVIP begins with requirements verification and proceeds through actual system integration with the goal of identifying defects as early as possible. By incorporating virtual integration throughout the lifecycle, system and software engineers are able to prevent these defects from propagating (including side effects of change), and removal reduces future defect insertions. The early discovery of defects improves quality, reduces schedule (and cost) for the development and qualification of these systems, and ultimately enables fielding capabilities earlier. Even for defects that escape ACVIP analysis in the design phase, when physical integration issues arise, the design specification crafted in AADL speeds root cause analysis and predicts side effects of resolutions, enabling efficient resolution rather than multiple “test and fix” cycles.

The use of ACVIP has shown utility and benefit from previous U.S. DoD and non-DoD projects. It is certain that the additional effort to build the product models (to enable ACVIP) will be higher than to not make that investment. However, since as much as 80% of the defects are not found until physical integration when using traditional methods, early discovery and remediation will result in signficantly lower overall cost as a result of using ACVIP. Additionally, establishing and maintaining the virtual environment that is required to use ACVIP will reduce the cost of future block upgrades to the platforms.

Reducing risk likelihood is generally achieved through a set of engineering and management activities in a step-wise progression that does not reduce the risk consequence. By incorporating an ACVIP approach, the program also reduces the consequence of problems that can impact cost to complete or reduction in delivered functionality. Figure 2 depicts how virtual integration and related analyses could impact both the likelihood of integration risk and late-stage discovery. The governance and workflow integration of ACVIP also reduces the consequence when those risks are realized by finding them early and working through design and integration problems well before the design is finalized.

ACVIP works for capability changes injected into the program throughout the lifecycle as well. Once a product is matured through initial fielding or Minimum Viable Capability Release, ACVIP continues to positively impact the program—as long as the models and analyses are kept up to date with the fielded implementation. We show this transition from metaphysical certainty and dire impact (i.e., it happens to nearly every program) to moderate likelihood and manageable consequence. It would be irresponsible to assert low likelihood or consequence of these risks until these practices have been put into place and quantitative analysis on affected programs can be performed.

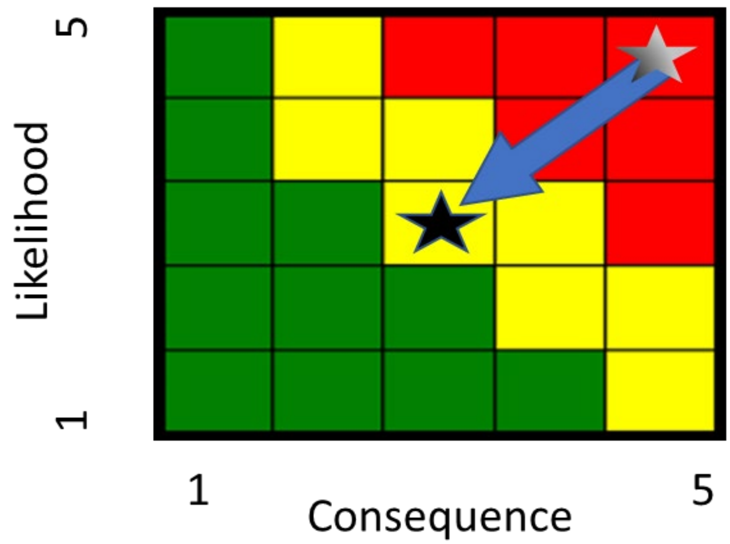


Figure 2. Value Engineering Goals for ACVIP

ACVIP augments and enhances model-based systems engineering (MBSE), along with the Systems Modeling Language (SysML), as a systems engineering practice. While SysML applies broadly to the entire system, ACVIP applies primarily to the runtime analysis of embedded computing systems. ACVIP uses model entities described in SysML together with model components expressed in AADL and augments them with the structure and additional properties that result in the specification of the software and hardware system. This formal representation of the architecture allows a precise and repeatable analysis capability that will verify a set of system quality attributes expressed as requirements, especially those about an implemented software deployed onto one or many computing platforms. The analysis capability spans a number of crosscutting system qualities that include safety, reliability, cybersecurity, resources, and performance in terms of signal latency and computing resource utilization, among others. The associated MBSE SysML models, along with other models that have been analyzed and updated because of performing ACVIP analyses, will have higher quality and will integrate with fewer issues or defects.

One of the ways that ACVIP would benefit a systems engineer is the representation and analysis of interface descriptions that trace down to the run-time level. The ACVIP analysis helps the systems engineer to understand the dependencies between systems and subsystems. Failing to identify interfaces, or failing to precisely specify their properties, is a common reason for products to fail and not to meet stakeholder expectations. Missing or incorrectly defined interfaces are a major cause of cost overruns and product failures (Chilenski & Kerstetter, 2015).

Similar issues caused the stability control system of the Lexus SUV to not slow the vehicle when it entered a turn at a high speed. In another example, the explosion of the Ariane 5 rocket's first test occurred because of a malfunction in the control software. This is an example of a failure to represent interfaces consistently between two pieces of software. The cause was a data conversion/representation issue among two software components. This resulted in the system detecting an out-of-bounds pressure on the rocket, causing the rocket to self-destruct (Inquiry Board, 1996).

These are examples of interface defect types that can be detected with ACVIP. Analysis of the architecture model identifies interface inconsistencies that include data type, data format, data update, data number, data range, and data units. These types of errors are detected continuously, beginning with the construction of the initial model, throughout the entire model as pieces become integrated—ensuring consistency at software and hardware integration time.

Where's the Value?

Figure 1 showed a bullseye, representing the time and money that we put on contract at the end of the project to finish the project. For the F-35 program, there were billions of dollars of cost and years in schedule borne by the government (GAO, 2021). Of course, we cannot ascribe all of the cost overruns and schedule delays to embedded computing system issues. There will also be late changes to the scope (or late changes to the design), to the components, and possibly a reallocation of functionality.

However, significant value to the acquisition stakeholder will come from the early identification and elimination of the showstoppers that arise late in lifecycle. Showstoppers are best characterized as issues that force a rewrite of the software, most likely due to a safety concern or a certification issue. Figure 3 summarizes issues from several safety critical systems.

> Despite best build-then-test practices, system-level faults due to software have increasingly dominated the rework effort for faults discovered during system integration and acceptance testing. Several studies of safety-critical systems show that 70% of errors in embedded safety-critical software are introduced in the requirements (35%) and architecture design phases (35%). At the same time, 80% of all errors are not discovered until system integration or later. The rework effort to correct a problem in later phases can be as high as 300-1,000 times the cost of in-phase correction. Therefore, it is desirable to discover such problems earlier in the lifecycle and increase the chance of tests passing the first time around. (Feiler et al., 2013)

Our claim is that the extra time and money (the bullseye) we spend at the end of the contract to "finish" the system should be planned for and managed as part of the acquisition risk. Instead of doing emergency appropriations for money in future years, to the detriment of other programs, program managers should avoid risk realization through virtual integration. "We believe that fully embracing an ACVIP approach is effective risk mitigation for the inevitable unknowns that surface late in all cyber-physical system development projects" (Boydston et al., 2019).

Find and Resolve Showstoppers Early

The acquisition community in general, and programs that build highly interoperable CPSs in particular, should acknowledge the near certainty of schedule and cost overruns and build in reward mechanisms that incentivize behavior that avoids them. If, for example, we can identify and address 90% of the showstoppers during system development, allowing us to achieve the baseline schedule with no increase in funding or compromise in capability, we should reward that behavior significantly. For example, if we were to avoid a $100 million cost overrun, we should be able to perform a risk-sharing initiative that would harvest some smaller portion of that (e.g., $25 million) as an incentive. Cheap at twice the price.

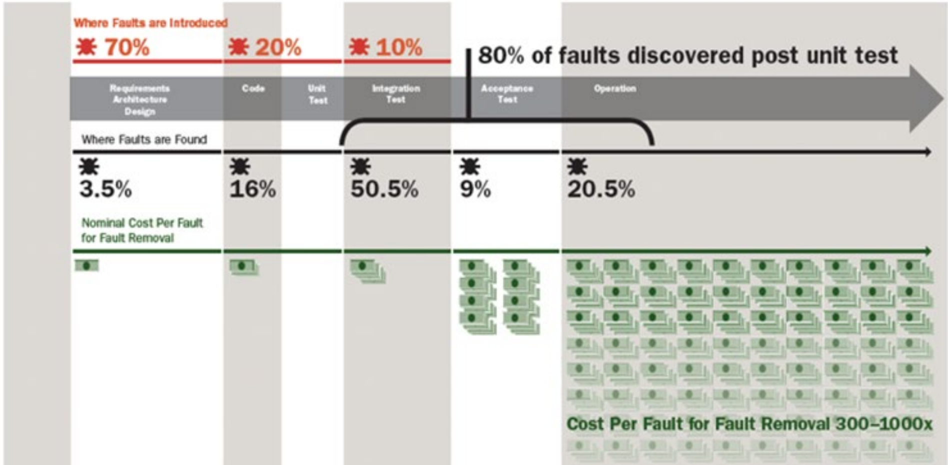


Figure 3. Cost Impact of Late-Stage Discovery (Feiler et al., 2013)

Figure 3 shows an analysis of the status quo of CPS development (Feiler et al., 2013). As much as 80% of issues are detected after unit test. It is hard to argue that this is the kind of result that we should be satisfied with when it comes to identifying and resolving problems early in lifecycle (when they are affordable to fix). As a result, we wind up paying many times more than what was estimated to resolve these issues. There is little direct financial incentive for the contractor to change associated behaviors because the government will make up the difference in total cost to get the project completed. A cynic would argue that the contractor is actually motivated to not change their development process. A comparison could be made to the acceptable quality level of an automobile manufactured in the United States up to the 1980s. Consumers would routinely accept substandard product quality and get their identified issues repaired after the initial purchase and hopefully during the warranty period. There was no incentive for the U.S. automotive manufacturer to update their process, and there was no public outcry for better cars. We just accepted the status quo. By contrast, Japanese companies during that time showed the world how to manufacture automobiles without quality issues and were rewarded with large changes in market share. This forced a change to everything in the U.S. automotive manufacturing industry. The expectation of the consumer had been changed. To get that kind of change in the environment for CPS, we need a different kind of forcing function through DoD policy changes to improve engineering processes and to insist on better outcomes through VE. A related change in the defense industrial base is needed to encourage risk-taking as well as rewarding attempts to improve.

So, in the context of VE for ACVIP, the U.S. government can incentivize the contractors to adopt ACVIP as part of their process (i.e., as a means to identify showstopper issues earlier in lifecycle). If the modeling and analysis is implemented in the right way, the contractor will avoid a significant number of showstoppers late in the lifecycle, which can be subject to risk-sharing incentives that benefit them and the government (Wrubel & Gross, 2015). This translates into reduction of acquisition risk. This will raise the bar for CPS developers.

Reward the "Right Kinds" of Behavior

We believe that contractors will not be motivated to change their process without incentive. We believe that there is plenty of opportunity to improve the status quo. This is what brings us back to the original concept of VE. We need to stimulate innovation within the contractor community by incentivizing changes to their workflow that take advantage of these novel methods. We want the incentive to reward the right behavior—that is, if the contractor is successful in avoiding showstoppers late in the lifecycle, there should be a significant reward.

We are not attempting to prescribe anything other than guiding principles, as we do not yet know enough to be prescriptive. We do know that we are talking about potentially hundreds of millions of dollars and a significant change to the development process. There are many ways that the rewards and incentives could be realized, which is a topic for future research.

Bridging the Systems Versus Software Divide

A long-standing hallmark in DoD system development is the mismatch between systems development activities and software development activities (Sheard et al., 2019). While we do not think that adopting ACVIP will break the cultural clash, we do think that it is a step in the right direction. MBSE is an established discipline. Potentially critical flaws in the system design can be found much earlier using a similar model-based approach for the design and implementation of the embedded computing system, allowing much more time for the evaluation of alternative approaches.

While the adoption of ACVIP by itself will not force an improved collaboration, the development of an architecture-centric virtual integration model will lead to the potential for a common ground for systems and software engineers to reach agreement. We refer to this as the "authoritative source of truth" (ASOT). Assumptions, constraints, and dependencies are all things that need to be represented and incorporated into the model, which will be fed by both the systems engineers and the software engineers. The ASOT will serve as a clearinghouse for the design issues. As the model evolves, with increasing fidelity over time, the model will serve as a fundamental resource for both groups, providing an environment that facilitates the rapid resolution of the showstoppers.

Contracting and Managing Incentives for ACVIP

One of the most effective communication tools that the U.S. government has with industry is its requests for information (RFI) and requests for proposals (RFP) and the resulting contracts. The structure of the requirements, evaluation criteria, and incentives send a strong message about what is important enough to be the difference between winning and losing. One challenge of managing a program is to balance the drive to perform as soon as possible with the hard decisions of increasing near-term costs for long-term gains. For this discussion, there will be a focus on cost-type contracts, which are best suited for development of cyber-physical products where there is an inherent lack of precision on which the outcomes can be predicted (Kendall, 2013).

The projected future cost of any contract is analyzed up front through mechanisms like the government's Independent Cost Estimate (ICE) that is developed prior to releasing an RFP. These projections are, in large measure, validated by the range of responses provided by industry to those solicitations. The cost structure of those bids goes through rigorous cost modeling and price justification internally before being presented as a part of the company's response to the RFP. These activities could set the benchmark from which incremental and final VE incentives can be set. The appropriation to fund VE incentives would need to be added to the program budget and included with the overall funding appropriation. Anecdotally, we have seen cost overruns that exceed 50% of the original ICE (GAO, 2020). It seems reasonable to set an initial threshold for the incentives at 25% of that. Notionally, if the idea works, the program would save 75% of their overrun.

There are a variety of contract award and incentive structures applicable to the activities needed to manage the technical and business objectives of building major weapon systems (DoD Open Systems Architecture and Data Rights Team, 2013). Development of new products is, by its nature, uncertain with respect to the input requirements and ultimate design synthesis. As such, the guidance is focused on cost-plus contracts and structures associated with awards that are aligned to the type of work being performed. The recommendations associated with integration contracts are especially valuable to consider.

While any new RFP released by the DoD is a communication tool, the evaluation criteria for the contract award is the part of the vehicle that speaks the loudest. To make ACVIP and VE effective in driving positive outcomes across the contract period, evaluation criteria must show that performing virtual integration will be rated very highly as a discriminator for award. In fact, typical evaluation factors include cost and performance. It is likely that ACVIP could be a significant contributor to both of these.

To ensure that industry remains engaged in using ACVIP, contract incentives should be in place for continuing to reward the right behavior: identifying showstoppers and reducing acquisition risk. Incentive Fee type cost contracts are suitable, but Award Fee types may provide the most utility. In such contract types, there is usually an award fee board that can adjust the criteria to match the state of the program and the types of performance improvements that are most sought after by program management. The best carrot should be at the end of the contract period, where the VE improvements captured during the contract period can be evaluated in the near hindsight of contract closure to the mutual benefit of both parties.

Well-established patterns of schedule slippage and cost overruns in software programs can also be mitigated through incentive structures. Metrics generated from performing ACVIP during early stages of the design process can be used to inform contractor fees, incentives, and terms. Programs can use these early design and synthesis stages to reduce program execution risk, and the contractor can benefit financially through the incentive structures—a win-win environment of transparency and collaboration.

There is a great degree of creativity in this space where customer and supplier can seek to deliver the greatest product for the money. However, the thrust of ACVIP is primarily focused on avoiding future unforeseen cost, not for reducing the profit available to industry. Juxtaposing total cost management with reasonable profit motives requires thoughtful attention in order to avoid unintended consequences of artificially inflated starting bids or short-sighted cost cutting measures, both of which ultimately hurt program execution and reduce taxpayer value.

Integrating ACVI Into Workflow

The introduction of a new technology generally provides opportunity for success and for failure. We learn from our mistakes, and many failures are often experienced before success can be claimed. In the context of the acquisition of CPS, ACVIP has been applied with mixed results. One significant lesson learned is that the ACVIP modeling and analysis has often been applied too late in the development process to have had a significant impact on the outcome. ACVIP is performed primarily with embedded computing systems because these systems require attention to CPS interaction, are timing constrained, have specific security and safety requirements, and require resource and performance analysis throughout their development. ACVIP specifically addresses these qualities. We believe the early surfacing of issues in these areas will translate into a more predictable and easier product development process.

Recommendations for Acquirers and Contractors

The context of the application of ACVIP strongly influences the planned workflow. Consider a contractor-oriented scenario where ACVIP analyzes process threads that are loaded on to a processor to optimize throughput and assess potential component deployment options. Compare this to a use case for ACVIP being applied by the U.S. government to support an acquisition. For example, a potential acquisition scenario would be to acquire a new and improved widget (e.g., a new global positioning system device) for integration into a legacy fleet.

In the acquisition scenario, ACVIP could be applied by the acquirer prior to the contract award to assess the likelihood that there will be problems during system integration and test. Contractors and acquirers will need to determine how to apply ACVIP to their existing development or acquisition processes, which are highly context dependent.

ACVIP practices are not widely adopted by the DoD's CPS contractors. There is an awareness of the benefits of ACVIP, and some contractors have more experience than others, but for the most part, the level of ACVIP expertise within the contractor community is low.

As contractors think about how to modify their workflow to take advantage of ACVIP, they will need to consider what (and how) they plan to measure the impact of ACVIP on their process. Both for their own purposes, and for their government customer, industry will need to measure the effect of a change to their workflow on their process baseline. It should be noted that actual, validated process performance baselines that would support this type of analysis are rare within the contractor community. Instrumentation and metrics are essential to provide the opportunity to analyze the anticipated workflow for the project and strategically insert the ACVIP practices as early as possible in the product lifecycle to try to achieve maximum effect.

A popular way to facilitate process instrumentation is to write down questions that one would want to be able to answer with the collected data. This is the key activity in the Goal Question Metric Approach (Basili et al., 1994), as it ties the measured data to the achievement of a business goal. As the questions are elaborated, it becomes clear to the developer whether their existing process instrumentation provides the data needed to answer the questions. This might generate an entire activity by the developer to establish a baseline for their process, the required first step before embarking on any process improvement. In the case of ACVIP, the key benefit is the ability to identify showstopper defects and issues earlier in lifecycle, and with an established baseline, some possible questions might be

- Were the defects identified by the new workflow not identifiable by previous methods (i.e., not normally detectable until physical devices were present)?
- Should the defects that escaped the new workflow (i.e., not detected until physical devices were present) have been detected by ACVIP? If so, why weren't they?

Figure 4 shows the key activities that we expect would benefit from ACVIP and where the data come from (or goes) for each activity. There is an intentional open aperture for delineation between organization (U.S. government, contractor, subcontractor) or by role (e.g., system engineer, software engineer, hardware engineer, test engineer) to open flexibility to alternative implementations and evolving practices. So there are a lot of options for how to actually integrate these practices or modify them to take advantage of the new process capability. There are nontrivial issues to consider, such as

- product line considerations
- proprietary data
- government furnished information (GFI) detail
- Modular Open System Approach (MOSA) requirements

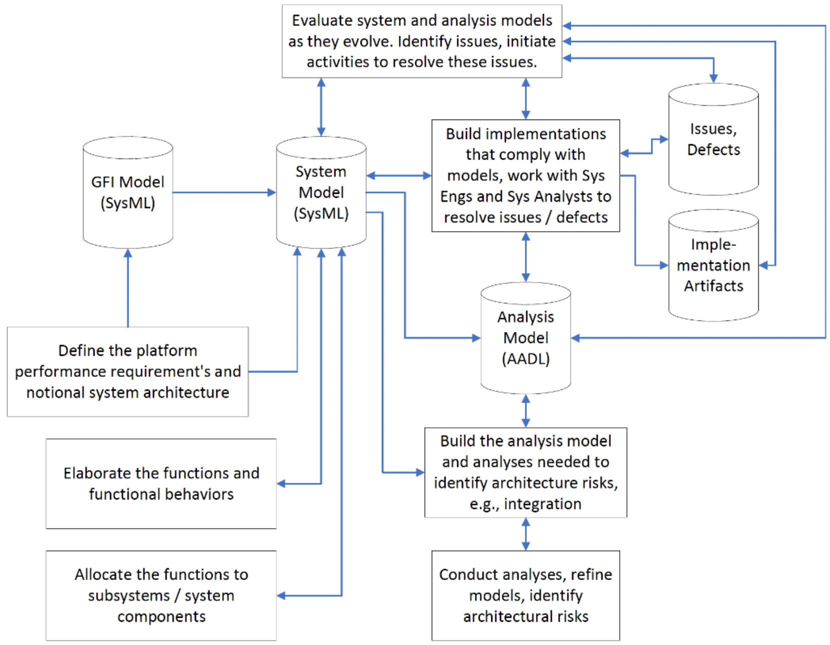


Figure 4. Workflow Activities and Repositories

Road to ACVIP Adoption

Adoption of any new approach must have the enabling environment set and be mature enough to be ready for full use. The enabling environment is in place:

- The Army and Carnegie Mellon University's Software Engineering Institute (CMU, SEI) have evolved the concept and practice of ACVIP.
- The SAE International® has nurtured the AADL, which is in use by a variety of international industries. Supporting tools have matured and include commercial products that facilitate the use of the Object Management Group's SysML standard; the Open Source AADL Tool Environment (OSATE) has been developed and released to the public by CMU, SEI and an array of international participants.

Adoption will inform the target community of the opportunities that are available to them along with the standards and associated tools.

- There are a set of training resources available to the community to aid in understanding the enabling environment elements such as AADL, SysML, VE methods, and associated tools.
- VE principles are robust and well-used practice in a variety of industries and government organizations but need to be tailored for ACVIP.

Adoption of these practices are mutually beneficial. As discussed in the contracting and incentive section above, this could be done by incorporating ACVIP as a contract award criterion. This would require the contractor to challenge their status quo and come up with creative and novel applications of ACVIP. Similar to the adoption of the microwave oven (as described above), technologies, tools, and techniques will evolve to shape ACVIP over time. The ACVIP content would appear in multiple places within the proposal document and be threaded through incremental and final contract evaluation activities.

We need to build some experiments that have hypotheses; producing data enables the evaluation of alternative implementations (with/without ACVIP) to verify/validate the hypotheses (i.e., demonstrate the value).

Bumps in the Road to Adoption

The synergy needed between software, hardware, and systems engineers will need to evolve in ways that have not been emphasized in the past. ACVIP has its greatest advantage when models are throughout product development *and* sustainment. Therefore, we will need to extend the discipline of creating and managing embedded computing systems engineering artifacts.

Putting these kinds of efforts into areas of the lifecycle, where they have not traditionally been, will require a change in culture. Changing culture is hard and takes time through shaping and supporting new behaviors. Incentives can be instrumental in helping to achieve those shifts in behavior. Integrating a mindset of seeking out value improvements as a part of daily work must replace environments where people feel like they are only supposed to follow the documented requirements. Success for this transformation will be built on trailblazers trying new approaches and showing that the high likelihood for a payoff for both customer and supplier transitions into added value that all sides can benefit from. The trailblazers need to be recognized for making good faith efforts, even if some do not pay off.

Transformations, by their nature, are initially unsettling to a status-quo environment. Committing to a new process can lead to underperforming before those approaches can wring out implementation challenges (often referred to as the learning curve). For any such initiative there are always start-up and scaling issues that need to be worked through (Dubner, 2020). Teams must be allowed to work on small improvements that can be scaled and grow such that they will eventually show skeptics positive examples that illustrate how to apply new technologies and process improvements focused on adding value.

Summary and Recommendations

After a time, repeated behavior patterns need to be accepted as a highly likely future event. Cost and schedule overruns for these highly complex and integrated cyber-physical systems are going to happen unless we change the way we acquire these products. The consequences of schedule slips and unfulfilled performance goals are manifold and widely known. This future cost realization can be turned into a value-benefit bogey such that mutual needs are met: improved business performance for industry and on-schedule acquisitions that fully deliver desired capability.

In CPS, “software is never done” (Defense Innovation Board, 2019) for three reasons: .1 capability changes are relatively affordable to introduce, 2. constant vigilance is needed to stay abreast of cybersecurity risks throughout the lifecycle of the product, and 3. software development is a constantly changing field where new methods and tools are also in motion. As such, process improvements will need to be managed for any CPS and should be accomplished in a way that is mutually beneficial to both supplier and customer. VE is a proven framework that can be extended into CPS to evaluate and introduce innovative new methods as a win-win partnership between industry and government.

Evaluating VE initiatives in an MBSE and ACVIP model-based environment will reduce uncertainty and risk for incorporating capability and process changes as well as clarify the VE benefits of the innovation. ACVI is a robust process that is supported by mature standards and tools on which to perform VE improvements. Embracing ACVIP also reduces design and integration risk for initial product development and for follow-on fact-of-life cybersecurity challenges and periodic capability improvement throughout the lifecycle (Chilenski & Kerstetter, 2015).

Integrating VE and ACVIP into the workflows of customers and suppliers engaged in developing CPS must be woven into the technical and business dimensions of the combined enterprise. This would include impacts to contract types, statements of work/objectives, development and operations processes, cost estimation, incentive mechanisms, and business processes.

Combining VE and ACVIP in this way is novel. As such, our first recommendation is to build a process performance baseline to be used to establish the rate at which showstopper defects escape the established process controls (e.g., peer reviews, unit tests, and hampering project execution). The data would be used as a reference to measure improvements, such as ACVIP implementations, that would be rewarded by VE. VE can only work when there is a solid reference point from which to measure. Even if you do not implement ACVIP, root cause analysis of the defects that escaped (and also analysis of the ones that were not detected) should be an essential part of all process improvement initiatives.

Our second recommendation is to pilot the application of ACVI and VE on selected acquisition programs to work out the details of the most effective metrics to be used, the most impactful analysis to be performed, and the value to be gained by all stakeholders. That pilot data will help develop measures and criteria for defining subsequent successful scaling of the initiative into other programs. Many innovations that have pleasing initial outcomes collapse under the weight of being fully deployed due to issues with scaling for wider use (Dubner, 2020).

Our third recommendation is to perform VE analysis at the enterprise level to determine the processes and products used for cyber-physical products operating in common domains and across related organizations to identify duplication and opportunities to gain greater efficiencies. Comparisons can be made to evaluate where points of leverage add overall portfolio value and determine systemic reuse strategies. VE can also be used to evaluate the positive or negative impacts of using/declining common products, modules, infrastructure, development, security, operations tools, and so on.

Taking on any process reengineering or transformational effort is fraught with risks and challenges, but they are worth doing where the benefits are high enough. We have identified areas to address and methods to employ that are replete with even greater benefits than the costs to bring them about.

Research Issues

- VE concepts can be applied to software development strategies.
- Products are developed across multiple organizations, but integrating across a wide array of actors makes it very difficult to get everyone on the same page.
- Value of virtual integration is harder to address due to different constituencies.
- Costing the scale for software teams is inconsistent and best-guess approach.
- Better systems engineering up front will ensure there is less expense in the long run.
- VE has yet to be connected to CI/CD for weapon systems.

Results

This analysis identified the aspects of VE that can be applied to the acquisition and lifecycle of CPS employing embedded computing resources. Programs will be able to identify the future cost of change and will have the ability to ensure that investments in modeling and analysis are preserved instead of traded off in the early stages of an acquisition when they do the most good.

References

Basili, V., Caldiera, G., & Rombach, H. D. (1994). *The goal question metric approach*. Semantic Scholar. https://www.semanticscholar.org/paper/The-Goal-Question-Metric-Approach-Basili-Caldiera/02e65151786574852007ecd007ee270c50470af0

Boydston, A., Feiler, P., Vestal, S., & Lewis, B. (2019, September). *Architecture centric virtual integration process (ACVIP): A key component of the DoD digital engineering strategy*. U.S. Army.

Chilenski, J. J., & Kerstetter, M. S. (2015). *SAVI AFE 61S1 report: Summary final report*. Aerospace Vehicle Systems Institute. https://savi.avsi.aero/wp-content/uploads/sites/2/2015/08/SAVI-AFE61S1-05-002_Summary_Final_Report.pdf

Clements, P., Kazman, R., & Klein, M. (2012). *Evaluating software architectures: Methods and case studies*. Addison-Wesley.

Defense Innovation Board. (2019, May 3). *Software is never done: Refactoring the acquisition code for competitive advantage*. https://media.defense.gov/2019/Apr/30/2002124828/-1/-1/0/SOFTWAREISNEVERDONE_REFACTORINGTHEACQUISITIONCODEFORCOMPETITIVEADVANTAGE_FINAL.SWAP.REPORT.PDF

De Graaf, R., Van der Linde, G., De Jong, H., & Vogt, B. (2019). Value engineering as a specialty for systems engineering: Exploring opportunities. *INSIGHT*, *22*(1), 41–44. https://www.doi.org/10.1002/inst.12237

DoD Open Systems Architecture and Data Rights Team. (2013, June). *Open systems architecture contract guidebook for program managers*. Under Secretary of Defense for Acquisition, Technology, and Logistics.

Dubner, S. J. (Host). (2020, March 19). Policymaking is not a science (yet) (No. 405) [Audio podcast episode]. In *Freakonomics.* Freakonomics Radio. https://freakonomics.com/podcast/scalability/

Feiler, P. H., Goodenough, J. B., Gurfinkel, A., Weinstock, C. B., & Wrage, L. (2013). *Four pillars for improving the quality of safety-critical software-reliant systems*. Carnegie Mellon University, Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=47791

GAO. (2019). *Weapon system sustainment: DoD needs to better capture and report software sustainment costs* (GAO-19-173). https://www.gao.gov/products/gao-19-173

GAO. (2020). *Defense acquisitions annual assessment: Drive to deliver capabilities faster increases importance of program knowledge and consistent data for oversight* (GAO-20-439). https://www.gao.gov/assets/gao-20-439.pdf

GAO. (2021). *F-35 joint strike fighter: DoD needs to update modernization schedule and improve data on software development* (GAO-21-226). https://www.gao.gov/products/gao-21-226

Hansson, J., Helton, S., & Feiler, P. H. (2018, April). *ROI analysis of the system architecture virtual integration initiative* (Report No. CMU/SEI-2018-TR-002). Carnegie Mellon University, Software Engineering Institute. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2018_005_001_517165.pdf

Inquiry Board. (1996). *ARIANE 5: Flight 501 failure.* http://sunnyday.mit.edu/nasa-class/Ariane5-report.html

International Council on Systems Engineering. (2015). *INCOSE systems engineering handbook*. Wiley.

Kendall, F. (2013, March/April). Use of fixed-price incentive firm (FIF) contracts in development and production. *Defense AT&L*. https://www.acqnotes.com/Attachments/Frank%20Kendall%20Use%20of%20Fixed-Price%20Incentive%20Firm%20Contracts%20Mar%202012.pdf

Office of the Under Secretary of Defense for Research and Engineering. (2018). *Design and acquisition of software for defense systems*. DoD Defense Science Board. https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

Rodriquez, S. (2014, December 2). *Top 10 failed defense programs of the RMA era*. War on the Rocks. https://warontherocks.com/2014/12/top-10-failed-defense-programs-of-the-rma-era/

SAVE International. (n.d.). *About Save International*. Retrieved April 2, 2021, from https://www.value-eng.org/page/About

Wrubel, E., & Gross, J. (2015). *Contracting for agile software development in the Department of Defense: An introduction* (Report No. CMU/SEI-2015-TN-006). Carnegie Mellon University, Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=442499

Integration of Production Management Into Software Development

Caitlin Kenney is a Systems Engineer and Agile Coach for International Systems Management Corp. She has an MS in Systems Engineering from the University of Maryland and a BS in Industrial Engineering from Northeastern University. Kenney is currently pursuing a PhD in Civil Engineering, Project Management at University of Maryland. She has over 10 years of experience supporting Department of Defense (DoD) projects, including two DoD Agile Pilot Programs and multiple agile project management adoptions. She is a licensed professional engineer in Illinois and Maryland. [Ckenney7@gmail.com]

Abstract

The application of the five production management levers addresses software projects as production systems, allowing for the application of well-established operations science techniques to improve throughput and predictability throughout the project life cycle. Traditional project management triple constraints of cost, schedule, and performance have long been the cornerstone of Department of Defense (DoD) software project management. Often, these constraints are managed independently, using past performance data to assess and predict future project performance, leading to highly variable outcomes. Traditional waterfall models for software project delivery exacerbate this variability and limit the ability to delivery capability at the speed of relevance. By treating projects as a production system, a deeper understanding of planning variables (product design, process design) and execution variables (capacity planning, limiting work in process, and variability) can be achieved and close the gap on project delivery performance. Further, the study connects industry best practices across software, manufacturing, and construction to improve DoD software project delivery. Operations science and production management principles and techniques are foundational to the agile and lean movement. The software industry has already adopted many of the operations science and production management techniques (such as limiting work in process, capacity-based sprint planning, software factories, etc.). A succinct process to apply, monitor, control, and report out on the five levers can encourage adoption and clarify execution of the Adaptive Acquisition Framework, DoD Instruction 5000.02, Software Acquisition Pathway.

Keywords: agile, production management, software

Introduction

The application of the five production management levers addresses software projects as production systems, allowing for the application of well-established operations science techniques to improve throughput and predictability throughout the project life cycle. Traditional project management triple constraints of cost, schedule, and performance have long been the cornerstone of Department of Defense (DoD) software project management. Often, these constraints are managed independently, using past performance data to assess and predict future project performance, leading to highly variable outcomes. Traditional waterfall models for software project delivery exacerbate this variability and limit the ability to delivery capability at the speed of relevance. By treating projects as a production system, a deeper understanding of planning variables (product design, process design) and execution variables (capacity planning, limiting work in process [WIP], and variability) can be achieved and close the gap on project delivery performance.

Further, the study connects industry best practices across software, manufacturing, and construction to improve DoD software project delivery. Operations science and production management principles and techniques are foundational to the agile and lean movement. The software industry has already adopted many of the operations science and production management techniques (such as limiting WIP, capacity-based sprint planning, software factories, etc.), while large capital construction projects are adopting production management to improve delivery on billion-dollar projects (Project Production Institute [PPI], n.d.). With a plethora of data both within the DoD and industry, a succinct process to apply, monitor, control, and report out on the five levers can encourage adoption and clarify execution of the Adaptive Acquisition Framework, DoD Instruction (DoDI) 5000.02, Software Acquisition Pathway.

Agile frameworks have been around for more than 20 years in software development. Providing a link between the operations science community in the manufacturing and construction industry and the software development communities is another step on the path of continuous improvement and delivery. It has been widely recognized that disconnected planning and execution affects project performance and acquisition throughout the DoD and beyond. This study helps the DoD take the lead within industry and revolutionize the way software projects are developed and delivered.

Research Issue Statement

This proposed study aimed to apply the five levers of production management—product design, process design, capacity, WIP, and variability (PPI, n.d.)—to the Adaptive Acquisition Framework, DoDI 5000.02, Software Acquisition Pathway to achieve increased control and monitoring within agile software development projects. Research questions included, Do current agile projects within the DoD (i.e., Agile Pilot Programs) currently employ any of the production management levers to manage and control software delivery? How can the production management five levers be applied to support agile software projects? How can the five levers be applied to the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD[A&S]) agile metrics?

Research Results Statement

While this research is the subject of the author's PhD studies at the University of Maryland, the study is ongoing. However, expected results are (1) processes to apply the five levers to improve cost, schedule, and performance monitoring and control, and (2) guidelines to apply production management levers to achieve OUSD(A&S) software metrics reporting requirements. This paper will share the progress made to date toward these objectives.

Technical Concept

The DoDI 5000.02, *Operation of the Adaptive Acquisition Framework*, published on January 23, 2020, led to incorporating Change 8 of the DoDI 5000.02T, *Operation of the Defense Acquisition System*," on September 15, 2020. The practices and standards within the DoDI 5000.02T are being updated incrementally to apply the adaptive framework, specifically for the software acquisition pathway. Major changes for the pathway were the elimination of the traditional waterfall Systems Engineering Technical Review (SETR) milestone decisions for an iterative, incremental delivery approach focused on minimum viable product (MVP) and minimum viable capability release (MVCR) deliveries (DAU, n.d.).

Within the updated DoDI 5000.02, there is a lack of detail related to how agile, lean, and DevSecOps approaches can be leveraged to improve monitoring and control of software project delivery, specifically during planning and execution. While a preliminary approach and concept for acquisition and project management documentation has been identified, a detailed approach to agile project management to support agile software development is needed. Agile software development is the application of the agile manifesto values and principles within development frameworks (i.e., Scrum, XP, etc.) and practices (i.e., pair programming, test driven development, etc.) to produce a software product. Agile project management is the application of the agile manifesto values and principles to project delivery (i.e., focusing on delivering value early and iteratively to customers; Agile Alliance, n.d.).

Application of Production Management

The proposed research develops processes to apply the production management five levers to improve cost, schedule, and performance monitoring and control during planning and execution, supporting an agile project management approach to software development. This effort will also establish guidelines to apply production management levers to achieve OUSD(A&S) software metrics reporting requirements by identifying how the five levers can be used to monitor and control software project planning and execution. The Adaptive Acquisition Framework software development and acquisition processes in the current DoDI 5000.02 are conceptual and do not fully address how agile, DevSecOps, and lean can be applied to a software development acquisition. Detailed processes to address how programs can tailor a software acquisition to adopt an agile framework, DevSecOps best practices, and lean techniques to achieve rapid development and deployment of software with cybersecurity built in are still needed. Specifically, the incorporation of combined planning and execution through the adoption of production management techniques, wherein planning occurs iteratively through the life cycle by the "doers" of the work; requirements are only defined as a need (what a user needs and why, not how); and schedule is timeboxed. Contrary to waterfall, in agile projects, time is fixed (timeboxed), cost and resources are fixed (capacity-based planning), and scope is flexible.

By framing the software project as a production system and applying agile frameworks and processes, project managers can achieve improved integrated planning and execution. The five levers to adopt agile project management include product design, process design, capacity, WIP, and variability (PPI, n.d.). Product design relates to what the product or project requires; in agile software development projects, these requirements are developed iteratively throughout the life cycle of the project. Process design is focused on the actual process and procedures to accomplish the work. Too often, the detailed processes required to accomplish individual tasks within a project are not well understood. Capacity-based planning is focused on planning work based on resources (personnel, materials, information, etc.) available versus planning to a hard date or milestone. This allows for more detailed work estimation and execution, reducing cost and schedule variability over time. WIP is the work that is in flight, meaning that work has started but is not yet finished. WIP represents tied up resources and capital; until WIP is released, it does not result in a direct outcome or value to the stakeholder. Finally, variability relates to anything that affects the overall outcome and performance of a project. Variability is largely driven and affected by the other levers previously mentioned. Controlling project variability, especially when it relates to cost and schedule, is a challenge on all projects, software or otherwise. Understanding the five production management levers forces project managers to look beyond the traditional "iron triangle" of cost, schedule, and performance. Instead, project managers need to understand the detailed level planning, design, and execution to control variability and improve overall project monitoring and controls.

Implications on OUSD(A&S) Software Metrics

The second expected result of this study is to provide guidelines for software project managers to apply production management levers to achieve OUSD(A&S) software metrics reporting requirements. OUSD(A&S) Software Policy Guidance identifies the following as a minimal set of metrics for software projects: process efficiency metrics, software quality metrics, software development progress, cost metrics, and capability delivery/value metrics (Brady & Rice, 2020). This study will provide guidelines on how the production management levers can be controlled to achieve the OUSD(A&S) metrics. For example, workforce capacity planning can be achieved using feature/story points. Variability can be controlled by tracking team velocity—adjusting story point commitments during a sprint as the team capacity flexes.

The proposed research is directly tied to the DoD National Defense Strategy 2018 goals and objectives to deliver performance at the speed of relevance, prioritizing the speed of delivery, continuous adaptation, and frequent modular upgrades to pace the threat (DoD, 2018). There have been multiple efforts within the DoD to support agile DevSecOps acquisitions, including the establishment of the Defense Innovation Board (DIB) and Office of the Secretary of Defense (OSD) guidance for the Agile Pilot Programs: *Software Acquisition Strategy: Agile Guidance* (OUSD[A&S], 2019e); *Contracting Considerations for Agile Solutions* (OUSD[A&S], 2019d); *Agile 101: An Agile Primer* (OUSD[A&S], 2019c); *Agile Software Acquisition Guidebook* (OUSD[A&S], 2020); *Minimum Viable Product (MVP) and Product Roadmap* (OUSD[A&S], 2019a); *Agile Metrics Guide* (OUSD[A&S], 2019b); and *DevSecOps Best Practices Guide* (Brady & Rice, 2020; DoD, 2020). The rewrite of the DoDI 5000.02 to reflect all of the lessons learned and to reflect an adaptive acquisition framework to support software projects is needed to ensure programs can adopt and execute agile principles and practices within the DoD acquisition environment. The application of production management levers through process and guidelines will further strengthen the agile initiatives within the DoD and transform not only software development but agile project management. The inclusion of production management levers and processes is aligned with the agile and lean movements seen within industry and provides the foundation needed within the acquisition community to understand and adopt agile principles and values for software delivery.

Research Questions and Objectives

Table 1. Research Questions and Objectives

Research Question	**Objectives**
Do current agile projects within the DoD (i.e., Agile Pilot Programs) currently employ any of the production management levers to manage and control software delivery?	Identify production management levers in use by Agile Pilot Programs and determine level of maturity
	Compare impacts to cost, schedule, and performance by assessing the throughput (delivery cycle time) of software deliveries
How can the production management five levers be applied to support agile software projects?	Determine how product and process design works in an iterative, agile approach
	Determine how fixed capacity planning results in increase control and throughput
	Identify how limiting WIP increases throughput
	Identify how variability can be controlled, not just monitored, by treating software projects as production systems
	Create a guideline for understanding software projects as a production system

How can the five levers be applied to the OUSD(A&S) agile metrics?	Identify how OUSD(A&S) metrics for agile software projects can be met using the production management levers
	Map the production management levers to the OUSD(A&S) metrics

Research Methodology

The initial phase of this research is focused on conducting a literature review of the Agile Pilot Program publicly available data. Since the introduction of the Agile Pilot Programs in the Fiscal Year (FY) 2018 National Defense Authorization Act (NDAA), multiple DoD projects have adopted agile principles and values with varying levels of maturity and success. Phase 1 will focus on a detailed literature review of the current state of agile adoption within the pilot programs, as well as interviews with the Agile Pilot Program offices to collect information related to application of production management levers.

There are multiple resources to review related to the Adaptive Acquisition Framework, the application of agile to DoD software projects, and the current state of agile adoption within the DoD. Additionally, there are potential data points from the Department of Homeland Security (DHS) that could provide additional lessons learned. The objective is to identify how the DoD has started adopting an adaptive framework for agile software projects; lessons learned regarding adoption; how production management levers are being used today; and their impact on OUSD(A&S) software metrics/success criteria.

The following is a list of materials that will be the basis for the literature review. Additional resources will be reviewed as identified during Phase 1 of the proposed research.

- DoDI 5000.02: *Operation of Adaptive Acquisition Framework*
- *Software Acquisition Strategy: Agile Guidance* (OUSD[A&S], 2019e)
- *Contracting Considerations for Agile Solutions* (OUSD[A&S], 2019d)
- *Agile 101: An Agile Primer* (OUSD[A&S], 2019c)
- *Agile Software Acquisitions Guidebook* (OUSD[A&S], 2020)
- *Minimum Viable Product (MVP) and Product Roadmap* (OUSD[A&S], 2019a)
- *OSD DevSecOps Best Practices Guide* (DoD, 2020)
- *AiDA: Acquisitions in the Digital Age* (MITRE, n.d.)
- Project Production Institute research (Arbulu et al., 2016; Shenoy, 2017; Shenoy & Zabelle, 2016)
- GAO reports on agile within the DoD and the DHS (GAO, 2020a, 2020b)
- Defense Innovation Board SWAP report (Defense Innovation Board, 2019)

Research Results

This research is the focus of the author's PhD studies at the University of Maryland, the results of which are ongoing. The goal of this paper is to share the preliminary results of the literature review using publicly available data and well-established production management and operations science techniques.

Importance of Flow

Production management is the application of operations management and science to production systems (PPI, n.d.). Traditional project delivery is focused on productivity measures. However, productivity does not equate to throughput or outcomes. Projects can have very high productivity measures and still fail to deliver (Shenoy & Zabelle, 2016). This is the result of poor flow through a system, failure to address and exploit bottlenecks, and failure to control WIP.

To improve project cost, schedule, and performance, a deeper understanding of what happens throughout the life cycle of a project is needed. The current approaches to project management largely adopted by the DoD and supported by organizations such as the Project Management Institute (PMI) have broken project management into siloed areas of responsibility, often creating large communication gaps that lead to project failures (Zabelle et al., 2018). The approach used today, even in some projects that have adopted agile, still often reflects the results of Conway's Law, which states that organizations' design systems mirror their communication structures. In other words, if the organization is not cross-functional, then the system or products will not be cross-functional, likely resulting in more changes late in the development process (Skelton & Pais, 2019).

To understand how this is all related to project performance, it is important to understand what is happening when projects are not delivering within cost and schedule. Little's Law explains how WIP impacts overall throughput and project performance. Limiting WIP (not eliminating) can provide more consistent results for throughput over time, as variability in cycle time (humans in the loop) can impact the flow (Choo, 2016).

$$\mathrm{TH} = \frac{\mathrm{WIP}}{CT}$$

Where,

TH = throughput (items/unit time)

WIP = work in process (# of items)

CT = cycle time (time units/item)

Production Management Levers

Production management is the application of operations science techniques to the management of the project or production process as a system. The idea is to look at project execution as a whole system, not as individual phases (Shenoy, 2017). It is a well-established concept that has been in practice within the manufacturing industry for decades (PPI, n.d.). It is also foundational to agile project management, which focuses on cross-functional, collaborative approaches to achieve a desired outcome or product. Too often projects are focused on productivity; however, productivity is not indicative of throughput or outcomes. An example of this would be classic project management monitoring and control tools used by the DoD today, such as Earned Value Management (EVM). Like any monitoring and control system, it is based on data, including estimations of work, cost, and value. The trouble is that these estimations are often made years in advance without the input of the people who will actually perform the work. The further the estimation is made, in both time and resources, the greater the variability in the measurement will be. At a certain point the data become nothing more than a point for contractors and government to argue over performance, providing little actual value or control.

One possible solution to these challenges is to leverage the five levers of production management throughout the life cycle of a project and to adopt a more agile framework to project management and execution. The five levers of production management include product design, process design, capacity, WIP, and variability (PPI, n.d.).

Product Design

Product design focuses on controlling the scope of a project through product requirements (PPI, n.d.). Specifically, product design focuses on developing the busines case, including the potential use cases and value it will deliver to the end user. Traditional project requirements are well defined and detailed up front as part of the planning phase. However, in agile software projects, the requirements start as a business case and user needs and are detailed and defined through iterative and incremental development cycles, improving over time based on user feedback (Agile Alliance, n.d.). Establishing processes to apply agile to the product design will lead to improved innovation and earlier user feedback, reducing the cycle time for software product delivery. Further, a set of guidelines on how to identify and define MVP from a DoD perspective would assist with software project planning.

Use Within the DoD Today

A full study of the applications in use within the DoD today is ongoing and currently limited to the results of publicly available data. However, based on the preliminary literature reviews, the following was found to support and demonstrate the use of the product design lever, specific iterative development of software requirements within DoD agile projects.

Many existing projects that have adopted agile practices still follow waterfall development practices when it comes to requirements, spending time and resources up front to develop and define detailed requirements a year in advance of when they will be developed. This is evidenced by the *Defense Acquisitions Annual Assessment*, published in June 2020, which stated that while many programs stated they were following agile practices, it was often more like a hybrid waterfall-agile approach (GAO, 2020b). In addition, while the statutory regulations have been modified and relaxed for certain types of programs, not all have made the transition and are still bound by contractual limitations, such as stage gate reviews, forcing less agile cadences and delayed releases. However, not all programs are experiencing these limitations to their agile approach, with many taking advantage of policy changes that allow for early prototyping (GAO, 2020b).

Application to OUSD(A&S) Agile Metrics

The OUSD(A&S) metrics include process efficiency metrics, software quality metrics, software development progress, cost metrics, and capability delivery/value metrics (Brady & Rice, 2020). Product design supports software quality metrics and agile product metrics. It also influences cost metrics and capability delivery/value metrics.

Product design supports software quality metrics by incorporating the user and stakeholders early and often throughout the product design and development activities. Recidivism, first-time pass rate, defect count, test coverage, and number of blockers all are informed or result from product design decisions. Capturing the rate of rejection or return on developed work provides project managers with the ability to monitor product design performance. Repeat issues, such as poor test coverage, can then be swarmed by the team to identify the root cause and make a change to the design approach to fix the gap (OUSD[A&S], 2019).

Process Design

Process design is key to understanding communication channels, flows, queues, and sequencing of work. There are many well-established agile frameworks that support this concept as a way to control and monitor throughput. Agile process design and practice adoption is a stepping stone to achieving DevSecOps continuous integration and continuous delivery of software. It is critical to establish processes and guidelines from a software acquisition adaptive framework standpoint. In the future, this study aims to provide a framework for process mapping within software projects to understand how technology, automation, and people can deliver software capability incrementally and iteratively.

Use Within the DoD Today

A full study of the applications in use within the DoD today is ongoing and currently limited to the results of publicly available data. However, based on the preliminary literature reviews, the following was found to support and demonstrate the use of the process design lever within DoD agile projects.

A study published in the DAU's *Defense Acquisition Research Journal* looked at five successful agile adoptions with the DoD. A common theme across the projects was the investment in agile coaches and process development. Understanding the flow of information and identifying bottlenecks and non–value-added activities within existing processes was critical to the success of the project's agile transformation (Kramer & Wagner, 2019). Further, according to the *14th Annual State of Agile Report*, which surveyed over 40,000 agile projects across industry and government, over half of respondents are implementing some form of value stream management as projects are adopting agile beyond software development, applying agile to their core business operations and project management as well (Digital.ai Software Inc., 2020).

Application to OUSD(A&S) Agile Metrics

OUSD(A&S) agile metrics identify a number of agile process metrics that aim to support the estimation, measurement, monitoring, and control of tasking at the lowest level possible. Metrics include story point estimation to support velocity calculation within small agile teams. Story points measure the estimate time and complexity of a task, taking into account dependencies, risks and unknowns, and skills required of a resource. The monitoring and control of theses metrics allows for greater predictability, as teams are responsible for estimating their own work, are accountable for documenting their progress, and are continuously reviewing and iterating to improve overall processes to support delivery (OUSD[A&S], 2019).

Capacity

Understanding capacity of not only the technologies employed but also the people assigned to do work is critical to creating an integrated approach to software development. Technology capacities related to software development and DevSecOps tools are well established and have been documented in the *DoD Enterprise DevSecOps Reference Design Version 1.0*, published August 12, 2019. However, workforce planning capacities are less understood within the current workforce. Years of waterfall development have created long workforce cycle times. Resources are often split between projects and efforts; throughput is stalled as people switch from task to task; the cycle time and takt time to deliver capability is measured in years, not hours. This study will focus on how software project managers can use agile thinking and processes to increase throughput and deliver quality software more frequently and reliably. Workforce planning, when done at the worker level, results in increased throughput and less variability of product outcomes, making it easier for project managers to track performance, cost, and schedule (actual vs. estimated).

Use Within the DoD Today

A full study of the applications in use within the DoD today is ongoing and currently limited to the results of publicly available data. However, based on the preliminary literature reviews, the following was found to support and demonstrate the use of the capacity lever within DoD agile projects.

The Reserve Component Automation System (RCAS) Army project tracks the capacity of their teams across the enterprise through story point analysis. Further, they have implemented a quarterly review to track overall performance using story points to estimate their team velocity. Velocity is then used to adjust their planning for the next increment (Kramer & Wagner, 2019). Capacity planning is a well-established method to support project monitoring and controls within agile projects. However, it is a major shift from the waterfall approach of surging resources to achieve scheduled dates. Further, while there are many policy guidelines across the government that have been modified in the last 5 years to recommend or encourage capacity-based planning, it is unclear how many programs are actually utilizing it today.

Application to OUSD(A&S) Agile Metrics

Capacity based planning most directly relates to velocity predictability and the cost metrics from the OUSD(A&S) metrics (OUSD[A&S], 2019). Planning work within the limitations of capacity allows for greater predictability when it comes to project cost and schedule. Individual team velocity normalizes over time, reducing variability and uncertainty and providing greater predictability of project performance. This supports the shift from traditional scheduling, which is often done separately by those independent of execution. With increased predictability, cost impacts and estimates become better over time.

Work in Process

WIP is a major driver of variability within any project, not just software. WIP ties up resources, slows down progress, and decreases throughput. It creates unnecessary dependences within a production system that can hinder delivery. However, WIP isn’t all bad. Understanding how WIP can be used to drive results is key to software delivery (PPI, n.d.). Using agile and lean tools, such as Kanban boards, sprint planning, and so on, software project managers can improve project predictability for delivering a capability.

Use Within the DoD Today

A full study of the applications in use within the DoD today is ongoing and currently limited to the results of publicly available data. However, based on the preliminary literature reviews, the following was found to support and demonstrate the use of the work in process lever within DoD agile projects.

While there are few data available on how programs are managing or governing WIP within their projects, there are some data available on the use of Kanban boards to control flow. The RCAS Army project used Kanbans for their program management office–related work (Kramer & Wagner, 2019). Kanban boards alone do not make a project agile. However, they are often used when there are continuous flow or continuous delivery releases (Rehkopf, 2021). In DoD projects, Kanbans can provide a way for programmatic and support activities outside of traditional developer teams to support an agile culture and approach for delivery.

Application to OUSD(A&S) Agile Metrics

There is overlap with many of the recommended OUSD(A&S) agile metrics, as WIP is a fundamental concept for supporting flow throughout the project. The guide specifically calls out WIP, cycle time, and throughput as flow metrics related to Kanban. However, WIP is a shared production management lever and agile concept that can be used in scrum and other agile applications as well. Controlling WIP will have a direct impact on the DevSecOps metrics related to deployment frequency and lead time. Further, WIP is also directly related to the Agile Product Metrics of delivered features/capabilities, as increased WIP will delay overall throughput and delivery of the desired features/capabilities. This is especially true if developer teams are constantly being re-tasked throughout the sprint due to emergent stakeholder needs. Careful planning and adherence to agile cultural changes is needed to ensure teams can limit WIP (OUSD[A&S], 2019).

Variability

Variability comes in many forms for software projects. The largest source of variability on any project is humans. No matter how standard or common processes, procedures, requirements, and so on are made, there is always variability during execution. The goal of using variability as a lever is to understand, monitor, and control variability at the source.

Traditionally, variability was monitored (not controlled) using tools such as EVM. However, there are limitations to this commonly used tool when it comes to software projects. EVM assumes that a project has a detailed list of established requirements up front, a detailed integrated master schedule (IMS) for the life cycle of the project, and detailed cost estimates. The issue with these assumptions and trying to apply tools such as EVM to software projects is that interactive and incremental development does not start with all this information up front. So, unless the project manager re-baselines every increment, there will be large variability in the EVM metrics. Further, the planning and execution resources are often independent of one another, making the cost estimates, schedules, and work breakdown structure disconnected from the people doing the work. If project managers want to truly control projects (not just monitor), then the estimates for the schedule and resources need to come from the lowest delegated level possible ("the doers").

Similar to how the DoD has adopted risk, issues, and opportunity management as a way to account for the good and bad of variability, production management treats variability as a way to address detrimental and beneficial variability within projects (Morrow, 2017). Sources of project variability include process variation/quality, time variations, and product variation. A common example of detrimental variation within a software project is rework of a product due to quality and testing issues; a common example of beneficial variability is completing work early or being able to pull work forward from the backlog (PPI, n.d.).

Use Within the DoD Today

A full study of the applications in use within the DoD today is ongoing and currently limited to the results of publicly available data. However, based on the preliminary literature reviews, the following was found to support and demonstrate the use of the variability lever within DoD agile projects.

There are multiple projects that are leveraging scrum and Kanban approaches. However, it is unknown how many are leveraging agile metrics, such as recidivism to track rework and impacts on quality. Additional surveys and data will need to be collected to understand how mature the agile applications within the DoD actually are, as many projects that reference agile as their approach also reported following traditional waterfall project management monitoring and control techniques, which is not aligned (GAO, 2020; Kramer & Wagner, 2019).

Application to OUSD(A&S) Agile Metrics

The OUSD(A&S) quality metrics are most closely related to the variability production lever. Specifically, recidivism (i.e., work returned to a team for rework), first-time pass rate, defect count, and test coverage are all examples of metrics that track detrimental variability. Beneficial variability will be captured via story completion rate, cumulative flow diagrams, and release burnup charts within the agile process metrics (OUSD[A&S], 2019). There are opportunities to add additional beneficial variability metrics. However, these will be heavily dependent on the agile approach and individual project applications.

Next Steps and Future Work

This paper represents the initial research and results to apply production management to agile projects within the DoD, as well as their adoption to support the application of the OUSD(A&S) software project metrics. Further research includes data collection related to the specific applications of agile within the DoD and related government projects. While guidelines and policy are being updated, there is little reporting available to demonstrate how programs are being held accountable for implementing agile. As evidenced by the latest Defense Acquisitions Annual Assessment, there are many programs (22) that claim to be practicing agile but are not delivering anywhere near the industry standard for iterative and incremental releases (GAO, 2020b).

Agile Pilot Program Interviews

A review of the DoD Agile Pilot Programs, their successes and lessons learned with agile frameworks, will be necessary to understand the level of maturity and success adopting agile. While there are some data available on programs that have adopted an agile approach or tools, there are few data published on the agile metrics being captured. Measures of agile performance and health are needed to understand where agile adoptions are failing within government projects. One-on-one interviews with project management, government product owners, scrum masters, and development teams are needed to understand the current application of production management levers in DoD project, including agile processes and metrics.

To support this objective, this study will create a standard set of questions to be shared with the Agile Pilot Programs to identify how production management levers and agile principles and values have impacted project cost, schedule, and performance. Challenges related to the delivery cycle time and throughput of software to the end user will be captured and used to identify processes for product and process design. Further, information related to agile software metrics for process efficiency, software quality, development progress, cost, and value will be assessed to identify enablers and challenges to agile adoption in the adaptive framework.

Process Development to Apply Production Management to Agile Software Projects

After Phase 1 has been completed and all data have been collected, a process to integrate production management levers for software project monitoring and control will be identified. The process will address how to apply product and process design, how to control WIP to increase throughput, and how to manage process, product, and resource variability to achieve throughput objectives for software incremental delivery. Additional guidelines to apply production management levers to deliver on the OUSD(A&S) software project metrics will be created, establishing a clear framework to achieve the full benefits of agile software delivery. There is no one-size-fits-all approach for metrics. However, it is important that they are well understood and applied properly within the acquisition community, especially within contracts.

References

Agile Alliance. (n.d.). *Agile 101.* Retrieved September 2020, from https://www.agilealliance.org/agile101/

Arbulu, M. R., Choo, P. H., & Williams, P. P. (2016). Contrasting project production control with project controls. *PPI Journal*, *1*. https://projectproduction.org/wp-content/uploads/2017/11/Contrasting-Project-Production-Control-With-Project-Controls.pdf

Brady, S., & Rice, C. (2020, March 17). *Software acquisition pathway.* DAU. https://www.dau.edu/Lists/Events/Attachments/193/SW%20Acq%20Policy%20Training_3.17.20.pdf

Choo, H. J. J. (2016, December 18). Little's Law: A practical approach to understanding production system performance. *Project Production Institute Journal*, *1*. https://projectproduction.org/journal/littles-law-a-practical-approach-to-understanding-production-system-performance/

DAU. (n.d.). *Execution phase*. Retrieved September 2020, from https://aaf.dau.edu/aaf/software/execution/

Defense Innovation Board. (2019, May). *Software acquisition and practices (SWAP) study*. Department of Defense. https://innovation.defense.gov/software/
Digital.ai Software Inc. (2020). *14th annual State of Agile report.* State of Agile. https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494
DoD. (2018). *Summary of the 2018 national defense strategy of the United States of America*. https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf
DoD. (2020, January 15). *OSD DevSecOps best practice guide.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/DevSecOps_Whitepaper_v1.0.pdf
GAO. (2020a, June). *Agile software development* (GAO-20-213). https://www.gao.gov/assets/710/707285.pdf
GAO. (2020b, June). *Defense acquisitions annual assessment: Drive to deliver capabilities faster increases importance of program knowledge and consistent data for oversight* (GAO-20-439). https://www.gao.gov/assets/710/707359.pdf
Kramer, J. D., & Wagner, L. J. (2019). Developmental test and requirements: Best practices of successful information systems using agile methods. *Defense Acquisition Research Journal*, *86*. https://www.dau.edu/library/arj/ARJ/ARJ88/ARJ-88%20_18-804%20Kramer.pdf
MITRE. (n.d.). *AiDA: Acqusition in the digital age*. Retrieved September 2020, from https://aida.mitre.org/
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019a, August). *Minimum viable product (MVP) and product roadmap: Version 1.0.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/ProductRoadmapAndMVP_WhitePaper%20V1.0%20FINAL.pdf
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019b, September 23). *Agile metrics guide: Strategy considerations and sample metrics for agile development solutions.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20Metrics%20v1.1%2020191122.pdf
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019c, November 18). *Agile 101: An agile primer.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20101%20v1.0.pdf
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019d, November 18). *Contracting considerations for agile solutions: Key agile concepts and sample work statement language.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Contracting%20Considerations%20for%20Agile%20Solutions%20v1.0.pdf
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019e, December 15). *Software acquisition strategy: Agile guidance.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Software%20Acquisition%20Strategy%20Agile%20Guidance.pdf
Office of the Under Secretary of Defense for Acquisition and Sustainment. (2020, February 27). *Agile software acquisition guidebook: Best practices and lessons learned from the FY18 NDAA Section 873/874 Agile Pilot Program.* DAU. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf
Project Production Institute. (n.d.). *Project Production Institute*. Retrieved September 2020, from https://projectproduction.org/
Rehkopf, M. (2021, April). *Kanban vs. scrum: Which agile are you?* Atlassian Agile Coach. https://www.atlassian.com/agile/kanban/kanban-vs-scrum
Shenoy, P. R. (2017). Defining "production system" from an operations science and project production management perspective. *PPI Journal*, *2*. https://projectproduction.org/wp-content/uploads/2017/09/PPI-JOURNAL-PPIPP-Defining-Production-System-from-an-Operations-Science.pdf
Shenoy, P. R., & Zabelle, T. R. (2016). New era of project delivery: Project as a production system. *PPI Journal*, *1*. https://projectproduction.org/wp-content/uploads/2016/11/New-Era-of-Project-Delivery-%E2%80%93-Project-as-Production-System-1.pdf
Skelton, M., & Pais, M. (2019). *Team topologies: Organizing business and technology teams for fast flow.* IT Revolution.
Zabelle, T. R., Choo, H. J., Spearman, M. L., & Pound, E. S. (2018). A "gap" in current project management and the impact on project outcomes. *PPI Journal*, *3*. https://projectproduction.org/journal/a-gap-in-current-project-management/

PANEL 22. DELIVER AT THE SPEED OF RELEVANCE

Thursday, May 13, 2021	
12:15 p.m. – 1:30 p.m.	**Chair: Pete Modigliani,** Senior Defense Capability Accelerator, MITRE Corp ***Buying for the Right Battle: Determining Defense Acquisition Strategies*** John Kamp, The George Washington University Amirhossein Etemadi, The George Washington University ***It's About Time: Toward Realistic Acquisition Schedule Estimates*** Raymond Franck, United States Air Force Academy ***Agile Improvements to Critical Path Method (CPM)*** Caitlin Kenney, University of Maryland

Pete Modigliani—is the Senior Defense Capability Accelerator within the MITRE Corporation enabling the DoD and Intelligence community to deliver innovative solutions with greater speed and agility. He works with acquisition and CIO executives, program managers, the Section 809 Panel,

Congressional staffs, and external groups to shape acquisition reforms, strategic initiatives and major program strategies. Pete champions digitally transforming the acquisition enterprise to modernize and accelerate operations. He launched MITRE's digital acquisition platform AiDA (aida.mitre.org).

Prior to MITRE, Pete was an Air Force program manager for C4ISR programs and an Assistant Vice President with Alion Science supporting the Air Force Acquisition Executive's Information Dominance division.

Pete holds a Bachelors of Science in Industrial Engineering from Rochester Institute of Technology, and a Masters of Business Administration in Information Technology from Boston College.

Buying for the Right Battle: Determining Defense Acquisition Strategies

Amir Etemadi—received a Doctor of Philosophy degree in electrical engineering from the University of Toronto, Toronto, Canada; a Master of Science degree in electrical engineering from Sharif University of Technology, Tehran, Iran; and a Bachelor of Science degree in electrical engineering from the University of Tehran, Tehran, Iran. He joined the faculty at the George Washington University in 2013, where he is currently an Assistant Professor in the School of Engineering and Applied Science in the Department of Electrical and Computer Engineering and the Department of Engineering Management and Systems Engineering. He teaches master- and doctorate-level courses in electrical engineering and engineering management.

Etemadi's research interests include distributed generation, power system dynamics and control, quantitative risk analysis, and engineering management. He is the Principal Investigator on several current research projects, including a National Science Foundation project on geomagnetic disturbance impacts on power system operation and an Acquisition Research Program grant on defense acquisition strategies. In addition, he is also a Faculty Graduate Research Advisor in the School of Engineering and Applied Sciences and has supervised over 50 master and doctoral candidates for degrees in both electrical engineering and engineering management. Etemadi is a member of the Institute of Electrical and Electronics Engineers (IEEE). [etemadi@gwu.edu]

John Kamp—received a Doctor of Engineering in engineering management from the George Washington University in 2019, a Master of Engineering in nuclear engineering from Iowa State University, and a Bachelor of Arts in mathematics and French from the University of Nebraska–Lincoln. Kamp joined the George Washington University research staff in 2019 and is supporting Etemadi's acquisition strategy research project. He is a retired naval submarine officer with extensive experience in research and development and program management. His research interests include engineering management, maritime systems, and acquisition system research. Kamp is a Fellow in the Royal Institution of Naval Architects and a member of several professional associations. [jckamp2018@gwu.edu]

Abstract

The Department of Defense (DoD) acquires operational systems via major defense acquisition programs (MDAPs). An average MDAP today will take about 8 years to deliver a new system (or new capabilities) to the operating forces using existing acquisition processes.

Cycle time is the duration between the start of system development until it is available for use. Programs can execute as planned when program cycle times are shorter than the pace of technology and adversary change. The pace of technology and adversary change is pushing the DoD to streamline acquisition processes and deliver products faster.

This paper presents a subset of research performed. It provides an overview of significant factors related to schedule and schedule growth. It classifies program acquisition strategies into three groups and identifies cycle time–related factors for these strategy groups.

Keywords: Acquisition strategies, Major Defense Acquisition Programs, program cycle times, decision frameworks, program management, predictor variable selection quantitative data analysis

Introduction

Former Secretary of Defense Mattis (2018) emphasized the need to deliver new capabilities at "*the speed of relevance"* (p. 10). The Department of Defense (DoD) will fast-track certain projects and focus priorities and resources to execute these projects. This research speaks to programs in the rest of the portfolio—those developing new capabilities that must accommodate changing priorities and resources and still deliver products on time and as promised.

Programs can execute as planned when program cycle times are shorter than the pace of technology and adversary change. However, the pace of technology and adversary change is pushing the DoD to deliver some capabilities sooner,[1] which often requires leadership involvement, greater risk, cost, effort, and acquisition process modifications.[2] These accelerated programs compete with other acquisition programs for resources and priorities, meaning some still-required programs will deliver required systems to the operating forces later and in smaller quantities than initially planned, unless they can make changes to reduce their cycle times.

Research Scope

This research focused on selected Major Defense Acquisition Programs (MDAPs)[3] active between 2007 through 2018 within the context of a defense-unique market with multiple government stakeholders and increasing demand for reduced cycle time[4] and capability delivery. Major policy changes[5] enacted between 2007 and 2018 provide context for the quantitative analysis of cycle time.[6]

Research Questions and Objectives

The research investigated policy and management issues related to accelerating DoD acquisition processes and addressed the following questions:

1. *What data reported in publicly released reports are significant predictors of program cycle time and schedule change*? In this research, cycle times are in months, and program start means approval to commence engineering and manufacturing development (also called Milestone B). *Schedule change* is the relative percent change relative to original cycle time since program start.
2. *How do these predictors change with acquisition strategies*? Acquisition strategies are detailed plans mandated by the Federal Acquisition Regulations (FAR 7, 2021). They are typically not publicly released but are inferred from observables such as solicitations, contract awards, budget and reporting documentation, and public reports of significant issues and events such as test failures and declaration of Initial Operational Capability (IOC).

The research produced several databases from publicly available sources suitable for research. These are available for research upon request from the authors. Simple regression

---

[1] Rapid Acquisition Offices can deliver interim solutions within 2 years of request.

[2] This is called streamlining or tailoring.

[3] MDAPs are weapon system programs with research and development expenditures greater than $300 million or procurement expenditures greater than $1.80 billion indexed to Fiscal Year 1990 constant dollars (MDAP, 2007).

[4] Cycle time is the duration between the start of system development until it is available for use, commonly identified by terms such as Initial Operational Capability (IOC) or Required Asset Availability (RAA).

[5] Specifically, the Weapons System Acquisition Reform Act of 2009, the 2016 National Defense Authorization Act (NDAA), and policy changes prior to January 2020 showcase this increasing demand for reduced cycle time.

[6] For example, the 2016 NDAA Section 804 changes require capability delivery within 5 years of program start to use these authorities.

models of cycle times and cycle time changes identified factors affecting cycle time reduction and growth from historical data.

We examined how these factors change with different acquisition strategies. This paper continues with an overview of recent literature, methodology, a summary of results, and conclusions.

Overview

The DoD executes MDAPs as life-cycle programs, where activities may be binned between development (which includes acquisition), procurement, and operations and support (O&S) phases of life. The F-14 spent 6 years in development, was produced for 22 years, and was operational for 33 years ("F-14 Tomcat," n.d.), or about 17% of its life in development, 61% in production, and 92% in service.[7] Platforms such as ships, aircraft, and vehicles are typically produced using hardware-based facilities with finite production capacities. For example, the Joint Light Tactical Vehicle program has a planned production buy of 58,306 vehicles (Dodaro, 2019). Full rate production at current budget levels is about 2,500 vehicles per year (Department of the Army, 2019), meaning production to meet inventory requirements could continue for over 20 years.

Acquisition Strategies

The DoD buys products, tangible and intangible items, and services collectively described as a *capability*. Acquisition *strategies* are developed by program offices and approved by senior leadership, and contain a statement of need for the capability, an estimated cost and schedule, and the contracting and support plans. Acquisition plans are *statutory* and regulatory documents that explicitly describe the *contracting and competition*[8] approaches (FAR 7.105, 2021). Specific statutory requirements vary depending upon the contracting strategy and include additional detail such as market surveys, performance criteria, and plans and requirements for technology development and risk management, test and evaluation, and security (FAR 7, 2021).

Schoeni (2018) defined three types of government acquisition strategies: coercion, public–private partnerships, and Competition using Open System Architectures . He found that only competition results in innovation (Schoeni, 2018).

The DoD categorized[9] acquisition strategies as acquisition models (Kendall et al., 2015). These include common structures such as hardware and software development, production, and operation and significant program phases and milestones. Programs are encouraged to tailor these models to the planned MDAP's "unique character" (Kendall et al., 2015). The GAO found variations within these models, such as planning to declare initial operational capability before completing initial operational test and evaluation (Dodaro, 2019).

An acquisition strategy defines *production and performance* requirements delivery plans spanning *single-step* or *incremental* schedules. Selected Acquisition Reports (SARs) describe and document production and performance requirements delivery, including production (hardware production line or software replication) and requirements fulfillment (complete or

[7] The F-14 program overlapped development and production and had concurrent production and operational service.

[8] The Federal Acquisition Regulations emphasize full and open competition and fixed-price contracts (FAR 7.105, 2021).

[9] This research was prior to implementation of the Agile Acquisition Framework.

incremental). Incremental upgrades[10] are production expressions of evolutionary acquisition strategies (Sylvester & Ferrara, 2003). By managing production and deployment configurations, incremental upgrades can be used to align production lots with deliveries of capabilities that mature between production versions (Mortlock, 2019).

Mortlock (2019) examined acquisition strategy development based upon assessed technical risk, approved requirements, and planned funding—using data from an actual program history, leading participants through decisions that a program office would make during program strategy development. He found that affordability concerns drive cost and schedule constraints, and despite preferences for single step acquisitions, incremental development is one of few choices for managing risk (Mortlock, 2019).

Acquisition strategies may include multiple acquisitions operating with varying degrees of coordination and interaction, such as unconstrained or complex systems (Stuckey et al., 2017). Rendon et al. (2012) identified system-of-system–related acquisition issues, such as control and program office staffing, and how these issues translate into modifications to contracting and organizational structures.

Georgiev (2010) analyzed defense acquisition strategy from a national perspective as a method to achieve policy goals. He classified defense acquisition strategies into those seeking technology innovation (active or offensive) or those adapting strategies to the current environment (passive or defensive) and the intended technology position (leader, follower, or outsider). He provided a hybrid of strategic and balanced scorecards to improve management decisions and results.

Existing regulations and statutes[11] define DoD rapid acquisition strategies. These limits result in limited scope and quantified objectives, senior leadership support and oversight, resource prioritization ahead of other programs, and extensive customization of existing processes to achieve program objectives. Tate (2016) postulated that only a few acquisition strategies are capable of rapid fielding—specifically, *using already mature or developed systems*, *incremental development and production* of limited or narrow capability improvements, and modular upgrades.

Acquisition strategies are generally not publicly available; however, some elements are in publicly available documentation,[12] such as single-step or evolutionary acquisition, technical maturity choices, and constraints on cost, schedule, and performance.

Schedule Growth Predictors

Better Buying Power was a process improvement initiative started by Secretary of Defense Robert Gates in 2010 (Layden, 2012) and expanded by the under secretary of defense for acquisition, technology, and logistics, Frank Kendall, in 2014 (Kendall, 2014)—with policy and direction to "buy more with no more" (Sethi, 2015). The initiative emphasized incentive-type contracts, affordability, and cost savings and realism (Kendall, 2014). Three parameters important to Better Buying Power are procurement quantities, unit cost, and cycle times; this research considers these the *functional objectives* of an acquisition strategy.

Cost growth is related to acquisition strategy factors such as prototyping, contract incentives in development and production, production competition, schedule concurrency, and

---

[10] Incremental upgrades are also referred to in the literature as block upgrades or release versions.

[11] See https://aida.mitre.org for more information.

[12] These include SARs, annual reports, and budget submissions.

schedule slip (Arena et al., 2006). Foreman (2007) identified longitudinal cost and schedule predictor variables based on SAR data. He showed that cost growth changes are related to procurement quantity changes and depend weakly on schedule growth between production decision (Milestone C) and IOC.

Lorell et al. (2017) compared six MDAPs with extreme cost growth and four with low cost growth and identified five salient program characteristics.[13] They noted, "Most of the extreme cost-growth programs' problems stemmed from a gross underestimation of the complexities and uncertainties … in designing, developing, integrating, and producing very challenging technological systems" (p. 97). While their findings are specific for cost growth, technical maturity and integration complexity are also related to schedule growth (Kamp, 2019).

Holloman et al. (2016) used SAR summary variance data to create cost, time and technical system-level *degree of difficulty* indicators and GAO Annual Assessments of Selected Weapon Systems maturity assessment data to indicate *achieved* technical performance. Using these indicators enables program managers to characterize acquisition performance risk *during execution* from monitoring and control processes such as Earned Value Management.[14]

Jimenez et al. (2016) conducted a literature review to find historical schedule growth predictors and identified statistically significant schedule-related predictors from MDAP SARs. Two variables were positively correlated to schedule growth between program start (Milestone B) and a production decision (Milestone C): research and development funds at program start, and program start on or after 1985. Two additional variables were negatively correlated with growth between Milestones B and C: percent research and development funds at program start, and program being a modification of an existing program or system (Jimenez et al., 2016).

Random forest methods have been used to create predictive contracting performance models (Gill et al., 2019) and provide an efficient method to identify important variables for use in a regression model (Grömping, 2009). Specific implementation issues are in the Methodology section of this paper. Wauters and Vanhoucke (2017) applied K-nearest neighbor methods to forecast project schedule and control methods and found that K-nearest neighbor methods work best for repetitive projects or those with accurate variability estimates. They also found that earned value/earned schedule approaches are best for controlling projects with high uncertainty.

This research fills the gap relating schedule prediction to acquisition strategies and will test two research hypotheses:

- Program cycle time may be predicted from programmatic resources and acquisition strategy decisions (H1).
- Percent change in program cycle time may be predicted from programmatic structural changes (H2).

---

[13] These are "insufficient technology maturity and higher integration complexity than anticipated; unclear, unstable, or unrealistic requirements, unrealistic cost estimates, adoption of acquisition strategies and program structures that lacked adequate processes for managing risk through incrementalism or through provision of appropriate oversight and incentives for the prime contractor, (and) use of a combined MS B/C milestone (assuming) that little or no RDT&E is required" (Lorell et al., 2017, pp. ix–x).

[14] Earned value tools relate project cost and schedule at the work breakdown structure level For a discussion of Earned Value Management, see Wei et al. (2016).

Program Functional Objectives

Program functional objectives—the cost, time to deliver, quantities delivered, and system performance—are the planned outcomes or results of the acquisition strategy. Cantwell et al. (2012) used a systems dynamics model to examine different cost-reducing strategies; all resulted in performance reduction and fewer delivered systems, three of four responses reduced total cost, and three of four achieved required schedule. This research used cycle time, unit cost, and procurement quantities as explicit program functional objectives.

Capili (2018) developed a system dynamics model of how factors such as contract types, schedule, and requirements and policy issues[15] can affect the ability of the government to implement Agile software development. Agile contracting scopes the number of requirements or *story points* during a fixed period of performance and cost (level of effort). Adding requirements during the Agile process results in trades and reductions of story points delivered to stay within schedule and cost constraints. Capili argued that the government acquisition constraints eliminate the ability of Agile processes to adapt to program changes.

Blair et al. (2011) provided examples from National Aeronautics and Space Administration (NASA) systems development and fielding and argued that most problems[16] in aerospace systems are due to problems with technical integration or system engineering deficiencies and failing to understand interactions. However, they also showed that institutional mandates, such as minimizing crew risk, bound what may be eliminated and add time and cost to acquisition strategies.[17]

Wong (2016) argued that the Mine Resistant Ambush-Protected vehicle program was delayed due to two institutional factors: validating the urgency of need and the decision to acquire systems meeting a long-term need or reacting to an urgent threat. These are analogous to Joint Capabilities Integration and Development System (JCIDS) deliberations and approvals of new capability requirements (McKenzie, 2018).

Methodology

This section reviews the research data collected, summarizes the response and predictor variables, and explains the supporting quantitative methods used in the research.

Research Design Overview

The study used data from publicly available reports on MDAPs from 2007 to 2018, from both the GAO annual assessments and Director of Operational Test and Evaluation (DOT&E) annual reports. We used Minitab 18, SPSS, and selected R libraries for statistical analyses. All data sets are in comma-separated variable formats and available from the authors upon request.

Data Collection

The data set contains 162 observations in an Excel spreadsheet. Observations had reports from both the GAO and the DOT&E during the fiscal year. This reduced the quantity of

---

[15] As an example, information assurance may be at the same time a story point, a policy, and a contract requirement.

[16] Blair et al. (2011) assert that 80% of problems are due to integration or system engineering failure without substantiation.

[17] They also provided several examples where designs were limited by physics or system engineering maturity (Blair et al., 2011, pp. 87–98) and required extensive systems engineering efforts to deliver the intended system performance.

observations but ensured two independent assessments of program status. SAR data and SAR summary data supplemented GAO information.

The data set was cleaned, filtered, and tested for consistency, correlation, and independence (Marshall & Russell, 2018). These were inspected by variable for correct entry and tested using Dixon's r22 outlier test (Minitab, n.d.) for outliers.

Statistical Methods

We characterized variables and their distributions by means, medians, standard deviations, and skew, kurtosis, and proportions (for categorical variables). We also calculated correlation coefficients and significance for continuous and categorical variables. We used graphing for qualitative data assessment.

Ensemble modeling (Ray, 2017) using the R randomForest package (Liaw, 2018) was used to identify important predictor variables. We compared variable importance from three R random forest packages—randomForest, cforest (Hothorn, 2020), and VSURF (Genuer, 2019)—to estimate variable importance and regression model performance (Grömping, 2009).

We created regression models in Minitab and SPSS relating cycle time to predictor variables. We adjusted models to maximize the predicted coefficient of determination (R-sq[pred]) with the fewest number of terms and inspected residual plots to identify any trends and verify regression assumptions. We tested association and independence of categorical factors to shorter or longer cycle times using chi-square tests. Finally, we used K-means clusters to classify acquisition strategies, subsetted the data into groups, and created regression models by groups to interpret differences in significant variables.

Results and Analysis

Cycle Time and Percent Change Cycle Time Regression Results

Random forest analysis identified starting predictor subsets for cycle time and percent change in cycle time regression models. We created a validation data set with a random draw of 44 (27%) of the full data set. A manual step-wise regression on the remaining 118 observations removed predictors with *p* values greater than 0.05 one at a time. Variance inflation factors (VIFs) were all less than 5, indicating no collinearity issues. The final models satisfied all regression assumptions. Table 1 summarizes significant predictors.

Table 1. Significant Regression Predictors

Cycle time (factor unit change = Δ months)	**% cycle time change** (factor unit change = % Δ)
R&D budget: (+)	Procurement budget % Δ (+)
Software approach Agile (-), Hybrid/NA (-) [relative to Waterfall]	DoDI 5000.02 Acq model: Model 4 (-), Models 2,5,6 (+) [relative to Model 1]
Joint (-) Depends on other MDAPS (+)	SVC Army, DoD (-), Navy (+) [relative to AF]
Reuses in-service technology (-) Uses commercial technology (-)	Integration issues (-) # Critical Technology Elements (CTEs) (+)
Financial instability (+)	Financial instability (+) Restructured (+) NM Breach (+)
R-sq(pred)~ 58%	*R-sq(pred)~ 59%*

Factors marked as (+) are associated with increasing cycle times. The factors highlighted in *green* and *yellow* are structural factors. Factors in *red* are either external or caused by external issues. Factors marked as (-) are associated with shorter cycle times.

Strategy structure factors associated with reducing cycle time include execution as a joint program use of an Agile or hybrid (including incremental) software development strategy, use of commercial technologies, and reuse of developed military technology. These factors may be changeable by program offices during execution.

We divided the data set cycle times into four quartiles to test categorical factor associations to MDAP cycle-time historical performance. We used chi-square association tests to test the trained regression model categorical predictors against cycle-time quartiles (Q1 = 1st quartile, Q4 = 4th quartile).

One factor, DoD acquisition model, was retired during this research. Table 2 summarizes the results of testing the association of the now-retired DoD acquisition model types[18] to cycle-time quartiles.

Table 2. DoD Acquisition Plan by Quartile

5000,02 model	Q1	Q2	Q3	Q4	P
1	21	20	22	26	0.000
2	0	1	2	6	
4	4	0	0	2	
5	10	16	17	4	
6	8	1	1	1	

xx	overrepresented in Q
xx	underrepresented in Q

Model 3, Incrementally Deployed Software Intensive Program, is not in this table as the data set had no such programs. P is the *p* value for the likelihood ratio test and shows a statistically significant association between cycle-time quartiles (columns) and DoD acquisition model (rows). Table 3 shows the quartile grouping of programs by the remaining categorical factors.

[18] One example of a now-retired DoD acquisition model type is Model 1, "Hardware Intensive Program" (Kendall et al., 2015).

Table 3. Quartiles Versus Regression Factors: Full Data Set

Factor	Category	Q1	Q2	Q3	Q4	P
Software approach	Waterfall	17	15	20	26	0.001
	Agile	21	23	20	7	
	Hybrid/NA	5	0	2	6	
Joint	no	32	29	34	34	0.475
	yes	11	9	8	5	
Depends on other MDAPs	no	28	12	14	7	0.000
	yes	15	26	28	32	
Reuses in-service technology	no	12	19	13	15	0.176
	yes	31	19	29	24	
Uses commercial technology	no	21	21	31	35	0.000
	yes	22	17	11	4	
Financial instability	no	27	15	15	6	0.000
	yes	16	23	27	33	

Two factors, joint and reuses in-service technology, did not show an association between the factors and cycle-time quartiles (P greater than 0.05). Programs with Waterfall-type software approaches were associated with longer cycle times (Q4). Programs using Agile and Hybrid-type[19] software approaches were associated with shorter cycle times (Q1). Commercial technology use was associated with shorter cycle times (Q1); programs that did not use commercial technology had longer cycle times (Q4). Finally, two program factors were associated with shorter cycle times (Q1) when they were not present: (a) dependence on other MDAPS and (b) financial instability (i.e., budget change greater than 10%).

Predictor Change with Acquisition Strategies

We classified the research data set into three acquisition strategies using K-means clustering with standardized variables on the functional objectives of cycle time, unit cost, and procurement quantities,[20] as shown in Table 4.

Table 4. Acquisition Strategy Clustering Summary

Mean	Cycle time (mo)	Unit cost ($M)	Procurement quantity	Count
High-End	**168**	**2,833**	364	56
Focused	92	175.8	274	64
Volume	99	1.4	**56,044**	42

Each row represents a functional objective, and the mean functional objective value is in the respective column. The maximum value of each column is in **bold**. Two of the three clusters

[19] "Hybrid" included incremental and mixtures of Agile, incremental, evolutionary, or some other approach.

[20] Clustering was on cycle time and the natural log transform of unit cost and procurement quantities.

align with column maxima—acquiring exquisite (“high-end”) capabilities and high unit cost, and acquiring product in large quantities (“volume”). The third cluster represents balancing functional objectives (“focused”) to deliver a capability requiring intermediate quantities and costs. The count column shows the number of observations in the data set in each cluster. Figure 1 shows this strategy grouping.

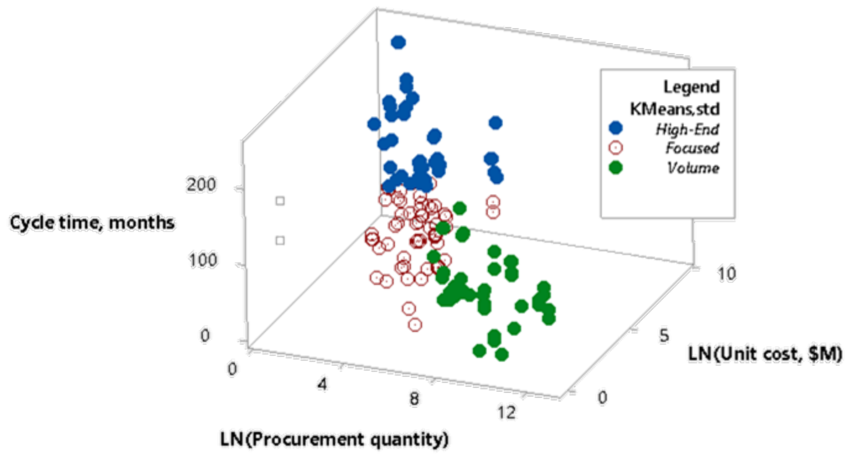


Figure 1. Acquisition Strategy Clusters

This classification strategy is dependent upon the values of the functional objectives and would be different if the classification used other objectives or used different approaches—such as weighting objectives or changing the clustering method. We tested factor significance using the same factors as in Table 1. Table 5 summarizes significant predictors by strategy.

Table 5. Cycle Time Significant Factors by Acquisition Strategy Cluster

High-End	Focused	Volume
R&D Budget (+)	R&D Budget (+)	~~R&D Budget (+)~~
Software approach*: Agile, ~~hybrid~~ (-)	~~Software approach*:~~ ~~Agile, hybrid (-)~~	Software approach*: Agile, hybrid (-)
~~Joint (-)~~ Depend on other MDAPs (+)	~~Joint (-)~~ ~~Depend on other MDAPs (+)~~	Joint (-) ~~Depend on other MDAPs (+)~~
Reuse existing DoD tech (-) ~~Use commercial tech (-)~~	~~Reuse existing DoD tech (-)~~ Use commercial tech (-)	~~Reuse existing DoD tech (-)~~ Use commercial tech (-)
~~Financial instability (+)~~	Financial instability (+)	~~Financial instability (+)~~
R-sq(pred): 27% No Box-Cox transform No outliers removed	R-sq(pred): 46% Box-Cox rounded λ = 0.5 One outlier removed[22]	R-sq(pred): 82% Box-Cox rounded λ = 0.5 Two outliers removed[21]

* Relative to Waterfall

[21] The outliers were removed to meet regression residual linearity assumptions.

Crossed-out and red factors (~~example~~) were not significant. Two groups had fewer significant predictors and a similar or better R-sq(predicted). The High-End group needed additional factors to improve model performance.[22] Table 6 shows percent cycle-time change for these same groups with the same coding as before.

Table 6. Percent Change in Cycle Time for Strategy Groups

High-End	Focused	Volume
Procurement Budget Change (+)	Procurement Budget Change (+)	~~Procurement Budget Change (+)~~
~~DoD 5000.02 (old) model **: 2,5,6 (+), 4 (-)~~	~~DoD 5000.02 (old) model **: 2,5,6 (+), 4 (-)~~	~~DoD 5000.02 (old) model **: 2,5,6 (+), 4 (-)~~
Service ~~(relative to AF):~~ Navy (+), Army~~, DoD (-)~~	Service (relative to AF): Navy (+), Army, DoD (-)	Service (relative to AF): Navy (-), Army, DoD (-)
~~Integration issues (-)~~ ~~# Critical Tech Elements (+)~~	~~Integration issues (-)~~ # Critical Tech Elements (+)	~~Integration issues (-)~~ # Critical Tech Elements (+)
~~Financial instability (+)~~ Restructure (+), ~~NM breach (+)~~	Financial instability (+) ~~Restructure (+),~~ ~~NM breach (+)~~	Financial instability (+) ~~Restructure (+),~~ ~~NM breach (+)~~
R-sq(pred): 61% Two outliers removed	R-sq(pred): 55%	R-sq(pred): 48%

Use of commercial technology, dependence on other programs, the number of critical technologies, and research budgets tended to differentiate between High-End, Focused, and Volume strategies. Volume strategy cycle time and change in cycle time were not related to research budgets and procurement budget changes but did reflect software development process differences. High-End strategies were not sensitive to the number of critical technology elements. Focused strategies were sensitive to financial instabilities.

Conclusions and Future Work.

Conclusions

Initial structural and strategy decisions affect cycle-time outcomes. Program cycle times are related to program resources and acquisition strategy decisions, and percent change in program cycle time is related to program structural changes. Significant cycle-time predictors include the size of research and development budgets, deciding to use commercial or reuse existing in-service technology, and avoiding dependency on other programs. Schedule change

[22] In this case, unit cost (+) and change in unit cost (+) were significant predictors for High-End programs.

(as measured by percent change in cycle time) was related to changing procurement budgets, the number of critical technology elements in a program, and financial (budgetary) stability.

Classifying MDAP acquisition strategies based on functional objectives of cycle time, unit cost, and procurement quantities highlighted how different factors, such as use of commercial technology or reuse of existing technology, were associated with reduced cycle times and early program decisions. We showed how significant cycle time and schedule growth predictors changed with acquisition strategies.

Relevance and Contribution to the Practice

This research provided quantitative insight into acquisition strategy factors affecting program cycle times and cycle-time growth. The associations between objectives and structural factors affect cycle times and identified significant acquisition strategy choices made during program development related to cycle-time outcomes.

Recommendations

These results reflect the use of public data and may not apply to non-MDAP programs. Future research should re-perform this research on a larger government-controlled data set and internal program documentation and compare findings with open-source results.

This material is based upon work supported by the Acquisition Research Program under Grant No. HQ00341910004. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Department of Defense nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. government.

References

Arena, M. V., Leonard, R. S., Murray, S. E., & Younossi, O. (2006). *Historical cost growth of completed weapon system programs* (Document No. TR-343-AF). RAND Project Air Force. https://www.rand.org/pubs/technical_reports/TR343.html

Blair, J. C., Ryan, R. S., & Schutzenhofer, L. A. (2011). *Engineering the system and technical integration* (NASA/CR—2011-216472). National Aeronautics and Space Administration.

Cantwell, P. R., Sarkani, S., & Mazzuchi, T. (2012). The effect of using a systems approach to project control within the U.S. defense industry. *2012 IEEE International Systems Conference Proceedings*, 1–5. https://doi.org/10.1109/SysCon.2012.6189454

Capili, R. (2018). *Using system dynamics modeling to understand the impact of federal constraints in implementing the agile methodology within the U.S. federal government* [Doctoral praxis, The George Washington University]. ProQuest.

Department of the Army. (2019). *Department of Defense Fiscal Year (FY) 2020 budget estimates: Army justification book of other procurement, Army tactical and support vehicles, budget activity 1*. https://www.asafm.army.mil/Portals/72/Documents/BudgetMaterial/2020/Base%20Budget/Procurement/05%20Other%20Procurement%20-%20BA1%20-%20Tactical%20&%20Support%20Vehicles.pdf

Dodaro, G. L. (2019). *Weapon systems annual assessment limited use of knowledge-based practices continues to undercut DOD's investments* (GAO-19-336SP). Government Accountability Office. https://www.gao.gov/assets/700/698933.pdf

F-14 Tomcat. (n.d.). In *Wikipedia*. http://en.wikipedia.org/wiki/F-14_Tomcat

FAR 7, Acquisition Planning (2021). https://www.acquisition.gov/far/part-7

Foreman, J. D. (2007). *Predicting the effects of longitudinal variables on cost and schedule performance* [Master's thesis, Air Force Institute of Technology]. Defense Technical Information Center. https://apps.dtic.mil/sti/pdfs/ADA463520.pdf

General Services Administration. (2019). *Federal acquisition regulation*. http://farsite.hill.af.mil/reghtml/regs/far2afmcfars/fardfars/far/Far1toc.htm#TopOfPage

Genuer, R. (2019, July 18). *VSURF: Variable selection using random forests*. CRAN. https://cran.r-project.org/web/packages/VSURF/index.html

Georgiev, V. (2010). Modeling defense acquisition strategy. *Connections: The Quarterly Journal*, *9*(4), 53–68. https://doi.org/10.11610/Connections.9.4.04

Gill, D., Muir, W. A., & Rendon, R. G. (2019). Predicting federal contractor performance issues using data analytics. *Proceedings of the 16th Annual Acquisition Research Symposium*, *II*, 16–27. https://calhoun.nps.edu/handle/10945/62901

Grömping, U. (2009). Variable importance assessment in regression: Linear regression versus random forest. *The American Statistician*, *63*(4), 308–319. https://doi.org/10.1198/tast.2009.08199

Holloman, S. S., Stuban, S. M. F., & Dever, J. (2016). Validating the use of performance risk indices for system-level risk and maturity assessments. *Engineering Management Journal*, *28*(2), 109–119. https://doi.org/10.1080/10429247.2016.1163245

Hothorn, T. (2020, March 5). *party: A laboratory for recursive partytioning*. CRAN. https://cran.r-project.org/web/packages/party/index.html

Jimenez, C. A., White, E. D., Brown, G. E., Ritschel, J. D., Lucas, B. M., & Seibel, M. J. (2016). Using pre-Milestone B data to predict schedule duration for defense acquisition programs. *Journal of Cost Analysis and Parametrics*, *9*(2), 112–126. https://doi.org/10.1080/1941658X.2016.1201024

Kamp, J. (2019). *Integrating immature systems and program schedule growth* [Doctoral praxis, The George Washington University]. ProQuest.

Kendall, F. (2014). *Better Buying Power 3.0* [White paper]. Department of Defense. http://www.acqnotes.com/Attachments/Better-Buying-Power-3.0-White-Power.pdf

Kendall, F., Gilmore, M., & Halvorsen, T. (2015). *Operation of the defense acquisition system*. DoD.

Layden, P. J. (2012). *Better Buying Power: An Army program manager's perspective* [Senior service college fellowship report]. Defense Acquisition University. https://www.dau.edu/training/career-development/sscf/Documents/Layden.pdf

Liaw, A. (2018, March 25). *randomForest: Breiman and Cutler's random forests for classification and regression*. CRAN. https://cran.r-project.org/web/packages/randomForest/

Lorell, M. A., Payne, L. A., & Mehta, K. R. (2017). Program characteristics that contribute to cost growth: A comparison of Air Force major defense acquisition programs (Document No. RR-1761-AF). RAND. https://www.rand.org/pubs/research_reports/RR1761.html

Marshall, E., & Russell, J. (2018). *The statistics tutor's quick guide to commonly used statistical tests*. StatsTutor. http://www.statstutor.ac.uk/resources/uploaded/tutorsquickguidetostatistics.pdf

Mattis, J. (2018). *Summary of the 2018 National Defense Strategy: Sharpening the American military's competitive edge*. DoD.

https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf

McKenzie, K. F. (2018, August 31). *Charter of the Joint Requirements Oversight Council (JROC) and implementation of the Joint Capabilities Integration and Development System* (JCIDS; CJCSI 5123.01). DoD. https://www.jcs.mil/Portals/36/Documents/Library/Instructions/CJCSI%205123.01H.pdf?ver=2018-10-26-163922-137

Minitab. (n.d.). *Minitab support—Outlier tests. Methods and formulas for outlier test*. Retrieved from https://support.minitab.com/en-us/minitab/18/help-and-how-to/statistics/basic-statistics/how-to/outlier-test/perform-the-analysis/select-the-analysis-options/

Mortlock, R. F. (2019). *Acquisition strategy formulation: Evolutionary/incremental development approach* (NPS-PM-19-179). Naval Postgraduate School. https://dair.nps.edu/bitstream/123456789/2768/1/NPS-PM-19-179.pdf

Ray, S. (2017, September 9). *Essentials of machine learning algorithms (with Python and R codes)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

Rendon, R. G., Huynh, T. V., & Osmundson, J. S. (2012). Contracting processes and structures for systems-of-systems acquisition. *Systems Engineering*, *15*(4), 471–482. https://doi.org/10.1002/sys.21214

Schoeni, D. (2018). Three competing options for acquiring innovation. *Air & Space Power Journal*, *32*(4), 85–93.

Sethi, S. (2015, January 8). Challenges in getting Better Buying Power to Indian army. *Indian Defence Review*. http://www.indiandefencereview.com/challenges-in-getting-better-buying-power-to-indian-army/

Stuckey, R. M., Sarkani, S., & Mazzuchi, T. A. (2017). Complex acquisition requirements: Using a systems engineering approach. *Defense Acquisition Research Journal*, *24*(2), 266–301.

Sylvester, R. K., & Ferrara, J. A. (2003). Conflict and ambiguity: Implementing evolutionary acquisition. *Acquisition Review Quarterly*, *10*(1), 27. ProQuest.

Tate, D. M. (2016). *Acquisition cycle time: Defining the problem (revised)*. Institute for Defense Analyses. https://www.jstor.org/stable/resrep22689?seq=1#metadata_info_tab_contents

Wauters, M., & Vanhoucke, M. (2017). A nearest neighbour extension to project duration forecasting with artificial intelligence. *European Journal of Operational Research*, *259*(3), 1097–1111. https://doi.org/10.1016/j.ejor.2016.11.018

Wei, N.-C., Bao, C.-P., Yao, S.-Y., & Wang, P.-S. (2016). Earned value management views on improving performance of engineering project management. *International Journal of Organizational Innovation*, *8*(4), 93–111. ProQuest.

Wong, J. P. (2016). *Balancing immediate and long-term defense investments* (RGSD-378). RAND. https://www.rand.org/pubs/rgs_dissertations/RGSD378.html

It’s About Time: Toward Realistic Acquisition Schedule Estimates

Charles Pickar—is a member of the NPS faculty where he teaches project management, defense acquisition, and systems engineering. Before joining NPS, he led the Applied Systems Engineering Program Area at the Johns Hopkins University Applied Physics Laboratory. He is a retired Army officer with extensive experience in the U.S. defense industry, including director and VP levels at Lockheed Martin, Northrop Grumman, and SAIC. He is the current Chair of the Systems Education Technical Committee of the IEEE Systems Council. His research and published work focus on applying systems engineering and system dynamics analytical approaches to defense acquisition problems. [ckpickar@nps.edu]

Raymond Franck, Brig Gen, USAF (Ret.)—retired from the faculty of the Graduate School of Business & Public Policy (GSBPP), Naval Postgraduate School (NPS) in 2012. He retired from the Air Force in 2000 in the grade of Brigadier General. His active duty career included a number of operational tours staff positions, and head of the Department of Economics and Geography, USAF Academy. His published work includes a number of journal articles and research reports on military innovation and defense acquisition management. [cfranck215@aol.com]

Abstract

This paper is part of a research agenda outlined in Franck et al. (2016) directed toward improving the realism of defense acquisition schedules. Defense acquisition schedules have long been a difficult problem. In this particular effort, we consider primarily the case of the 737MAX, which has been a fortuitous example of the risks of scheduling-by-fiat. We analyze the 737MAX misadventure using Systems Dynamic and Root Cause Analysis methods.

Introduction

Interest in project management estimations—cost and scheduling—remains strong in the academic world, driven by the sometimes-spectacular cost and schedule overruns in general, but defense projects in particular. Unfortunately, notwithstanding the effort, projects continue to fail and overrun every established metric. Nevertheless, we continue to study in the hopes of a breakthrough—a cure—for all that ails the defense acquisition world.

No one believes it is possible to accurately estimate a schedule so there are no overruns. Sometimes we get scheduling right, but often we get it wrong. After the fact, we can determine what went wrong and why; however, we have not yet been able to prevent failure. However, we believe we must do better not only to stay on schedule, but to answer the overriding imperative in defense acquisition to deliver systems as fast as possible that work. There are many hypotheses about the why, but it could be we are ignoring the one constant factor in project management: the human being with all the complexity and imperfections.

Humans tend to think about project management in the context of cause and effect (Dörner, 1996). We consider cost and associated variables during the project planning process. We do the same when developing a schedule. The planning process allows us to visualize how the development will unfold. Once we start executing, however, our ability to visualize the interplay of variables, from stakeholder demands to supply chain issues to requirements changes, is limited. We then react to events in a serial “cause and effect manner,” solving the immediate problem, but often neglecting to consider feedback and second order effects of those decisions. Newton’s Third Law of Motion states, “For every action, there is an equal and opposite reaction.” In human activities from engineering to war, a corollary to that law adds the idea of a counteraction, or response to the *reaction*. This concept is well understood in military planning and is a basic concept in wargaming, but in planning for and managing projects we identify the cause and effect relationships, action-reaction, but don't consider the action-reaction-*counteraction* sequence.

Scheduling is unique as studies by operational research experts, systems analysts, and even mathematicians attest (Boyd & Mundt, 1995; Herroelen & Leus, 2004, 2005; Rodrigues & Williams, 1998; Vandevoorde & Vanhoucke, 2006). In fact, we can explain schedule—the how and the what—using mathematics. We can also use the same mathematics and probability to develop schedules. What we haven't been able to do is apply mathematics and probability to get scheduling right. System dynamics provides the opportunity to consider scheduling and schedule execution from the people perspective.

System dynamics was conceived and developed by Forrester in the 1960s (Forrester, 1971). In many ways, Forrester's approach was like that of Dörner in that both recognized not only the limitations in human ability, but also that social systems were far more complex and difficult to understand than any technology. Further, both saw the world in terms of systems. Although we may not always think of it, we treat a development project, whether commercial or military, as a system with both inputs and outputs, as well as constraints and mechanisms. Inputs represent those management, budget, policy, materials, and other variables that are transformed by the system into outputs. Constraints are those regulatory, legal, fiscal, and time variables that restrict the system. Mechanisms are the people and processes used by the system to transform to the outputs.

If a project plan is a mental model of a system development, it represents the project team's shared assumptions of how the development will proceed. It represents a system structure (Forrester, 1971). Forrester also recognized that the human mental model (including that of a system development) often fails because the human mind often draws the wrong conclusions about the consequences of that model. System dynamics thinking and a recognition of the criticality in considering the role and thinking of the human in project management in general and scheduling in particular offers a tool to examine the execution of aerospace system developments.

The Boeing 737MAX: Background

The reader is entitled to ask why a commercial project, like the 737MAX, is a legitimate topic for defense acquisition research. We believe the answer is in the three parts. First, the Boeing airliner is an aerospace program with technical, program management, and scheduling issues. Second, the 737MAX program (particularly the aircraft accidents) have been highly publicized. This public discussion has produced a fairly extensive airing of the relevant facts, and also some excellent analyses (which make research in some depth both possible and potentially illuminating). Finally, the 737MAX is a superb example of what can happen when program duration is dictated by considerations outside the development program.[1]

The Boeing Corporation and Airbus SE, a duopoly, are the largest commercial and defense aircraft producers in the world. Boeing's first successful commercial jet was the Boeing 707 (first flight 1957). Airbus became a major commercial aircraft player with the A320 (the major Boeing 737 competitor) in 1987. In a very real sense, the Boeing 737MAX was a product of the Boeing-Airbus competition. Boeing has been continuously incorporated (albeit under different names) since 1916; Airbus since 1970.

[1] This is what we term an "aspirational" schedule estimate, which we define more specifically later.

That rivalry has not been especially friendly. It has featured World Trade Organization complaints[2] and some hard-fought contests in various market segments. These have included aerial tankers (Boeing KC-46 vs. NG/Airbus A330 MRTT). However, the center of their competition has been narrow-body civil airliners—the contenders being the Airbus A320 and Boeing 737 families. Both have been major commercial successes and significant contributors to both companies' profits. Deliveries by year are shown in Figure 1. While the competition has been intense, both companies have been highly successful in the narrow-body market—so far.

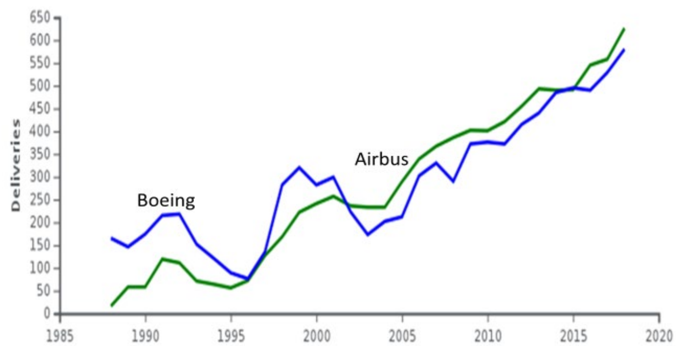


Figure 1. Boeing 737 and Airbus 320 deliveries by year (since A320 first flight)

By 2010, Boeing had reason to believe the Boeing 737 was becoming obsolescent, and was considering a new, clean-sheet replacement aircraft. The most promising enhancements then available were new turbofans, but they offered fuel efficiency improvements that were likely in single-digit percentages, certainly no more than 15% (Aviation Week, 2010). Boeing management reckoned that prospective customers would not be sufficiently interested in recapitalizing their fleets to purchase airplanes of this sort.

Thus, the near future for the narrow-body competition likely featured continued production of essentially the same aircraft (Boeing's 737NG) and the most recent versions of the Airbus 320 family. The next generation of narrow-body passenger airliners would appear around 2020.

At the time, that position was plausible, but proved to be wrong. New engines available (CFM International Leap 1B and the Pratt & Whitney PW 1000G) offered fuel efficiency increases of about 14%, and customers were indeed interested in the fuel efficiencies the A320neo (new engine option) offered**.** At the Paris Air Show of 2011, Airbus presold 667 A320neos in one week. This, and related developments, convinced Boeing of a time-sensitive need to respond to the re-engined Airbus models. In response, Boeing promised in 2011 to deliver a narrow body fairly quickly (Gelles et al., 2019).

Boeing entered development of a new narrow-body product (named the 737MAX) at a double disadvantage. First, Airbus had started its program sooner. Second, the Airbus 320 (first flight in 1987) was a newer design than the Boeing 737 (1967). In particular, it had more vertical distance between the (wing) engine mounts and the tarmac. This is shown in Figure 2.

---

[2] This dispute has surfaced again recently, with U.S. threats to impose tariffs on EU goods because of a WTO finding of illegal subsidies for Airbus (Peker & Zumbrun, 2019). One previous chapter in this long-running story is recounted in Franck et al. (2011, pp. 8–9).

Basically, Boeing found itself in the position of having to produce a new narrow-body airliner that would be ready (soon enough) close to the A320neo launch with fuel efficiency improvements that were sizeable (good enough) to cause customers to remain with Boeing rather than moving to Airbus. Airbus was going to have the A320neo available by 2017, with 12–15% improvement in fuel efficiency (relative to the A320). Boeing's response was to promise *quick delivery* of a new model 737 with a new fuel-efficient engine.

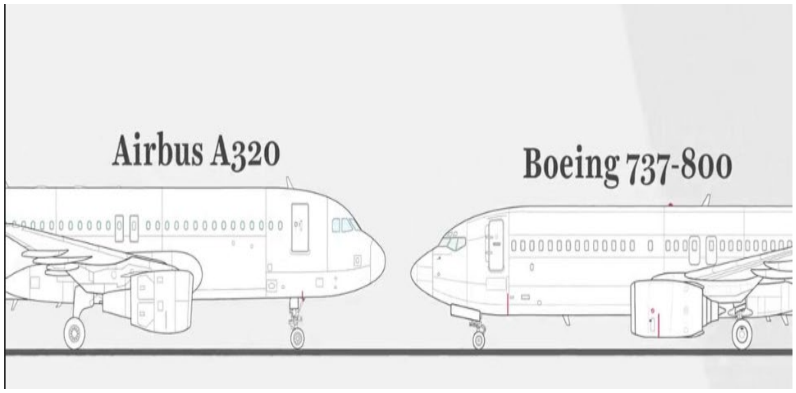


Figure 2. Engine ground clearance on A320 family vs. Boeing 737 family (bing.com/images)

In summary, the logic that appears to have driven the Boeing 737MAX strategy is as follows:

- Competitive pressure from Airbus for single-aisle aircraft forced Boeing to do something quickly in order to remain in that line of business.
- Fuel efficient engines are necessary to sell passenger jets in the current global market. They were also necessary to tempt airline customers to buy new aircraft.
- The schedule would be driven by the perceived need for an in-service time as close to the Airbus A320 neo as possible.
- The schedule, over cost and performance, seems to have been Boeing management's driving factors, rather than being driven by the time required for the necessary engineering.
- Use of 737 airframe meant lower production costs, simpler FAA certification, and lower training costs (driven by pilot familiarity with existing 737 fleet).
- Larger engines did not fit on the existing 737 wings, so design modifications were needed.
- Engine modifications changed the aerodynamics of the airplane.
- However, the airplane needed to match the pilot qualification requirements of in-service B737 aircraft so that aircrew training did not significantly delay introduction to commercial service. (This last constraint proved to be particularly consequential.)[3]

It appears the modifications to this aircraft originally designed in the late 1960s to make it a competitor in the 2010s were greater than originally anticipated. Relocated, more powerful engines significantly changed the aircraft handling characteristics in some flight regimes. Any new requirements resulting from the changes probably should have driven new requirements

[3] This would mean that the 737NG and 737MAX would carry the same "type rating" and pilots could be qualified to fly both. Campbell (2019) explains all this succinctly and well.

which would have increased the schedule an unacceptable amount of time. This management reaction to competitive pressure seems a classic case of what we call an aspirational schedule. Figure 3 shows a system view of the 737 upgrade. The arrows show the interrelationship of the system variables, as well as the feedback those variables can cause.

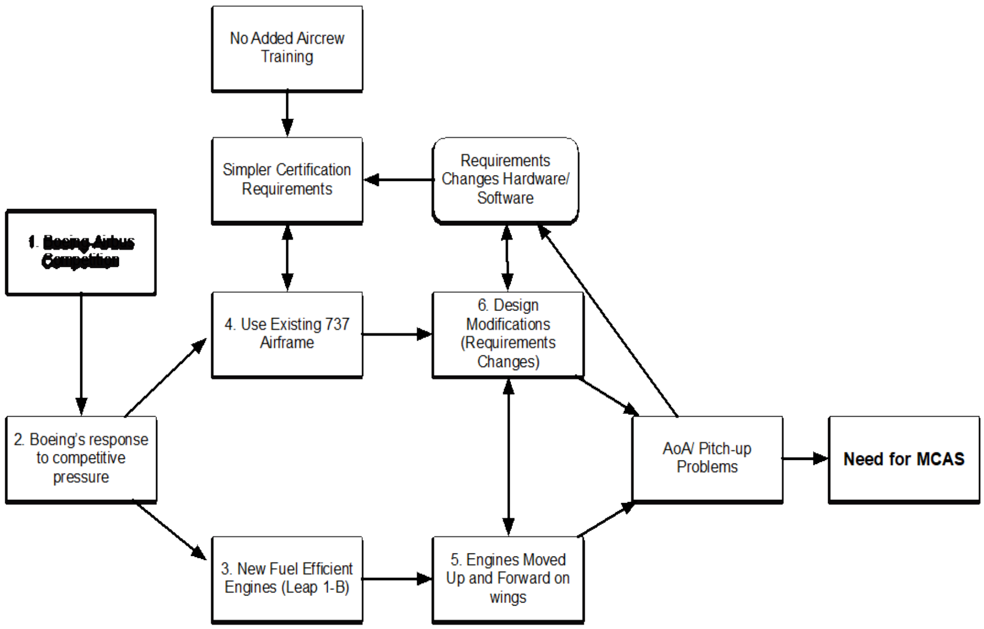


Figure 3. The 737 Upgrade “System”

The Race to the Swift

Completing development of the Boeing aircraft to the Airbus timeline was a schedule challenge, as shown in Table 1. The major program objective was to deliver a more fuel-efficient narrow-body airliner quickly and at a relatively low cost before the market was saturated by Airbus.

The new engine was larger in diameter than those on current 737s.[4] This difficulty was overcome by modifying the nose landing gear and positioning the new engine forward and upward relative to the wing. The overall effect was to keep the same ground clearance while taxiing. However, moving the engines changed the flight characteristics of the aircraft.

Table 1. Major Milestones (Actual) for the Airbus A320neo and Boeing 737MAX

EVENT	**A320 neo**	**737MAX**
Program Announced	December 2010	August 2011
First Aircraft Produced	July 2014	December 2015
First Flight	September 2014	January 2016
First Customer Delivery	January 2016	May 2017

[4] The 737 was well designed for a low-bypass turbofan, like the JT8 (48” in diameter). A high-bypass turbofan like the CFM-56 (60”) necessitated an oval cowling to preserve ground clearance. An advanced high-bypass turbofan like the LEAP 1-B (69”) also necessitated repositioning the engine for the 737MAX family.

Schedule Pressure: Speedy Execution

The schedule pressure had technical ramifications, but also impacted the work environment. According to several published reports (e.g., Broderick, 2019; Gelles et al., 2019; Nicas et al., 2019), the 737MAX development program proceeded on something of a forced march schedule. Boeing employees perceived a compelling need to finish within the company's time frame.

In search of faster progress, an approach described as "compartmentalized" came into being.[5] (This, of course, raises the question whether this approach contributed to the Maneuvering Characteristics Augmentation System [MCAS] problems that arose later.) Some concluded in retrospect that the development schedule had been "stretched to the breaking point" (e.g., Campbell, 2019).

The Problematic Maneuvering Characteristics Augmentation System

> Scientists and engineers seem able to predict only a fraction of the difficulty they are likely to face in a specific project. Much of it simply crops up unexpectedly. (McNaugher, 1987, p. 66)

In fact, even programs intended to be simple can have complications, and the 737MAX was deceptively complicated. The A320neo is mostly a re-engined A320. The new 737MAX required more complex modifications to reduce drag and a need to reposition the engines. The new engines (LEAP 1B) were almost 40% larger and weighed almost double those of the 737NG (CFM-56). The new plane was longer and had a wider wingspan. What Boeing couldn't change was the height above airport ramps without having to redesign the landing gear, which would have threatened both the development schedule and quick FAA certification (Tkacik, 2019).

A later part of the testing program revealed the aircraft tended to pitch up, because the Center of Gravity (CG) and the Center of Lift (CL) were too close together due to the new engine location (Coughlin, 2020). The change in engine position is shown in Figure 2.

Acknowledging the challenges with the engine repositioning, one proposed solution involved modifications to the airframe itself (Langewiesche, 2019). However, given the schedule pressure, Boeing chose instead a software solution.[6] It was named the Maneuvering Characteristics Augmentation System (MCAS; Gelles, 2019; Broderick, 2019). The first version of MCAS (MCAS1[7]) was intended to input automatic, corrective control inputs to situations involving relatively high airspeeds (and G forces, in the form of 0.6 degrees of pitch-down trim applied in 10 seconds, with maximum trim change limited to 5 degrees; Gates, 2019).

However, later flight tests also revealed some difficulties at normal G forces and low airspeed. This led to increased realm of engagement parameters to include low speeds, high angles of attack (AoAs) and "normal" G-loadings. MCAS pitch changes were increased to 2.5 degrees (Campbell, 2019). Moreover, the resulting MCAS2 could engage any number of times (Gates, 2019; Langewiesche, 2019). The overall effect was to make the MCAS2 "more aggressive and riskier" (Nicas et al., 2019).

---

[5] This is analogous to problem decomposition in the operations research literature, and also bears resemblance to "concurrency" in the acquisition literature.

[6] There were, however, other reasons to favor a software solution.

[7] We term the earlier version as MCAS1 and the later (more aggressive version) as MCAS2. This is our own terminology and adopted for expository clarity.

From a software engineering perspective, Johnston et al. (2019) suggest there were four key errors in the development and fielding of MCAS: poor documentation, a rushed release, delayed software updates, and humans out of the loop. The poor documentation refers to not only the lack of documentation on MCAS, but that the documentation was printed instead of digital. MCAS1 was regarded (correctly, we think) as an "innocuous" feature that could or would seldom emerge as a problem. And, should it occur (in either version), treating the incident as a runaway-trim malfunction would solve the problem. Thus, the flight crews involved in the Lion and Ethiopian accidents had little, if any, knowledge of MCAS2 operation or potential consequences should an AoA indicator malfunction. According to Pastzor (2019): "One senior Boeing official said the company had decided against disclosing details about the system that it felt would inundate the average pilot with too much information—and significantly more technical data—than he or she needed or could realistically digest."

The rushed release was a product of the marketing driven strategy Boeing pursued—release a product so as not to lose business (Johnston & Harris, 2019). Statements attributed to Boeing employees assigned to the project included "intense pressure cooker," "fast turnaround" environment, and work at "double the normal rate." One technician reported that he had received "sloppy blueprints" with a promise of future fixes. However, that remedy was still incomplete in early 2019 (Gelles et al., 2019).

The delayed software updates were affected by some things Boeing could not control; the U.S. government shutdown in 2017 caused updates to be delayed by at least 4 months (Johnston & Harris, 2019; Pasztor, 2019). In at least one case, Boeing submitted a software fix to the FAA for certification 7 weeks before the Ethiopian Airlines crash. It is impossible to know whether a less rushed, more robust software design process would have made a difference (Johnston & Harris, 2019).

The fourth issue, the "human out of the loop" problem, resulted from the MCAS2 being activated by a single AoA sensor (Nicas, 2019). Choosing to rely on one indicator when two were readily available could only be regarded as a "bewildering mistake" in retrospect (Langewiesche, 2019).[8] A related mistake was allowing the more powerful MCAS2 to be activated an unlimited number of times. As Langewiesche (2019) noted, "No one I spoke to from Boeing, Airbus or the N.T.S.B. could explain the reasoning here."

It is clear in hindsight that Boeing's haste led to mistakes or miscommunications. Those out of the loop were not limited to pilots. For example, relevant FAA officials were not informed (Gates, 2019); discussion of the MCAS system was deleted from the 737MAX pilot manual (Tangel, 2019); and furthermore, the more aggressive version, MCAS2, was not well shared with interested parties, including airworthiness certification authorities (Tangel, 2019).

---

[8] We regard Langeiwesche as the best single source on the B737 fatal accidents, particularly as to what happened and why.

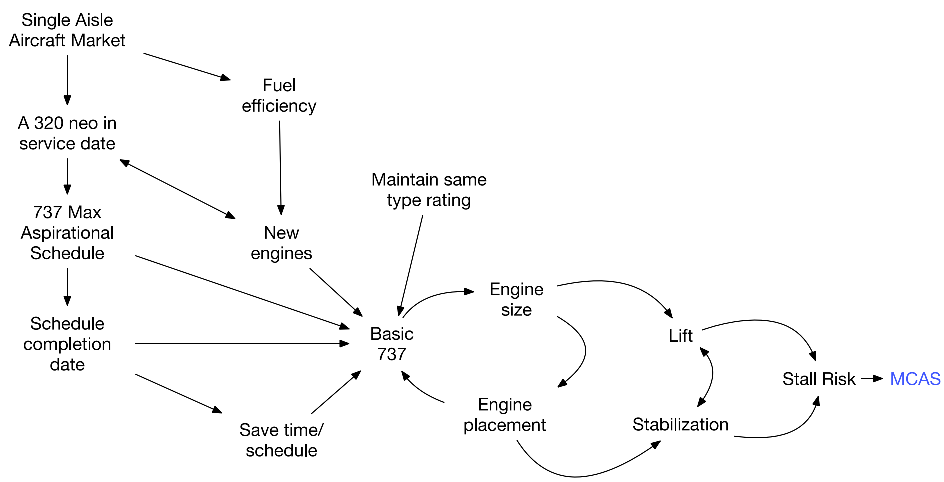


Figure 4. How the B737Max Came to Be

Our summary of the 737Max story is shown in *Figure 4*. The key drivers were time perceptions based on the market rather than engineering estimates. It's worth noting that getting to "MCAS decisions" include both technical and communications issues (especially not fully informing the pilots). Figure 5 shows our assessment of the dynamics of the development.

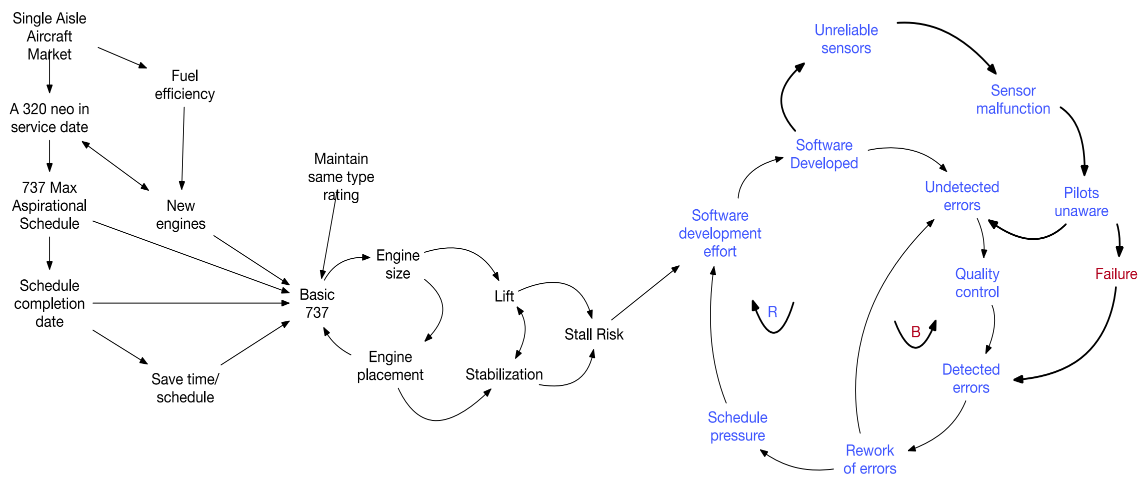


Figure 5. The Dynamics of the MCAS Development

THE 737MAX CRASHES

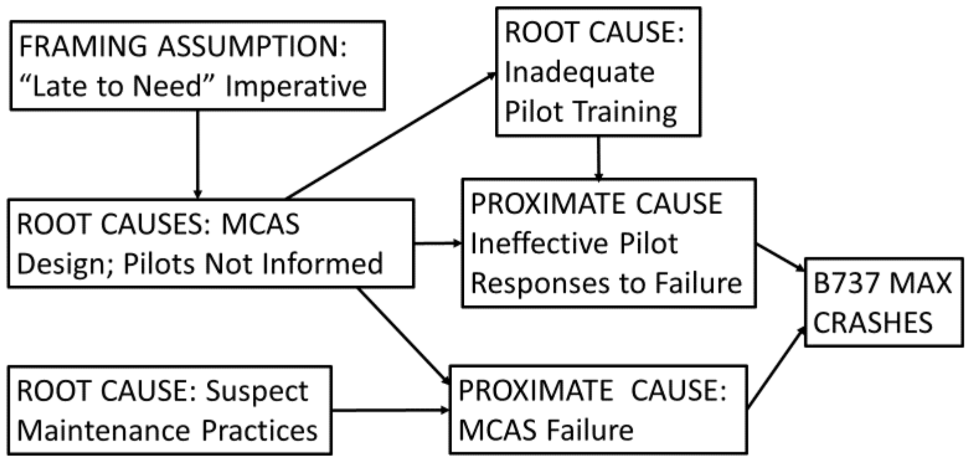


Figure 6. Hypothesized Causes of 737MAX Mishaps.
(Langewiesche, 2019; Pasztor, 2019; authors' interpretations)

The circumstances of the Lion and Ethiopian Air accidents with the 737MAX have been discussed extensively. For those interested in learning more about the events surrounding the crashes, we recommend Langewiesche (2019) and Tangel et al. (2019) as good starting points. In addition, reports by various agencies (e.g., House Committee on Transportation and Infrastructure, 2020; JTAR, 2019; NTSB, 2019) are very informative.

Overall Comments on This Case

1. What trades were made?

- Schedule got major (close to exclusive) emphasis—with attendant design constraints. Strict (at least fairly strict) adherence to original schedule was pervasively enforced.

2. Consequences of the trades (results of schedule emphasis)?

- A new narrow-body type was conclusively ruled out; schedule constraint dictated a 737 variant with a new, but already developed model.
- In development there was a lack of overall program review and oversight.
- In at least some cases, multiple concurrent tasks were completed with insufficient regard for overall aircraft safety. (Nothing apparently went wrong with "performance" or apparently with cost.)
- Time pressures throughout the 737MAX development effort.
- Some steps were skipped.
- Some steps were done concurrently.
- Most notable specific example of unknown work (ex-ante) is that a bigger engine on old airframe design resulted in pitch-up problems noted at low speeds and high angles of attack.
- Although not intended, the program pace detracted from the operating safety of the airplane delivered for commercial service.

3. How and why were the trades made?

- Schedule emphasis due primarily to commercial success of A320neo, which put Boeing at a major strategic disadvantage in the narrow-body airliner market.
- Recovery strategy focused on a quickly developed 737 variant.

4. What is the evidence or rework? Primarily responding to the pitch-up problem:

- MCAS (Maneuvering Characteristics Augmentation System, part of 737MAX software suite) was changed to deal with aircraft handling issues in high-speed flight.
- Further rework arose with the need to resolve pitch-up problems at low speed with high power settings. A solution (MCAS2) appeared late in the game (within the corporate-dictated schedule). Good information on this issue (both quality and quantity) is available in publicly available sources.

Some Further Observations

An interesting, but so far somewhat neglected, question was how Boeing was caught wrong-footed in 2011 with the Airbus 320neo underway, with no planned response until 2020. One report (Broderick, 2019) has it that the A320neo family was originally intended as a defensive response to the potential threat from Bombardier's Canadian Regional Jets.[9]

If that is indeed the case, then why did Boeing not feel a similar need to likewise defend itself? For Boeing, like Airbus, the narrow-body airliner family is its leading source of profit. How did Boeing look at the same market environment (with the same contestability concerns) and reach a very different conclusion—especially when the B737 design was closer to the end of a long run than was the A320. The bottom line might well be that the fundamental root cause of the current 737MAX difficulties was a strategic miscalculation a decade in the past.

As noted, this miscalculation led to a difficult problem for Boeing in two parts: (1) coaxing additional competitive life from a half-century-old design, and (2) doing so in a manner responsive to the threat posed by the Airbus A320neo. This first part was due primarily to an old design originally intended for low-bypass turbofans. The second part would have been less difficult had Boeing not made the strategic miscalculation noted previously and had started its 737-replacement program sooner.

In short, Boeing launched a program which (like all new programs) had both competitive and technical aspects. In this case, a timely response to the A320neo dictated the form (re-engining) of its narrow-body program at a pace driven by the A320neo family. The MCAS consequences were partly a matter of bad luck, but the design team might well have made other miscalculations for which the consequences have been nonexistent or less serious.

Finally, and most relevant to our current purpose, is that the 737MAX case is an object lesson concerning the hazards of aspirational schedules, especially if they're taken too seriously.

---

[9] This seems unlikely at first glance but is understandable. For example, Franck et al. (2012) contains an analysis of various potential competitors to the Boeing-Airbus narrow-body duopoly – of which Bombardier and Embraer regional jets were reckoned the most serious. This hypothesis is supported, *inter alia*, by Airbus acquiring the Canadian Regional Jet production facilities, and Embraer's regional jets now being a joint venture with Boeing (which holds an eighty-percent interest).

A Final Word—Aspirational Schedules

While this study examined a commercial aircraft project, the similarities of complex system development are strikingly like the broader aerospace and defense (A&D) sector. Indeed, Boeing is both commercial aircraft manufacturer and defense contractor. We posit the aerospace-defense world practices two inherently different kinds of system development scheduling. There are two kinds of scheduling. The first is based on a structured planning (e.g., Critical Path Method [CPM] or Program Review Evaluation Technique [PERT]). The second kind of scheduling is that described in this paper—aspirational scheduling. We define an aspirational schedule as one defined by a political or business desire, aim, or goal rather than accepted scheduling techniques.

Aspirational schedules are driven by political and commercial processes and decisions. It is an example of making engineering development fit a strategy, rather than allowing the engineering discipline to define the time needed. In and of itself, the idea of political or commercial events driving developments is not new. Developments from the Manhattan Project to the Lockheed U-2 to Polaris are examples of political requirements driving development. What may be lacking in this latest move to aspirational schedules, however, is acknowledgement of the challenges of aspirational scheduling and an acceptance of the necessity for reasoned trades in a development.

Aspirational schedules now appear to be highly fashionable, and they have also attracted powerful institutional advocates. One recent example is the Air Force's Digital Century Series initiative. Its chief advocate is the current Assistant Secretary of the Air Force for Acquisition, Will Roper. He advocates rapid development and production of a series of fighter aircraft, such as the Air Force procured starting in the 1950s. His rationale includes a more agile response to peer competitors, enabled by new generations of design simulation software (Freedburg, 2019). This is to take place in a less risk-averse acquisition culture. With these technologies and development of new combat aircraft types in 5 years or less, this looks a lot like a large-scale adoption of aspirational schedules (Insinna, 2019).

A second example emerging is the Ground-Based Strategic Deterrent (Minuteman ICBM replacement). The program has been declared "late to need," and is proceeding apace despite potential complications with an ongoing Federal Trade Commission investigation (Censer, 2019; Clark, 2019; Erwin, 2018; FTC, 2018).

References

Airbus Newsroom. (2010, December 1). *Airbus offers new fuel saving engine options for A320 family*. https://www.airbus.com/newsroom/press-releases/en/2010/12/airbus-offers-new-fuel-saving-engine-options-for-a320-family.html

Airbus Newsroom. (2014, September 25). *First A320neo successfully completes first flight*. https://www.airbus.com/newsroom/press-releases/en/2014/09/first-a320neo-successfully-completes-first-flight.html

Airbus Newsroom. (2016, January 20*). First A320neo delivery opens new era in commercial aviation*. https://www.airbus.com/newsroom/press-releases/en/2016/01/first-a320neo-delivery-opens-new-era-in-commercial-aviation.html

Aviation Week. (2010, September 6). *Re-engine no-go?*

Boeing 737.org. (2019). *Boeing 737 MAX: Differences*. http://www.b737.org.uk/737maxdiffs.htm

Boeing Media Room. (2011). *Boeing launches 737 new engine family with commitments for 496 airplanes from five airlines*. https://boeing.mediaroom.com/2011-08-30

Boyd, D. S., & Mundt, B. D. (1995). Schedule estimating relationships for the engineering and manufacturing development of bomber, transport, tanker, and surveillance aircraft systems. *The Journal of Cost Analysis*, *12*(1), 131–154. http://doi.org/10.1080/08823871.1995.10462295

Broderick, S. (2019). *The MAX saga, one question at a time*. https://aviationweek.com/max-saga

Brown, D. P. (2016). The Boeing 737 MAX completes first flight (and landing). *Airliner Reporter.* https://www.airlinereporter.com/2016/01/the-boeing-737-max-completes-first-flight-and-landing/
Competition between Airbus and Boeing. (2019, September 17). In *Wikipedia.* https://en.wikipedia.org/wiki/Competition_between_Airbus_and_Boeing
Campbell, D. (2019). The many human errors that brought down the Boeing 737 Max. *The Verge.* https://www.theverge.com/2019/5/2/18518176/boeing-737-max-crash-problems-human-error-mcas-faa
Censer, M. (2019). Northrop says it has received "civil investigative demand" from Federal Trade Commission. *Inside Defense.* https://insidedefense.com/insider/northrop-says-it-has-received-civil-investigative-demand-federal-trade-commission
Clark, C. (2019). FTC investigates Northrop: GBSD. *Breaking Defense.* https://breakingdefense.com/2019/10/ftc-investigates-northrop-gbsd/
Coughlin, D. (n.d.). *Crashing the 737 MAX.*
Dörner, D. (1996). *The logic of failure: Why things go wrong and what we can do to make them right.* Metropolitan Books.
Erwin, S. (2018, June 5). Acquisition of Orbital ATK approved, company renamed Northrop Grumman Innovation Systems. *Space News.* https://spacenews.com/acquisition-of-orbital-atk-approved-company-renamed-northrop-grumman-innovation-systems/
Hashim, F. (2017, May 25). *Malindo operates world's first 737 Max flight*. Flight Global. https://www.flightglobal.com/orders-and-deliveries/malindo-operates-worlds-first-737-max-flight/124109.article
Flight Global. (2006, March 3). *Boeing firms up 737 replacement studies by appointing team*. https://www.flightglobal.com/boeing-firms-up-737-replacement-studies-by-appointing-team/66022.article
Franck, R. E., Lewis, I., Matthews, D., & Udis, B. (2011). *Emerging patterns in the global defense industry* (NPS-AM-11-001). Naval Postgraduate School. https://my.nps.edu/documents/105938399/108619268/NPS-AM-11-001.pdf/f0f69e09-399e-485b-b987-9665eeae9069
Franck, R., Lewis, I., & Udis, B. (2012). *Global aerospace industries: Rapid changes ahead?* (NPS-AM-12-014). Naval Postgraduate School. https://my.nps.edu/documents/105938399/108614878/NPS-AM-12-014.pdf/ac1f3375-a07d-4e0c-9fdd-e70fe24c3798
Forrester, J. W. (1971). Counterintuitive behavior of social systems. *Technology Review*, *73*(3), 52–68.
Freedberg, S. J., Jr. (2019). A new century series? Will Roper takes Air Force back to the future. *Breaking Defense*. https://breakingdefense.com/2019/04/a-new-century-series-will-roper-takes-air-force-back-to-the-future/
Federal Trade Commission. (2018, December 3). *Federal Trade Commission decision and order: In the Matter of Northrop Grumman Corporation, a corporation; and Orbital ATK, Inc., a corporation* (Decision and Order Docket No. C-4652). https://www.ftc.gov/system/files/documents/cases/181_0005_c-4652_northrop_grumman_orbital_atk_modified_decision_and_order_12-4-18.pdf, 3 December.
Gates, D. (2019, March 18). Flawed analysis, failed oversight: How Boeing, FAA certified the suspect 737 MAX flight control system. *Seattle Times.* https://www.seattletimes.com/business/boeing-aerospace/failed-certification-faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lion-air-crash/
Gelles, D., Kitroeff, N., Nicas, J., & Ruiz, R. R. (2019). Boeing was "go, go, go" to beat Airbus with the 737 Max. *New York Times.* https://www.nytimes.com/2019/03/23/business/boeing-737-max-crash.html
Herroelen, W., & Leus, R. (2004). The construction of stable project baseline schedules. *European Journal of Operational Research*, *156*(3), 550–565. https://doi.org/10.1016/S0377-2217(03)00130-9
Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, *165*(2), 289–306. http://doi.org/10.1016/j.ejor.2004.04.002

House Committee on Transportation and Infrastructure. (2020). *The Boeing 737 MAX aircraft: Costs, consequences, and lessons from its design, development, and certification: Preliminary investigative findings*. Democratic Staff.
Insinna, V. (2019). *The U.S. Air Force's radical plan for a future fighter could field a jet in 5 years.* Defense News. https://www.defensenews.com/digital-show-dailies/2019/09/16/the-us-air-forces-radical-plan-for-a-future-fighter-could-field-a-jet-in-5-years/
Johnston, P., & Harris, R. (2019). The Boeing 737 MAX saga: Lessons for software organizations. *Software Quality Professional*, *21*(3).
Joint Authorities Technical Review. (2019, 11 October). *Boeing 737 MAX flight control system* (Report submitted to FAA). https://www.faa.gov/news/media/attachments/Final_JATR_Submittal_to_FAA_Oct_2019.pdf
Kingsley-Jones, M. (2008, February 27). A320 completes first IAE V2500 SelectOne. *Flight Global.* https://web.archive.org/web/20180618175734/https:/www.flightglobal.com/news/articles/picture-a320-completes-first-iae-v2500-selectone-221880/
McNaugher, T. L. (1987). Weapons procurement: The futility of reform. *International Security, 12*(2), 63–104.
Peker, E., & Zumbrun, J. (2019). U.S. to impose tariffs on EU goods after WTO's Airbus ruling. *Wall Street Journal.* https://www.wsj.com/articles/u-s-can-levy-tariffs-on-eu-exports-over-airbus-wto-says-11570025040
Nicas, J., Kitroeff, N., Gelles, D., & Glanz, J. (2019, June 2). Fatal flaw in Boeing 737 Max traceable to one key late decision. *Irish Times.* https://www.irishtimes.com/business/manufacturing/fatal-flaw-in-boeing-737-max-traceable-to-one-key-late-decision-1.3912491
National Transportation Safety Board. (2019). *Safety Recommendation Report: Assumptions used in the safety assessment process and the effects of multiple alerts and indications on pilot performance* (Accident Number: DCA19RA017/DCA19RA101, Boeing 737 MAX 8, 29 October 2018, 10 March 2019, ASR 19-01, 19 September). https://www.ntsb.gov/investigations/accidentreports/reports/asr1901.pdf
Pasztor, A., & Tangel, A. (2019, March 28). How Boeing's 737 MAX failed. *Wall Street Journal.* https://www.wsj.com/articles/how-boeings-737-max-failed-11553699239
Rodrigues, A. G., & Williams, T. (1998). System dynamics in project management: Assessing the impacts of client behaviour on project performance. *The Journal of the Operational Research Society*, *49*(1), 2–15. http://doi.org/10.1057/palgrave.jors.2600490
Tangel, A., Pasztor, A., & Maremont, M. (2019, August 17). The four-second catastrophe: How Boeing doomed the 737 MAX. *Wall Street Journal.* https://www.wsj.com/articles/the-four-second-catastrophe-how-boeing-doomed-the-737-max-11565966629
Tkacik, M. (2019, October). How Boeing's managerial revolution crash course created the 737 Max disaster. *The New Republic*.
Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*(4), 289–302. http://doi.org/10.1016/j.ijproman.2005.10.004

Agile Improvements to Critical Path Method

Caitlin Kenney —is a Systems Engineer and Agile Coach for International Systems Management Corp. She has an MS in Systems Engineering from University of Maryland and a BS in Industrial Engineering from Northeastern University. Kenney is currently pursuing a PhD in Civil Engineering, Project Management at University of Maryland. She has over 10 years of experience supporting DoD projects, including two DoD Agile Pilot Programs and multiple agile project management adoptions. She is a licensed professional engineer in Illinois and Maryland. [Ckenney7@gmail.com]

Abstract

Agile project managers have a difficult time communicating schedules in traditional formats. For example, integrated master schedules, work breakdown structures, and so on are expected by project leadership, especially if they are new to agile. Agile does not include detailed level planning, years in advance. Instead, it focuses on small increments and iterates on the scope until the stakeholder need is met. Detailed planning is only done within the confines of an increment. Higher level milestones are captured for future efforts, but the detailed level planning is not done until nearing start of the work. Limitations of traditional project management techniques are nothing new. However, management of such projects still has not changed or significantly improved upon monitoring and controlling methods since their inception. The traditional Critical Path Method (CPM) is an example. CPM has the following known limitations. There is an assumption that durations can be accurately estimated and forecast. Estimates are based on historical data and made by separate planning departments, not the "doers"—the people closest to the work. Interdependencies are not always clear up front—they evolve as more information becomes available and work is further broken down, making it difficult to differentiate between scope creep and scope refinement. This can result in lengthy and costly contractual battles between the contractor and developer. Schedule logic in CPM is largely driven by finish-to-start dependencies, making other types of dependencies difficult to forecast and the critical path difficult to identify without software. Software requires user training/results in a designated "scheduler," removed from the work. Management assumes projects can add resources to speed up completion, failing to account for learning curves, increased complexity, and additional bottlenecks. By applying production management and agile principles, projects can improve on the critical path method and develop specific agile applications.

Keywords: agile, software, cpm, production management

Introduction

Traditional project management processes aimed to monitor and control cost, schedule, and performance are based on rigid estimating techniques applied to a single baseline early in a project life cycle. An agile approach allows for rolling-wave planning and execution, resulting in decreased variability and improved forecasting. This paper explores how agile principles of iterative and incremental planning and execution applied to traditional monitoring techniques for cost and schedule, such as Critical Path Method (CPM) scheduling can benefit from an iterative, agile approach. When projects are treated as a system, planning and execution are not siloed to respective parties; production design and processes are better understood, leading to improved capacity-based planning, reduced variability, reduced work in process, and increased throughput. ISE and agile share many common principles and methodologies. Applying these to project management results in improved monitoring and control.

Research Issue Statement

How can projects use CPM in an agile project to create a detailed enough plan for project managers? Forecasting is only as good as the data you put in. However, if projects adopt an agile approach to CPM, projects can leverage the benefits of iterative, incremental develop to forecast progress and completion. Many traditional project management organizations that adopt agile try to impose traditional methods of control (CPM, Earned Value Management [EVM], etc.). They can be applied, but they must account for the iterative approach of agile. Applying an agile approach to CPM could improve on the method and provide agile project managers another means of measuring and presenting schedule data.

Research Results Statement

A deeper understanding of the process design, production design, capacity planning, inventory control, and variability is needed in order to improve CPM. Production process and design should be used to develop the scheduling model accounting for capacity of resources (people, equipment, etc.). Capacity-based planning and velocity-based planning need to be the drivers behind the scheduling duration estimates (rates drive dates). By adopting agile, projects can adapt the CPM approach to account for rolling-wave planning, iterative development, and incremental changes, resulting in improved project performance.

CPM Background

The critical path method (CPM) is defined as “a method used to estimate the minimum project direction and determine the amount of schedule flexibility on the logical network paths within the schedule model” (Project Management Institute [PMI], 2017b). CPM uses a schedule network analysis technique, independent of resources, to calculate the estimated project duration based on early start/finish and late start/finish activity dates. A forward and backward pass is conducted to determine the earliest start/finish and late start/finish based on the project activity estimated durations and sequencing. By creating a network diagram of activities based on predecessor and successor requirements and estimated task duration, the critical path and float (flexibility) of a project can be estimated (PMI, 2017a).

Schedule Logic and Precedence Diagraming

There are four types of logical relationships and dependencies in CPM that affect the project schedule duration. The following definitions summarize the relationships (PMI, 2017a):

- **Finish-to-start (FS):** predecessor activity must finish before successor activity can start
- **Finish-to-finish (FF):** successor activity cannot finish until a predecessor activity has also finished
- **Start-to-start (SS):** successor activity cannot start until predecessor activity has also started
- **Start-to-finish (SF):** successor activity must start before predecessor activity can finish

In a schedule network diagram, an activity node will contain the activity name, early start, early finish, late start, late finish, total float, and duration (see Figure 1).

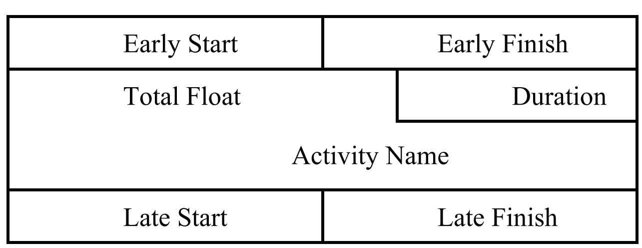


Figure 1. Activity Node in CPM Network Diagram

Float

Float, or slack, is the amount of time an activity can be delayed without an impact on the downstream activities within a schedule. There are two types of float: total float and free float. Total float is the "amount of time that a schedule activity can be delayed or extended from its early start date without delaying the project finish date or violating a schedule constraint" (PMI, 2017a). Free float is the "amount of time a schedule activity can be delayed without delaying the early start date of any successor or violating a schedule constraint" (PMI, 2017a). Since the critical path is the longest path through a project schedule, the critical path will have zero float.

Limitations of CPM Methods

CPM activity durations and network diagrams are based on estimates to create a baseline schedule to monitor project performance against. Ideally, this should allow project managers the opportunity to control the project performance by allocating additional resources and crashing or fast-tracking activities to complete a project on time. However, in practice this is very difficult to achieve. CPM has the following known limitations (Fischer et al., 2020):

- There is an assumption that durations can be accurately estimated and forecast. Estimates are based on historical data and made by separate planning departments, not the doers—the people closest to the work (Habibi et al., 2018).
- Interdependencies are not always clear up front; they evolve as more information becomes available and work is further broken down, making it difficult to differentiate between scope creep and scope refinement. This can result in lengthy and costly contractual battles between the contractor and developer (Quah & Prabhakar, 2008).
- Schedule logic in CPM is largely driven by finish-to-start dependencies, making other types of dependencies difficult to forecast and the critical path difficult to identify without software. Software requires user training/results in a designated "scheduler," removed from the work (Ouelhadj & Petrovic, 2009).

Management assumes that projects can add resources to speed up completion, failing to account for learning curves, increased complexity, and additional bottlenecks (Brooks, 1975).

Approach

By applying agile principles to the CPM approach, a modified approach to CPM can be used for agile software projects. However, this approach may not be appropriate for all projects; there is a continued need to have integrated schedules and milestones within government projects. Often, an agile software project only represents one piece of an overall system of systems, making it critical for agile software projects to have a method or approach to address CPM.

Agile is not a process but rather an umbrella term used to refer to several methodologies that emphasize the four agile methods and 12 principles of the Agile Manifesto developed in the early 2000s. Agile focuses on iterative and incremental development and improvements by embracing change as new information becomes available (Agile Alliance, n.d.). Traditional change management approaches are too slow and do not allow projects to respond to change rapidly.

Agile also promotes collaboration and self-organization. This means that individuals are empowered to self-organize and take ownership of their work. Decision-making is delegated to the lowest level of authority possible to enable the people doing the work to make more realistic estimates about the time and resources required to accomplish the work. To apply agile methods to project controls, CPM has to account for rolling-wave and capacity-based planning, iterative development cycles, and incremental changes.

Production Management Concepts

All CPM schedules start with a basic list of tasks, predecessors, and durations. A network diagram to capture the overall flow of work is used to demonstrate the process from the end to end. The limitations of CPM, as listed in Limitations of CPM Methods, create the need for deeper understanding of the work that is planned and estimated to be accomplished. In order to achieve this level of detailed planning in an iterative, agile approach, production management concepts such as the production design, process design, capacity planning, inventory or work in process (WIP) control, and variability need to be taken into account during CPM planning (Project Production Institute [PPI], n.d.).

Production and process design are closely related. *Production design* is a compilation of the requirements that will be used to design and build a product (PPI, n.d.). In the case of an agile software project, the requirements are often user-based and developed in an iterative manner over time, incorporating user and stakeholder feedback throughout the design and development process. *Process design* is the design of the processes and procedures to accomplish the work within the project (PPI, n.d.). In the case of agile software projects, the agile ceremonies and cadences of sprint planning, execution, demo/review, and retrospective are examples of processes that support the development of a product. You cannot fast-track or shortcut these processes to deliver a product faster; they are necessary for achieving the requirements in a cost-effective and iterative manner. In the case of CPM, production and process design impact the overall workflow and network diagram used to build the schedule. An understanding of both the product requirements and the process requirements is needed to capture all the applicable work within a timebox, such as a sprint.

Capacity planning is the approach of developing sprint plans based on resource availability (PPI, n.d.). It requires estimation and assignment of work based on the inputs available (data, models, etc.), the number of hours available by each team member, and the complexity/scope of the work to be accomplished. Work is estimated and planned to the appropriate scope within the sprint based on the team’s capacity (Cohn, 2005). In the case of CPM, capacity drives the duration estimates, accounting for resource hours available, dependencies of the work within the network, and possible blockers (such as external reviews).

By adopting agile principles, project managers can redesign the CPM approach to account for rolling-wave planning, iterative development, and incremental changes, resulting in improved project performance.

Rolling-Wave and Capacity-Based Planning

Rolling-wave planning is an iterative and incremental approach to project cost and schedule estimation. The estimate is done progressively over time, refined through rolling-wave planning methods that deliver cost and schedule estimates at the task level just in time. As more information becomes available and plans become more detailed, the estimate is updated and refined (Slinger, 2012). Capacity-based planning is the delegation of task estimation to the individuals conducting the work. This allows for a more detailed estimate based on the individuals’ hours available, skill sets, and personal understanding of the work complexity.

Story point analysis involves assigning story points, a unit of measurement for expressing a software development effort, a point value for the effort to complete the item. The individual values are not important. However, the relative value that the software development team assigns to the story points reflects the effort of the tasks involved. Something that is assigned a 2 should be roughly twice as much effort as something assigned a 1 (Cohn, 2005).

There are many agile estimation techniques, such as story point estimating, that allow for task estimation to account for not only the estimate hours, but also the complexity and risk associated with the task requirements (Agile Alliance, n.d.). For example, if a task is not well defined and an individual does not have the experience associated with the task, the estimate would be high, indicating to the project manager that there is an immediate need to spend more time on the task definition. Before the work even starts (resources, cost, etc.), the project manager can control the project direction by addressing the issues in real time versus finding out about it later in a monthly CPM report.

Iterative Development Cycles and Incremental Changes

Iterative development cycles and incremental changes are similar to Plan-Do-Check-Act (PDCA) cycles. An example of an agile application of PDCA is through the scrum sprint cycles (see Figure 2). A sprint is a timeboxed period to do work, usually between 2 to 4 weeks long. A sprint starts with a planning session focused on near-term and priority work from the project backlog. Capacity-based estimates are used to determine what resources are available and how much work to plan into the sprint cycle based on the hours and skills available. The team then commits to a set amount of work for the sprint and works down the sprint backlog until the sprint ends. Even if all the work is not completed, the sprint will end on a set date. At the end of the sprint, the team reviews their work completed with their stakeholders, providing an opportunity to either continue forward or make an incremental change. This allows for rapid, responsive changes before too many resources and too much time have been committed to a flawed or problematic solution. It also provides an opportunity for stakeholders to clarify their requirements as more information becomes known, allowing for a collaborative solution to be developed (Agile Alliance, n.d.).

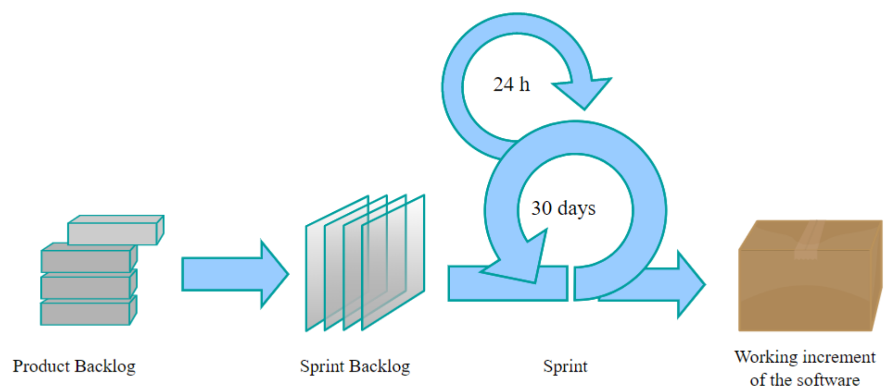


Figure 2. Scrum Sprint Cycle

Work in Process, Inventory/Backlog, and Resource Controls

Planning work in sprints allows projects to better manage and control WIP, inventory, and resources. Large amounts of WIP leads to less throughput, as too many tasks are started but not necessarily finished. This commits resources, time, and money unnecessarily and inefficiently, which increases project variability. As project variability increases, monitoring and control methods lose their effectiveness to control, becoming historical snapshots (backward-facing), instead of forward-looking. In addition, large amounts of inventory also tie up resources and drives up project costs, as inventory requires space and resources to store, track, and manage. Finally, since agile promotes rolling-wave and capacity-based planning, work is committed based on resource availability, skill set, and capacity. Resource surges may still be needed on occasion, but they are the last resort, relying instead on the people doing the work to only commit to what they are capable of achieving within the sprint.

Agile Approach for CPM

An agile approach for CPM also begins with rolling-wave planning. It is unlikely that most projects can produce a detailed level sequence of activities months and years in advance. However, creating the network diagram during a sprint timebox will generate less variability because resources commit to work based on their capacity and skill set. Sequencing of activities also improves because the inputs/outputs of near-term work must be well understood to estimate and commit to a sprint. If there is uncertainty related to a specific activity, then it is raised to the attention of the project manager during planning, which can then be communicated back to the stakeholders. This iterative approach to problem solving continues until the team reaches an understanding of what the requirements are and can commit to a path forward.

The critical path can be determined at the sprint and project level but is tracked differently. At the sprint level, the estimated duration can be inferred from the story point estimates for each task. The earliest start/finish and late start/finish are calculated the same way but require that tasks within a sprint have a network diagram in place as part of planning. This means identifying the predecessor and successors of each task within a sprint. It does require a modification to traditional CPM, though, in that the resources and their capacity are directly tied to a task and its story point estimate, which will affect the network diagram development and requires review as part of the sprint planning session. Further, traditional scheduling tools, like integrated master schedules, are often outdated before they are even finished. A roadmap is an agile approach for long-term planning that captures high level milestones and project targets but is constantly updated from sprint to sprint based on how the work is progressing and based on feedback from stakeholders.

Application to Software Projects

Software Development Tools That Support Agile CPM

There are many software tools available that can provide an agile CPM approach for software development projects. Listing of these tools is not meant as an endorsement but rather a demonstration of how agile CPM can achieve meeting the needs of both the project team and stakeholders. According to the *14th Annual State of Agile Report*, Jira remains the most popular agile project management tool across industry (Digital.ai Software Inc., 2020). Jira allows projects to plan, organize, and track work throughout the life cycle of project. Work breakdown and dependencies are captured and managed in real time within the Jira environment, allowing the team to see what is happening in a connected (versus disconnected) state (Atlassian, n.d.). This is a significant improvement over previous project management tools such as Microsoft Excel or Project.

Further, Jira plugins such as SoftwarePlant's (2021) BigPicture and ALM Works's (n.d.) Structure both allow software development projects to pull project tasking into a Gantt-style chart/network diagram view. An agile work breakdown structure provides the basis for organizing and breaking the work down. Tasks can be linked in various relationships—such as end-to-end, end-to-start, start-to-start, and start-to-end—similar to Microsoft Project or Primavera scheduling software. Milestones can be added, and work can be connected to these dates, allowing project managers to pull reports on the critical path. These data are captured and updated in real time as teams work within the system to update their tasking, eliminating the disconnect of having separate planners and doers.

Applications Within Government Software Projects

According to the latest *Defense Acquisitions Annual Assessment*, while 22 of the DoD's weapons programs reported agile development as their software project delivery approach, there were inconsistencies across the programs in terms of execution and delivery (GAO, 2020). Out of the 22 programs, only six met the industry standard for agile delivery practices, which is delivery every 6 weeks or less (Freedberg, 2020). While this seems discouraging, it is important to note that the adoption of agile within government projects is still relatively new, and true agile transformation of a project or program can take years.

As the DoD is transitioning to more agile practices, there are multiple programs and projects that are adopting agile practices and software successfully. For example, Joint Space Operations Center (JSpOC) Mission System (JMS), Air Force; Distributed Common Ground System–Navy (DCGS-N) Increment 2, Navy; Global Combat Support System–Joint (GCSS-J), Defense Information Systems Agency (DISA); and Catapult/Attack the Network Tool Suite (ANTS), Joint Improvised-Threat Defeat Organization (JIDO) programs have all adopted Jira project management software to support iterative and incremental development practices (Kramer & Wagner, 2019). While it is unknown what, if any, Jira plugins are being used to track the critical path, the adoption and native functions within the basic version of Jira support the overall goals and objectives of CPM by allowing task tracking in an integrated and networked fashion.

Conclusion

Agile methods allow projects to adapt to rapid and constant change. Traditional methods for project monitoring and control are based on theories and assumptions that change and variability are rare in projects, which results in project systems that can only monitor the past, not control the present or forecast the future. By applying agile methods—such as rolling-wave and capacity-based planning and iterative development cycles and incremental changes—and by focusing on controlling WIP, inventory, and resources, CPM can be improved to provide control, not just monitoring.

In an agile construct, the intent of CPM—tracking the critical path of a project—can be done within native software project management tracking tools, such as Jira. Further adoption of Jira and Jira plugins, such as BigPicture and Structure, could allow government projects to capture the detailed level schedule and critical path in real time, eliminating or greatly reducing the barriers in place today with traditional CPM approaches and tools. After all, agile and CPM share a common goal of delivering a product or project with transparency of the time/cost throughout the life cycle.

References

Agile Alliance. (n.d.). *Agile 101*. Retrieved 2018, from https://www.agilealliance.org

Atlassian. (n.d.). *Jira software*. Retrieved April 2021, from https://www.atlassian.com/software/jira?&aceid=&adposition=&adgroup=56999361780&campaign=1439934479&creative=429724101024&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p34164036185&ds_e=GOOGLE&ds_eid=700000001558501&ds_e1=GOOGLE&gclid=Cjw

ALM Works. (n.d.). *Structure: Project management at scale*. Atlassian Marketplace. Retrieved April 2021, from https://marketplace.atlassian.com/apps/34717/structure-project-management-at-scale?hosting=cloud&tab=overview

Brooks, P. F. (1975). *The mythical man-month.* Addison-Wesley Publishing Company.

Cohn, M. (2005). *Agile estimating and planning.* Pearson Education.

Digital.ai Software Inc. (2020). *14th annual state of agile report.* State of Agile. https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494

GAO. (2020, June). *Defense acquisitions annual assessment: Drive to deliver capabilities faster increases importance of program knowledge and consistent data for oversight.* https://www.gao.gov/assets/710/707359.pdf

Habibi, F., Birgani, O. T., Koppelaar, H., & Radenovic, S. (2018). Using fuzzy logic to improve the project time and cost estimation based on project evaluation. *Journal of Project Management*, 183–196.

Fischer, G., Ballard, H. G., Spearman, M., & Parsons, M. (2020, October 21). *Does CPM scheduling work?* Project Production Institute. https://projectproduction.org/events/does-cpm-scheduling-work/

Freedberg, S. J., Jr. (2020, June 12). *DoD "agile" software development still too slow: GAO*. Breaking Defense. https://breakingdefense.com/2020/06/dod-agile-software-development-still-too-slow-gao/

Kramer, J. D., & Wagner, L. J. (2019). Developmental test and requirements: Best practices of successful information systems using agile methods. *Defense Acquisition Research Journal*, *86*. https://www.dau.edu/library/arj/ARJ/ARJ88/ARJ-88%20_18-804%20Kramer.pdf

Ouelhadj, D., & Petrovic, S. (2009). Survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417.

Project Management Institute. (2017a). *A guide to the project management body of knowledge (PMBOK Guide)* (6th ed.). Project Management Institute.

Project Management Institute. (2017b). *PMI lexicon of project management terms v3.2*. https://www.pmi.org/pmbok-guide-standards/lexicon

Project Production Institute. (n.d.). *Project Production Institute*. Retrieved September 2020, from https://projectproduction.org/

Quah, J., & Prabhakar, G. P. (2008). Scope creep in software development. *Journal of Social Management*, *6*.

Slinger, M. (2012). *Agile estimation techniques* [Presentation]. Project Management Institute Global Congress, Vancouver, Canada.

SoftwarePlant. (2021, April). *BigPicture: Project management & PPM*. Atlassian Marketplace. https://marketplace.atlassian.com/apps/1212259/bigpicture-project-management-ppm?hosting=cloud&tab=overview

PANEL 23. TOOLS AND STRATEGIES FOR ACQUIRING INFORMATION TECHNOLOGY

Thursday, May 13, 2021	
12:15 p.m. – 1:30 p.m.	**Chair: Reuben Pitts,** President, Lyceum Consulting, LLC ***Automated Comparison of Software Versions*** Bruce Allen, Naval Postgraduate School Neil Rowe, Naval Postgraduate School ***Technology Trust: The Impact of Trust Metrics on the Adoption of Autonomous Systems used in High-Risk Applications*** Michael Anderson, Naval Postgraduate School Johnathan Mun, Naval Postgraduate School ***Applying Agile Principles Beyond Information Technology and Computer Software*** Terrence Leary, MITRE Corporation Virginia Wydler, MITRE Corporation ***A Comparative Analysis of Advanced Methodologies to Improve the Acquisition of Information Technology in the Department of Defense for Optimal Risk Mitigation and Decision Support Systems to Avoid*** Thomas Housel, Naval Postgraduate School Timothy Shives, Naval Postgraduate School Johnathan Mun, Naval Postgraduate School Raymond Jones, Naval Postgraduate School Ben Carlton, Marine Corps Systems Command

Reuben Pitts—is the president of Lyceum Consulting. He joined the Naval Weapons Lab in Dahlgren, VA, in June 1968 after graduating from Mississippi State University with a BSME. His early career was spent in ordnance design and weapons systems. He subsequently served on the planning team to reintroduce the Navy to Wallops Island, VA, currently a multiple ship combat, over-the-water weapons testing lab for Surface Ship Combat Systems, Fighter Aircraft, and live missile firings. His outstanding service as the deployed science advisor to commander, U.S. Sixth Fleet, was recognized with the Navy's Superior Civilian Service (NSCS) Award and the Navy Science Assistance Program Science Advisor of the Year Award.

Pitts was selected to lead the technical analysis team in support of the formal JAG investigation of the downing of Iran Air Flight 655 by USS Vincennes, and participated in subsequent briefings to CENTCOM, the chairman of the joint chiefs, and the secretary of defense. As head, Surface Ship Program Office and Aegis program manager, Pitts was awarded a second NSCS, the James Colvard Award, and the John Adolphus Dahlgren Award (Dahlgren's highest honor) for his achievements in the fields of science, engineering, and management. Anticipating the future course of combatant surface ships, Pitts co-founded the NSWCDD Advanced Computing Technology effort, which eventually became the Aegis/DARPA-sponsored High Performance Distributed Computing Program, the world's most advanced distributed real-time computing technology effort. That effort was the foundation for the Navy's current Open Architecture Initiative. In 2003, Pitts accepted responsibility as technical director for PEO Integrated Warfare Systems (IWS), the overall technical authority for the PEO. In September of that year, he was reassigned as the major program manager for Integrated Combat Systems in the PEO. In this position, he was the program manager for the Combat Systems and Training Systems for all U.S. Navy Surface Combatants, including aircraft carriers, cruisers, destroyers, frigates, amphibious ships, and auxiliaries. In July 2006, Pitts returned to NSWCDD to form and head the Warfare Systems Department. While in this position, he maintained his personal technical involvement as the certification official for Surface Navy Combat Systems. He also served as chair of the Combat System Configuration Control Board and chair of the Mission Readiness Review for Operation Burnt Frost, the killing of inoperative satellite USA 193.

Pitts has been a guest speaker/lecturer/symposium panelist at many NAVSEA-level and DoD symposiums and conferences and at the Naval Postgraduate School, the Defense Systems Management College, and the National Defense University. For 19 years, Pitts was the sole certification authority of all Aegis Combat System computer programs for fleet use. He retired from the U.S. Civil Service in September 2008, with over 40 years of service to the Navy.

Using Texture Vector Analysis to Measure Computer and Device File Similarity

Bruce Allen—is a Faculty Associate at the Naval Postgraduate School where he performs research and supports students in various topics relating to computer science. His specialties include data analytics, robotics, cyber security, digital forensics, and GUI design. [bdallen@nps.edu]

Abstract

Executable programs run on computers and digital devices. These programs are pre-installed by the device vendor or are downloaded or copied from a storage media. It is useful to study file similarity between executable files to verify valid updates, identify potential copyright infringement, identify malware, and detect other abuse of purchased software.

An alternative to relying on simplistic methods of file comparison, such as comparing their hash codes to see if they are identical, is to identify the "texture" of files and then assess its similarity between files. To test this idea, we experimented with a sample of 23 Windows executable file families and 1,386 files. We identify points of similarity between files by comparing sections of data in their standard deviations, means, modes, mode counts, and entropies. When vectors are sufficiently similar, we calculate the offsets (shifts) between the sections to get them to align. Using analysis on these shifts, we can measure file similarity efficiently. By plotting similarity vs. time, we track the progression of similarity between files.

Introduction

Software of unknown pedigree abounds. This is partly due to software being distributed as executable code or a "binary," and evaluating the contents of a binary is technically challenging.

Numerous updates to a binary can occur over the useful life of the executable to address new software requirements, fix software defects, or port the software to a different computing platform. Each of these requires recompilation and results in a new binary.

Executable code can be analyzed using reverse-engineering tools that recover information about the binary's structure, function, and behavior. Some tools recognize data regions inside the code, while more advanced tools analyze the machine instructions to make inferences about the code's function. Because of the differences in instruction set architectures (ISAs), tools use models of ISAs. However, reverse engineering of a binary can be resource-intensive and can be stymied by deliberate anti-reversing techniques used to protect the binary file.

Executable code is vulnerable to malware. By replacing machine instructions with malicious ones, executable code can be transformed into malware. Malware can divert execution of code to perform one or more malicious tasks. Detection of malware contained in adversarial malware binaries is technically challenging, even with the use of artificial-intelligence techniques such as deep learning (Kolosnjaji et al., 2018).

We introduce here an approach based on texture vectors to allow executables to be compared against each other without requiring reverse engineering of the binaries. Our approach can be used as a first step to determine whether reverse engineering is needed.

Background

Contents of an Executable File

A binary contains more than just executable code. It includes fixed data, reserved space, and links to executable code that is external to the file (Josse et al., 2014). Similarities in fixed data and fixed links are easiest to find because they can be matched directly. Reserved space usually consists of bytes with zero values and is found in many places in a typical executable file. It can complicate similarity measurements since there can be many false matches with zero bytes.

The portion of an executable file that contains the actual executable code consists of machine instructions and their associated operands. When executable code is modified, many machine instructions remain the same but usually their locations shift. Then the memory addresses encoded in their operands may change to compensate for this shift unless the code uses addressing relative to a register. However, register arguments encoded in operands may also shift.

Identifying File Similarity

Numerous approaches exist for identifying similarities between files. They can be used on text files, binary files, images, video, and audio. A few apply to files containing executable code. Some of these executable-analysis tools visualize software evolution in source code using version-control information or source-code file analysis (Arbuckle, 2008). A three-dimensional graph can show where code accesses the operating system or other information about code flow, and graph how these numbers change over the evolution of a software product. The Code Time Machine tool (Aghajani et al., 2017) does this to show the evolution of code metrics for a given file. It shows values along a time-line for the number of lines of code, number of methods, and cyclomatic complexity (i.e., the number of paths the code can take given the possible conditions written into the code). A three-dimensional graph of files and file relations between versions relates files. Circles represent releases, squares represent files, and edges represent associations (Koike & Chu, 1997). Other tools that graph code evolution are CVSScan (Voinea et al., 2005) and EPOSee (Burch et al., 2005).

There are many types of files. Three important ones are:

- **Text:** Text typically consists of words arranged in sentences. It may also be fragmented because of formatting, as in a formatted PDF file, or may be in short phrases in data tables or in the data section of executable code. We can measure text similarity by comparing words.
- **Arbitrary bytes:** What may appear as arbitrary bytes may be numeric data, compressed data, or executable codes. Numeric data often has low entropy because many of the bytes tend to be zero. Compressed data has high entropy because unused byte patterns in the data are removed.
- **Audio and video:** Audio and video data consists of bytes arranged in sequences. Bytes can be compared by aligning the sequences.

There are many algorithms for identifying similarities in data. Some work better than others given the type of data being compared. Methods used in comparing files are:

- **Comparing byte sequences:** Comparing content of text files to identify similarity is a common operation. One approach tries to find the longest common subsequences (Bergroth & Hakonen, 2000), where text that does not match is identified as new or deleted content. Algorithms for efficient string matching include Knuth-Morris-Pratt (Wikipedia, 2020), which allows searching without backtracking when a near match is found, and Boyer-Moore (Cole, 2018), which skips alignments when searching for specific text. Although intended for text, both algorithms may be used with executable code. Another popular algorithm for text files is implemented in the “diff” utility developed for the Linux operating system (Shotts, 2019).

- **Comparing executable bytes:** Comparing bytes in files is similar to comparing text in files. However, bytes of files containing executable code are unlikely to match on operands and thus only the operators should be compared. For Intel architectures, operands are usually spaced at 4-byte or 8-byte intervals. Because of this, Cabezas and Mooij (n.d.) says that "binary file analysis by both binary diffing and cryptographic hash signatures comparison is a very limited approach to identify source code being re-used" and suggests metadata analysis. Regardless, for Intel architectures, it is useful to compare at every fourth or eighth byte because this will often align runs of comparisons with operators.
- **Comparing histograms:** We can identify common sequences of N bytes (N-grams) between files and contiguous sequences of bytes of a given length and compute a histogram of them. In Jang and Brumley (2009), 5-grams are used, and a Bloom filter is used as an efficient data structure for storing N-gram patterns that are found. Overall file similarity can be measured with the Jaccard index, the count of N-grams in common divided by number of distinct N-grams in both files. To take frequencies of the N-grams into account in measuring similarity, the cosine similarity or the Kullback-Leibler divergence can be used (Rowe, 2018).
- **Transforming values before comparison:** It may work better to measure similarity on transforms of the data values. This is commonly done for audio and video data; perceptual hashing (Hadmi et al., 2012) provides a similar hash output if features are similar. For images, we can transform the image to frequency space, or apply convolutions to it to enhance features. For signals, we can apply the Fourier transform to obtain frequencies.
- **Comparing metadata:** Initial comparisons of files can use their descriptive data to decide if they are sufficiently related to be worth further analysis. For example, if we know two executable files are built to run on a Microsoft Windows system using the same Intel instruction-set architecture, they are worth comparing. We can also compare metadata about sections and data structures within the files (Cabezas & Mooij, n.d.). Metadata includes:
    - File types and subtypes.
    - Data compression parameters. Cloning is indicated if the compressed size is significantly smaller than the combined size of its parts (Cabezas & Mooij, n.d.).
    - Mentions of precompiled libraries.
    - Hashcodes on the files.
- **Comparing decompiled data:** Executable files can be decompiled into text, and we can compare this text. Disassemblers and decompilers can do this, though they are not perfect. Disassemblers turn the bytes of executable code into corresponding machine code mnemonics and symbolic names, addresses, and offsets. Decompilers go further by turning bytes into source code.

Identifying similarity specifically between versions of source code can be accomplished in several ways:

- **Object-oriented analysis:** Software objects in source-code versions can be visually compared using a difference graph (Seemann & von Gudenberg, 1998). A graph of each version can be created where nodes are classes and node attributes are class methods and variables. Edges connect nodes where attributes of one node reference attributes of another. Then a class relation diagram is constructed that highlights differences in class relations in two software versions.

- **Software-diagram analysis:** Software diagrams created during design may be compared if available (Rho & Wu, 1988).
- **Version control analysis:** Many products used by the software industry manage source code versioning with a repository (Swierstra & Lh, 2014). Then there is often documentation of the differences between versions.

Calculating File Similarity

In this section we present our texture-vector approach. We perform three layers of calculations to make inferences about similarity and how and where files are similar. Our steps are:

1. Calculate texture-vector datasets from the two files to be compared.
2. Compare texture-vector datasets to identify similarity offsets and produce a similarity offset histogram.
3. Calculate statistics from the heights of the similarity offset histogram to produce a single similarity measure for the comparison of the two files.

This process is illustrated in Figure 1.

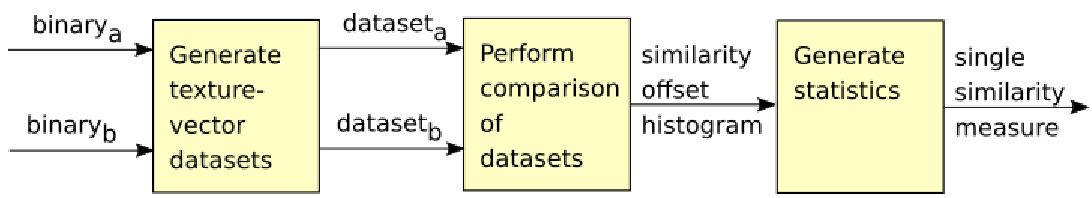


Figure 1: Inference Process

Calculating Texture-Vector Data

Texture vectors are calculated from the byte values of contiguous sections of binary data. Although many transform algorithms are possible, we are specifically interested in transforms that can both represent some unique characteristic of the data and possess a value that can be meaningfully compared to other values to measure similarity.

Sections measured as similar by many transforms have stronger similarities than others. We tested the following transforms for calculating texture vectors on the integer values of the bytes:

- **Standard Deviation:** The standard deviation of the byte values in a section of binary data. Two sections with a similar amount of deviation may be similar.
- **Mean:** The average byte value in the section. When executable code changes, operators may remain the same and help maintain the same mean.
- **Mode:** The most frequent byte value in the section. Often this value was zero in our data. This value is nonmetric and can only be used in computing similarity distances in the sense that it is identical or not.
- **Mode Count:** The count of occurrence of the most frequent byte value in the section.
- **Entropy:** The Shannon entropy of the byte values in the section. Two sections may be similar if the amount of randomness in each section is similar.

Calculating Texture-Vector Distance

Two texture vectors are defined as similar when the first texture-vector is within a threshold of closeness to the second texture vector by the weighted square of the L2 (Euclidean) distance metric (Defant , 2011). The similarity can be thought of as 1/d2 where d is distance, calculated as: d = w1(dv1)2 + w2(dv2)2 + w3(dv3)2 + w4(dv4)2 + w5(dv5)2 where dv is the difference at a given vector element and w is the weight for a given vector element. For example if texture-vector 1 has values [100, 30, 220, 50, 80], texture-vector 2 has values [101, 32,225, 51, 80], and weights [w1, w2, w3, w4, w5] are [0.25, 0.25, 0.0, 0.25, 0.25], then the L2 distance d2 is 0.25 ∗ 12 + 0.25 ∗ 22 + 0.0 ∗ 52 + 0.25 ∗ 12 + 0.25 ∗ 02 = 0.25 + 1.0 + 0 + 0.25 + 0 = 1.5. A threshold of similarity was used for our graphics; for instance, if the acceptance threshold is 1.0, these vectors are not similar because 1.5 $ 1.0. We set weight values by experiment as explained in the Tuning Rejection Thresholds section. A good threshold identifies numerous correct similarities between the sections of data from which the texture vectors were calculated while excluding non-similarities.

Calculating Similarity Off Between Sections

We calculate similarity offsets by comparing all the texture vectors in one file against all the texture vectors in another file and counting the offsets between the files where the texture vector distance is within the threshold of closeness. When there are many offsets with the same value, this gives high confidence in those byte matches.

We implemented a display to show consistently strong offsets between two files. The display draws lines connecting similar texture vectors. The pattern and quantity of similarity lines indicates the nature and degree of file similarity. Figure 2 shows an example of two very similar versions of executable code, where the texture vector pattern of each file is shown across the top and bottom, and the lines between them indicate points of similarity. The files are both roughly 220 KB in length.

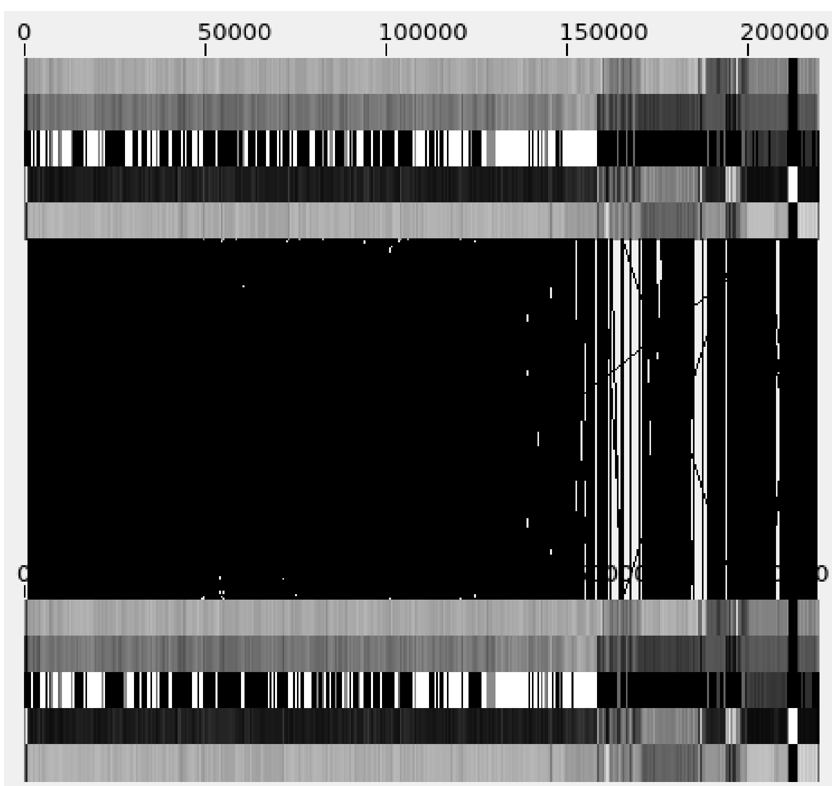


Figure 2. Texture Patterns of Two Very Similar Executable Files and Lines Connecting

Them Indicating Points of Similarity

Calculating Similar-Section Off Histograms

We calculate a similarity offset histogram from the set of offsets identified when searching for sufficiently similar texture vectors. There can be many thousands of offset values where similar-section matches can occur. To quantify this distribution of offsets, we create a similarity offset histogram and distribute calculated offset values across approximately 400 buckets, which sufficiently categorizes offsets in a viewable form. Consistent offset values are found as peaks on the histogram of offset values and represent likely meaningful similarities.

We calculate the measure of similarity between two files from the heights in the similarity offset histogram to provide a numeric measure of similarity between files. A large spread in heights suggests similarity at specific offsets, indicating similarity, while minimal spread in heights suggests a random distribution of similarity offsets, likely a result of false positives.

Calculating Similarity Measures Between Files

We calculate the measure of similarity between two files from the magnitude of the standard deviation of the heights of the compensated histogram as described in the Calculating Similar-Section Off Histograms section. An example of calculated similarity measure, along with the texture vectors, similarity offsets, and similar-section offset histograms, is shown in Figure 3. The top part describes the files being compared, the weights used in calculating the texture-vector distance, and statistics about the view, including the calculated similarity measure of 334.3535. The middle part shows the two texture-vector patterns, which visually appear identical, along with the center region saturated black with similarity lines. The bottom part shows the similarity histograms, where the similar-section offset histograms have spikes and low points. We will conclude that these two files are nearly identical in the Results section.

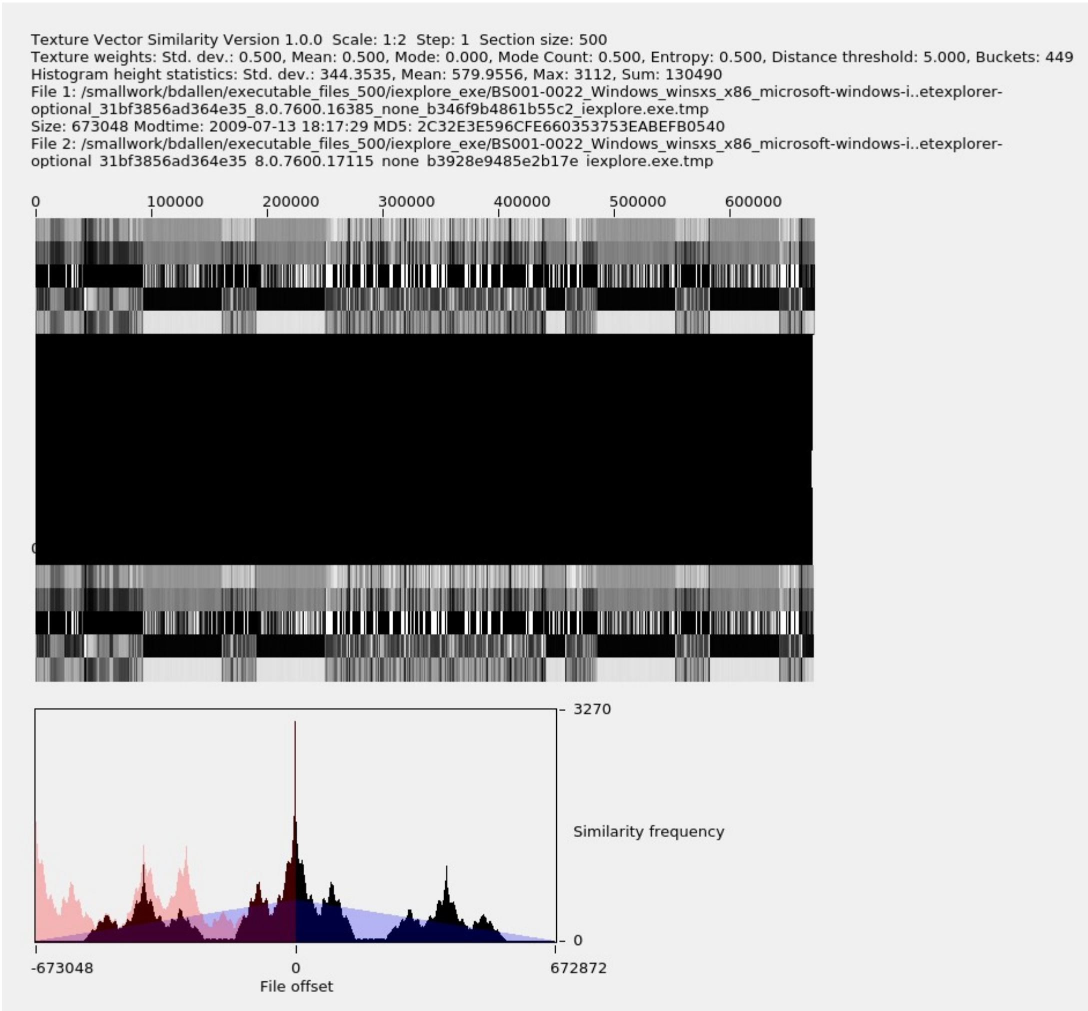


Figure 3. Example of High Value of High Similarity in File Family iexplore_exe

Tracking Versions of Executable Code

We can also graph a network of relationships between different versions of the same executable. By using the file modification time for the horizontal axis and the calculated similarity measure described in the Calculating Similarity Measures Between Files section as the vertical axis, we can show the relationships between versions. Files that have a larger similarity measure to the selected file are plotted higher on the vertical axis. Files whose similarity measure is below a user-selectable measure are not plotted. By adjusting the similarity threshold using the SD slider in the Texture-Vector Browser GUI tool, we can remove files with minimal similarity to reveal clusters of files that match with greater similarity. Using this graph, we can make inferences; for example, releases with a similar modification time may be a result of bug fixes or security updates; releases with a smaller similarity measure may have more functional differences or may have added malware. An example of this graph is shown in Figure 9.

Preparing the Dataset of Executable Files

The dataset we studied consisted of executable files, texture-vector files, and similarity-graph files. The initial set of files was a sample of executable .exe and .dll files extracted from the Real Data Corpus (Garfinkel et al., 2009). The Real Data Corpus consists of “images” (copies) of used disk drives and other devices obtained from non-U.S. countries. The files were extracted using the icat extraction tool from The Sleuth Kit forensics tool, (https://forensicswiki.org/wiki/The_Sleuth_Kit). Prof. Rowe picked 23 representative families of executables defined by a file name for each. Since many of the files were faulty, he used a software wrapper that loaded files for each distinct file contents (as indicated by its hash code) until the wrapper found a non-faulty copy. Names were changed from the original ones to distinguish files with the same names and different contents. The initial set consisted of 1,386 files. Of these, 162 were excluded because their size was greater than 1 MB and 55 were excluded because their size was less than 1 KB. Of the remaining 1,169 files, 35 were excluded because they were identical based on their MD5 cryptographic hash, leaving 1,134 files in our dataset. Figure 4 shows the distribution of file sizes. Note that since all files are from various countries and no files are from the United States, our collection may exclude important versions of software.

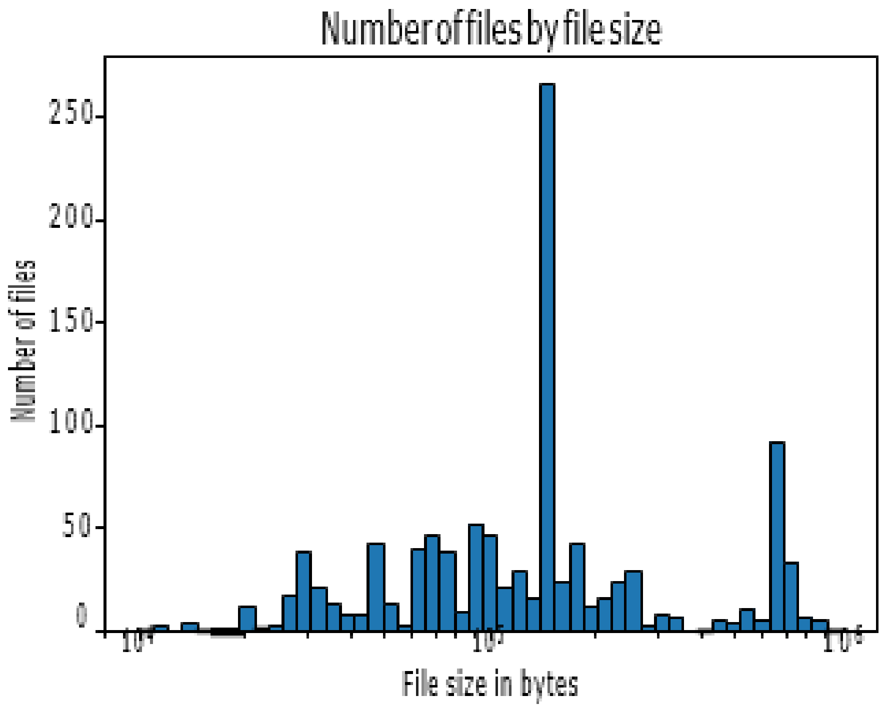


Figure 4. Histogram of File Sizes for our Dataset

The distribution of files by file modification time is shown in Figure 5.

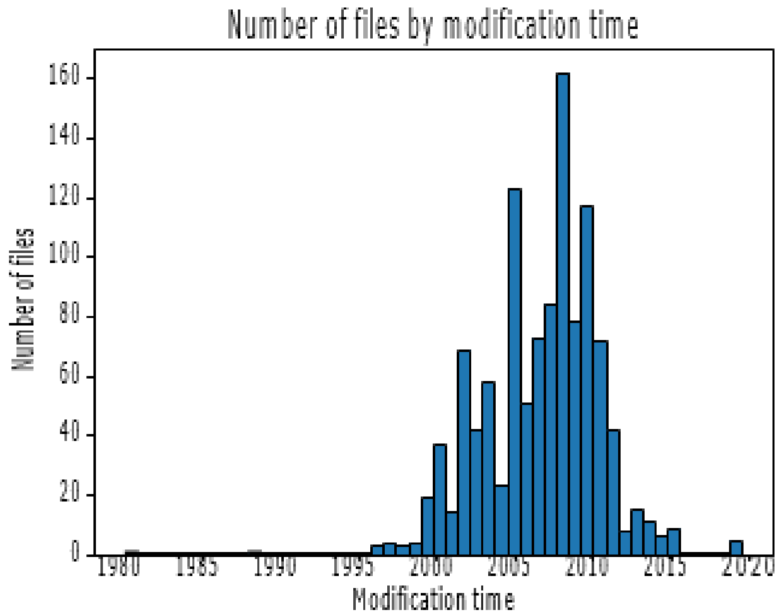


Figure 5. Histogram of File Modification Times for our Dataset

Statistics on the 23 file families that we studied are shown in Table 1. This includes source-code family tabulate_drive_data_py, which allows us to compare some versioned source-code files too.

Preparing the Texture-Vector Files

We created the texture-vector .tv files with the sbatch_calc_tv.bash program, which is part of the Texture-Vector Generator toolset. Due to the computational burden, we calculated texture vectors on the Naval Postgraduate School (NPS) Hamming supercomputer using sbatch parallel processing. We then copied these .tv files to the Texture-Vector Similarity repository, renaming them to their MD5 cryptographic hash value, for access by the Texture-Vector Similarity GUI tool.

Tuning Rejection Thresholds

Similarity is indicated when the square of the L2 distance measure is less than an acceptance threshold, as described in the Calculating Texture-Vector Distance section 3.1.1. We performed our tuning with two arbitrarily selected larger files in the ccalert_dll file family. We began with a default weight of 0.5 for the standard deviation, mean, mode count, and entropy transforms and, after some experimentation, we selected a distance rejection threshold of 5.0 because it resulted in reasonable similarity offsets without an oversaturation of matches. We selected a default weight of 0.0 for the mode because mode values do not quantifiably compare with each other, though an alternative could be to set distances between modes to 0 for identical values and 1 for nonidentical values.

Table 1. Files by File Family

File Family	File count	Min file size	Max file size	Mean file size	Standard deviation of file size
a0003775.dll	14	1591	853504	258271.6	318135.5
bthserv.dll	37	1067	92160	31455.4	19509.8
ccalert.dll	23	189560	267880	225524.2	21199.8
cdfview.dll	244	1178	409600	144513.2	39662.1
dunzip32.dll	34	11091	149040	114370.9	26991.3
hotfix.exe	33	53248	112912	94098.4	13263.9
iexplore.exe	216	3506	903168	461304.5	277712.7
mobsync.exe	80	8192	970752	156818.5	141438.6
msdc.dll	6	159232	194048	174983.3	15696.5
nvshu.dll	32	151552	262144	240128.0	33724.4
pacman.exe	2	165594	241693	203643.5	53810.1
policytool.exe	104	1224	787508	54764.8	84605.1
powerpnt.exe	19	2310	676112	366290.8	236454.6
rtinstaller32.exe	4	135168	158312	146740.0	9843.3
safrslv.dll	29	1582	65536	41681.3	12648.2
tabulate.drive.data.py	23	18647	47544	34090.3	7213.7
typeaheadfind.dll	2	35920	39856	37888.0	2783.2
udlaunch.exe	4	118784	118784	118784.0	0.0
vsplugin.dll	8	65606	118801	88180.2	15049.3
webclnt.dll	80	1261	611328	96930.6	92513.1
winprint.dll	7	12048	44544	29627.4	13120.4
wmplayer.exe	120	2864	520192	142871.3	101072.6
xrxwiadr.dll	13	8192	311296	123327.4	75040.4

We examined our tuning of weight values by setting all weight values to 0.0 and then, one weight at a time, examined the saturation of matched offsets as we adjusted the weight for each texture contribution from 0.0 to 1.0. For each weight adjustment, we observed that the quantity of similarity offsets identified would vary as we changed the weight and also that there was a visually understandable quantity of similarity at weight 0.5. Given this, we accepted our weight and rejection threshold values as our default values. These defaults are shown in Table 2.

Table 2. Default Texture-Vector Threshold Settings

Setting	Type	Value
Standard Deviation	Weight	0.5
Mean	Weight	0.5
Mode	Weight	0.0
Mode Count	Weight	0.5
Entropy	Weight	0.5
Rejection threshold	Threshold	5.0

Preparing the Similarity-Graph Files

We created the similarity-graph files by running the sbatch_ddiff_tv.bash program which is part of the Texture-Vector Similarity toolset. We calculated the similarity metrics on the NPS Hamming supercomputer using sbatch parallel processing with a job queue size of 700, resulting in a graph of 1,134 nodes and 463,486 edges from which we can create a similarity matrix across all file families. We compared files across file families in order to measure similarity between known dissimilar files. There are 642,411 possible edges, but we dropped 178,925 of them because they had less than two similarity matches. This processing took about 15 hours. Runtime of each file pair varied because file sizes varied.

Table 2. Default Texture-Vector Threshold Settings

Setting	Type	Value
Standard Deviation	Weight	0.5
Mean	Weight	0.5
Mode	Weight	0.0
Mode Count	Weight	0.5
Entropy	Weight	0.5
Rejection threshold	Threshold	5.0

Node data consists of the node index, filename, file family, file size, file-modification time, and file MD5 hashcode. Edge data consists of the edge's source and target file node indexes along with the standard deviation, mean, maximum, and sum similarity metrics described in the Calculating Similarity Measures Between Files section.

Results

To evaluate the ability of our tools to identify similarities between executable files, we examined the 642,411 texture-vector similarity measures calculated for each pair of files for the 1,134 files. Of the 642,411 possible comparisons, 463,486 of them produced nonzero similarity values. Similarity measure values varied from zero to about 300. The distribution of these 463,486 similarity values across all files in our dataset is shown in Figure 6. Due to the uneven distribution of these values, a similarity threshold cannot be calculated using a normal gaussian distribution. Most similarity measure values were less than 10, which is where the curve becomes level. This suggests that actual similarity between two files may be indicated when their similarity measure is greater than 10.

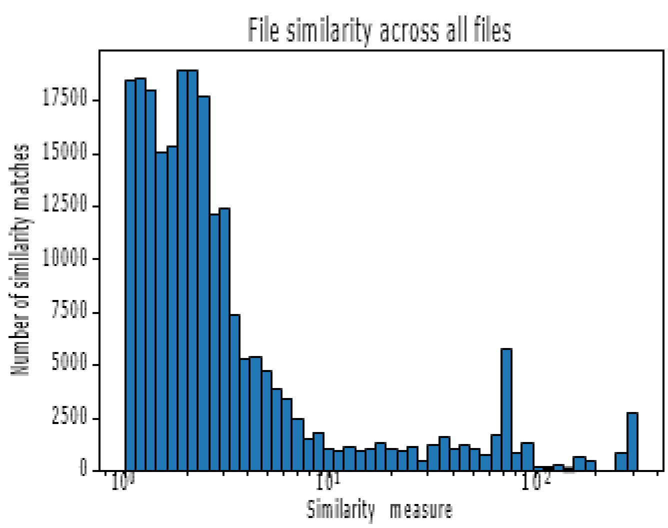


Figure 6. Histogram of Similarity Matches Across All Files in Our Dataset

Evaluating Similarities

To establish a baseline of what the similarity measure values are for similar files, we calculated the mean similarity measures within and across file families. We tested whether the similarity measure between files of the same file family was higher than the similarity measure between files in different file families. The confusion matrix for file similarity across all file families in our dataset is in Table 3. Rows and columns represent file families using the numbers in the second column. The mean similarity measures between files within file families is typically greater than the mean similarity between files in other file families, showing that our approach for identifying file similarity is useful.

Table 3. Mean File Similarity Between File Families

Family	No.	1	2	3	4	5	6	7	8	9	10	11	12
a0003775.dll	1	4.5	1.2	5.4	2.1	3.8	3.0	3.6	2.8	2.2	3.3	5.1	2.3
bthserv.dll	2	1.2	3.7	1.2	0.7	0.7	0.7	0.5	0.7	0.6	0.3	0.9	0.6
ccalert.dll	3	5.4	1.2	11.4	2.5	3.9	3.2	2.2	2.9	3.6	4.3	4.8	2.2
cdfview.dll	4	2.1	0.7	2.5	10.0	1.3	1.1	1.6	2.2	1.8	0.9	1.7	0.8
dunzip32.dll	5	3.8	0.7	3.9	1.3	5.1	2.3	4.0	2.1	2.0	3.4	3.6	1.6
hotfix.exe	6	3.0	0.7	3.2	1.1	2.3	8.5	1.3	2.0	1.4	3.8	3.1	1.6
iexplore.exe	7	3.6	0.5	2.2	1.6	4.0	1.3	130.2	9.3	1.6	2.4	1.5	7.5
mobsync.exe	8	2.8	0.7	2.9	2.2	2.1	2.0	9.3	6.1	1.5	2.3	2.7	1.6
msrdc.dll	9	2.2	0.6	3.6	1.8	2.0	1.4	1.6	1.5	4.5	1.4	2.0	0.9
nvrshu.dll	10	3.3	0.3	4.3	0.9	3.4	3.8	2.4	2.3	1.4	32.9	6.2	2.1
pacman.exe	11	5.1	0.9	4.8	1.7	3.6	3.1	1.5	2.7	2.0	6.2	1.5	2.2
policytool.exe	12	2.3	0.6	2.2	0.8	1.6	1.6	7.5	1.6	0.9	2.1	2.2	2.6
powerpnt.exe	13	3.5	0.4	2.2	1.1	3.1	1.5	41.2	5.8	1.4	2.6	2.2	4.6
rtinstaller32.exe	14	3.4	0.9	4.1	2.0	3.6	2.0	1.6	2.3	2.2	2.3	2.8	1.2
safrslv.dll	15	1.9	0.9	2.2	1.1	1.2	1.6	1.1	1.1	0.7	2.0	2.0	1.0
tabulate_drive_data.py	16	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	-	0.1	-	0.3
typeaheadfind.dll	17	0.9	0.6	1.3	0.7	0.4	0.3	0.4	0.5	0.7	0.1	0.7	0.5
udlaunch.exe	18	2.9	0.4	3.3	1.1	2.5	-	1.3	1.7	1.8	3.3	3.0	-
vsplugin.dll	19	3.0	0.6	3.4	1.0	2.0	2.5	4.0	1.8	1.2	3.2	3.0	1.6
webclnt.dll	20	3.3	1.0	3.6	1.1	2.3	1.3	1.8	1.8	1.5	2.2	2.8	1.0
winprint.dll	21	0.8	0.5	0.9	0.4	0.6	0.5	0.4	0.5	0.5	0.4	0.6	0.6
wmplayer.exe	22	3.1	0.4	3.1	0.9	2.4	2.0	21.7	3.6	1.3	3.0	2.8	2.4
xrxwiadr.dll	23	11.5	0.8	12.1	2.5	9.2	4.1	3.2	4.6	3.3	12.9	13.0	3.8

Family	No.	13	14	15	16	17	18	19	20	21	22	23
a0003775.dll	1	3.5	3.4	1.9	0.1	0.9	2.9	3.0	3.3	0.8	3.1	11.5
bthserv.dll	2	0.4	0.9	0.9	0.1	0.6	0.4	0.6	1.0	0.5	0.4	0.8
ccalert.dll	3	2.2	4.1	2.2	0.1	1.3	3.3	3.4	3.6	0.9	3.1	12.1
cdfview.dll	4	1.1	2.0	1.1	0.1	0.7	1.1	1.0	1.1	0.4	0.9	2.5
dunzip32.dll	5	3.1	3.6	1.2	0.1	0.4	2.5	2.0	2.3	0.6	2.4	9.2
hotfix.exe	6	1.5	2.0	1.6	0.1	0.3	-	2.5	1.3	0.5	2.0	4.1
iexplore.exe	7	41.2	1.6	1.1	0.2	0.4	1.3	4.0	1.8	0.4	21.7	3.2
mobsync.exe	8	5.8	2.3	1.1	0.1	0.5	1.7	1.8	1.8	0.5	3.6	4.6
msrdc.dll	9	1.4	2.2	0.7	-	0.7	1.8	1.2	1.5	0.5	1.3	3.3
nvrshu.dll	10	2.6	2.3	2.0	0.1	0.1	3.3	3.2	2.2	0.4	3.0	12.9
pacman.exe	11	2.2	2.8	2.0	-	0.7	3.0	3.0	2.8	0.6	2.8	13.0
policytool.exe	12	4.6	1.2	1.0	0.3	0.5	-	1.6	1.0	0.6	2.4	3.8
powerpnt.exe	13	76.0	1.5	1.0	0.2	0.2	1.5	2.8	1.7	0.3	12.6	8.2
rtinstaller32.exe	14	1.5	13.4	1.1	0.1	0.4	3.1	1.9	2.0	0.6	1.7	6.3
safrslv.dll	15	1.0	1.1	3.3	0.1	0.8	-	1.4	1.2	0.6	1.1	2.6
tabulate_drive_data.py	16	0.2	0.1	0.1	2.8	0.1	-	0.2	0.2	-	0.1	0.3
typeaheadfind.dll	17	0.2	0.4	0.8	0.1	2.3	0.2	0.5	0.8	0.4	0.2	0.6
udlaunch.exe	18	1.5	3.1	-	-	0.2	-	2.1	0.9	0.5	2.2	3.6
vsplugin.dll	19	2.8	1.9	1.4	0.2	0.5	2.1	3.2	1.7	0.5	2.7	3.5
webclnt.dll	20	1.7	2.0	1.2	0.2	0.8	0.9	1.7	3.8	0.7	1.5	5.0
winprint.dll	21	0.3	0.6	0.6	-	0.4	0.5	0.5	0.7	1.1	0.4	0.6
wmplayer.exe	22	12.6	1.7	1.1	0.1	0.2	2.2	2.7	1.5	0.4	9.1	5.7
xrxwiadr.dll	23	8.2	6.3	2.6	0.3	0.6	3.6	3.5	5.0	0.6	5.7	15.9

Although the greatest average similarity for a given file family is usually within that file family, there are exceptions as between file families a0003775_dll and xrxwiadr_dll. This inconsistency could be due to the differences in file size or to other attributes within the files in these two file groups. An example similarity analysis plot illustrating the problem is Figure 7. Ranges of homogeneous texture vectors contain similar low mode counts and moderately high entropy values, suggesting that our similarity measure is primarily attributed to regions of compressed data rather than similarity in code. The few similarity matches in other regions suggest that there is actually little similarity between these two files.

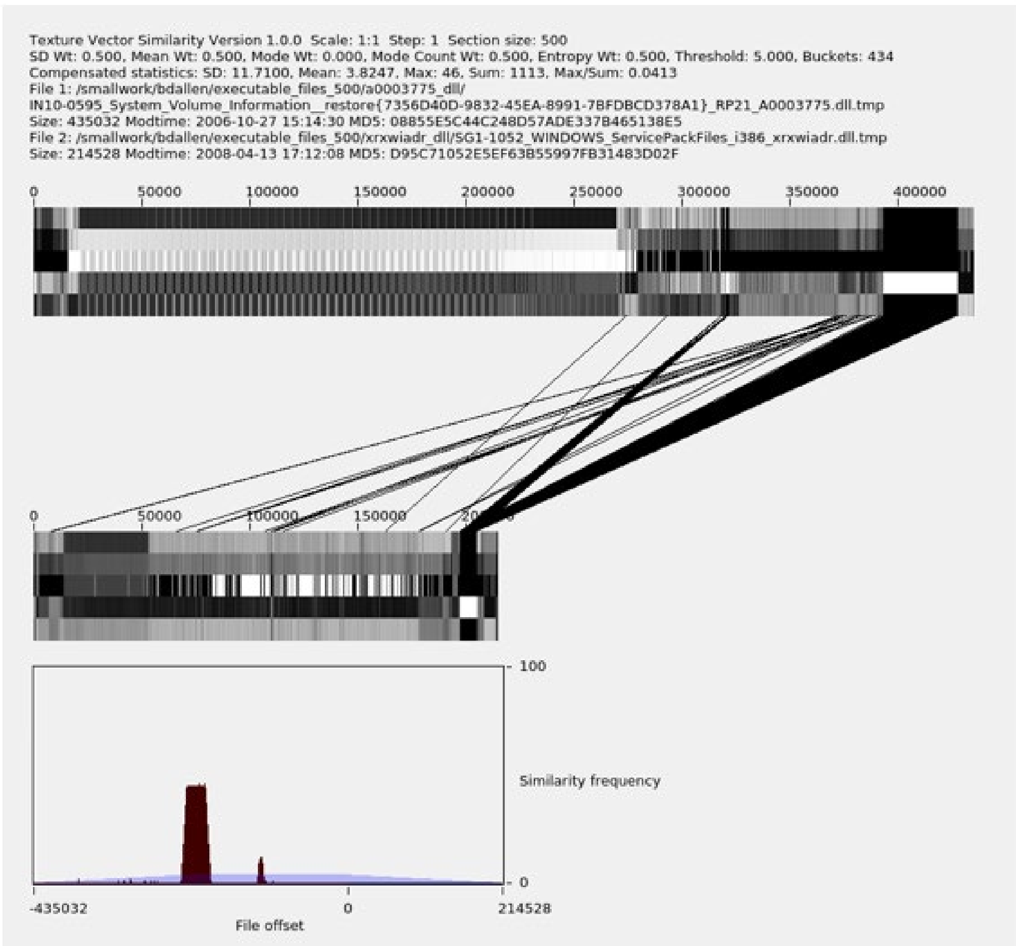
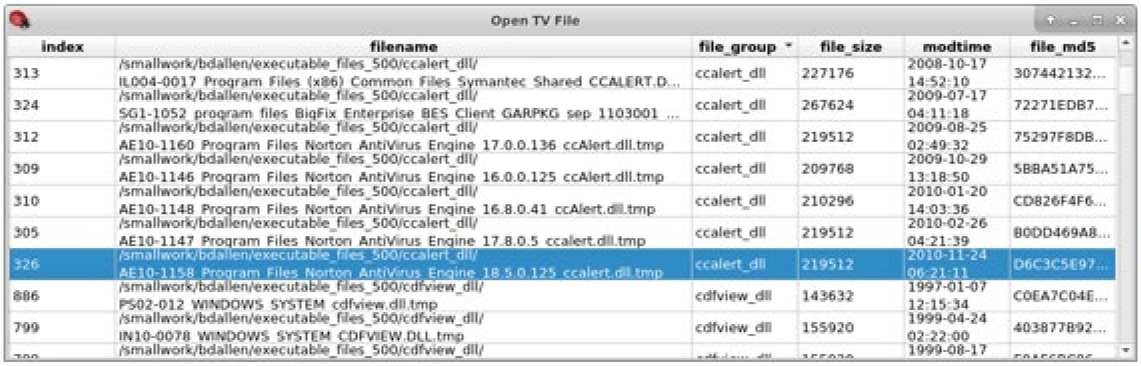


Figure 7. False-Positive Similarity Between Two Files Caused by Homogeneous Compressed Data

Examining Similarity Using the Texture-Vector Browser GUI Tool

Our Texture-Vector Browser GUI tool can examine trends in file similarity based on file creation times and file-similarity measures. Figure 9 shows an example. The horizontal axis is the file modification time. This can be the time the file was created if it was never modified, or the time it was modified by update or by contamination with a virus. The vertical axis is the measure of similarity between the file the user selects and the other files in the view, which if the Stay in group mode is selected, will be files within its family. Files higher up on the vertical axis are more similar to the selected file than files lower down on the vertical axis, where the similarity measure, as described in the Calculating Similarity Measures Between Files section, is the value on the vertical axis. By clicking on a node, the focus of the view changes to show the similarities between the file associated with the clicked node and other files. By clicking on an edge, the view shows the similarity graph involving the two files associated with the edge.

Using the node listing capability in the Texture-Vector Browser GUI and by sorting the list by file group and modification time, we find and select the file in the ccalert_dll file group with the latest timestamp, as shown in Figure 8.



Open TV File

index	filename	file_group	file_size	modtime	file_md5
313	/smallwork/bdallen/executable_files_500/ccalert_dll/ IL004-0017_Program_Files_(x86)_Common_Files_Symantec_Shared_CCALERT.D...	ccalert_dll	227176	2008-10-17 14:52:10	307442132...
324	/smallwork/bdallen/executable_files_500/ccalert_dll/ SG1-1052_program_files_BigFix_Enterprise_BES_Client_GARPKG_sep_1103001_...	ccalert_dll	267624	2009-07-17 04:11:18	72271EDB7...
312	/smallwork/bdallen/executable_files_500/ccalert_dll/ AE10-1160_Program_Files_Norton_AntiVirus_Engine_17.0.0.136_ccAlert.dll.tmp	ccalert_dll	219512	2009-08-25 02:49:32	75297F8DB...
309	/smallwork/bdallen/executable_files_500/ccalert_dll/ AE10-1146_Program_Files_Norton_AntiVirus_Engine_16.0.0.125_ccAlert.dll.tmp	ccalert_dll	209768	2009-10-29 13:18:50	5BBA51A75...
310	/smallwork/bdallen/executable_files_500/ccalert_dll/ AE10-1148_Program_Files_Norton_AntiVirus_Engine_16.8.0.41_ccAlert.dll.tmp	ccalert_dll	210296	2010-01-20 14:03:36	CD826F4F6...
305	/smallwork/bdallen/executable_files_500/ccalert_dll/ AE10-1147_Program_Files_Norton_AntiVirus_Engine_17.8.0.5_ccalert.dll.tmp	ccalert_dll	219512	2010-02-26 04:21:39	80DD469A8...
326	/smallwork/bdallen/executable_files_500/ccalert_dll/ AE10-1158_Program_Files_Norton_AntiVirus_Engine_18.5.0.125_ccalert.dll.tmp	ccalert_dll	219512	2010-11-24 06:21:11	D6C3C5E97...
886	/smallwork/bdallen/executable_files_500/cdfview_dll/ PS02-012_WINDOWS_SYSTEM_cdfview.dll.tmp	cdfview_dll	143632	1997-01-07 12:15:34	C0EA7C04E...
799	/smallwork/bdallen/executable_files_500/cdfview_dll/ IN10-0078_WINDOWS_SYSTEM_CDFVIEW.DLL.tmp	cdfview_dll	155920	1999-04-24 02:22:00	403877B92...
[illegible]	/smallwork/bdallen/executable_files_500/cdfview_dll/	[illegible]	[illegible]	1999-08-17	[illegible]

Figure 8. Sorted Node Listing with Node 326 Selected

In our dataset, this file is named AE10-1158_Program_Files_Norton_AntiVirus_Engi ne_18.5.0.125_ccalert.dll.tmp, indicating that it is on drive AE10-1158 from United Arab Emirates. It is indexed in our similarity graph dataset as node 326 (in green). The file naming convention is explained in the Peparing the Dataset of Executable Files section. This graph shows node 326 and its similar neighbors and similar edges, where the similarity measure, described in the Calculating Similarity Measures Between Files section, is 1.0 or more. The horizontal axis is the file modification time and the vertical axis is the relative similarity between file (node) 326 and the other files. In this graph, we see two clusters of similarity.

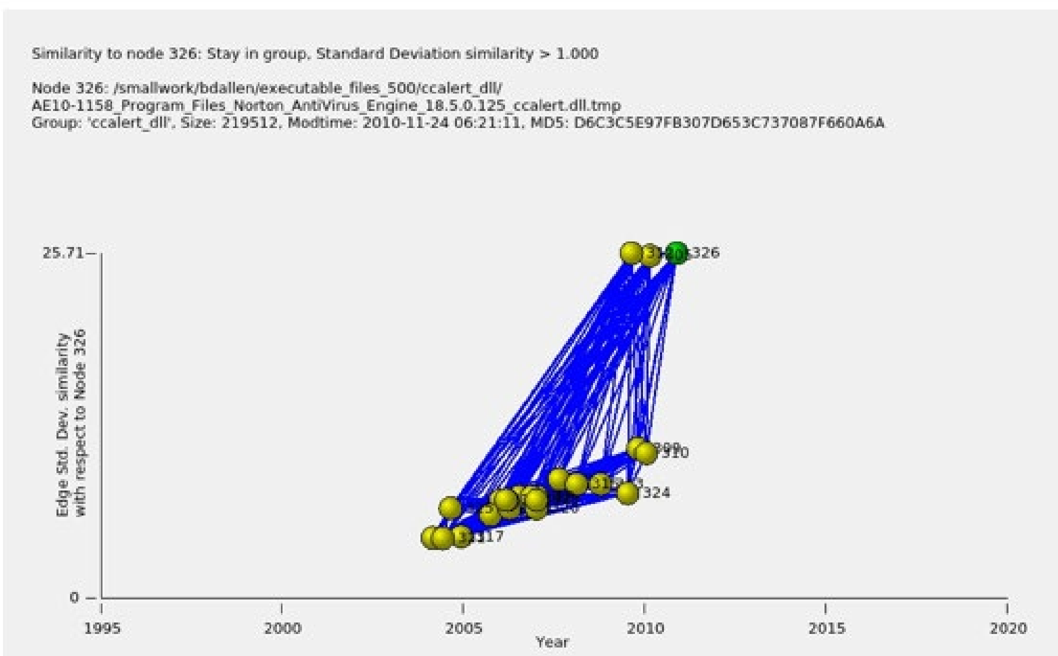


Figure 9. Files (Nodes) and Similarity Measures (Edges) Associated With File Node 326 Showing Modification Times and Similarity to Node 326

Figure 10 shows the analysis of the edge that connects nodes 326 and 312, corresponding to Program Files/Norton AntiVirus Engine 18.5.0.125_ccalert.dll on drive AE10-1158 and Program Files Norton AntiVirus Engine 17.0.136_ccAlert.dll on drive AE10-1160. This display was obtained using the GUI by clicking on the edge shown in Figure 9 that connects these two files. The texture vector patterns appear very similar and the similarity histogram spikes with a similarity count of nearly 370 near file offset 0, a large number, indicating that these two files are similar. We can click on any of the yellow dots in the GUI to select the file corresponding to it to compare other files against it.

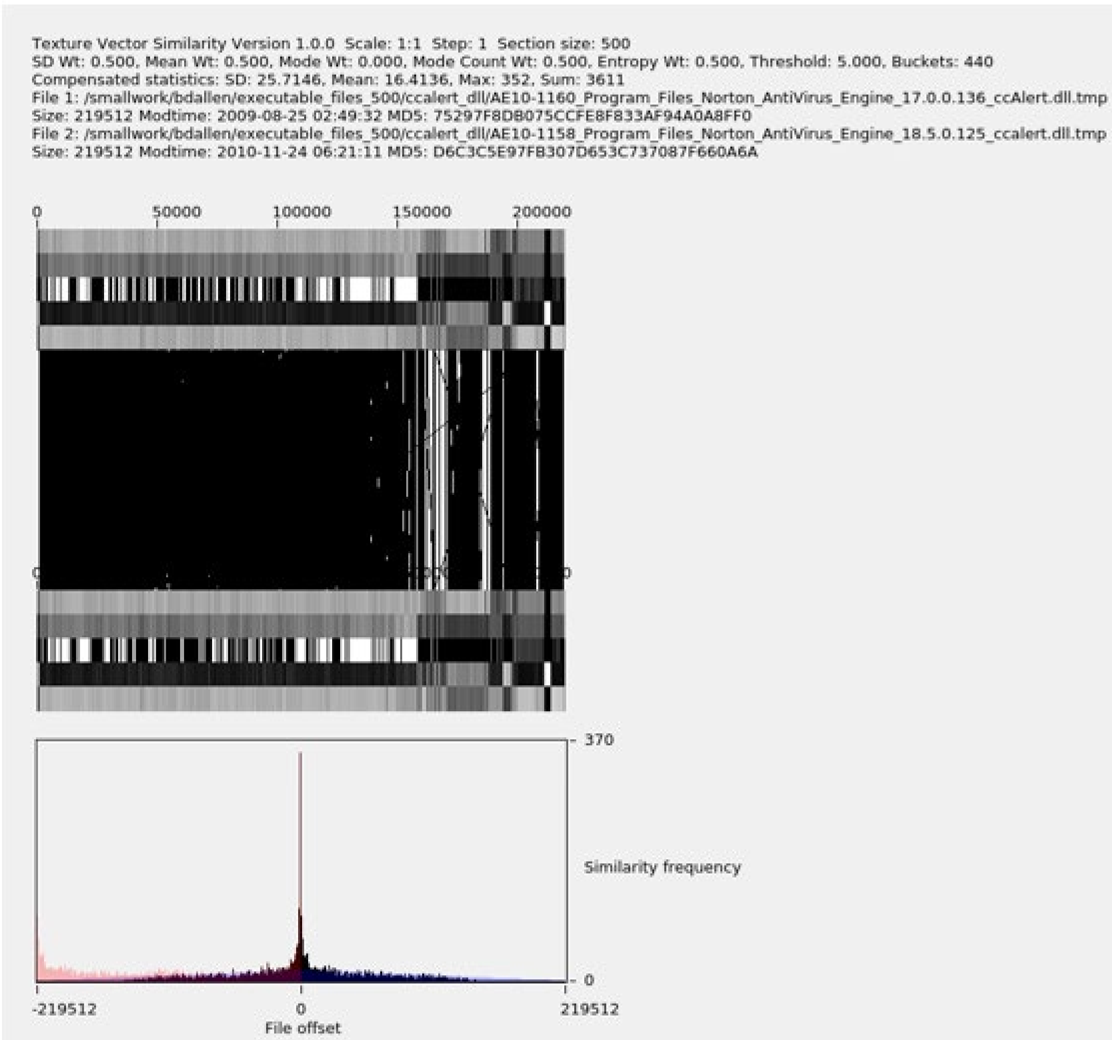


Figure 10. A Detailed Comparison of Files 312 and 326 Showing a High Degree of Similarity

Composition Analysis

By looking at the five bands in the texture-vector diagram, we can make inferences about the regions of executable code files being compared, in particular the locations of header, code, and data sections. For Figure 10, for the first two textures, covering the first 1,000 bytes, the standard deviation, mean, mode, and entropy values are lower than the values in other regions, while the mode count is higher. We infer that this represents a header, and the transition in the texture represents a transition to another type of content. The region from approximately byte 1,000 to byte 160,000, contains relatively medium values of the standard deviation, mean, and entropy, mode values that are either very high or very low, and consistently low mode counts. We infer that this is the code section. The third region, from approximately byte 160,000 through to the end at byte 219,512, usually has a low mode value while values in the other four statistics vary but consistently with the two files. We infer that this is a region of data mostly unchanged between version. We also infer that the additional 10 KB added in the newer version was new code.

Progressive Time Similarity

Software files tend to be most similar to the previous version. Figure 11 shows an example for the nvrshu_dlll file family. Here, the file with the latest timestamp, WINDOWS system32 nvrshu.dll from the MY01-023 drive from Malaysia, is selected. We see sporadic measures of similarity between 10 and 30 for files before year 2005, but for files after 2005, we see a gradual increase in similarity over time from about 40 to 61.

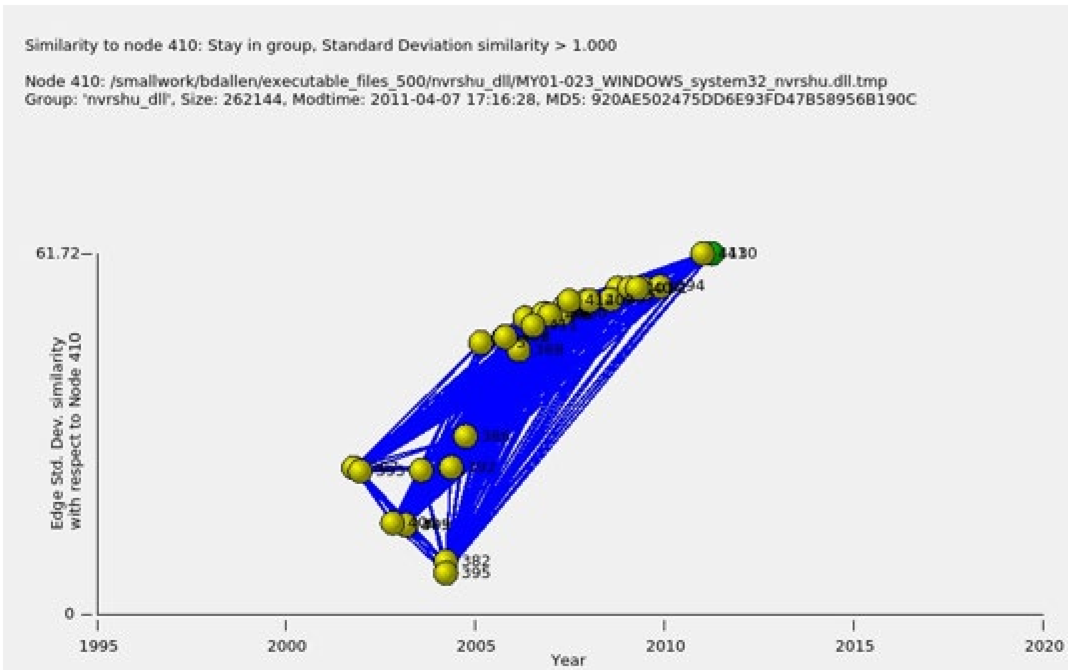


Figure 11: Similarity Increases as Versions Approach the Latest Version

Version Analysis

With these diagrams, we can study on the origin and evolution of versions of files. Although an original file should have the earliest file creation time, file creation times can be modified inadvertently or maliciously. Another clue is that the original file often has the least amount of code. Node 326 in Figure 9, file Program Files/Common Files/Sym antex/Shared ccAlert.dll.tmp from drive PA002-049 from Panama is likely the original file in its group because its file modification time is earliest and its similarity to latest files decreases over time.

A newer version of code that introduces new features is likely to contain more code than the version before it. A newer version that is only a bug fix will be similar in size to the version before it and will have similar texture-vector patterns as in Figure 10.

Files released at approximately the same time may be targeted for different operating system platforms or different feature sets. For example 13 files in the webclnt_dll file family were released over 2 days, 2006-01-03 and 2006-01-04. This is too clustered to be in response to new functionality or bug fixes. These files could be a response to a virus because some of their file sizes are the same and their texture-vector patterns appear identical. However, bear in mind our sample is incomplete and important versions of software may be missing.

Examining Similarity using Gephi

Although the Texture-Vector Browser GUI tool was specifically designed for examining network graphs created from the dataset of similarity-graph files, graph analytics can also be done with popular open-source tools such as the Gephi graph-visualization tool.

Conclusions and Future Work

Conclusions

This thesis proposed applying a vector of transforms to executable code to create texture-vector data, and then using analytics to identify similarities between executable files. We tested a sample of executable code files with our methods. Our experiments showed files within file families had greater average similarity than files across file families. We found that the visual patterns in the texture vectors were effective in identifying similar regions in two files as well as sections that may be compressed.

Future Work

This work used texture vectors calculated from a section size of 500 bytes. A large section size might reveal similarity across a larger section of data, equivalent to applying a low-pass filter to texture-vector values. A section size that is a power of two or is aligned to the size of fixed-size data might naturally align better with the section boundaries from which texture-vectors are calculated.

Texture vectors may be useful for classifying file types or detecting types of data embedded within a file. Further work in this direction might consist of defining data patterns that map to particular data types.

The open-source tool Gephi offers many capabilities such as filtering and neighbor analytics that can be used to augment the similarity analytics provided by our tool.

Future work might use it to obtain additional insight about file similarity.

References

Aghajani, E., Mocci, A., Bavota, G., & Lanza, M. (2017). The code time machine. In 2017 IEEE 25th International Conference on Program Comprehension (ICPC).

Arbuckle, T. (2008). Visually summarising software change. In 12th International Conference Information Visualization.

Bergroth, T. R. L., & Hakonen, H. (2000). A survey of longest common subsequence algorithms. In Proceedings of the International Symposium on String Processing Information Retrieval (SPIRE ’00), 39–48.

Burch, M., Diehl, S., & Weigerber, P. (2005). Visual data mining in software archives. In SoftVis ’05 Proceedings of the 2005 ACM Symposium on Software Visualization, 37–46.

Cabezas, D., & Mooij, B. (n.d.). Detecting source code re-use through a binary analysis hybrid approach. Forensic Magazine. https://www.forensicmag.com/article/2013/02/detecting-source-code-re-use-through-binary-analysis-hybrid-approach

Cole, R. (2018). Tight bounds on the complexity of the Boyer-Moore pattern matching algorithm (1st ed.). Forgotten Books.

Defant, A. (2011). Classical summation in commutative and noncommutative Lp-spaces (1st ed.). Springer.

Garfinkel, S., Farrell, P., Roussev, V., & Dinolt, G. (2009, August). Bringing science to digital forensics with standardized forensic corpora. Digital Investigation, 6, S2–S11.

Hadmi, A., Puech, W., Said, B. A. E, & Ouahman, A. A. (2012). Perceptual image hashing. In University of Montpellier II, CNRS UMR 5506-LIRMM, France.

Jang, J., & Brumley, D. (2009). Bitshred: Fast, scalable code reuse detection in binary code. CMU-CyLab-10-006.

Josse, S., Bachaalany, E., Gazet, A., & Dang, B. (2014). Practical reverse engineering: x86, x64, ARM, Windows Kernel, reversing tools, and obfuscation (1st Ed.). Wiley.

Koike, H. & Chu, H.-C. (1997). Vrcs: Integrating version control and module management using interactive three-dimensional graphics. In Graduate School of Information Systems University of Electro-Communications Chofu, Tokyo 182, Japan.

Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., & Roli, F. (2018, December). Adversarial malware binaries: Evading deep learning for malware detection in executables. In Proceedings of the 26th European Signal Processing Conference, Rome, Italy.

Rho, J., & Wu, C. (1988). An efficient version model of software diagrams. In Proceedings 1998 Asia Pacific Software Engineering Conference (Cat. No.98EX240).

Rowe, N. C. (2018). Associating drives based on their artifact and metadata distributions. In Proceedings of the 10th International EAI Conference, ICDF2C 2018, New Orleans, LA, USA.

Seemann, J., & von Gudenberg, J. W. (1998, March). Visualization of differences between versions of object-oriented software. In Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering.

Shotts, W. (2019). The Linux command line (2nd ed.). No Starch Press.

Swierstra, W., & Lh, A. (2014). The semantics of version control. In Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2014), 43–54, New York, NY, USA.

Voinea, L., Telea, A., & van Wijk, J. J. (2005). Cvsscan: Visualization of code evolution. In SoftVis '05 Proceedings of the 2005 ACM Symposium on Software Visualization, 47–56.

Wikipedia. (2020, February 4). Knuth-Morris-Pratt algorithm. https://en.wikipedia.org/wiki/Knuth-Morris-Pratt algorithm

Technology Trust: The Impact of Trust Metrics on the Adoption of Autonomous Systems Used in High-Risk Applications

Michael Anderson is a PhD Candidate at the Naval Postgraduate School Information Sciences Department. His research interests include Technology Trust. Mr. Anderson has an extensive background in the design of wireless communication systems and the use and deployment of military technologies. His past experience includes over 10 years of work as adjunct faculty teaching university students and professional engineers and scientists in the areas of digital design and computer architecture. Mr. Anderson holds a BS and MS in Electrical Engineering from the University of Michigan, Ann Arbor. [-mail address: mganders@nps.edu]

Dr. Johnathan Mun and is a specialist in advanced decision analytics, quantitative risk modeling, strategic flexibility real options, predictive modeling, and portfolio optimization. He has authored 24 books; holds 22 patents and patents pending; created over a dozen software applications in advanced decision analytics; and has written over a hundred technical notes, journal articles, and white papers. He is currently the CEO of Real Options Valuation, Inc., and his prior positions include vice president of Analytics at Oracle/Crystal Ball and a senior manager at KPMG Consulting. Dr. Mun holds a PhD in Finance and Economics from Lehigh University, an MBA and MS from Southeastern University, and a BS in Physics and Biology from the University of Miami. [E-mail address: jcmun@nps.edu]

Abstract

As autonomous systems become more capable, end users must make decisions about how and when to deploy such technology. The use and adoption of a technology to replace a human actor depends on its ability to perform a desired task and on the user's experience-based trust that it will do so. The development of experience-based trust in autonomous systems is expensive and high risk. This work focuses on identifying a methodology for technology discovery that reduces the need for experience-based trust and contributes to increased adoption of autonomous systems. Initial research reveals two problems associated with the adoption of high-risk technologies; 1) end user's refusal to accept new systems without high levels of initial trust and 2) lost or uncollected experience-based trust data. The main research hypothesis is that a trust score, or trust metric, can influence the initial formation of trust by functioning as a surrogate for experience-based trust, and that trust in technology can be measured through a probability-based prediction of risk.

Introduction

We had better be quite sure that the purpose put into the machine

is the purpose which we desire.

- Norbert Wiener, Some moral and

technical consequences of automation

The use of technology by the Department of Defense (DoD) depends on its ability to perform a desired task. There are many issues associated with trust in technology that are increasing in importance as the U.S. military begins to acquire and deploy autonomous systems. In order to ensure the effective adoption of new innovations in technology, there is a need to establish a system of metrics that justify a level of technology trust. This proposed research has the explicit goal of investigating and recommending trust metrics by applying advanced analytical methodologies to increase the speed and effectiveness of the adoption of new technologies. This investigation proceeds by participating in an evaluation of technologies for use in evolving high-risk military applications. The trust metrics are measured in terms of the technology acceptance versus system control.

Technology Trust

Devitt (2018) implies that in order to meet the DoD requirements for increased speed of adoption for new technologies, there is a need to replace the model of developing trust over longer periods of time with a justifiable metric of trust. This research studies the effectiveness of establishing and introducing trust metrics on the evaluation and selection of technologies. The work participates in an ongoing assessment of autonomous systems for use in high-risk military applications throughout fiscal year 2019. A model is developed that optimizes the cognitive impacts of these trust metrics as they relate to the technology selection and adoption process. The approach will be extensible and can be adopted into private industry.

Research Problem

The recent increase in the use and deployment of commercial technologies by other countries is a disruptive threat to the United States' technological superiority. The rapidly changing technology landscape requires DoD laboratories to increase the speed at which they adopt new technologies (David & Nielsen, 2016). With declining budgets in research, it is imperative that the DoD establish new methods for rapidly adopting and effectively deploying new and emerging technologies whenever possible.

Research Purpose

As autonomous systems begin to surpass the capabilities of humans, there is a need to establish a level of confidence in a technology's ability to perform as expected. The complexity of modern systems makes it difficult to establish a comprehensive metric of trust. Past research in technology trust focused on automation and methods to measure interpersonal person-to-firm relations, such as trust in a Web vendor or a virtual team member (McKnight et al., 2011). This research has the goal of establishing and measuring a comprehensive trust metric for individual pieces of technologies, such as autonomous systems, used in high-risk military applications. The development of a trust metric serves two purposes: first, as a surrogate for experience-based trust by contributing to the formation of initial trust and, second, as a collection tool for capturing experience-based trust data.

Research into a "trust-discovery" methodology contributes to improved understanding of human-machine trust formation and the development of a technology-literate workforce capable of accurately assessing new technology for a given operational scenario. This work first establishes a baseline definition of what it means to "trust" technology. It concludes with the development of a methodology leading to trusting relations between humans and technology. This work contributes to the literature in areas of trust in autonomous systems, technology adoption, and technologies intended for use in high-risk applications where failure or improper application can lead to severe consequences.

Research Questions

This study attempts to answer the following questions:

1. How do varying levels of system control affect the development of trust in technologies used in high-risk military applications? The constructs researched include:
    a. Perceived ease of use
    b. Perceived usefulness
    c. Intent to use
2. How do anthropomorphic metrics affect the development of trust in technologies used in high-risk military applications? The constructs researched include:
    a. Hardware
    b. Algorithms
    c. Links

Research Approach

The following research approach is used:

1. Study the evaluation process of autonomous systems for use in high-risk military applications.
2. Develop a conceptual framework for trust metrics that optimizes the technology evaluation process.
3. Observe and record the results of both laboratory and field experimentation.

The basic tenets of the experimental design are realized through a 2 x 3 factorial design (Table 1-1).

Table 1-1. 2 x 3 Factorial Design

			SYSTEM CONTROL		
			LOW	MID	HIGH
TRUST METRIC		NOT USED	…	…	…
		USED	…	…	…

Contribution

The concept of a technology trust metric has applicability beyond the DoD. Private industry can greatly benefit from the concepts and methodologies developed in this research by applying trust metrics to the research and development of existing or new consumer technologies such as machine learning (ML), artificial intelligence (AI) systems, smart algorithms, and embedded technologies. These intelligent systems are transformative areas that will eventually integrate into all industries (e.g., self-driving cars, delivery drones, big data analytics, and the Internet of Things, where algorithms, machines, and computer systems are continually learning and evolving).

This research also contributes to trust theory and provides an increased understanding of military technology acceptance. The recommendations provide a conceptual framework for how a military community develops trust in technologies for high-risk missions and how varying factors influence the development of such a relationship. Currently, there is an effort to perform such trust analytics within the DoD in which this current research will participate.

Organization

Section 2: Literature Review

This review investigates existing literature that includes terms such as *technology trust and risk, decision making,* and *technology-adoption models*. A review of current and past theory on technology trust and decision making is developed, which is then used to develop a comprehensive metric for assessing technology trust within the DoD. A proposed framework for a comprehensive trust metric is identified and introduced to the technology evaluation process.

Section 3: Experimental Design

Both lab and field experiments are conducted to identify trust metrics. This research intends to leverage an ongoing DoD experiment reviewing and selecting a series of new autonomous systems. The existing data is collected from DoD active-duty technology end users, as well as civilian scientist support staff. The study investigates how varying levels of trust influence cognitive decision making as well as technology adoption. The primary product of this investigation is the experimental data obtained.

Literature Review

The purpose of this section is to understand the formation of trust, as well as analyze the constructs of a trust relationship. The idea of trust metrics is broken down into quantifiable segments based on leading theories. We conclude by presenting a conceptual framework for a technology trust metric based on what was learned from the literature, as well as what is missing from the literature.

This research was initiated through informal interviews that attempted to identify the factors that contribute to the use of technology in high-risk environments. The participants were a small group of active-duty military and veterans that deploy, or have deployed, with technology that posed great risk of physical harm should it fail. A number in this group experienced significant injury due to the failure of technology, and the potential for bias was noted. The open-ended questions were based on what the users did or did not like about using technology in high-risk scenarios. The initial coding of interviews revealed the following themes:

1. Hands-on experience with technology is critical for establishing trust, and team-based reputation for a technology is as important as personal experience.
2. Users favor simple technology containing only the features needed to accomplish a mission, and users reject new technology in favor of older and more trusted systems.
3. Personal investment in a mission is key to learning how to use new technology.

These themes all have implications for the adoption of autonomous systems within the DoD. Advanced robotic systems have the ability to improve performance in a number of military roles while reducing risk to humans, and it is important to understand how to improve the adoption of such systems within the DoD. This initial research focused on technology in dangerous environments and reveals that adoption is highly dependent on the ability of the user to obtain the knowledge necessary to develop trust. This theme led to our initial literature review on understanding trust, and how it applies to technology adoption.

The literature review was developed through searches on both Web of Science and Google Scholar using combinations of search terms such as *trust, knowledge-trust, technology trust, human-computer, human-robot, technology acceptance, trust attribute, trust risk,* and *risk score*. The literature results were narrowed to 93 relevant articles.

Knowledge

The process of obtaining knowledge is fundamental to the establishment of trust. We therefore briefly review the epistemologies, or the processes for how a person gets to know something, as concepts important to this work. Early philosophers presented the two opposing views of the source of knowledge: rationalism or empiricism.

The French Philosopher Rene Descartes was an early rationalist who believed that we can only know something through reason, and that the only thing we can truly know is that we have consciousness. Descartes presented a methodology for knowing what is real that rejects a construct needed for the establishment of technology trust. He established a dualism that reduces our understanding to distinct areas of consciousness and matter but does not account for the senses. Our sense perception, he believed, is easily prone to error due to subjective interpretation. He believed that the senses are meant to simply get us around in the world rather than lead us to truth. In order to test our hypothesis of trust in technology we must identify constructs that permit measurement of human interaction with technology, and technology interaction with its surroundings.

John Locke later introduced empiricism that, contrary to rationalism, stated that all knowledge must be obtained through experience. The empiricists claimed that the senses were the only way to true knowledge, and that experience is much more accurate than anything the mind could ever reproduce through memory or reason. The theories presented by rationalism and empiricism both stand to contribute to the formation of trust through the application of reason-based knowledge and experience-based knowledge. (However, there is a limitation in that we lack a method for integrating these two forms of obtaining knowledge.)

Further review reveals that modern philosophers reject the idea that knowledge is obtained exclusively through either rationalism or empiricism. The philosopher Immanuel Kant provided a synthesis between the two opposing theories. First, he noted that reason lacks the ability to create sensory experience; it is only through reason that we are able to accurately analyze the stimuli received through the senses. This theory represents a foundation for understanding the development of trust. The Figure 2-1 represents a causal model based on our finding in the literature that includes a synthesizing feedback loop to represent how we come to know something.

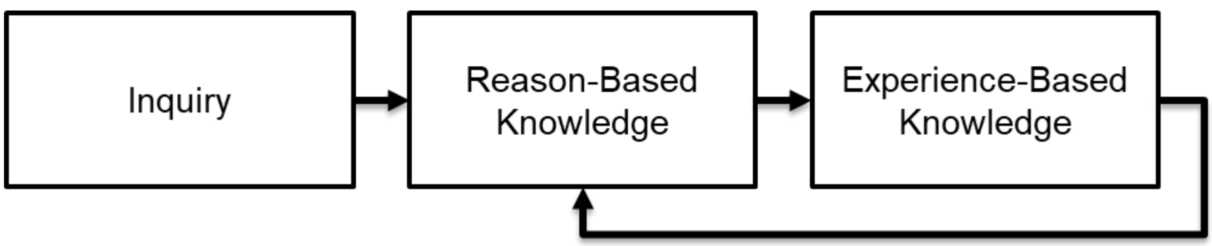


Figure 2-1. A Model of Inquiry Leading to Knowledge

Trust

Castelfranchi and Falcone (2010) review over 72 definitions of what it means to know something well enough to have trust, and their work reveals a great deal of confusion and ambiguity surrounding the use of the term. The concept of trust appears to be subjective in nature, and the literature does not provide a commonly accepted definition across research disciplines. Agreement in the literature was found for the definition of trust in two small areas: 1) the basic premise of trust involves two actors, and 2) trust is a relationship in which one entity relies on someone, or something, based on a given criterion. Research into the meaning of a "given criterion" reveals an interchangeable use of the terms *trust* and *confidence*. The only noticeable difference in the use of these terms is that trust is based on decisions involving risk, whereas confidence involves decisions devoid of consequence.

This literature review furthers its investigation into trust through researching interpersonal relationships. Leading theories on interpersonal trust present vulnerability and risk as the contributing factors unique to the development of such a relationship. Cho et al. (2015) surveyed the meaning of trust across academic disciplines and identified that it follows a basic premise involving risk. For example, they found that academic researchers of trust in psychology assess the probability that individual behaviors are repeatable in situations that entail risk, and in sociology researchers of trust assess the probability that one party will perform an action that will not hurt the interests of a dependent party or expose them to risk due to ignorance or uncertainty.

Rousseau et al. (1998) define interpersonal trust as a psychological state of a trustor accepting vulnerability in a situation involving risk, based on positive expectations of the intentions or behavior of the trustee. Boon et al. (1991) simplify the definition of trust as a state involving confident predictions about another's motives in situations entailing risk. The majority of early research on trust involves person-person relationships and provides a starting point for our understanding of the process of developing trust. Figure 2-2 presents an operational model of interpersonal trust formation based on the reviewed literature.

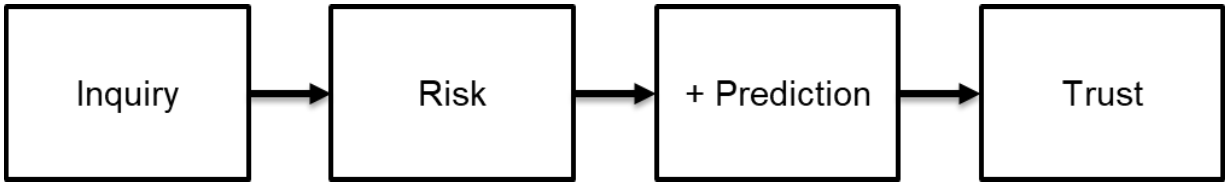


Figure 2-2. A Model of Interpersonal Trust Formation

Adams and Webb (2002) describe two broad processes of developing trust between two individuals. The first is defined as "person-based trust", which develops through repeated engagements, and the second is called "category-based trust", which develops in the absence of direct experience. These definitions parallel the theories identified in our previous research into the epistemologies. Consequently, we modify interpersonal trust terminology to match our research by replacing "category-based" with "reason-based", and "person-based" with "experience-based".

Kramer and Tyler (1996) assess reason-based trust and presents it as useful for understanding how one develops a trusting relationship when personal or social interaction is not possible. This type of trust often develops through someone's membership in a familiar group or category. The factors contributing to reason-based trust can be social roles, training, or experience. In reason-based trust, the relationship is most commonly developed through a reputation that serves as a proxy for personalized knowledge and direct experience. These concepts lead to our first research hypothesis regarding the experience-based trust relationships.

H1: An experience-based proxy will influence the tendency to trust or distrust.

Rempel, Holmes, and Zanna (1985) assess that experience-based trust relationships develop over a long period of time through personal interaction. In their early research on trust they describe three factors that influence the development of trust as competence, benevolence, and integrity. They also discuss the significance of the mental motivation behind the desires to establish a relationship and found it was strongly correlated to the factors that influence trust. Their work confirms a theme identified in our early interviews with users of technology in risk-application that emphasized the importance of personal investment. It also leads to our second hypothesis relating motivation to technology acceptance.

H2: Increased personal motivation will increase technology acceptance.

There appears general agreement in the literature reviewed that interpersonal trust consists of two categories: first, that trust is both reason-based and experience-based and, second, the strength of the trust bonds may differ. The concept of initial trust involves the development of a relationship based purely on reason and represents a weaker connection that can be explained by first impressions. The second category of experience-based trust involves direct knowledge and regular interaction. This type of trust represents a stronger connection and is explained by relationships that develop over a longer period of time through an experience-reason feedback loop. Figure 2-3 presents a model of interpersonal trust.

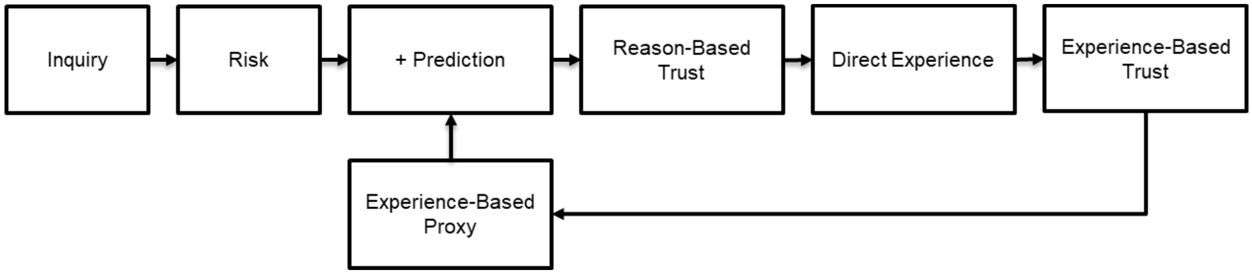


Figure 2-3. Interpersonal Trust Life Cycle

Technology

The past research on interpersonal trust applies in many ways to trust in technology. This study sought out literature that contributes to the development of a methodology of technology discovery leading to person-technology trust. The potential for integrating interpersonal trust research into technology trust was discussed by McKnight (et al., 2011). This research found that interpersonal trust is based on a trustee's expectations and reliance on a trustor to perform as expected through benevolence, even though the trustor possesses the volition to choose to do what is right or what is wrong. Since technology does not possess volition (ability to choose), some researchers went as far as to dismiss the idea of trust in technology as irrelevant. However, recent advances in artificial intelligence refute the claims that technology lacks volition. This is confirmed in the vast amount of current research into how autonomous systems make decisions that can either harm or protect human life.

Technology trust research is further represented in multiple disciplines of engineering and science. The major fields of technology trust research include, but are not limited to, artificial intelligence, command and control, human-computer interaction (HCI), data fusion, human-machine fusion, cyber security, and automation. Multiple models for researching trust are presented in the literature that combine both humanlike and systemlike terminology. Technology trust is a multifaceted area of research that integrates both humanlike measures and systemlike measures. Three of the most frequently used humanlike terms used to model technology are *competence, benevolence,* and *integrity*. The work by McKnight et al. (2011) and Lankton et al. (2015) consider the systemlike alternate terms for technology trust as *reliability, functionality,* and *helpfulness*. A number of systemlike measures of technology trust were identified that are outside the scope of this work but still important to ongoing trust research. These potential systemlike measures include supply chain management, past vendor performance, hardware/software-oriented security, and network security.

The majority of the language used to describe interpersonal trust can apply to technology trust. For example, the word *benevolence* is a very humanlike attribute that is likely to appear in future literature on the decision-making capabilities of self-driving cars. A total of 86 factors and attributes related to interpersonal and technology trust were collected from the literature to form a random nomological network of trust terms. A *factor* is described as situational consideration of technology use that has the potential to influence trust, such as risk and time to operate. An *attribute* is a characteristic inherent to the technology such as its speed, power, and processing capability. The combined and unsorted list is presented in Table 2-1. Future experimentation involves understanding the influence of these terms in the following areas:

1. Factors that measure reason-based and experience-based technology trust
2. Attributes that characterize technology trust as a proxy for experience

Table 2-1. Nomological Network of Trust Factors and Attributes (*Sources:* Cho et al., 2015; DeVitt, 2018; Hoff and Bashir, 2015; McKnight et al., 2011; Schaefer, 2016)

Ability	Character	Disappointment	Importance	Process	Skills
Adaptive	Communication	Disposition	Incompetent	Protect	Stability
Adoption	Competence	Dynamic	Integrity	Purpose	Supportive
Adversarial	Completeness	Easy	Intelligibility	Rationality	Teammate
Altruism	Confidence	Expectation	Intent	Recency	Trainable
Attractive	Contract	Experience	Knowledge	Reciprocation	Transparency
Autonomous	Control	Faith	Learning	Regret	Uncertain
Availability	Cooperation	Faults	Likeable	Relational	Understandability
Awareness	Credibility	Fear	Monitored	Relevance	Unstructured
Belief	Credit	Feeling	Motives	Reliability	Utility
Benevolence	Decisive	Frequency	Perception	Relief	Validity
Capability	Delegation	Frustration	Performance	Responsive	
Capital	Dependability	Helpfulness	Popular	Risk	
Centrality	Difficult	Honesty	Power	Robust	
Certainty	Directability	Hope	Predictability	Similarity	

Figure 2-4 represents the integration of technology trust with the interpersonal trust factors and attributes included in our nomological network of terms.

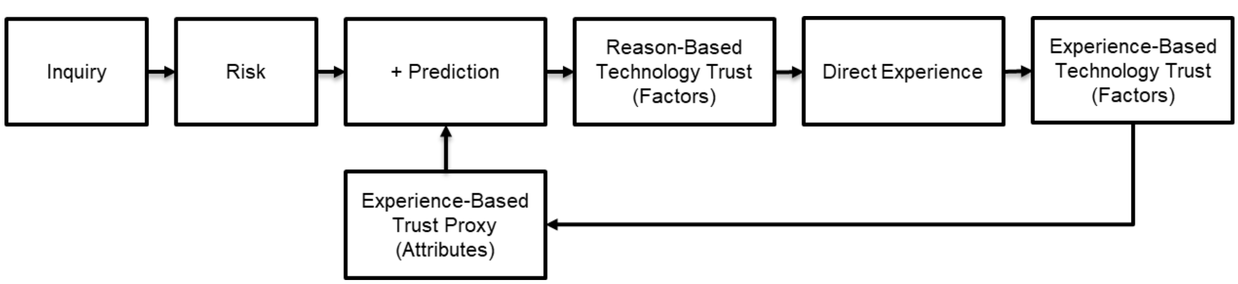


Figure 2-4. Technology Trust Life Cycle

A theory relevant to measuring and characterizing trust is found in the technology acceptance model (TAM) developed by Fred Davis nearly 30 years ago. This model plays a significant role in the majority of research investigating the factors and attributes that influence the acceptance of a technology. In the work by Venkatesh and Bala (2008), they present the TAM's ability to predict individual adoption and use of technology. The TAM assesses the behavioral intention to use a technology through two constructs: perceived usefulness (PU), which is defined as the extent to which a person believes that using a technology will enhance his or her job performance, and perceived ease of use (PEOU), which is defined as the degree to which a person believes that using a technology will be free of effort. These two variables are used to establish a relationship between external influences and potential system usage (Gefen et al., 2003). In the work by McKnight et al. (2002), it was experimentally determined that the TAM variables do not predict continued use of a technology outside of initial acceptance, and that trust in a vendor's past technology does not translate to acceptance of subsequent technologies.

Tétard and Collan (2009) address the challenges of adopting new technology in their work on the lazy-user theory. This theory states that a user will select the technology that demands the least amount of effort to do the job. This theory also addresses one of the themes identified in our early grounded theory study interviewing operators of technology in high-risk scenarios. The application of this theory places technology users at a disadvantage, particularly in high-risk military applications where trustors are known to avoid more capable technology for systems that are easier to understand. If an experience-based proxy can improve the accuracy of developing trust through increased technology literacy, it may lead to increased acceptance of more complex and capable technologies thereby reducing the influence of the lazy-user theory. This leads to our third research hypothesis.

H3: An experience-based proxy will decrease the influence of the lazy-user theory on technology acceptance.

Conclusions

One intent of this section is to identify gaps in research on trust in autonomous systems. It appears that a methodology of technology discovery that leads to trust is not available. This review reveals a clear distinction between reason-based trust and experience-based trust. It also suggests that users are willing to trust technology in high-risk environments, and that an experience-based proxy may increase the quality of such a relationship and the pace at which it is established. Based on the finding in literature, Figure 2-5 illustrates a conceptual framework for a causal methodology of technology adoption by introducing an experience-based proxy that is hypothesized to improve technology adoption. The impact of a proxy introducing inaccurate information is noted as significant but is outside the scope of this work.

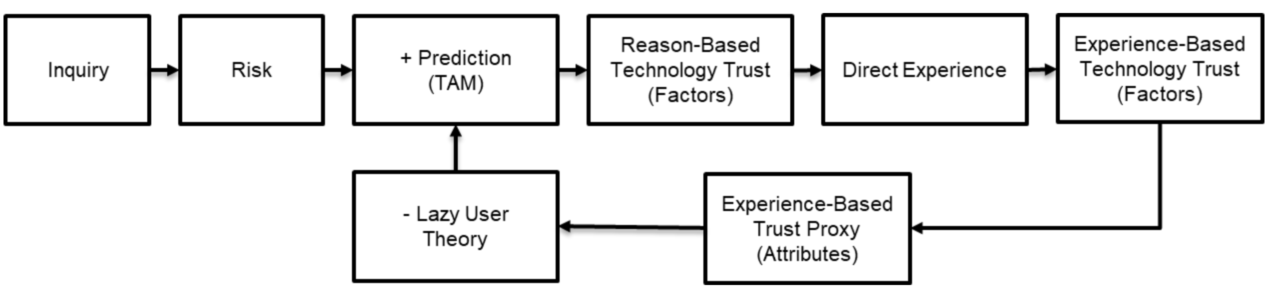


Figure 2-5. Conceptual Framework for Methodology Leading to Technology Trust

Experiment Methodology

This experiment investigates the formation of trust in technology and how it influences the adoption of autonomous systems for use in high-risk military applications. The formation of trust in technology is governed by two constructs: reason-based trust and experience-based trust. Existing literature presents the case for increased accuracy in technology selection through the development of experience-based trust. However, the development of experience-based trust is financially burdensome and takes much longer to form. In most military scenarios, developing experience-based trust presents high levels of risk for physical injury and harm.

Introduction

This experiment is designed to identify trust metrics and how they influence the formation of reason-based trust in autonomous systems used in high-risk military applications. The desired outcome of this work is the identification of attributes that can replace some of the burden required to develop experience-based trust. This research does not intend to demonstrate the validity of the theories behind technology acceptance. Rather, this work investigates potential causal relationships between the manipulation of information and its effect on trust in technologies.

The experiment is conducted in two-phases. Phase one is a group-administered experimental survey that employs manipulations of multiple theories of technology acceptance in order to collect data on reason-based trust in autonomous systems. Phase Two consists of administering the same survey following extensive field testing and experimentation of the phase one systems to provide external validity.

Metrics

The goal of this work is to study the influence of trust metrics on the acceptance of autonomous systems in high-risk applications. However, the complexity of modern technology makes it difficult to establish generalizable metrics that can function as a proxy for experience-based trust. One area of research relevant to establishing such metrics involves the use of anthropomorphism, the attribution of human traits to nonhuman entities, to increase a trustor's ability to accept and utilize technology. Waytz et al. (2014) discuss the need for humanlike mental models to consider technology as a trustworthy teammate. There are reported cases (Pak et al., 2012) where the tendency to anthropomorphize technology leads to situations in which humans give a higher degree of trust to a technology than is warranted. The inverse of this situation also exists in the development of a lack of trust in a human teammate caused by the introduction of technology with more capability and reliability. The work conducted by Waytz et al. (2014) includes a study that found test subjects were quicker to forgive a trustee's mistakes and stay calm in high-stress situations when the trustee was a technology with humanlike attributes. This work provides a foundation for the establishment of our technology trust metrics.

HAL Score

In this work we hypothesize that statistically significant differences will result in technology trust by anthropomorphizing an experience-based proxy. This hypothesis is based on leading theory used to increase cognition in students enrolled in a college-level computer architecture course. Over a period of ten years, the author of this paper provided instruction to university year-three engineering students on the topics of digital design and computer architecture. The predominant challenge reported by students in end-of-year course evaluations was difficulty synthesizing the highly complex components of a computer into a usable system. Based on student feedback, a method for reducing complexity was developed by anthropomorphizing the components of a computer. This theory provided students with the context needed to understand how the pieces of a computer function together to create a whole system. The work resulted in increased student comprehension and an ability to describe a computer from the elemental circuits up to the most advanced concepts of computer engineering such as compilers and operating systems.

To develop the measurement system needed for an experience-based technology trust proxy, we introduce the anthropomorphic technology categories of *hardware*, *algorithms*, and *links* (HAL) as illustrated in Figure 3-1.

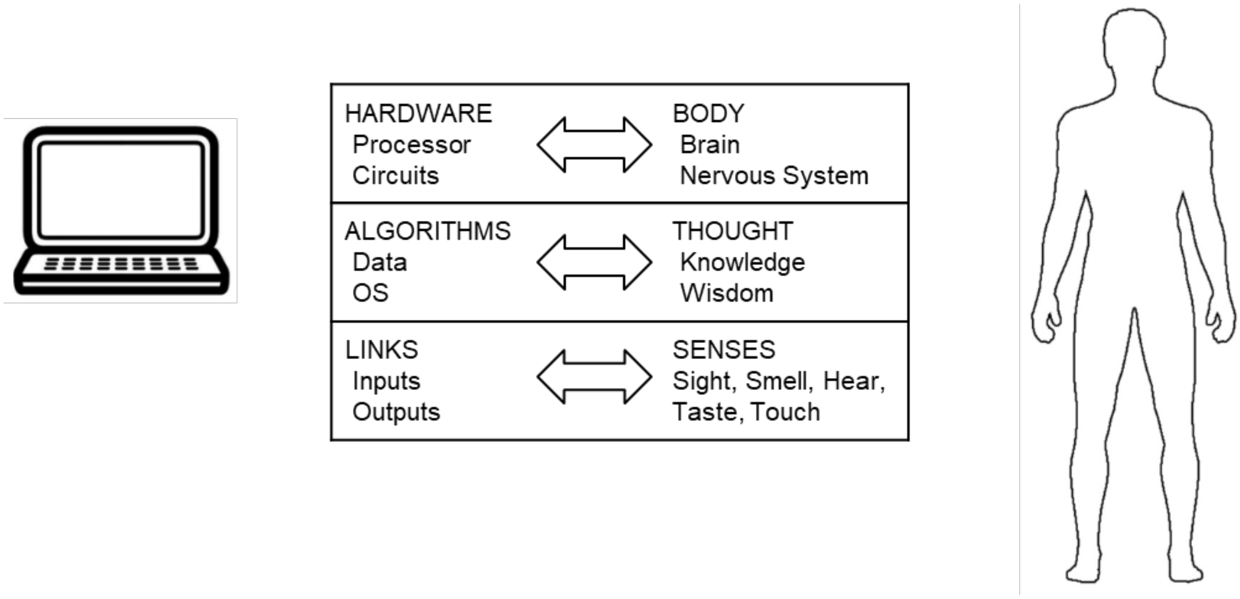


Figure 3-1. Anthropomorphic Technology Trust Metrics

In order to increase the familiarity for military end-users, the metrics are established through the HAL scoring system. The values of each HAL subsystem initially range from 0 to 100 and lead to an equally weighted maximum score of 300. This scoring system is identical to the Physical Fitness Test (PFT) employed by the United States Marine Corps. The PFT scores three physical fitness tests each scored from 0 to 100. The individual tests are pull-ups, crunches, and a 3-mile run that result in a maximum combined score of 300. Future research intends to identify weights for the HAL score that accurately reflect the overall impact on trust. For the purposes of this experiment we integrate the HAL score as a proxy for experience-based trust as shown in Figure 3-2.

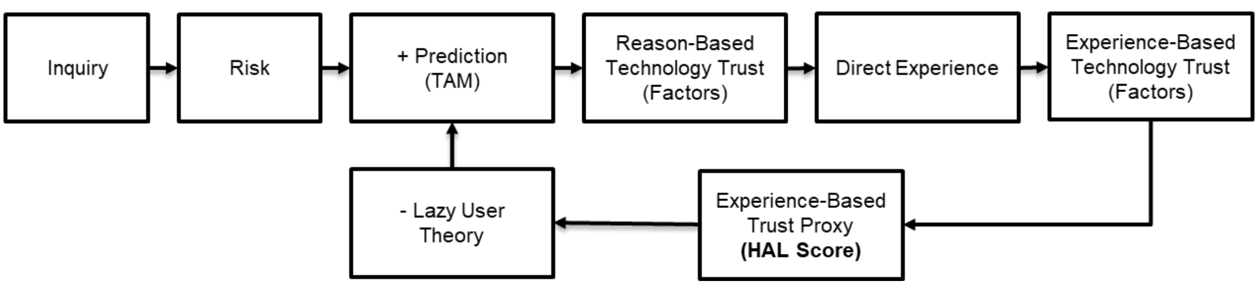


Figure 3-1. HAL Score Experimental Model

Data Analysis Plan

This study will employ Repeated Measures ANOVA. The variables in this study create a mixed design scenario. The first manipulated variable, metric, is a between-subjects factor and applies a treatment between two groups. The second manipulated variable, System Control, is a within-subjects factor, and each subject receives all three treatments of low (autonomous), medium (remote-control), and high (tethered control).

There are validity concerns due to fixed-effects seen in a repeated measure study. The participants may weight the system control variable based solely on whether or not they like the accompanying technology. To correct for such effects, techniques such as multilevel modelling may be employed in place of repeated measures analysis.

Success in this research is realized through statistically significant results leading to a new theory on the causal relationship between anthropomorphic trust metrics and the intent to use an autonomous system.

Proposed Schedule

Date	Process
Mar–April 2019	Data Collection
April–May 2019	Data Analysis
May 2019	Initial Findings
July–Aug 2019	Field Testing
Sep 2019	External Validity Data Analysis
October 2019	Final Report

Conclusion

The topic of trust in technology is increasingly important to the DoD as outlined in the Defense Science Board Study on Autonomy (David &Nielsen, 2016) that states, "There is a need to build trust in autonomous systems while also improving the trustworthiness of autonomous capabilities. These are enablers that align RDT&E processes to more rapidly deliver autonomous capabilities to DoD missions."

This work involves the introduction of novel ideas to existing theories that relate to the formation of trust. This research focuses on the impact of trust toward the adoption of autonomous systems. We have established that trust involves a user assuming some level of risk. The only literature available on technology trust involves situations that expose users to insignificant levels of risk. We posit that our research conducted on technology used in high-risk military application will reveal causality not identified in previous trust research.

References

Adams, B. D., & Webb, R. D. (2002). Trust in small military teams. In *7th international command and control technology symposium* (pp. 1–20).

Boon, S. D., Holmes, J. G., Hinde, R. A., & Groebel, J. (1991). *Cooperation and prosocial behavior*. Cambridge University Press Cambridge.

Castelfranchi, C., & Falcone, R. (2010). *Trust Theory: A socio-cognitive and computational model*. John Wiley & Sons.

Cho, J.-H., Chan, K., & Adali, S. (2015). A survey on trust modeling. *ACM Computing Surveys (CSUR)*, *48*(2), 28.

David, R. A., & Nielsen, P. (2016). *Defense science board summer study on autonomy*. Defense Science Board Washington United States.

Devitt, S. K. (2018). Trustworthiness of autonomous systems. In *Foundations of Trusted Autonomy* (pp. 161–184). Springer.

Gefen, D., Karahanna, E., & Straub, D. W. (2003). Trust and TAM in online shopping: An integrated model. *MIS Quarterly*, *27*(1), 51–90.

Hoff, K. A., & Bashir, M. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors*, *57*(3), 407–434.

Kramer, R. M., & Tyler, T. R. (1996). *Trust in organizations: Frontiers of theory and research*. SAGE.

Lankton, N. K., McKnight, D. H., & Tripp, J. (2015). Technology, humanness, and trust: Rethinking trust in technology. *Journal of the Association for Information Systems*, *16*(10), 880.

McKnight, D. H., Choudhury, V., & Kacmar, C. (2002). Developing and validating trust measures for e-commerce: An integrative typology. *Information Systems Research*, *13*(3), 334–359.

McKnight, D. H., Carter, M., Thatcher, J. B., & Clay, P. F. (2011). Trust in a specific technology: An investigation of its components and measures. *ACM Transactions on Management Information Systems (TMIS)*, *2*(2), 12.

Pak, R., Fink, N., Price, M., Bass, B., & Sturre, L. (2012). Decision support aids with anthropomorphic characteristics influence trust and performance in younger and older adults. *Ergonomics*, *55*(9), 1059–1072.

Rempel, J. K., Holmes, J. G., & Zanna, M. P. (1985). Trust in close relationships. *Journal of Personality and Social Psychology*, *49*(1), 95.

Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of Management Review*, *23*(3), 393–404.

Schaefer, K. E. (2016). A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems. *Human Factors*, 24.

Tétard, F., & Collan, M. (2009). Lazy user theory: A dynamic model to understand user selection of products and services. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on* (pp. 1–9). IEEE.

Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, *39*(2), 273–315.

Waytz, A., Heafner, J., & Epley, N. (2014). The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology*, *52*, 113–117.

Applying Agile Beyond Information Technology and Software

Terry Leary— Acquisition Principal at The MITRE Corporation—supports Intelligence Community, Defense, and Federal sponsors with their acquisition strategies and source selections. He has more than 25 years of experience in program management, acquisition, and contracting. He is a former Air Force Program Manager for major systems acquisitions. He holds an MS in Astronautical Engineering from The George Washington University and a BS in Astronautical Engineering from the United States Air Force Academy. [tleary@mitre.org]

Ginny Wydler—Chief Acquisition Scientist at The MITRE Corporation—supports federal sponsors with their acquisition strategies and source selections. She has more than 25 years of experience in acquisition, both federal government and commercial. She holds an MS in National Security Strategy, Industrial College of the Armed Forces; an MS in Acquisition, Naval Postgraduate School; and a BS in Business Administration, University of Maryland. She is a Certified Professional Contracts Manager and Fellow with National Contract Management Association. [vwydler@mitre.org]

Erin Schultz—Center for Acquisition Management and Sciences Technical Director at The MITRE Corporation—provides expertise in contracting, source selection, and acquisition strategies for a variety of customers. Schultz also served as the Vice President of Proposal Development for a Fortune 500 IT corporation. She has more than 25 years of contracting experience with both federal government and industry. She holds a BA from the University of Virginia [eschultz@mitre.org]

Sobe Ahmed—Lead Cost Analyst at The MITRE Corporation—supports sponsors with deem cost analysis for their programs. He has worked as a consultant in private industry, as a cost analyst, a construction engineer, and a project manager and civil engineer. Ahmed holds three degrees from Virginia Polytechnic Institute and State University: MS in Information Technology, an MBA, and a BS in Civil Engineering. [smahmed@mitre.org]

Abstract

This research explores where Agile practices used for software development within the information technology domain, can be successfully applied in other domains. Private industry is already applying Agile principles in manufacturing, construction, health, research and development, and finance. These approaches are captured in this research in terms of best practices and lessons learned. This research explores where accepted Agile principles and practices can accommodate organizational agility and program management environments.

This research identifies suggested guidance where Agile can be applied more broadly, especially when adopting the four Agile values.

1. Determine if Agile is a good fit.
2. Determine the scope of the Agile effort.
3. Consult with experts.
4. Tailor Agile values and principles to the situation.
5. Develop an approach.
6. Leverage lessons learned.

Introduction

This paper presents research on whether Agile development can be successfully applied to domains beyond information technology (IT) and software. This research will explore where accepted Agile values, principles, and practices have been applied in other technical domains and offers guidance for organizations to adopt Agile within their organization.

Background

Agile software development is an overarching term that encompasses numerous modern methods for software development that adhere to a common set of values and principles. When those values and principles are applied, Agile projects, programs, or organizations display shared characteristics such as collaboration; small, self-organizing, and cross-functional teams that include customers and users; adaptive planning; early and continuous delivery; and rapid and flexible response to change.

Agile Values and Principles

When discussing the history of Agile, everyone acknowledges the significance of the 2001 meeting of 17 software developers in Snowbird, UT, that led to publishing the *Manifesto for Agile Software Development* that specified four values, and the *Principles Behind the Agile Manifesto* that defined 12 principles, which are cornerstones for all Agile methods.

Agile Values (Beck et al., 2001a)

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan.

Agile Principles (Beck et al., 2001b)

1. Customer satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Deliver working software frequently (weeks rather than months).
4. Close, daily cooperation between businesspeople and developers.
5. Projects are built around motivated individuals, who should be trusted.
6. Face-to-face conversation is the best form of communication (co-location).
7. Working software is the primary measure of progress.
8. Sustainable development, able to maintain a constant pace.
9. Continuous attention to technical excellence and good design.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. Best architectures, requirements, and designs emerge from self-organizing teams.
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly.

Software development methods that are now part of the Agile umbrella pre-date the 2001 publications. Some of these methods include: Rapid Application Development (RAD) from 1991, Unified Process (UP) from 1994, Dynamic Systems Development Method (DSDM) from 1994, Scrum from 1995, Extreme Programming (XP) from 1996, and Feature Driven Development (FDD) from 1997. Today there are many methods of software development that are considered Agile. Some of the most popular methods include adaptive software development, Agile modeling, UP, disciplined agile delivery, DSDM, XP, FDD, Lean software development, Lean startup, Kanban, RAD, Scrum, Scrumban, and Scaled Agile Framework (SAFe; "Agile Software Development," 2019).

Agile methodologies did not derive from software development or IT in the first place. Many trace agile concepts back to 1930s when Walter Shewhart of Bell Labs began applying the Plan-Do-Study-Act (PDSA) cycle for product and process improvement. He taught PDSA to his mentee, Edwards Deming, who later applied it extensively in post-WWII (1950s) Japan to develop the famous Toyota Production System, which became the primary source for today's "lean" thinking. In an ironic twist, after Agile was defined, some purists refused to recognize software development based on lean as Agile until they improved their focus on customer collaboration. Ultimately, Agile methods based on lean such as Lean software development, Kanban, and their hybrids (Scrumban and Lean scrum) gained acceptance (Rigby et al., 2016).

Understanding the scrum process and terminology will be particularly useful, as it is the most widely implemented Agile framework and our research found that it served as the starting point for adaptation in most domains beyond IT and software.

Given the history that Agile does not originate from software and that it produced a wide variety of methodologies in practice today, it naturally leads to the research question—can Agile be, or has it been, adopted to acquisition domains beyond IT to improve speed of delivery and increase user satisfaction just as it has done for software?

Agile Environments

Numerous references discuss the modern business atmosphere that has evolved over the past several decades and led to conditions where Agile is optimized to excel. The International Council on Systems Engineering (INCOSE) describes an Uncertain, Unpredictable, Risky, Variable, Evolutionary (UURVE) environment that drives the need for an Agile approach (Dove & Schindel, 2016). This is nearly identical to the concept of a Volatile, Uncertain, Complex, and Ambiguous (VUCA) environment, a term popularized by the U.S. Army in the 1990s that also supports the need for Agile (Kirkpatrick et al., 2019). *The Scrum Fieldbook* makes the case that things change so rapidly, the old ways of working are breaking down and that complexity is not the exception but has become the norm (Sutherland, 2019). Henrik Kniberg, in "Agile Everywhere!" his keynote to Agile Tour Montreal, provides a simple graphic on when Agile is most needed that summarizes these concepts by comparing the spectrums of understanding on what to deliver and how to deliver it (Kniberg, 2016). There is a great deal information about when and where Agile is applicable—as in a "good fit."

The *Harvard Business Review* article "Embracing Agile" (Rigby et al., 2016) provides a thorough summary of the right conditions for Agile, outlining favorable and unfavorable conditions, as shown below.

Table 1. When Is Agile Right?

Conditions	Favorable	Unfavorable
Market Environment	Customer preferences and solution options change frequently.	Market conditions are stable and predictable.
Customer Involvement	Close collaboration and rapid feedback are feasible. Customers know better what they want as the process progresses.	Requirements are clear at the outset and will remain stable. Customers are unavailable for constant collaboration.
Innovation Type	Problems are complex, solutions are unknown, scope is not clearly defined. Product specifications may change. Creative breakthroughs, time to market is important. Cross-functional collaboration is vital.	Similar work has been done before, and innovators believe the solutions are clear. Detailed specifications and work plans can be forecast with confidence and should be adhered to. Problems can be solved sequentially in functional silos.
Modularity of Work	Incremental developments have value, and customers can use them. Work can be broken into parts and conducted in rapid, iterative cycles. Late changes are manageable.	Customers cannot start testing parts of the product until everything is complete. Late changes are expensive or impossible.
Interim Mistakes	Interim mistakes provide valuable learning.	Interim mistakes may be catastrophic.

The Government Accountability Office (GAO) Report 12-681, *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods* (GAO, 2012) identified 32 practices and approaches as effective for applying Agile software development methods to IT projects. Ten of the practices were found to be common across all five agencies reviewed:

- Start with Agile guidance and an Agile adoption strategy.
- Enhance migration to Agile concepts using Agile terms, such as user stories (used to convey requirements), and Agile examples, demonstrating how to write a user story.
- Continuously improve Agile adoption at both the project level and the organization level.
- Seek to identify and address impediments at the organization and project levels.

- Obtain stakeholder/customer feedback frequently.
- Empower small, cross-functional teams.
- Include requirements related to security and progress monitoring in your queue of unfinished work (the backlog).
- Gain trust by demonstrating value at the end of each iteration.
- Track progress using tools and metrics.
- Track progress daily and visibly.

These two references from HBR and GAO point to the need for conditions to be right for Agile principles to be adopted and integrated into the business atmosphere to be successful.

Agile Applied to Management and Organizations

Agile Management

The report "Unleashing the Power of Small, Independent Teams" from the *McKinsey Quarterly*, July 2018, proposes that the key to Agile management is found in the enactment of small independent multi-functional teams that are given a specific task/goal/product that matters greatly to the customer, empowered to self-organize and make decisions, and provided with the tools to move fast. Further, the teams must be supported by organizational management that takes on roles and responsibilities quite different from the traditional model. Management must learn to direct teams to the best opportunities, define outcomes and let teams chart their own course, staff teams with the best personnel, oversee with a light touch, focus on removing roadblocks and providing support, ensure proper governance, provide teams a clear view of the customer, allocate resources up front and hold teams accountable, and commit to retraining/coaching managers for their redefined roles (Bossert et al., 2018).

Organizational Agility

Organizational Agility is a term used to refer to the idea of Agile management. The MITRE Corporation, Enterprise Strategy and Transformation Technical Center, developed a model for Organizational Agility (Kirkpatrick et al., 2019a). The model defines Organizational Agility as, "the capacity to adapt quickly and effectively in response to or in expectation of changes in the organization's environment" (Kirkpatrick et al, 2019a, p. 4). This work found that attributes of traditional organizations include: a hierarchical structure, leaders that sense/scan the environment every 2 or 3 years, decisions are made by those at the top, a less educated workforce is required, less customer input, and strategy is set at the top while power trickles down. These traditional characteristics no longer work well in the VUCA environment most businesses faced in the late 20th century. The Organizational Agility model found that three routines best characterize Agile organizations in the VUCA environment: sensing, interpreting, and responding. The Organizational Agility Model applies the sensing, interpreting, and responding routines across seven organizational dimensions, and compares the varying approaches between traditional and Agile organizations (Kirkpatrick et al., 2019b).

Table 2. Organizational Agility Model

Organizational Dimensions	Traditional Organizations	Agile Organizations	
		Stability	Flexibility
Organizational structure	Hierarchical structure based on function	Flat organizational structure with permanent teams	Temporary teams form, re-form, and dissolve when needed
Knowledge sharing and experimentation	The environment is scanned annually by leaders	Routine processes are in place to scan the environment, widely share and store information	Employees reflect on information to make sense of it and determine the impact to the mission
Decision making	Leaders make decisions and solve problems	Decisions are communicated upward to leaders, who provide resources and oversight	Decisions are made at the lowest level where expertise resides
Leader actions	Leaders restrict information sharing and inhibit learning	Leaders are informed about decisions made at lower levels	Leaders encourage employees to share what they learn
Process management	Processes are either overly rigid or ill-defined	Some processes are well-defined and continually improved	Some processes are flexible; experiments are conducted to facilitate learning and innovation
Roles	Roles are ill-defined or not flexible	Some roles are stable and clearly defined	Some roles are intentionally flexible to quickly meet changing conditions and customer needs
Norms and expectations	Norms are to retain information, not collaborate, and view learning as a cost	Leaders create climate of psychological safety. Employees are expected to view “mistakes” as learning, share information, and collaborate. Learning and training are viewed as an investment; employees are trained proactively in anticipation of future skill needs.	
Overall	Respond to change slowly, ineffectively	Anticipate as well as respond to changing environmental conditions to carry out their missions quickly and effectively	

There are two important strengths in this model. First, it acknowledges that some aspects of most organizations are best suited for a stable approach that is closer to traditional than Agile. The model allows for those areas to be tailored for each organization on a case-by-case basis. By not requiring an “all or nothing” commitment to Agile, the framework mitigates some of the key barriers for those contemplating the use of Agile (e.g., inertia, risk of breaking already efficient processes, and time to train and develop necessary skills). Secondly, in an Agile organization, personnel in authority support these routines by providing access to resources and by removing obstacles, but most important, leaders at all levels must provide psychological safety.

Psychological safety is like trust, and both terms are used in describing attributes of Agile. When psychological safety is not present, team members are more reluctant to take risks, ask questions, or offer new ideas for fear of embarrassment or reprisals (Kirkpatrick et al., 2019b). The concept of psychological safety leading to the absence of fear is further supported and elaborated on in *The Scrum Fieldbook*. Sutherland feels so strongly about the need for psychological safety (and trust) that he addresses the importance in three of his five scrum values: openness, respect, and courage. He also emphasizes its importance in empowering personnel to highlight problems. Sutherland supports a principle of the Toyota Production System that not having any problems is the worst problem because they always exist and not knowing what they are makes it impossible to fix them (Sutherland, 2019, pp. 110–112 and 131–136).

Agile Applied to Other Domains

Manufacturing

Aircraft—Saab Fighter Jets

Saab made the commitment to build the Gripen E fighter jet using Agile practices that are "implemented at every level and in every discipline: software, hardware and fuselage design. The Saab Agile framework contains practices from Lean, Scrum, Kanban, XP and others" (Furuhjelm et al., 2015). This is one of few examples of applying Agile to the design and manufacturing domain writ large and not just in the software development component of a production item. The following describes a summary of the characteristics of the Saab Agile framework as derived from two sources: Team Autonomy, Common Iterative Project Master Plan, Continuous Improvement, Clarity, Commitment, Transparency, Continuous Delivery, Prioritization, Living Strategic Plan (Furuhjelm et al., 2016; Sutherland & Justice, 2020).

Saab's Agile framework includes several catalysts (additional characteristics) atypical to scrum but critical toward implementing Agile in the aircraft manufacturing domain: Modular Architecture, Modeling and Simulation, and Co-located Pilots.

Result: The use of Agile on all aspects of the Gripen E program resulted in a fighter jet with 50 times lower development costs compared with the F-35, 10 times lower unit costs compared with the F-35, and the lowest cost per flight hour of any modern fighter jet (Kniberg, 2016).

Motor Vehicles—WIKISPEED and Tesla

The **WIKISPEED SGT01** car was an entrant in the Progressive Insurance Automotive X Prize contest. The contest goal was to produce a full-sized, road-legal car, getting 100+ mi/gal (2.25L/100 km). WIKISPEED used an Agile design approach that demanded that modules were loosely coupled and could be tested apart from the whole. The WIKISPEED car developed major sub-assemblies, such as suspension, motor, and body, that can be replaced in the time it takes to change a flat tire. The modularity supported many of the design and process principles. It allowed for rapid iterations and experimentations during development and testing. The loosely coupled components, with simple and well-defined interfaces, minimized system interdependencies that enabled more rapid and reliable software and hardware development. While most of WIKISPEED's Agile implementation adapted standard principles, one atypical aspect is that WIKISPEED was able to complete the work using a distributed team (Socha et al., 2020).

WIKISPEED Result: WIKISPEED successfully built one prototype to prove concepts using Agile principles. The Agile methods resulted in a product cycle times measured in weeks, not years (as compared to traditional motor vehicles). The company is working on other prototypes for a mail delivery vehicle, next generation taxicab, ultra-light racing vehicle, and five-seat sedan for ARPA-E's (Advanced Research Projects Agency-Energy) LiteCar Challenge.

Tesla's Agile approach is disrupting the automobile industry by shifting the core of its products from mechanics to software, and the company is using that advantage to skip beta testing on its latest vehicle, Model 3, and going straight into production. Antonio Patti, a digital strategist at the Transport Company of Milan, wrote in a 2017 blog post:

> "Skipping the beta testing for a traditional car-maker is a sort of suicide, but if you are Tesla it is the first application of its agile car development framework approach.

> Tesla's cars are different from the other cars. Its engines, chassis, interiors and all the other components are way simpler [than] other cars so, once tested and standardized, they don't need to be tested again for all the models." (Patti, 2017)

The data collected from the same hardware used previously on the Model S and Model X is sufficient to justify going directly to production and skip beta testing.

Tesla no longer views the Model 3 as a new concept car that requires beta testing but instead as the next company model, an iteration of the product where the important hardware parts have already been tested and the easily replaceable components and less critical functions are still in development (Patti, 2017).

Tesla Result: Tesla shifted its products from mechanics to software to skip beta testing and go straight into production. This approach allows enacting product updates and mechanical improvements through updated software directly to the customer, avoiding extensive recalls and utilizing large amounts of data drawn from customers to continually improve products.

Automotive Parts, Power Tools, and Agriculture Sensors—Bosch

The example of Bosch's transition to an Agile company is included in this research not because it contains any unique methods but due to the sheer size and diversity of the company. Bosch is an enormous company with 390,000 employees located across 60 countries that has been in existence since 1886 and is well known as an innovator. In May 2017, when commenting on why Bosch was turning to Agile across all parts of its business, CEO Volkmar Denner said, "For Bosch [A]gility is crucial, it allows us to adjust to the increasing speed of change around us. Agility allows us to remain in a position as an innovation leader" (Howard, 2017). Denner realized that the world environment had radically changed and his company needed to react faster, provide more customized solutions, and be more connected (everything is part of the Internet of Things). He is clearly underscoring the UURVE and VUCA environments introduced earlier as driving factors for why every program/project or organization should be considering Agile. Bosch homed in on the approach of implementing Agile as a leadership-driven transformation.

Result: Bosch cut its automotive parts development time in half; for home and garden power tools their innovation increased and employee engagement improved; and for agricultural sensors, 10 new innovations were developed in 1 month when it traditionally took 6 to 8 months for each innovation (Howard, 2017). Perhaps, more important, Bosch provided some valuable lessons learned through its two false starts and demonstrated that with persistence and commitment, Agile could be applied even in the largest, most widely dispersed companies.

Shipbuilding and Submarines—GDEB and HII/NNS

General Dynamics Electric Boat (GDEB) and Huntington Ingalls Industries' Newport News Shipbuilding (HII/NNS) both have achieved success in manufacturing surface ships and submarines by applying Agile principles. A *Naval Engineers Journal* article and several GAO reports provide excellent detail on recent efforts to adapt Agile principles into shipbuilding and submarine manufacturing processes. For completeness, the following list of comprehensive reports providing extensive industry reviews is provided:

- "Benefits and Challenges of Implementing Agile Development in Modular Shipbuilding," *Naval Engineers Journal*, June 2019 (Castell et al., 2019)
- "Research on Systematization and Advancement of Shipbuilding Production Management for Flexible and Agile Response for High Value Offshore Platform," *Journal of Naval Architecture and Oceanographic Engineering*, September 2011 (Song et al., 2011)

- "Can a Shipyard Work Towards Lean Shipbuilding or Agile Manufacturing?" *Sustainable Maritime Transportation and Exploitation of Sea Resources* (Alves de Moura & Botter, 2012)

Result: Numerous reports cite excellent details on recent efforts in shipbuilding and submarine manufacturing to adapt Agile principles in their processes.

Construction

Uranium Enrichment Plant—Centrus Energy Corp

Construction projects are generally considered poor candidates for Agile because they are thought to require sequential phases (i.e., initiation/planning, design, construction, testing, turnover, and closeout) and any changes in the plan get progressively more expensive later in the life cycle. Centrus Energy Corp. elected to apply Agile concepts to a construction project to build the next generation U.S. uranium enrichment technology. The project required building, installing, operating, and testing a large cascade of machines and centrifuges. Centrus chose to use Agile because there was a mandated aggressive schedule and funding was being provided in tranches, both of which rendered the traditional construction approach unworkable. To implement Agile, Centrus effected numerous changes not part of conventional construction.

Result: Centrus met an aggressive schedule, was under budget, had no safety issues or deficiencies from any oversight and regulatory agencies, and had a significantly higher than average productivity for a nuclear project or even for a less complex construction project (Straçusser, 2015).

Prefabrication—PCL Constructor

PCL Constructor, Canada's largest general contractor company, in 2012, formed PCL Agile to implement Agile principles including prefabrication as a form of incremental development. PCL Agile now regularly provides offsite modular construction in large warehouses to assemble smaller components and larger built-up assemblies, to overcome site logistical challenges where lay-down (working real estate) areas are scarce (Caulfield, 2018). Many vertical construction projects are located within dense downtown areas that lack ample space necessary for construction staging and equipment maneuvering. More architects are using prefabricated solutions to ensure projects are completed on time while mitigating unforeseen change conditions. For PCL Agile, this approach led to better site access in the field, improved quality control within a climate-controlled environment, reduced congestion and trade stacking at the site, increased worker safety, decreased waste due to reusable parts, and reduced transportation costs, which strengthened contractor and customer relationships.

Result: PCL Agile now regularly provides offsite modular construction in large warehouses to assemble smaller components and larger built-up assemblies, to overcome site logistical challenges where lay-down (working real estate) areas are scarce.

Construction Project Management—Construction Blog

In a NTaskmanager.com blog post, Fred Wilson notes that despite not seeming to be a good fit for construction, Agile is proving its effectiveness and efficiency in this industry. Wilson sees Agile principles already being incorporated and feels the existing roles in construction can easily map to Agile roles—for example, the superintendent acting as a scrum master coordinating work and ensuring task completion, the project manager acting as the product owner prioritizing tasks and interfacing with the customer, and workers completing the physical tasks acting as the development team. Project management software is now commonly used in construction, allowing real-time collaboration (between architects, engineers, subcontractors, owners/clients), monitoring progress, facilitating quicker decisions, and saving costs in materials and overtime (Wilson, 2018).

Rachel Burger observed that Agile construction is best suited for the pre-design and design phases where customer involvement can be increased and the complex project can be broken into easier-to-manage subprojects, enables collaboration, sharing of information, re-prioritization of tasks due to unforeseen circumstances, coordination for major installs between all trades affected, and feedback for continuous improvement on productivity and profitability (Burger, 2019).

Result: Agile construction also applies to project management where the roles and responsibilities must be redefined. It has already made its way into project management software tools, allowing for real-time collaboration, and affords the potential to improve speed, reduce costs, and increase customer satisfaction.

Health

Big Data—3M HIS

Daniel Keys Moran is a computer scientist, novelist, and big data expert who popularized the saying, "You can have data without information, but you can't have information without data" (Sutherland, 2019). 3M Health Information System (HIS) is a research and development (R&D) program that is trying to address the big data problem of how to turn huge amounts of data into actionable information for hospitals, insurance companies, and health plans. 3M HIS integrates inputs such as doctors' notes, lab reports, and demographic data to support proactive versus reactive actions. In May 2015, 3M was faced with the daunting task of updating its system from International Statistical Classification of Diseases and Related Health Problems Version 9 (ICD-9) with ~14,000 codes to ICD-10 with ~141,000 codes within 6 months (by October 2015) and the efforts were not going well. 3M HIS had ~5,000 customers that used its system daily to determine the correct codes and properly reimburse hospitals and clinics from insurance companies, so the program decided to adopt Agile and hire Scrum Inc. to consult. 3M met the deadline and continued to improve its adoption of Agile over the years.

Result: 3M recognized that it needed to change to meet an aggressive schedule and that it did not possess the necessary expertise to make the needed change. 3M used an expert consultant to help prioritized the work, instituted stabilized teams, worked off the backlog, reduced interruptions, and changed the scrum process to improve velocity, and successfully meeting deadlines.

Mental Health—Monash Health

Monash Health is the largest public health service provider in Melbourne Australia. Dr. Melissa Casey, Director of Psychology at Monash Health, led the change to Agile after realizing their system was broken from observing her teams struggle to meet key performance indicators for emergency care and measuring a significant increase in patients presenting in crisis to emergency rooms. In 2013, after several weeks of design, clinician workshops, and collaborating with Agile consultants the first prototype Agile Psychological Medicine Clinic (APMC) opened. The methods APMC found most useful were adopting an Agile ethos, using Agile consultants, developing flexible treatment programs, and instituting rapid delivery of care (Faucher, 2019).

Result: APMC was so successful that Dr. Casey and her team won a State-level award for delivering innovative mental health care the first year after opening. In addition, four more clinics modeled on the prototype opened near other hospitals. Over a 12-month period patient data showed a 23%–46% improvement on several standardized measures following treatment, a significant decrease in the number of clinicians and case managers involved in each case (handoffs), and an increase in clinician job satisfaction.

Research and Development

Laboratories/Academic Research—Broad Institute of MIT/Harvard

The Broad Institute of Massachusetts Institute of Technology (MIT) and Harvard is one of the world's leading genomic research centers, with over 1,300 employees across 30 academic laboratories, each with a different research focus but all working to enhance health and human medicine. Within each laboratory are numerous teams that share a common research theme but conduct individualized experiments. At Broad, Kendra West co-founded AgileAcademia a group of more than 80 members across 22 teams that look to improve team function through Agile practices.

West and her AgileAcademia group identified several methods that helped their teams embrace Agile. The methods they highlighted were introducing Agile in a comprehensible manner because academics have a hard time getting started and try to understand the whole process and avoiding using intimidating Agile terminology (e.g., project reflecting instead of retrospective and check-in vs standup). With some effort, the same intentions behind Agile were applied for great success in the R&D environment (West, 2018).

Result: Broad Institute identified typical challenges R&D laboratories adopting Agile would face. They found methods to address roadblocks and successfully incorporated Agile values.

Innovation—3M HIS

Another R&D example echoes many of the same tips recommended by the Broad Institute. This example comes from the 3M HIS work previously described under the health domain. 3M HIS is made up of dozens of Agile teams and some of those teams also work in the R&D environment where they must design, create, prototype, and refine new and innovative approaches. These teams are faced with the dilemma of how to best adapt Agile to research projects. They discovered implementation was not insurmountable, given that the research process is already naturally iterative with many false starts and incomplete finishes.

The 3M HIS R&D teams decided to customize the SCrum fOr REsearch (SCORE) method proposed by Hicks and Foster (2010). The techniques that worked for them included reducing the daily scrum to two meetings per week and eventually to one per week and making all other meetings "on demand" between team members and scrum master to troubleshoot or review results (Butterfield & LaBrec, 2019; LaBrec & Butterfield, 2016).

Result: The 3M HIS R&D teams adopted a more efficient use of time for faculty, improved both morale and productivity for student researchers due to the transparency and accountability, and generated a group identity for a research community for shared knowledge to the benefit of all.

Product Research—HPE

Ruly Weisbach was the R&D director at Hewlett-Packard Enterprises (HPE) in 2015 and led six R&D groups through the transition to Agile. His experience in the R&D environment was that early on the teams used the terminology, held the meetings, and followed processes but were still not Agile. In the R&D domain, his teams tended to start on too many features at once, so the work in progress (WIP) was always overloaded, making it impossible for a product owner to get feedback and react to it, and the list of features would take years to deliver.

For HPE R&D, Weisbach tackled this by (a) reviewing the backlog with the team and ruthlessly removing items that didn't fit the definition for a minimum viable product (MVP), which they defined as the top 5 themes and 50 features; (b) structuring the organization by features not domains (functional expertise); (c) tracking features in addition to user stories so that teams were consistently delivering new functionality to the customer; and (d) planning short term for MVP, not long term on features that were not the priority, which kept the backlog manageable and prevented feature creep (Weisbach, n.d.).

Result: The HPE solution for Agile R&D was to force all teams to focus on prioritizing the backlog and delivering the most important feature first, which is identical to the swarm attribute discussed in *The Scrum Fieldbook* about the 3M HIS work.

Finance—ING

ING started a pilot Agile transformation in 2010 that was limited to three IT teams. Those teams showed enough success to convince the company to adopt it across the entire software development department in 2011. By 2014, executives recognized the success in software and IT but noted it had not translated to business gains (Siroky, 2020). That realization was the motivation for ING's full implementation of Agile. Starting in 2014, ING spent 8 to 9 months examining what worked at companies such as Zappos, Google, and Spotify and planning how to adapt those concepts for ING. The full transition started in June 2015 when ING started focusing on the customer journey (i.e., an individual may go to a branch for financial advice but wants to go online to make the investment) or ING risked becoming irrelevant.

A McKinsey & Company article (Schlatman & Jacobs, 2017) based on interviews with the chief information officer (CIO) and chief operating officer (COO) provides the best insights into their reasons to move to Agile, the ideas they adapted from other companies, how they implemented them, and the challenges and risks they faced. Some of the main points of the article are summarized below, but the full source is highly recommended for anyone contemplating an organizational shift to Agile.

ING adapted a squad/tribe/chapter construct from Spotify. To apply Agile at scale across the company, ING organized its ~2,500 employees into 350 squads no larger than 9 people, all co-located, formed to work toward a specific client-related objective with end-to-end responsibility, and ready to dissolve once its mission was complete. The squad/tribe/chapter approach is how ING adapted Agile for its own use, but even more important are some of the adaptations and observations the company made in applying Agile company-wide beyond IT and software (Perkin, 2017; Schlatman & Jacobs, 2017).

Result: ING has been quicker to market, has increased employee engagement, releases software every 2 to 3 weeks versus 5 to 6 times per year previously, and has increased customer satisfaction and employee engagement scores.

Lessons Learned from Adopting Agile

Table 3 presents a synopsis of the successful results of the domain research on companies that adopted Agile into their processes and methods from the Agile Applied to Other Domains section of this paper. Most companies improved quality, decreased production time, and/or reduced costs. They also improved collaboration across the organization, breaking down barriers to communication, which resulted in increased employee satisfaction. They generally improved their viability and relevance within the market under changing market or technology conditions, enabling company longevity.

Table 3. Agile Application Lessons Learned

Company	Agile Results
Saab	The use of Agile on the Gripen E program resulted in a fighter jet with 50 times lower development costs compared with the F-35, 10 times lower unit costs compared with the F-35, and the lowest cost per flight hour of any modern fighter jet.
WIKI SPEED	Successfully built one prototype to prove concepts using Agile principles. The company is now working on other prototypes: mail delivery vehicle, next generation taxicab, ultra-light racing vehicle, and five-seat sedan for ARPA-E's LiteCar Challenge.
Tesla	Shifted products from mechanics to software to skip beta testing and go straight into production. Enacting product updates and mechanical improvements through delivery of updated software directly to the customer, avoiding extensive recalls to continually improve products.
Bosch	Cut development time in half; divided into cross-functional teams leading to increase innovation and improve employee engagement; developed 10 new innovations in 1 month when it traditionally took 6 to 8 months for each innovation.
Shipbuilders	Numerous reports cite excellent quality improvements in shipbuilding and submarine manufacturing by adapting Agile principles.
Centrus	Met an aggressive schedule, under budget, with no safety issues or deficiencies from oversight and regulatory agencies. They had a significantly higher than average productivity for a nuclear construction project or a typical less complex construction project.
PCL	PCL Agile now regularly provides offsite modular construction in large warehouses to assemble smaller components and larger built-up assemblies, to overcome site logistical issues.
Construction Projects	Agile has already made its way into the construction domain through software tools and real-time collaboration to improve speed, reduce costs, and increase customer satisfaction.
3M HIS	3M prioritized its work, instituted stabilized teams, reduced backlog, reduced interruption, and changed its scrum process to improve velocity.
Monash Health	So successful four more clinics modeled on the prototype opened near other hospitals. Over a 12-month period patient data showed a 23%–46% improvement on several standardized measures following treatment, a significant decrease in the number of clinicians and case managers involved in each case (handoffs), and an increase in clinician job satisfaction.
Broad Institute of MIT/Harvard	Incorporated tailored Agile values into its laboratory environment; utilized flexible sprints and adopted the right tools and techniques to fit each research lab space.
3M HIS R&D	3M HIS R&D teams adopted a more efficient use of time for faculty, improved both morale and productivity for student researchers with transparency and accountability, and generated a group identity for a research community and shared knowledge to the benefit of all.
HPE	The HPE solution for Agile R&D allowed all teams to prioritize the backlog and delivering the most important feature first. This approach used the swarm attribute referenced in *The Scrum Fieldbook*.
ING	Quicker to market, increased employee engagement, and released software every 2 to 3 weeks versus 5 to 6 times per year. Customer satisfaction and employee engagement scores are all up.

Apply at scale: The benefits of adopting Agile will not extend throughout a company and business units unless Agile is applied at scale. If Agile can be applied on a small scale to an individual project/program or organizational unit, the benefits will be limited to that scope. If a company decides to adopt Agile at scale, the change needs to be leadership driven, applying the Agile management information presented in this research.

Organizational adoption: Organizations should use Agile principles to transform their company and decide how to apply Agile. They need to outline core principles and transform roles to define outcomes. A company that draws a detailed plan, schedule, and milestones for Agile migration, falls into the trap of waterfall project management while attempting to become Agile.

Teams: Organizations need to release the power of small cross-functional teams with clear purposes, close connections to the end user, and process iteration with short feedback cycles (sprints). Scrum is the most widely used form of Agile, consistently at the top of the State of Agile™ survey, with ~72% of respondents saying they practice Scrum or a hybrid Scrum (VersionOne, 2019).

Experts: Organizations should consider using an Agile outside consultant, rather than trying to adopt Agile alone. Companies can learn from their expertise and experience.

Common Characteristics of Adopting Agile

The research showed several common characteristics of companies that successfully applied Agile in their domains. These characteristics reflect on the Agile values or principles. When possible, companies adopting Agile should cultivate these common characteristics such as: a) iterative, b) adaptive and flexible, c) collaborative, small teams, d) user/customer focus, e) visual progress, f) short feedback loop, g) continuous improvement, h) trust/psychological safety, i) simplify, j) change in leadership roles, and k) build in quality.

Conclusion

Our research concludes that Agile can be applied successfully across many domains. Several studies and surveys have confirmed that notion. *The Scrum Fieldbook* references surveys. CHAOS Report from Standish Group since 1985 covered many companies and showed that Agile projects are successful/are challenged/fail at 42%/50%/8%, compared with 26%/53%/21% for traditional projects, or a 60% greater success rate and 260% lower failure rate (Sutherland, 2019, p. 40).

The benefits of adopting Agile development practices are clear. Between 2016 and 2019, thousands of Agile practitioners from companies and countries were surveyed, and 50%–85% of those surveyed identified the following benefits (in approximate descending priority): a) user/stakeholder engagement, b) allows for change, c) transparency/visibility, d) faster time to market, e) increased team productivity, f) project predictability (deliveries, cost, and schedule), g) improved team morale, h) business and IT alignment, i) improved quality, and j) reduced risk (fail early; VersionOne, 2015, 2019).

The research concludes that an organization or program can adopt Agile in a domain beyond IT and software. There are no definitive answers for the optimum approach. The following general guidance for implementing Agile is provided:

- **Determine if Agile is a good fit**: There is no point starting down the path to Agile if an organization cannot meet or otherwise substitute for core values and principles such as access to users and the ability to iteratively deliver a useable product (see the Background section).
- **Determine the scope of the Agile effort**: Agile can be applied to all or portions of an organization or program. If applying at scale, Agile management is a necessary component (see the Agile Applied to Management and Organizations section).
- **Consult with experts**: Organizations are unlikely to have Agile experts on hand ready to assist with this journey. Organizations should make sure they have access to experts (*The Scrum Fieldbook* and Agile experts to consult with such as those identified in the Agile Applied to Other Domains section).
- **Tailor Agile values and principles to the situation**: This sounds simple but is where organizations or programs need to put the most thought (see the Background section).
- **Develop an approach**: Remember not to fall into the trap of developing a detailed plan and milestones; instead, use Agile to iterate and refine the approach (see the Agile Applied to Other Domains section).

- **Leverage lessons learned**: Use research, the references provided, and other organizations that already transitioned to Agile (see the Lessons Learned from Adopting Agile section).

Federal agencies face different challenges than private industry when adopting agile. Many constraints, such as budget authorizations, program approvals, leadership change, and different agendas can impact full adoption of Agile, and certainly influence scale. Some of these challenges as well as best practices are included in the GAO Report 12-681. We suggest that Federal Agencies review these items as they adopt Agile practices within their organizations.

References

Agile Software Development. (n.d.). In *Wikipedia*. Retrieved October 11, 2019, from https://en.wikipedia.org/wiki/Agile_software_development

Alves de Moura, D., & Botter, R. C. (2012). Can a shipyard work towards lean shipbuilding or agile manufacturing? In *Sustainable maritime transportation and exploitation of sea resources*. Taylor & Francis Group.

Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Ward, C., Kern, J., Thomas, D., Fowler, M., & Marick, B. (2001a). *Manifesto for agile software development*. https://agilemanifesto.org/

Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Ward C., Kern, J., Thomas, D., Fowler, M., & Marick, B. (2001b). *Principles behind the agile manifesto*. https://agilemanifesto.org/principles.html

Bossert, O., Kretzerg, A., & Laartz, J. (2018, July). Unleashing the power of small, independent teams. *McKinsey Quarterly*.

Burger, R. (2019, October 22). Agile construction management. *The Balance Small Business*.

Butterfield, R., & LaBrec, P. (2019, June 21). The return of agile methods for research. *Inside Angle*.

Castelle, K., Dean, A., & Daniels, C. (2019, June). Benefits and challenges of implementing agile development in modular shipbuilding. *Naval Engineers Journal*.

Caulfield, J. (2018, August 6). More contractors are turning to offsite production for speed and quality. *Building Design + Construction*.

Dove, R., & Schindel, B. (2016, July). Agile systems engineering process features collective culture, consciousness, and conscience at SSC Pacific unmanned systems group. In *Proceedings from 26th Annual INCOSE International Symposium*.

Faucher, C. (2019, February 4). Agile outside IT: Being well, being agile. *ScrumAlliance*.

Furuhjelm, J., Segertoft, J., Justice, J., & Sutherland. J. J. (2015, September). *Owning the sky with agile: Building a jet fighter faster, cheaper, and better with scrum.* Scruminc.com.

GAO. (2012, July). *Software development: Effective practices and federal challenges in applying agile methods* (GAO Report 12-681).

Hicks, M., & Foster, J. (2010, September 18). *Adapting scrum to managing a research group* [University of Maryland, Technical Report #CS-TR-4966].

Howard, A. (2017). *Case study: Scrum@Scale at Bosch: Embracing agility.* Scrum@Scale.

Kirkpatrick, S., Miller, S., Terragnoli, A., & Sprenger, A. (2019a, November). *Development of an organizational agility assessment for government and non-profit organizations* [MITRE Public Release #19-03347-5].

Kirkpatrick, S., Miller, S., Terragnoli, A., & Sprenger, A. (2019b, November). *MARS agency player application guide briefing, "A serious game to overcome disruptive change"* [MITRE Public Release #18-03570-5].

Kniberg, H. (2016, November 16). *Agile everywhere!* [Keynote to Agile tour Montreal].

LaBrec, P., & Butterfield, R. (2016, June 28). Using agile methods in research. *Inside Angle*.

Patti, A. (2017, May 3). *Tesla Model 3 agile car development framework* [Blog].

Perkin, N. (2017, October 25). *Agile transformation at ING: A case study*. Agile Business Manifesto.

Rigby, D., Sutherland, J., & Takeuchi, H. (2015, April 20). The secret history of agile innovation. *Harvard Business Review*.

Rigby, D., Sutherland, J., & Takeuchi, H. (2016, May). Embracing agile. *Harvard Business Review*.

Schlatman, B., & Jacobs, P. (2017, January). ING's agile transformation. *McKinsey Quarterly*.
Siroky, D. (2020, March 6). *Agile and DevOps are failing in Fortune 500 companies*. Plutora Blog.
Socha, D., Folsom, T., & Justice, J. (n.d.). *WIKISPEED—Applying agile software principles and practices for fast automotive development.* Business Agility Institute. Retrieved March 27, 2020, from https://businessagility.institute/learn/wikispeed-applying-agile-software-principles-and-practices-for-fast-automotive-development/
Song, Y.-J., Woo, J.-H., & Shin, J.-G. (2011, September). Research on systematization and advancement of shipbuilding production management for flexible and agile response for high value offshore platform. *Journal of Naval Architecture and Oceanographic Engineering*, *3*(3), 181–192.
Straçusser, G. (2015, October 10). *Agile project management concepts applied to construction and other non-IT fields* [Paper presentation]. PMI® Global Congress 2015—North America, Orlando, FL, United States.
Sutherland, J., & Schwaber, K. (2017, November). *The scrum guide™*. Scrum.org.
Sutherland, J. J. (2019). *The scrum fieldbook: A master class on accelerating performance, getting results, and defining the future*. Random House.
Sutherland, J. J., & Justice, J. (n.d.). *Agile in military hardware: How the SAAB Gripen became the world's most cost-effective military aircraft.* Scrum Alliance. Retrieved March 27, 2020, from https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Global%20Scrum%20Gatherings/2017%20San%20Diego/Presentations/JusticeJoe_Agile_In_Military_Hardware_SGSD.pdf
VersionOne. (2015). *10th annual state of Agile™ report*. VersionOne.com.
VersionOne. (2019, May 7). *13th annual state of Agile™ report*. CollabNet.
Weisbach, R. (n.d.). *How I led 6 R&D groups through an agile transition.* TechBeacon. Retrieved March 27, 2020, from https://techbeacon.com/app-dev-testing/how-i-led-6-rd-groups-through-agile-transition
West, K. (2018, August). *Reinventing research: Agile in the academic laboratory*. Agile 2018 Conference, Agile Alliance.
Wilson, F. (2018, October 15). Agile construction project management—How to be an agile construction team? *NTaskmanager.com*.

A Comparative Analysis of Advanced Methodologies to Improve the Acquisition of Information Technology in the Department of Defense for Optimal Risk Mitigation and Decision Support Systems to Avoid Cost and Schedule Overruns

Thomas J. Housel—is a tenured, full professor for the Information Sciences (Systems) Department of the U.S. Naval Postgraduate School. He received his PhD from the University of Utah in 1980. He specializes in valuing intellectual capital, knowledge management, acquisitions research, telecommunications, information technology, value-based business process reengineering, and knowledge value measurement in profit and nonprofit organizations. He has managed a $5 million+ portfolio of field studies, educational initiatives, and industry relationships. His current research focuses on determining the most promising methodologies for improving the acquisitions life cycle and developing a physics-based model to predict revenue growth in knowledge-intensive companies.

Johnathan Mun—is a research professor at the Naval Postgraduate School and teaches master's and doctoral courses as well as executive seminars in quantitative risk analysis, decision sciences, real options, simulation, portfolio optimization, and other related concepts. Mun received his PhD in finance and economics from Lehigh University. He also has an MBA in business administration, an MS in management science, and a BS in biology and physics. He has taught at universities all over the world and is considered a leading world expert on risk analysis/real options analysis with over 23 patents. He has authored over 22 books.

Raymond D. Jones—is a professor of the Practice in Acquisition Management at the Graduate School of Defense Management at the Naval Postgraduate School. He retired as a colonel from the U.S. Army in 2012. His last assignment in the Army was as the Deputy Program Executive Officer for the Joint Tactical Radio System (JTRS). He has a BS degree in aerospace engineering, an MS in aeronautical engineering from the Naval Postgraduate School, an MBA from Regis University, and an MA in national resource strategy from the Industrial College of the Armed Forces. He is currently an IS PhD candidate at the Naval Postgraduate School.

Timothy R. Shives—is a DoD civil servant in the field of information technology (IT) management serving at the Defense Manpower Data Center. He is also an officer in the United States Marine Corps and has served more 15 fifteen years in both the active duty and reserve components. He holds a Doctor of Education (EdD) and an Education Specialist (EdS) postgraduate degree, an MS in information technology management, an MBA from the Naval Postgraduate School, and an MA in National Security and Strategic Studies from the U.S. Naval War College. Shives is currently an IS PhD student at the Naval Postgraduate School.

Benjamin Carlton, Major, USMC—is an active-duty Marine officer currently serving as a project officer for the Global Combat Support Systems-Marine Corps (GCSS-MC) at the Marine Corps Systems Command. Carlton contributed to this research while completing his studies for an MS in information technology management at the Naval Postgraduate School from 2018 to 2020.

Abstract

This study examines five advanced decision support methodologies—Lean Six Sigma (LSS), Balanced Score Card (BSC), Integrated Risk Management (IRM), Knowledge Value Added (KVA), and Earned Value Management (EVM)—in terms of how each can support the information technology (IT) acquisition process. In addition, the study provides guidance on when each methodology should be applied during the acquisition life cycle of IT projects. This research includes an in-depth review of each methodology in the context of the acquisition life cycle. All acquisition projects within the Department of Defense must go through the acquisition life cycle. While each acquisition project is unique, all must pass a series of common hurdles to succeed. Understanding how and when the methodologies can be applied to an IT acquisition is

fundamental to its success. The study concludes with a set of recommendations for the use of each methodology in the acquisition life cycle of IT projects.

Problem Statement

A recurring problem at the U.S. Department of Defense (DoD) is that acquisitions of information technology (IT) have been fraught with schedule and cost overruns. High-profile programs such as the Joint Strike Fighter, Coast Guard Deepwater program, Army Comanche, and the Navy A-12 demonstrate the need for improvement within the acquisition process. The current suite of management tools does not seem to adequately provide sufficient early warning and high enough fidelity into the root causes of fiscal overruns in order to provide the program manager (PM) time to adequately respond to program issues. This is a problem because the capabilities promised to the warfighter are not provided in a timely manner and the over-budgeted resources used to provide the capabilities could be more efficiently allocated to other programs. In essence, the learning time that traditionally is found in production does not exist in DoD acquisitions. Thus, the “Job Shop” size for many of these major acquisition procurements becomes a lot size of “one” due to the fact that each version of the new production is a unique procurement with customized outputs and places a higher premium on each reoccurring item. A further problem is that the current methodologies do not include a defensible way to measure the value of the proposed acquisition of an IT system. Without a ratio level measure of value, using portfolio management to optimize IT investments is problematic. Several of the methodologies (i.e., Knowledge Value Added [KVA] and Integrated Risk Management [IRM]) do provide ratio scales for the value metrics they use.

The problem is that current analysis and management tools based on continuous production are not adequate to address the evolving requirements of complex IT systems. This is a problem because acquisitions of increasingly complex IT systems require a broader set of management and analysis tools to ensure successful acquisitions in a Job Shop production context. The purpose of this study was to suggest a set of criteria for selecting management and analysis tools that could help acquisition professionals successfully navigate the acquisition life cycle of Job Shop products. This research is important because acquisition managers need a wider variety of tools to help them optimally analyze and manage their increasingly complex acquisition of IT-based portfolios.

There are several analytical and decision-support methods that can be used to improve the acquisition life cycle of IT investments. This study provides an approach that will aid practitioners in selecting the best decision support method for a given phase of the acquisition life cycle for IT systems. The methodologies that were reviewed for this study included Lean Six Sigma (LSS), Balanced Score Card (BSC), Integrated Risk Management (IRM: Risk Simulation), Knowledge Value Added (KVA), and Earned Value Management (EVM).

Research Questions and Objectives

The research questions are as follows:

1. When should the methodologies be used in the acquisition life cycle to ensure successful IT acquisitions?
2. How should the methodologies be used in the acquisition life cycle to ensure successful IT acquisitions?
3. What are the risks and limitations of using each of the methodologies for IT acquisitions?

The objective of the research is to provide a set of pragmatic recommendations, based on comparison of the proposed methodologies, that focuses on when and how each method can be applied to improve the acquisition life cycle of IT investments.

The Five Methodologies

There are other management tools (aside from the five methodologies) that might be applied to the IT acquisition life cycle (e.g., activity-based costing and Total Quality Management [TQM], to name two). However, a review of the literature supported the focus on the five main analytical methodologies identified for this study. Expanding the potential scope of this research to include other methodologies was deemed to add minimal value given that these five approaches are in current use in acquisitions management and research. It was also assumed that starting with these five methodologies would provide a platform for inclusion of other approaches in future research.

Reviews of each of the methodologies follow, beginning with LSS followed in order by BSC, IRM, KVA, and ERM. The focus on this paper is providing a brief introduction of the methodologies. For a more thorough discussion, see the full report of this research that provides a detailed explanation as well as examples of and prior research on each methodology (see Housel et al., 2019).

Lean Six Sigma

Currently employed to help justify the future use of an IT system to incrementally improve process productivity within the DoD, Lean Six Sigma (LSS) is a combination of two complementary concepts, Lean and Six Sigma, designed to eliminate waste and variation to attain customer satisfaction in the areas of quality, delivery, and cost (Salah et al., 2010). Six Sigma evolved from the TQM program and is focused on reducing variability and removing defects within a process (Apte & Kang, 2006). The Lean concept centers on reducing waste and increasing the speed of a process (Apte & Kang, 2006). In the past, practitioners often chose one concept or the other, believing the two approaches to be contradictory in nature (Apte & Kang, 2006). However, many managers now view the concepts as synergistic (Apte & Kang, 2006). Together they lead to the goal of a continuous process flow via a cycle of iterative improvement.

LSS is an effective technique to improve the processes within a system. A detailed understanding of a procedure is required prior to implementing any changes to a process. This acumen could give decision-makers insight into the as-is system, that is, the current process or system the acquisition program is seeking to improve. Having a firm grasp on the as-is system may assist the PM when deciding the best course of action to fulfill stated requirements. LSS offers the most benefit when applied to processes that are already stablished. Incrementally improving procedures during the operations and support phase may provide significant cost savings and improved performance over the life of an acquisition.

Balanced Score Card

A strategic planning and management methodology developed by Kaplan and Norton (1996), BSC includes financial metrics as well as nonfinancial performance measures, such as leadership, customer satisfaction, and employee satisfaction, to achieve a balanced view of an organization's performance (Kaplan & Norton, 1996; see also Niven, 2008). The BSC helps to strategically align an organization's actions to its vision and strategy, improve internal and external communications, and monitor organization performance against strategic goals.

BSC could provide valuable perspective to the DoD when determining how to fill a specified need. Linking the various categories to acquisition categories could help determine the best solution for an Information System (IS) or IT need. Rather than looking at each acquisition as an individual system, a BSC approach could help decision-makers assess the needs of the organization rather than just state requirements for a single program. However, the DoD Decision Support System does incorporate some of these considerations already, specifically in the interaction between JCIDS and the Defense Acquisition System, which may diminish some advantages typically gained from using BSC.

Integrated Risk Management

IRM is a comprehensive methodology that is a forward-looking risk-based decision support system incorporating various methods such as Monte Carlo Risk Simulation, Parametric Forecast Models, Portfolio Optimization, Strategic Flexibility, and Economic Business Case Modeling. Economic business cases using standard financial cash flows and cost estimates, as well as non-economic variables such as expected military value, strategic value, and other domain-specific Subject Matter Experts (SME) metrics (e.g., Innovation Index, Conversion Capability, Ability to Meet Future Threats, Force Structure, Modernization and Technical Sophistication, Combat Readiness, Sustainability, Future Readiness to Meet Threats) can be incorporated. These metrics can be forecasted as well as risk simulated to account for their uncertainties and modeled to determine their returns to acquisition cost (e.g., return on investment for innovation, or return on sustainability). Capital investment and acquisition decisions within IT portfolios can then be tentatively made, subject to any budgetary, manpower, and schedule constraints.

The IRM methodology is a systematic technique to determine the best possible projects to pursue based on the statistical likelihood of their success. Using historical knowledge of defense acquisition programs and IT systems in both the government and commercial realms could improve the budgeting and scheduling processes. Determining the likely range of outcomes through dynamic statistical modeling may improve the program's performance. By better understanding the risk associated with various components, a more appropriate schedule and budget could be developed. IRM may also help determine which real options should be included in acquisition contracts. A high-risk program may need more options, such as the options to abandon, delay, or expand, based on its actual performance. Finally, IRM could prove useful in portfolio management, helping decision-makers determine which programs to initiate when viewing the portfolio of other programs in progress and used operationally.

Knowledge Value Added

As the U.S. military is not in the business of making money, referring to revenues throughout this paper may appear to be a misnomer. For nonprofit organizations, especially in the military, we require the KVA methodology to provide the required "benefits" or "revenue" proxy estimates to run a true ROI analysis. ROI is a basic productivity ratio with revenue in the numerator and cost to generate the revenue in the denominator (i.e., ROI is revenue-cost/cost). KVA generates ROI estimates by developing a market comparable price per common unit of output multiplied by the number of outputs to achieve a total revenue estimate. The presumption is that the output of a process, at a given point in time, is the thing of value because it was desired by the process owner regardless of how the process owner may decide to change the process at some future point in time.

In this way, KVA follows the general historical accounting model as a measure of cost (i.e., historical cost accounting model) per common unit of output. Standard accounting is based on historical measures of cost based on the cost to use resources (i.e., human, machine, raw, and infrastructural) to produce outputs. Generally accepted accounting practice (GAAP) does not provide any way to allocate revenue backward/historically within the enterprise. KVA goes a step further by adding a historical common unit measure of value (i.e., ratio level metric for common units of value via the KVA methodology). In a for-profit enterprise, this addition to

GAAP allows for the allocation of revenue throughout the enterprise based on the outputs that core processes or functional areas produce at a given point in time providing an estimate for ROI. And, using KVA, it has been shown that internal ROIs are a defensible metric to use as a surrogate for capital asset price in estimating volatility over time (Housel et al., 2007). Armed with this new information, it is possible to use standard financial investment metrics that require measures of volatility (i.e., risk in financial terminology) over time.

In application to measuring the general productivity of organizational resources, KVA is a methodology whose primary purpose is to describe all organizational process outputs in common units. This provides a means to compare the current and potential future outputs of all assets (human, machine, information technology) regardless of the aggregated outputs produced. For example, the purpose of a military process may be to gather signal intelligence or plan for a ship alternation. KVA would describe the outputs of both processes in common units, thus making the ROI performance of any of the processes comparable.

KVA differs from other nonprofit ROI models because it allows for revenue estimates, enabling the use of traditional accounting, financial performance, and profitability measures at the suborganizational level. KVA can rank processes by the degree to which they add value to the organization or its outputs. This assists decision-makers in identifying how much processes add value. Value is quantified in two key metrics: Return on Knowledge (ROK: revenue/cost) and ROI (revenue-investment cost/investment cost). As previously noted, the KVA method has been applied to numerous military core processes across the services. It was originally developed to estimate the ROI on IT acquisitions in the telecommunications industry at the subcorporate level and has been used for the past 17 years in the DoD, with emphasis on the Navy, to assess the potential value added by IT acquisitions to core DoD processes.

With the KVA methodology, the value concept has a different meaning than it does for EVM or LSS. Using the KVA methodology, the value concept is based on complexity theory. This methodology values organizational processes in terms of their ability to change inputs into outputs using a given process. Thus, these changes are the units of value (Housel & Kanevsky, 1995). Elementary changes can be represented by common units of computational complexity; see Kolmogorov complexity theory explanation in Housel and Kanevsky's (1995) original treatise. These common units of complexity can be described in terms of the knowledge required to execute these units in a process. And, the amount of knowledge (i.e., computational complexity) can be described in terms of the learning time for a common reference point learner (i.e., common units of learning time is proportionate to the amount of knowledge contained in a process by the process change-making resources: people and machines).

KVA is potentially an extremely valuable tool for inclusion in the Defense Acquisition System. Since the DoD is not a for-profit company, it does not have revenue to judge the effectiveness of its programs. Instead, it relies on various metrics and evaluations that are not comparable for system to system. If the DoD implements the KVA methodology, PMs may have an objective measure to compare various technological solutions to fulfill requirements. Understanding the value that a system or process provides in direct comparison with the value of other systems, whether they are similar or unrelated processes, could provide beneficial information in the decision-making, budgeting, and planning processes.

Earned Value Management

EVM is used by the DoD and industry for the planning and management of projects and programs. It provides cost and schedule metrics to track performance in accordance with an acquisition project plan during the developmental phase of the acquisition life cycle after the engineering development contract is awarded. It uses a work breakdown structure (WBS) to try to measure the performance of a program based on the amount of planned work that is done at any point in the program management baseline (PMB). EVM uses cost and schedule metrics that aid in performance trend analysis with a focus on identifying any budget and schedule deviations from the plan to allow the project team to take action as early as possible. It has been used for process improvements, but its strength is in providing a disciplined, structured, objective, and quantitative method to integrate performance, cost, and schedule objectives for tracking contract performance (DoD, 2015). It is important to note the term *value* in EVM does not have the same meaning as in other methodologies, such as Knowledge Value Added. Within the context of EVM, *value* is defined as the work accomplished towards completion of the project. There is no reference to the quality of the completed work or additional (or missing) benefits the work might provide to a system. The value is assumed because the specifications were defined in the project requirements.

EVM has proven to be a reliable system to manage cost and schedule performance for manufacturing in both defense and commercial industries. However, as systems become more complicated and IT and IS gain a more prominent place within even traditional manufacturing projects, EVM may need additional information from additional methodologies to improve its capabilities. Better incorporating the strategic guidance associated with a program, the value gained from subcomponents and subprocesses, the risk associated with developing subcomponents of a system, and incrementally improving a process may help improve the Defense Acquisition System as a whole.

Research Methodology

A review of each of the methodologies was conducted as well as a high-level review of the current phases of the acquisition life cycle (i.e., DoDI 5000 series). The methodologies were evaluated in terms of each major phase of the acquisition life cycle to suggest how they might be used to enhance the likelihood of successful completion of the phase. Analysis included a review of how the general overall acquisition life cycle approach might be modified to incorporate the benefits from the methodologies, including the original motivations for the IT acquisition per the problems/challenges identified prior to the beginning of the acquisition process. It was presumed that it was possible that the acquisition life cycle should include a formal review of the need for the IT in the first place. It also was presumed that it was possible that the acquisition life cycle should not end when the IT is acquired. What follows are a review of the generic IT acquisition life cycle and the mapping of this generic life cycle to the existing DoD acquisitions framework; a review of the benefits and challenges of using each of the five methodologies with final recommendations about how to use each within the generic acquisition life cycle; a statement of the limitations of this study; and remarks on future research.

Acquisition Life Cycle

This study developed a basic framework for placing the five methodologies within the generic IT acquisition life cycle as shown in Table 1, which can be mapped to the standard DoD acquisition framework. Doing so allows a comparison of where the two general frameworks match up and provides some preliminary guidance for how the five methodologies might be used in the standard 5000 series acquisition framework.

Table 1. Five Approaches: When to Apply in the Methodologies in Tech Investment Life Cycle

Pre-Investment	Strategic Goal Alignment	Implementation	Post Implementation
KVA (As-Is)	BSC (Align strategy with performance metrics)	EMV (Monitor cost and schedule, adjust as needed)	KVA (Monitor ROI, ROK)
LSS (Identify waste, value added)	IRM (Identify the strategic options for IT investments)	KVA (To-Be, ROI, ROK)	LSS (Assess and monitor cost, waste reduction)
Other	Other	IRM (Use the project management tools within the IRM suite)	

As shown in Table 2, the Defense acquisition life-cycle framework mirrors the generic technology investment acquisition life cycle in that there exists a planning phase that includes activities consistent with pre-investment and strategic alignment, an execution or implementation phase, and an operations and support phase, generally considered the post-implementation phase of a program. The DoD defines these phases as the Materiel Solution Analysis phase, Technology Maturation and Risk Reduction phase, Engineering and Manufacturing Development phase, Production and Deployment, and the Operations and Support phase. Figure 1 is a visual representation of these phases as they are defined in DoDI 5000.02.

Table 2. Aligning the Generic and 5000 Series Life Cycles

Pre-Materiel Solutions Analysis	Materiel Solutions Analysis	Technology Maturation and Risk Reduction	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
-Strategic goal alignment -Pre-investment	Pre-Investment	Pre-investment	Implementation	Implementation	Post-implementation

Materiel Solution Analysis Phase

The Materiel Solution Analysis (MSA) phase assesses potential solutions for a needed capability in an Initial Capabilities Document (ICD), which was developed during the defense requirements generation process known as the Joint Requirements Capability Determination System (JCIDS). The MSA phase is critical to program success and achieving materiel readiness because it is the first opportunity to influence systems supportability and affordability by balancing technology opportunities with operational and sustainment requirements. During this phase, various alternatives are analyzed to select the materiel solution and develop the Technology Development Strategy (TDS) that will be further assessed in the TMRR phase and eventually executed during Engineering and Manufacturing Development (EMD).

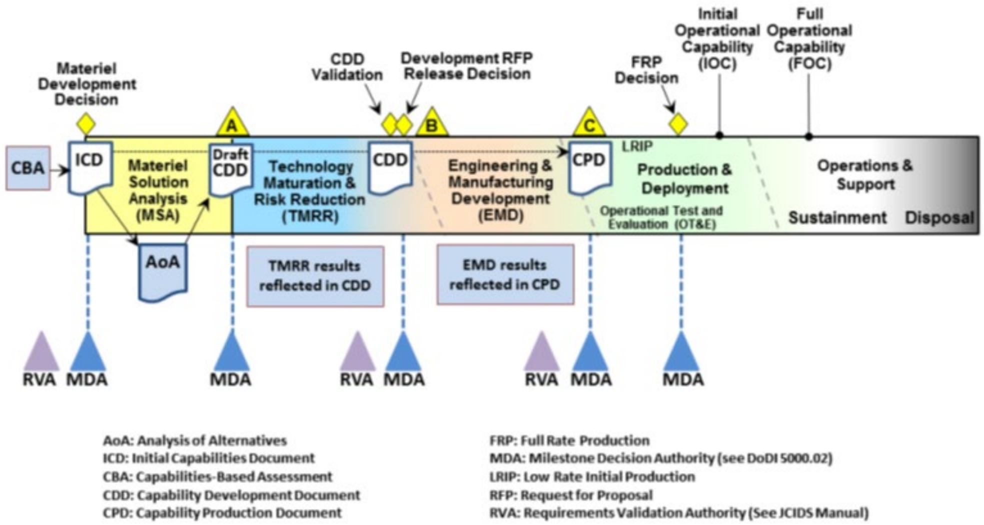


Figure 1. The 5000 Series Acquisition Life Cycle (DoD, 2017)

The MSA phase also includes identifying and evaluating affordable product support alternatives with their associated requirements to meet the operational requirements and associated risks. Consequently, in describing the desired performance to meet mission requirements, sustainment metrics are defined that will impact the overall system design strategy. One of the principal tasks that must be completed during this phase is the Analysis of Alternatives (AoA), suggesting that tools that offer robust trade-off analysis might be better suited for this phase.

Significant events within the MSA and other phases of the acquisition life cycle are listed in Table 3. While this is not an all-inclusive list of events during each phase, important steps within a program's development are incorporated.

Technology Maturation and Risk Reduction Phase

The Technology Maturation and Risk Reduction (TMRR) phase is designed to reduce technology risk, engineering integration risk, life-cycle cost risk and to determine the appropriate set of technologies to be integrated into a full system. The objective of the TMRR phase is to develop a sufficient understanding of a solution to make sound business decisions on initiating a formal acquisition program in the EMD phase. This phase lends itself well to management tools that provide all the program manager (PM) needs to conduct technical and business process trade-off analysis studies relative to cost and schedule.

Table 3. Key Events Within the Phases of the 5000 Series

MSA	TMRR	EMD	P&D	O&S
Analysis of Alternatives	Preliminary Design Review	Complete detailed design	Low rate initial production	Lifecycle Sustainment Plan (LCSP)
Initial funding estimates	Capability Development Document	System-level Critical Design Review (CDR)	Initial Operational Test & Evaluation (IOT&E)	System Modifications
Technology Development Strategy	Competitive prototyping	Establish project baseline with Performance Measurement Baseline (PMB)	Full rate production decision	Sustainment
	Acquisition Program Baseline (APB) established		Initial and Full Operational Capability (IOC and FOC)	Disposal

Engineering and Manufacturing Development Phase

The Engineering and Manufacturing Development (EMD) phase is where a system is developed and designed before going into production. The EMD phases is considered the formal start of any program and the point at which a development contract is awarded based on a specific statement of work (SOW). The goal of this phase is to complete the development of a system or increment of capability and evaluate the system for technical maturity before proceeding into the Production and Deployment (PD) phase. This is the phase in which cost and schedule variance models that help the PM to better understand technical issues are best employed since requirements are fundamentally solidified and represented in the SOW. If requirements are shown to be less than optimal or there are other mitigating issues during this phase that impact cost and schedule, then decision support tools to facilitate trade-offs may be used to help the PM maintain the program baseline and deliver user-defined capability.

Production and Deployment and Operations and Support Phases

These two phases (PD and Operations and Support [OS]) are necessary for the PM to ensure that the product being manufactured meets the operational effectiveness and suitability requirements for the user or customer. While the design is pretty well set at this point in the program, there may still be some trade-offs that take place prior to the full rate production decision and fielding of the system. The PM is less concerned with managing cost and schedule variance at this point since the contract types typically revert to a fixed price strategy. The biggest concern for the PM at this point is correcting any final deficiencies in the system and establishing a stable manufacturing and sustainment process.

The four generic phases listed in Table 1 align with the current DoD structure, as shown in Table 2. As the scope of this research is limited to the 5000 series, the pre-materiel solutions analysis column is for informational purposes only. The JCIDS process accomplishes strategic goal alignment, determining the necessary additions to the DoD's capabilities portfolio prior to the 5000 series. The ICD generated in the JCIDS process describe the high-level needs that the user requires, and these needs are assessed in the AoA process during the MSA phase. Within the scope of this paper, the DoD acquisition life cycle and generic IT acquisition life cycle begin with pre-investment during MSA.

Risk Management Framework

If one discounts basic scheduling and cost management practices, the primary tools to monitor progress of an acquisition program during the MSA and EMD phases are EVM and the Risk Management Framework (RMF). Figure 2 shows the seven steps that comprise the RMF, repeating in a cyclical pattern—prepare, categorize, select, implement, assess, authorize, and monitor.

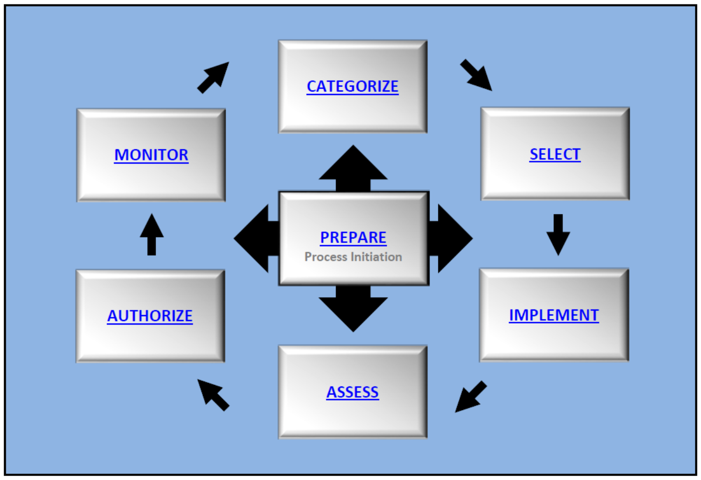


Figure 2. Seven Steps in the Risk Management Framework (Joint Task Force Transformation Initiative, 2018)

Preparation initiates the process, ensuring organizations are ready to execute RMF and giving context and priorities for managing risk (Joint Task Force Transformation Initiative, 2018). Categorization consists of organizing the system and the information used by the system based on an impact analysis (Joint Task Force Transformation Initiative, 2018). The risk manager then selects the appropriate security controls, tailoring them as necessary (Joint Task Force Transformation Initiative, 2018). The controls must then be implemented into the system and its operating environment before assessing the controls' effectiveness and authorizing the use of the information system (Joint Task Force Transformation Initiative, 2018). Finally, the manager must monitor the security controls on a continual basis, repeating the cycle as necessary when deficiencies are discovered (Joint Task Force Transformation Initiative, 2018). EMD is the first point at which PMs use EVM in an official capacity. The appropriate decision-makers approved a schedule and budget for the program creating the Acquisition Program Baseline. Future progress is now measured against this benchmark. Even using these proven tools, cost and schedule overruns occur regularly, illustrating the need for a different approach.

The RMF is a broad analysis that covers multiple types of risk and is used throughout the entire life cycle of a new development system. Implementing other tools into the process could help PMs better understand the risk involved at various points throughout the program. Within an acquisition there is an interdependence of risk. As the program progresses (and using the EVM methodology) and the ACWP increases, there are increasing levels of aggregation and abstraction of risk. For instance, to award an EMD contract, the technology involved must be at a Technology Readiness Level (TRL) of 6, indicating the technology performed adequately in a relevant test environment (Assistant Secretary of Defense for Research and Engineering [ASD(R&E)], 2011). However, the technology is not yet completed and requires significant improvement before production. The current risk assessment program does not account for the possibility that this categorization is incorrect and may not lead to a fully operational system. As a result, PMs proceed with the assumption the technology will continue development as planned. Any lack of progress will not become apparent until the ACWP begins to vary from the BCWP. It is often too late to make the appropriate corrections to the program in order to remain on budget by the time the discrepancy is discovered using EVM metrics.

Early risk management that focuses on the validity of the decision-making process using the RMF framework might introduce a higher level of understanding of the subordinate processes. For example, if at a particular milestone, the technology is not at the level of readiness it is being portrayed, then the consequences are x, y, and z. The results of each statement can be expressed in terms of time and money, or, keeping with the already established EVM terminology, potential CV. A PM can then assign a probability of success estimate to the state of the program that might drive a deeper understanding of the various interdependent program management processes.

Generic Framework and 5000 Series Integration

Table 4 shows when each methodology might be used in the 5000 series phases. This table reflects the reality that there are multiple tools for the various phases that should be used in concert and that certain tools are more appropriate for a particular phase than others. It is incumbent on the PM to use the tools appropriately in that they provide more information for a complex environment. The tools themselves do not provide the solutions to potential problems; they are simply indicators of underlying performance issues and, as such, are tools that can provide better insight into the life cycle of a program.

Table 4. Methodologies Within the 5000 Series

Materiel Solutions Analysis	Technology Maturation and Risk Reduction	Engineering and Manufacturing Development	Production and Development	Operations and Support
BSC	IRM	EVM	EVM	KVA
IRM	KVA	IRM	IRM	LSS
KVA	LSS	KVA	KVA	
LSS				

Understanding the extent to which a particular tool might provide greater insight into program performance across the life cycle, one should consider the level of analysis required and the viability of a particular tool to provide sufficient insight at that level of analysis. Three levels of analysis were considered for this initial survey: organizational, business process, and task analysis (see Table 5).

Table 5. Management Tool Selection Criteria Based on Level of Analysis, Focus of Analysis, and Acquisition Phase

Level of Analysis	Focus of Analysis	Time Horizon	Acquisition Phase
Organization	-Strategic competitive advantages: BSC, IRM -Value=Revenue: BSC, IRM strategic options	LSS: 3+ months (depends on level of process complexity)	MSA/TMRR/P&D/O&S
Business Process	-Cost savings: LSS, EVM, BSC, IRM -Schedule: EVM Value: KVA outputs	-EVM: 5+ months set up time (depends on requirements) -KVA: 2 days – 1 month (depends on level of analysis)	MSA/TMRR/P&D/O&S
Task Analysis	-Cost savings: LSS, IRM -Value=Cost+schedule cycle time: LSS, BSC	-BSC, IRM: 3-6 months	TMRR/EMD/P&D

It is clear from Table 4 that a variety of tools are required across the life cycle for the PM to gain a more robust view of the program performance. As shown in Table 5, the selection of the tool will depend on the particular focus and time horizon with which the tool is able to provide relevant information about the program. Table 6 illustrates different benefits and challenges of each methodology. Simply relying on one tool will not allow the PM to adequately manage the program. Planning for the type and depth of the management tool is started early in the life cycle and should be part of the overall acquisition strategy. Additionally, selecting contractors that are able to implement and manage these tools is critical in the decision-making process.

BSC is an excellent tool when viewing a system holistically. It provides a way for managers to examine a project from a systems-thinking approach. It may be most useful when strategizing about the potential use of an IT acquisition and how it might fit into the DoD's higher-level strategic goals prior to developing a requirements document. The statements derived from the BSC for general dissemination among all levels of the organizational structure must be translated into a simpler form presented in set of objectives and targets that are clear for all levels within the organization. It is also important to understand that leadership is central to ensuring any IT acquisition will support the organization's overall strategy enumerated in the BSC. This is true in the DoD as well as in any organization's implementation of a BSC (Llach et al., 2017). Without leadership support and guidance, the BSC is unlikely to succeed, and the organization will not be able to generate acceptable returns on its IT investments.

Table 6. Benefits and Challenges of the Five Methodologies

	Extensible, Quantitative Value Measurement	Time to Perform	Cost	Bottleneck Analysis
BSC	No, subjective measurement (revenue is exception)	3-6 months (depends on level of analysis)	Accounting-based financial metrics only	None
EVM	No, cost measurement only	5+ months setup time (depends on requirements)	Cost of resources and time	No, linear tracking only
LSS	No, nominal value only	3+ months (depends on level of process complexity)	Activity-based costing approach	Direct bottleneck analysis
KVA	Yes	2 days – 1 month (depends on level of analysis)	Common units of cost	Elapsed time versus work time
IRM	Yes, KVA	3-6 months (relatively quick once initial steps completed)	Cost accounting and KVA cost metrics	Monte Carlo simulation

The use of BSC can result in a cursory review of KPIs during the traditional acquisition life-cycle management process. BSC also avoids overreliance on financial KPIs by viewing the effects of each KPI on the other parts of the scorecard. While financial KPIs are reviewed with BSC, the other segments are separated from a purely financial analysis, allowing managers to use their judgement in determining how the proposed solution will affect the scorecard as a whole. The problem is that without a quantifiable common-units performance metric that allows the practitioner to determine the relative value between the different scorecards, it is difficult to determine which course of action would be optimal. There is no performance ratio that tells the manager that by performing a given action, the financial KPIs will improve by a given amount, the stakeholder engagement will decrease by this amount, and the internal process will change by this amount. Instead, it is more of a conceptual thought exercise to ensure managers consider the effects of their decisions on the entire range of KPIs. Because of this, BSC works best during the strategic goal alignment phase of the generic IT acquisition life cycle and the pre-MSA portion of the DoD acquisition life cycle. The MSA phase also includes aligning the stated requirements with the possible solutions to the capability gap during the AoA. An all-inclusive view of the effects of the various IT solutions that are being considered will assist in the selection of the most appropriate option to continue towards acquisition. BSC is recommended for implementation during the MSA phase.

EVM provides users with an easily understandable report of a project's advancement towards completion. Comparing the BCWP and the ACWP gives a clear view of how a system is progressing within the anticipated budget. The metrics used for cost and time are also clearly delineated. This delineation allows managers to compare the performance at different points throughout the project, which can assist in determining where a project has changed trajectories. There are numerous challenges when using EVM as well. While cost is measured and tracked regularly, the value of the project is not monitored as closely. Despite the name, the amount of work performed does not tell a manager the actual quantifiable value (in a common-units measurement) the project has accrued at a given point. There is no quantifiable measure of value within the methodology. The only quantitative measures of performance are measures of cost and time.

The ACWP assumes the outputs from all work were perfect on completion. If there are issues with the results from earlier efforts, they must be reworked, changing the ACWP calculation. If the technology does not improve as expected because the TRL was not accurately portrayed, a PM will believe the project is on schedule despite the “earned value” lagging behind what the numbers are projecting. Additionally, and in some instances because of this assumption, EVM outputs are not timely. Conducting an accurate analysis of a program is time consuming and does not provide useful predictive information. By the time EVM alerts a PM to a variance, the variance has already occurred. All corrections are reactive to bring the ACWP back to the baseline, which has proven to be a nearly impossible task in practice. EVM will only be effective when the baseline plan is well researched and accurate. Otherwise, the ACWP is compared against flawed data. EVM does provide valuable information to project managers during the EMD phase but should be supplemented with some of the other methodologies (LSS, KVA, IRM) throughout the project management cycle.

Successfully implementing LSS into a process will lower the cost of the project by reducing the variation in a product run and the waste associated with its production. When additional steps or unnecessary waste is reduced, additional resources become available for use in other processes. In identifying a bottleneck, LSS can address multiple problems simultaneously depending on how the project is defined. By creating improvement in one area and freeing resources, other areas may benefit from an improved process workflow. However, LSS can be costly to implement. The analysis requires a great deal of time and information to develop meaningful understanding of any problems. LSS’s definition of value is at the nominal scale level: an item either adds value to a project or it does not. Reality is not often as black or white. There are required steps that must be conducted that do not necessarily add value to a product from the user’s standpoint. For instance, accounting departments do not attempt to directly add value to a final product, but any organization recognizes the need for accounting, suggesting the accounting department does add value. LSS is time-consuming when applied on a large scale, as would be the case in a DoD acquisition. Defining the problem and determining appropriate measurements in a step-by-step manner is a major undertaking. However, acquisition professionals can use it to ensure the project is defined and measured appropriately.

The greatest benefit from KVA is a quantifiable (common units) value metric that can be compared across various aspects of a project (Housel & Bell, 2001). If the value of an intermediate step is quantified, managers can compare the outputs of a component instead of simply the effort measured by time and cost that were inputs. KVA provides a value measurement for both tangible and intangible assets, making it especially well-suited for use with IT. A KVA analysis can be accomplished in a relatively short period of time in comparison with the other methodologies. A quick, rough-cut KVA analysis can provide rapid guidance for the project before sinking valuable time and resources into a more comprehensive examination. KVA is primarily a measurement tool that provides performance information to decision-makers. It is not a system that will drive an acquisition project towards the goal on its own. As in the other methodologies mentioned thus far, KVA has limited value in making predictions for future value, focusing instead on the current value of systems in development. There must be another methodology employed with KVA to ensure a project’s success.

IRM provides a foundation to incorporate the risk associated with a decision into a quantitative decision process. IRM’s core premise maintains there is a probability for success and failure with every decision option during a project’s life cycle. Using statistical simulations, real options, and optimization will improve the quality of information a PM has to determine the course of a project. Real options analysis can be used to frame strategies to mitigate risk, to value and find the optimal strategic pathway to pursue, and to generate options to enhance the value of the project while managing risks. IRM’s drawback is that the analytical methods can sometimes be difficult to master. But with the requisite knowledge and training, coupled with the correct tools, the IRM methodology can provide a plethora of value-added information for making strategic and tactical decisions under uncertainty.

Comparison of Key Attributes

Choosing a methodology should depend on the nature of the project under consideration, specifically, the commitment needed from the organization, the organization's desire to align strategic goals with the project, the predictive capability of the methodology, the flexibility required, and the time available. Table 7 compares these categories across the five methodologies. While others in the organization need to understand the concepts to comprehend status reports, EVM only needs the management team to track the cost and schedule of the project compared to the baseline as there is no goal alignment with the organization. While the CPI and SPI can help estimate the final cost and schedule, there is no true predictive ability associated with EVM since the assumption is that the schedule will proceed according to the baseline, regardless of previous performance. Adherence to the baseline is essential in EVM, and changing requirements can drastically alter a baseline, reducing the effectiveness of the methodology. Setting up, monitoring, and reporting the performance of each work package within the WBS can be a time-consuming and expensive task.

Based on the strategic goal alignment and the department-specific metrics, the entire organization is committed to any BSC efforts. The underlying assumption within BSC is that measuring something will improve its performance. As such, leaders are predicting improvement in the areas being measured, although BSC does not give a numerical estimate of the improvement. BSC is flexible in that the same key areas can lead to different metrics depending on the specific department's tasks. These tasks and metrics can also change as the organization shifts its vision or strategy. However, doing so can take a significant amount of time as every level must adjust its metrics and can do so only after the immediate superior has updated the metrics for that level.

Table 7. Comparison of Key Attributes

	EVM	BSC	KVA	IRM	LSS
Organizational Commitment Required	Management team	Entire organization	Analyst and process owner	Analyst, project and portfolio manager, and leadership	Leadership, champion, project manager, process managers, LSS team members
Organizational Goal Alignment	None: Tracks completed work vs baseline	Every level to organizational goals	None: Objective measurement of output	Portfolio management	Requires commitment to techniques but not an overall shift in organizational strategy
Predictive Capability	Limited: CPI and SPI can be used to estimate final cost and timeline	Limited: Assumes high marks in chosen metrics indicates positive future performance	As-Is to To-Be predictive improvements	High: Probabilities based on historical data	Limited: Incremental improvement predictions
Flexibility	Not flexible after baseline established. Requirements ideally remain constant	Can develop different metrics for each department	Can be adapt language of description used for common units of output	Real Options provide flexibility after learning and implementation	Creates iterative changes to processes
Time Requirement	Time consuming	Time consuming	Rough cut analysis done quickly	Relatively quickly, depending on data collection for first steps	Time consuming

KVA needs only the analyst and the process owner as the subject matter expert to determine the value of a process's output, eliminating the need to align the project with an organization's goals. Using this analysis, they can establish the current as-is process and compare it with the to-be process in development, predicting the improvement between systems. Since KVA can be used with any language of description to define the process, analysts can choose whichever method is most beneficial for the particular system in question, providing flexibility. This analysis can be completed quickly, potentially providing a rough-cut assessment within a few days.

IRM requires the organizational leadership, portfolio and project managers, and the analyst to determine how a project fits within an organization's portfolio, the Present Value of the project, and potential real options. By analyzing and simulating various scenarios, IRM provides a prediction of a project's likely performance, which allows managers to build in flexibility via real options at the appropriate locations. Assuming the data necessary for the analysis is available, the process can be completed in a relatively quick manner.

Leadership, project and process managers, a project champion, and LSS team members must all be involved for an LSS initiative to have success. Leadership is needed to provide funding for black and green belt training to ensure improvements made to processes remain in place and additional areas with potential enhancements are identified. While the overarching goals of the company will not change because of LSS, some business practices will be adjusted to make iterative improvements. There is limited predictive capability within the methodology other than that the areas from which waste and variation are removed will produce a more efficient product. LSS makes numerous incremental changes that can be time consuming before a process is optimized.

Methodologies in IS Acquisition

As previously discussed, the five methodologies all have strengths and weaknesses, making them more suitable in certain applications than others. Table 8 depicts some of these considerations when conducting an acquisition of a software-intensive system, hardware-intensive system, upgrade to a legacy system, or a complete organic build. The biggest challenge in using EVM when acquiring IS is the iterative nature of software development. EVM needs clearly stated, detailed requirements for intermediate steps to be most effective. While the outputs of software programs are defined well, the steps required to build the software are not, leading to issues when developing cost and schedule estimates. If the software is not complex or consists of known processes, EVM can sufficiently monitor the progress. Integrating software and hardware is also complicated with EVM since there are numerous pieces of the program that must be combined to meet the goals, resulting in additional debugging and recoding. EVM is more efficient when used to manage the physical creation of systems or infrastructure. It can monitor the progress of software work packages but is not as useful at estimating the earned value of those programs until the requirements have been delivered.

Table 8. Methodology Performance in Different IS Acquisition Cases

	EVM	BSC	KVA	IRM	LSS
Software Intensive Systems	Not well adapted to iterative system development lifecycle	Aligns organizational goals with system development given appropriate metrics	Provides value and cost estimate enabling productivity and ROI on IT estimates	Includes KVA capabilities. Allows iteration of the value of real system options	Can be used in software fixes or improvements after system is operational
Hardware Intensive Systems	Useful provided the IT component is relatively non-complex	Aligns organizational goals with system development given appropriate metrics	Provides value and cost estimate enabling productivity and ROI on IT estimates	Includes KVA capabilities. Allows iteration of the value of real system options	Can improve hardware manufacturing and sustainment processes
Legacy System Upgrades	Useful for manufacturing based updates of programs	Difficult to adapt changes in vision/strategy to existing hardware and software	Determine value of components. Helps manager decide how to use resources to improve system	Includes KVA capabilities. Allows iteration of the value of real system options	Can improve sustainment process and determine system bottlenecks for future upgrades
Organic Builds	Useful for manufacturing based acquisitions not involving complex software development	Helps ensure new system alignment with strategic goals	Can help manager estimate future value of the system	Quantifies risk and assigns probabilities of success, allowing for real options analysis	Useful after system is operational

BSC can assist mangers in aligning the goals of the organization with those of their individual program, whether they are dominated by hardware or software. This is especially true during an organic build, ensuring the entire IS under development is created with the strategy and vision of the acquisition community in mind. However, it can be difficult to change the vision when implementing updates to existing hardware and software systems already in use if the original strategy differs greatly from the strategy already in place. For example, if the Littoral Combat Ship (LCS) needs updates in the future through acquisition programs and the future vision of the DoN focuses on redundancy for combat operations versus the current vision of IS replacing manpower, it will be difficult, if not impossible, to redesign the ship with the necessary modifications.

KVA can provide an objective, ratio scale measure of value and cost for each subprocess within any of the IS systems. Using the two measurements, managers can then analyze productivity ratios, such as ROI, to determine the effectiveness of a process compared to the resources used to achieve the output. This can help the manager decide how to use resources to update systems or estimate the future value of a system being acquired. Combining the KVA results with IRM allows managers to iterate the value of real options analysis through simulation and other techniques. IRM can also quantify risks and assign probabilities of success for programs and components of programs using historical data. It is a tool to assist with the investment strategy, making it useful when acquiring all types of information systems. However, it is not designed to help manage the actual acquisition of a program or determine how to meet its detailed requirements.

LSS is best used after a process has reached its steady-state operational capability. Then it can be used to analyze any of the systems to reduce waste and variation within the processes. The corrections made to the sustainment process are done incrementally, gradually improving the efficiency of the program over time. While elements of LSS, such as mistake proofing, may be beneficial during the acquisition process, LSS as a whole works better after the program is operational and can improve the system holistically.

Research Discussion, Recommendations, and Conclusion

The central question of this research was, “How should the methodologies be used in the acquisition life cycle to help ensure successful acquisition of IS technologies?” It should be noted that EVM is required for all programs with a contract value greater than $20 million. Regardless of this requirement, EVM offers a structured approach to the acquisition of IT via program management processes that track schedule and cost. While there are some significant limitations when using EVM for IS acquisitions, this was the only program management methodology required by the government and military program managers and can be useful in ensuring that an acquisition stay on schedule and within cost estimates.

The major weakness of EVM for IT acquisition is that it was not designed for managing IT acquisitions that follow a very iterative pathway. Organic IT acquisitions require a given level of flexibility to deal with the unknowns that arise during the development process. In addition, EVM does not provide a common unit of value metric to enable standard productivity metrics, such as ROI. When value is inferred by how consistent a program is with original baseline cost and schedule estimates, the performance of the program may sacrifice on the quality of the outputs when planned program activities become iterative, as in the development of many IT programs. For example, if an IT program is trending toward cost and schedule overruns, but the resulting value added of the modifications to the original requirements provides disproportionate increases in value, EVM is not designed to recognize this increase in value.

To remedy these shortcomings of EVM in IT acquisitions, the methodology should be combined with BSC, KVA, and IRM. BSC and KVA can be useful during the requirements phase of EVM by ensuring that a given IT acquisition is aligned with organizational strategy and that a baseline process model has been developed for establishing current performance before acquisition of the supporting IT. A future process model that estimates the value added of the incorporation of the IT can also set expectations that can be measured against the baseline model after the IT has been acquired. IRM can be used to value the real options that an acquired IT may provide so that leadership can select the option that best fits their desired goals for the IT inclusion. This kind of information can help guide the requirements analysis based on expected value added by the IT over time.

BSC is not recommended for use within the defense acquisition system as a means to ensure an IT acquisition aligns with the overall defense strategy for any given area or military service. The primary purpose of BSC is to ensure all levels of the organization are aligned to the organizational strategy and vision. The requirements process already produces outputs aligned with the strategic goals. Program managers must oversee their programs in accordance with the given requirements, which should force them to automatically align with the vision of the DoD. The “what you measure is what you get” theory is accounted for in the defense acquisition system. The specifications, cost, and schedule are the desired measurements that must be followed. While BSC might provide some benefit in aligning goals throughout the DoD or the entire acquisition process (i.e., using BSC to align requirements, budgeting, and acquisition together), using BSC exclusively within the defense acquisition system is not recommended.

KVA should be used in the acquisition of IT. Having an objective, quantifiable measure of value in common units will allow decision-makers to better understand and compare different options based on their value and the cost. Obtaining a return on investment of IT systems can only be done when using KVA to determine the value embedded in the system. This information provides insight to PMs and gives them a more complete perspective regarding the performance of both the current and the to-be systems.

Likewise, using IRM is recommended when acquiring IS through the defense acquisition system. Applying static and dynamic modeling techniques to predict likely outcomes can improve the risk estimates associated with the components and sub-components of a program. Analyzing various real options within the context of the models' outputs will help PMs make the most advantageous choices when determining a program's future.

LSS should also be used when acquiring IT. The incremental advancements LSS principles can discover may result in significant improvements in efficiencies and cost saving measures over the life of a program. Using the DMAIC process to eliminate waste and reduce variation will enhance program performance. The techniques can be applied to all types of processes, including both hardware and software-based systems. Improvements may be made to aspects of programs ranging from the software repair process to the depot level repair of the hardware in an IS. The military already has extensive experience with LSS, including education teams and a belt training system. This familiarity will make the introduction of the formal LSS methodology into the defense acquisition system easier than other options.

- How should the methodologies be used in the acquisition life cycle to ensure successful acquisition of IS technologies?

Program managers should use EVM only in the EMD phase, as is currently done. EVM will work best in hardware manufacturing solutions with technology that is fully mature prior to the program beginning. Since many IS acquisition programs consist of advancing the current technology and developing new software solutions to meet requirements, EVM is not perfectly suited for IS development. Nevertheless, PMs can use various agile EVM techniques to complete projects on baseline provided the appropriate steps are taken when establishing the baseline. Requirements must be broken into small, easily definable tasks with suitable risk and uncertainty factors accounted for within the schedule. Other methodologies should be used with EVM to ensure these factors are based on defendable metrics rather than simply guessing how much additional time and money may be necessary to complete complex tasks.

During the MSA phase, KVA will help determine the value of the different options considered in the AoA. KVA can objectively measure the value of the current, as-is system and the potential to-be systems under consideration. Using other factors such as cost, complexity, timeline, etc., the PM can then select an appropriate alternative. As the chosen solutions mature during the TMRR phase, an updated KVA analysis will reassess initial estimates and provide a projected return on investment for the IT solution prior to entering the EMD phase. In the OS phase, KVA will help decision-makers establish how a program is performing and use that information to make any adjustments or corrections that may be needed. KVA has limited prediction capabilities, so it should be used in conjunction with other methodologies, particularly IRM, to obtain the most benefit.

IRM techniques should be implemented during most of the acquisition phases. Ideally, portfolio management decisions were made during the requirements development process, although they should also be considered during MSA. Financial and value analysis derived from KVA, as well as simulation of possible outcomes, should occur during the MSA, TMRR, and EMD phases. The results of these simulations should be fed into the EVM baselines to account for risk across the program. Real options should be developed during the TMRR phase prior to awarding contracts, and the real options should be executed during the EMD and PD phases as appropriate.

LSS will best serve IS acquisitions after the product is implemented in the operational forces during the OS phase, which overlaps with PD. While individual manufacturers may use LSS in their manufacturing processes, PMs will not see the full benefits of this methodology until the program is in its steady state operation and the incremental improvements can have the greatest effect on process improvement and cost savings. LSS will help PMs evaluate the system through in-depth analysis of updates, upgrades, repairs, and other services that occur during OS. Elements of LSS may be useful in other phases of the Defense Acquisition System as most processes can be improved in some manner. However, formal LSS procedures should not be established until the system is in use, regardless of whether it is a hardware- or software-based system.

In conclusion, PMs should use the approach that fits with the selection criteria and point in the acquisition life cycle (constrained by time and cost). Thus, they should use continuous production economics for mature, simple products. Furthermore, in the case of the job shop and lot size of one, economics for complex products that can be "intelligensized."[1] In the era of Great Power Competition and the race with rising powers, there is a need for a new common unit of value to track upside of value of intelligensizing military products to stay ahead, because no current methodology in use provides adequate program value to risk-based forecasting. Therefore, this study seeks to address that gap in methodology.

Limitations and Future Research

This research examined only the 5000 series acquisition life cycle.[2] It is probable both the JCIDS and PPBE processes could benefit from the calculated implementation of some, or all, of the methodologies discussed. Improving one component of the Defense Acquisition Decision Support System will likely improve the outputs of the other two systems. Additional research into creating a quantifiable measure of risk will provide beneficial information that allows decision-makers to understand the probability of success for subcomponents within a project.

Future research in how the five methodologies might be useful for other areas of investment in IT and DoD acquisitions of IT might be beneficial in extending the current research study. The proposed five methodologies may be useful for researchers who are also interested in focusing on the following topics of acquisition research interest:

- Innovative Contracting Strategies—contracting at the speed of relevance (BSC, IRM)
- Breaking down silos, enterprise management (LSS, KVA)
- Rapid Acquisition and Decision Support (IRM, KVA)
- Effects of Risk-Tolerant and Risk-Averse Behavior on Cost, Schedule, and Performance (IRM, EVM)
- The Role of Innovation in Improving Defense Acquisition Outcomes (BSC, IRM, EVM)
- Applying Model-Based Systems Engineering to Defense Acquisition (IRM, KVA)
- Augmenting the Acquisition Decision Processes with Data Analytics (IRM)

---

[1] This is an operational term that the authors use to describe the processes laid out in this work.

[2] Given that the case studies of IT acquisitions exist in various existing data sources and written case studies, there is very little risk associated compared to the normal generation of new data sets that were required in the prior studies performed by the authors for the ARP. Access to acquisition subject matter experts (SME) at NPS reduced the risk associated in seeking other SMEs to discuss IT acquisitions and the use of the methodologies within the IT acquisition life cycle.

References

Apte, U., & Kang, K. (2006). *Lean six sigma for reduced life cycle costs and improved readiness*. Naval Postgraduate School.

Assistant Secretary of Defense for Research and Engineering. (2011, April). *Technology readiness assessment (TRA) guidance*.

DoD. (2015, January 7). *Operation of the defense acquisition system (change 4)* (DoD Instruction 5000.02). https://www.esd.whs.mil/Directives/issuances/dodi/

DoD. (2017, February 26). *Defense acquisition guidebook*. https://www.dau.mil/tools/dag

Housel, T., & Bell, A. (2001). *Measuring and managing knowledge*. McGraw Hill.

Housel, T., & Kanevsky, V. A. (1995). Reengineering business processes: A complexity theory approach to value added. *Infor*, *33*(4), 248. http://libproxy.nps.edu/login?url=https://search-proquest-com.libproxy.nps.edu/docview/228436959?accountid=12702

Housel, T., Little, W., & Rodgers, W. (2007). Estimating the value of non-profit organizations using the market comparables approach. In *Collected Papers of the European Institute for Advanced Studies in Management (EIASM), 3rd Workshop on Visualising, Measuring and Managing Intangibles & Intellectual Capital (Refereed Proceedings)*. Ferrara, Italy: European Institute for Advanced Studies in Management.

Housel, T., Mun, J., Jones, R., & Carlton, B. (2019). *A comparative analysis of advanced methodologies to improve the acquisition of information technology in the Department of Defense for optimal risk mitigation and decision support systems to avoid cost and schedule overruns* (NPS-AM-20-002). Naval Postgraduate School, Acquisition Research Project. https://dair.nps.edu/handle/123456789/2774

Joint Task Force Transformation Initiative. (2018). *Risk management framework for information systems and organizations: A system life cycle approach for security and privacy* (No. NIST SP 800-37r2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-37r2

Kaplan, R. S., & Norton, D. P. (1996, January–February). Using the balanced scorecard as a strategic management system. *Harvard Business Review*, *74*(1), 75–85.

Llach, J., Bagur, L., Perramon, J., & Marimon, F. (2017). Creating value through the balanced scorecard: How does it work. *Management Decision*, *55*(10), 2181–2199.

Niven, P. R. (2008). *Balanced scorecard step-by-step for government and nonprofit agencies.* John Wiley & Sons.

Salah, S., Rahim, A., & Carretero, J. A. (2010). The integration of Six Sigma and Lean management. *International Journal of Lean Six Sigma*, *1*(3), 249–274. https://doi.org/10.1108/20401461011075035

PANEL AI. INNOVATIONS IN DEVELOPING ARTIFICIAL INTELLIGENCE IN DEFENSE PROGRAMS

Wednesday, March 3, 2021	
11:00 a.m. – 12:15 p.m.	**Chair: Rear Admiral Kurt Rothenhaus, USN,** Program Executive Officer, Command, Control, Communications, Computers and Intelligence (PEO C4I)/Program Executive Officer, Space Systems (PEO Space Systems) ***Increasing Confidence in Machine Learned (ML) Functional Behavior during Artificial Intelligence (AI) Development using Training Data Set Measurements*** Bruce Nagy, Naval Air Warfare Center, Weapons Division at China Lake ***Artificial Intelligence Systems: Unique Challenges for Defense Applications*** Bonnie Johnson, Naval Postgraduate School ***Acquiring Artificial Intelligence Systems: Development Challenges, Implementation Risks, and Cost/Benefits Opportunities*** Thomas J. Housel, Naval Postgraduate School Johnathan Mun, Naval Postgraduate School Raymond D. Jones, Naval Postgraduate School Timothy R. Shives, Naval Postgraduate School Benjamin Carlton, Marine Corps Systems Command Vladislav Skots, U.S. Army ***Leverage Artificial Intelligence to Learn, Optimize, and Wargame (LAILOW) for Navy Ships*** Ying Zhao, Naval Postgraduate School Gabe Mata, Marine Operations Analysis Program Office, USMC Erik Hemberg, Massachusetts Institute of Technology Una-May O'Reilly, Una-May O'Reilly Nate Derbinsky, Northeastern University Bruce Cormany, NAVWARSYSCOM Andrew Haley, Naval Supply Systems Command (NAVSUP) Adam Hilliard, Office of the Chief of Naval Operations (N412)

Rear Admiral Kurt Rothenhaus, USN—is a native of New York City, New York. He received his commission in 1992 upon graduating from the University of South Carolina where he earned a Bachelor of Science degree. He also earned a Master of Science in Computer Science and a Ph.D. in Software Engineering from the Naval Postgraduate School and transferred into the Engineering Duty Officer community in 2003.

His operational assignments include serving as the combat systems/C5I officer on USS Harry S. Truman (CVN 75) and chief engineer on USS O'Brien (DD 975). Additionally, he served on the staff of Destroyer Squadron 15 and on USS Fife (DD 991). He completed an Individual Augmentee tour in Baghdad, Iraq.

His shore tours include: program manager for PMW 160, the Navy's Tactical Networks Program Office, at PEO C4I and commanding officer of Space and Naval Warfare Systems Center Pacific. He also served as the deputy program manager for the Navy Communications and GPS Program Office (PMW/A 170), the assistant program manager for the Consolidated Afloat Network Enterprise Services (CANES) in PMW 160, and the Maritime Tactical Command and Control (MTC2) assistant program manager in the Navy Command and Control Program Office (PMW 150).

In May 2020, he assumed duty as program executive office Command, Control, Communications, Computers, and Intelligence and program executive office Space Systems, Naval Information Warfare Systems Command.

His personal awards include the Legion of Merit, Meritorious Service Medal, Joint and Navy and various unit and service awards.

Increasing Confidence in Machine Learned (ML) Functional Behavior during Artificial Intelligence (AI) Development using Training Data Set Measurements

Bruce Nagy—is a Research Engineer at the Naval Air Warfare Center, Weapons Division at China Lake. His research focuses on advanced game theory techniques, artificial intelligence and machine learning applications for tactical decision aids. Mr. Nagy has earned four degrees: one in mathematics, two in electrical engineering, and one in biology from The Citadel and Naval Postgraduate School. He led the development of advanced algorithms and metrics that resolved national defense issues in satellite communications for DoD. At UCLA during postgraduate work, he investigated modeling brain stem communication with muscle groups at the cellular level, in cooperation with NIH.

Abstract

Both the commercial world and Department of Defense (DoD) are challenged with system safety issues when dealing with Machine Learned (ML)/Artificial Intelligence (AI) deployed products. DoD has a more severe issue when deploying weapons that could unintentionally harm groups of people and property. Commercial manufacturers are motivated by profit, while DoD is motivated by defense readiness. Both are in a race and can suffer the consequences from focusing too much on the finish line. Establishing formal oversight ensures safe algorithm performance. This paper presents a measurement approach that scrutinizes the quality and quantity of training data used when developing ML/AI algorithms. Measuring quality and quantity of training data increases confidence in how the algorithm will perform in a "realistic" operational environment. Combining modality with measurements determines: (1) how to curate data to support a realistic deployed environment; (2) what attributes take priority during training to ensure robust composition of the data; and (3) how attribute prioritization is reflected in size of the training set. The measurements provide a greater understanding of the operational environment, taking into account issues that result when missing and/or sparse data occur, as well as how data sources supply input to the algorithm during deployment.

Introduction

As opposed to traditional software development techniques, Machine Learned (ML)/Artificial Intelligence (AI) created functions have models that are configured using training data sets. Traditional code is used to manage the training process. Training sets are comprised of a combination of attributes, sometimes called features. When we refer to a feature within an image, we are describing a piece of information contained in the content of the image. In this case, the feature describes a certain region of the image, which has certain properties as opposed to another popular definition of a feature as a single pixel in an image. The aggregation of attributes can be contained in one source, e.g., a camera taking a facial picture, or from many sources, e.g., various sensor inputs, such as radar and communication links. In this paper, we will distinguish whether attributes are generated from one or multiple sources based on their modality. As will be described, understanding the type of modality and creating training data sets with the proper quality and quantity of instances/samples to replicate the variation, anomalies and noise experienced during deployment is key to algorithm behavioral confidence.

No Warning Labels

The first woman killed by an autonomous driven car (Schmelzer 2019) provided a reality check for the reliability of AI performance in a deployed environment. Interestingly, 1896 is when the first person was killed by a human driver. Who' was at fault? It was determined to be the driver. When an autonomous system makes a mistake, is it the car or the driver (Gurney 2013) that is at fault? There are many factory recalls of faulty mechanisms in cars, like brakes. Is it any different with AI software systems? The objective for many car developers is how safe can they make a car using autonomy as compared to other manufacturers (Griffith E 2016). That is their key to their advertising to create acceptance and sales from the consumer.

Elon Musk states a major concern in that AI systems might be developed in secret (Etherington 2012), thereby limiting oversight. For example, Microsoft has exclusive rights to OpenAI's text generation software (Hamilton 2020). This goes against the initial policy by Elon Musk as one of the founders of OpenAI with the goal of developing open source technology. Over the last decade and beyond, the primary motivator for car companies has been money. Over a 20-month period, a company producing technology for driverless cars was involved with 18 accidents (Wiggers 2020**)**. This company declined to support a conglomerate of major automotive developers focused on "safety first" guiding principles (Wiggers 2019**)** in autonomous vehicles. Instead, the company publicly stated that they support laws and regulations. From a legal standpoint, it is quite uncertain that existing laws will apply (Moses LB 2007). Because of that, car manufacturers may not have the proper incentives to develop safe systems (Cooter 2000). Even with this company's public being against a proactive safety focus, even with 18-accidents in 20 months, they were still able to raise over 3 billion dollars. Some legal thoughts support limited regulation, but with the caveat of incentivizing commercial manufacturers to only develop beneficial/useful AI (McGinnis JO 2010). For better or worse, discussions about economics, law, and philosophy (Russell 2015) are attempting to shape the answer to what is beneficial/useful. How to intrinsically motivate (Baum 2017) developers to create beneficial AI? The challenge is that people justify actions based on needs (Kunda 1990). Commercial manufactures must support their bottom-line, whereas DoD has a different set of goals.

The DoD Unique AI Challenge – It's Secret!

DoD has a different set of standards with regard to what is beneficial as compared to commercial needs. Yet, there are many things we can learn from industry. Certainly, DoD cannot afford an international incident regarding an autonomous system, especially a series of incidents that occur over a 20 month period. The significant challenge is that DoD must develop solutions in a sheltered and isolated environment, even from other classified projects. Even open source based oversight is limited. That is the reason system safety organizations are so vital in DoD and must have standards, measurements, policies and procedures to support their effort. Whether in the commercial world or DoD, AI functionality is considered to be unpredictable, unexplainable and goal uncertain (Yampolskiy, 2020). When we talk about AI safety issues for naval weapon systems, this has not typically included adversarial attacks that might affect functional performance. Given this perspective, AI adversarial network attacks using techniques like DeepFakes, putting an image/video into another image/video for mis-categorization (Chauhan 2018), was not included this research, but may be considered for future investigation. Unpredictable, unexplainable and goal uncertain is still an significant issue with AI deployed technology, even when developers are motivated and doing their best (Deci 1971, Krantz 2008). Even the best is still resulting in 18 incidents in 20 months.

A major challenge that both DoD and commercial manufactures face is a race to the finish line (Armstrong 2016) approach to development. Is there something to be learned from the nuclear arms race? The obvious lesson is that we need oversight in the early stages of development (Borrie 2014). AI may have the same dramatic effect, as did the nuclear arms race. Consider the issues of putting military drones and weaponry under the full control of AI systems (Bohannon 2015). Now consider Murphy's Law, "Anything that can go wrong, will." When it comes to what we expect computers to do and what they will actually do, especially when development gets more complex, unwanted incidents are more likely to occur (Joy

2000). Note that the majority of research that was been initiated over a decade ago involving robotic “decisions” and actions was being funded by the military (Lin 2011).

What DoD must ask is, “Can we deploy AI in safety critical functions, i.e., AI-enabled weapons acting autonomously?” The challenge to answering this question is in determining if an AI system can ever be “fixed”, become more reliable, to support safety needs, like brakes in a car.

For both commercial and government AI development, the need for safety standards is becoming more prominent (Ozlati 2017). The federal government has taken the initiative. National Institute of Standards and Technology (NIST) is focused on creating standards that provide oversight for AI development. In their 52 page report (NIST 2019), one of the nine areas of focus is on metrics. This paper is offered for consideration to be included in the NIST standards for AI development with regard to measuring the quantity/size and quality/composition of training data.

Navy System Safety for Weapons Deployment

In order to overcome the unique challenges of ensuring there is adequate safety and security in naval ordnance, the Naval Ordnance Safety and Security Activity (NOSSA) formed. NOSSA, the funding organization for this research, recognized the AI system safety might require a special set of policies, guidelines and metrics. Their concern was that ML/AI algorithms could not be analyzed using traditional hazard analyses approaches (MIL-STD 882E), nor would Federal Aviation Administration rigor guidelines (DO-178C) be adequate. NOSSA wanted to investigate requirements for unique analysis specific to AI development in military systems (Joint SSSEH v1.0). NOSSA also wanted to investigate if any new methodologies were needed to conduct adequate hazard analysis for AI deployed weapon systems (JS-SSA-IF Rev. A).

This research was motivated based on the six critical reasons why the Navy needs to establish measurable confidence in Machine Learned algorithms being deployed in weapons systems:

1. We cannot and should not expect the warfighter to accept and use AI as a social norm (Lapinski 2005), even when the best explainable AI techniques are available, without first having our Acquisition Community measurably establish confidence in Machine Learned algorithms being deployed in realistic operational environments.

2. Acquisition communities cannot identify and certify operational constraints of an ML algorithm for deployment without having confidence in the training data quality, including any negative side effects (Everitt 2018), that might result from the training process.

3. DoD Acquisition communities are limited when following commercial system safety guidelines because the commercial world does not have the same rigor requirements for ensuring AI functional behavior. Commercial manufacturers are driven by profit and may suffer from objective reasoning (Lewandowsky 2015) associated with the conflicting motivation to emphasize safety issues might result in lowering sales.

4. AI upgrades to Navy programs of record that were initially developed following a Capability Maturity Model for traditional software development (Shneiderman 2020) currently exclude ML/AI development differences. Acquisition communities need support and oversight to fill this gap.

5. It is imperative that “Speed to the fleet” deployment of AI systems must overcome their motivational limitations and consider safety impacts of AI using planning, oversight and continuous monitoring by knowledgeable review boards that includes retrospective analysis of disasters (Shneiderman 2016).

6. Navy Weapon System Explosive Safety Review Board (WSESRB) and other approval oversight authorities are limited in their assessment without adequate guidance and tools (Porter 2020, Jones 2019). Guidance and tools need to be a priority in DoD budgets.

AI has a potential of creating a technology leap (Eden 2013). That potential leap, especially when dealing with weapon systems, needs scrutiny. This scrutiny focuses on the specificity of the composition and size of the training data. This research will describe the needed scrutiny by oversight groups can use to increase safety and confidence in the deployment of AI functions.

The AI Acquisition Paradigm

An ML/AI function is selected because it can handle “noisy” inputs and still make a decision as to category or value of the output, the former being categorization and the latter being regression. The success rate of an ML/AI function is the primary measurement, but success is limited to the quality and quantity data input used to train the algorithm. Because of this dependency, “garbage in, garbage out” becomes a determining factor in the capability of the algorithm. For ML/AI functions, the degree of “garbage in” can affect how unpredictable, unexplainable and goal uncertain the algorithm performs (Amodei 2016).

Machine learning is a process where input data is used to train the algorithm to determine a correct answer. In general, training data sets have two parts: (1) the attributes that the function is learning to recognize, most times called instances, and (2) a truth label that describes the categorization of those attributes to train on correct answers. A trained ML function receives attributes and determines whether those attributes belong to a category such as a dog or cat. This research investigated measuring various aspects of attributes used for categorization. In the research, we divided attributes within the training set into three levels of significance: (1) primary, (2) secondary and (3) tertiary. In our sandbox analysis, we considered and determined that tertiary was unnecessary with regard to its modality. Our concern focused on the effects missing or sparse data occurrences had on the most significant attributes, i.e., a noisy operational environment where the unexpected happens. Unexpected examples might be communication link failures, sensor malfunctions or human data input error.

Training set size and composition (Foody 1995) is the principal ingredient that establishes the quality and quantity of a Machine Learned algorithm. No matter how exceptional is your Data Scientist development team, without the adequate quality and quantity of training data, the algorithm will never meet operational needs. The problem is that training data is a new paradigm for acquisition managers to consider. There are methods to test the output (Pei 2017) to determine incorrect corner case behaviors. Some of this tests are provide “whitebox” analysis. Yet, even these tests don’t provide insights into the composition and size of the input, again raising the concern about “garbage in, garbage out.” Key areas needing to be addressed regarding “negative training” (Rodríguez-Pérez 2017), i.e., things to not categorize, or how well noisy data occurs within a “realistic” operational environment cannot be addressed without looking at input. How well does input represent missing and sparse data issues and how much of the data set training consist of these examples. Testing the output or reviewing the array of weights inside the box might provide insights, but direct measurements of inputs will provide facts.

Instead of the output performance, our approach focused on measuring training data input as an approach to increase algorithm success rate reliability (Kim 2014). Preparing the training data for measurement is a form of curation within the Data Science field. This type of curation rigor of the input will aid the developer in thinking about how "noise" might affect the algorithm when deployed in its operational environment. Obviously, it is always important to measure output, but this research demonstrates the value of detailed measurements of the input.

Training ML algorithm based on operational environment "realism" was the primary motivation behind the development of these measurements. Since "realism" was the goal, it was necessary to create a program that represented products that would eventually be deployed in the operation environment. For the purpose, a "Sandbox" was created.

Our "Sandbox" – Because "Seeing is Believing"

Results of this research are based on using a "Sandbox" implementation approach that represents completion of three phases of a four-phase research approach. A "Sandbox" implementation approach means that a project was created, stakeholder requirements generated, architecture defined, design constructed, code developed and tests conducted within a confined, controllable environment for training, experimentation and analysis. Our Sandbox is designed to support seven different AI-enabled algorithms. To support realism, a mocked up acquisition program was created that consisted of five different AI-enabled algorithms supporting a mission planner and three different AI-enabled algorithms supporting two deployed autonomous vehicles. Both mission planner and autonomous vehicles had a full set of DoDAF system diagrams and UML Sequence Diagrams defining interfaces associated with software message transfer, SQL commands and application programming interfaces (APIs). These artifacts were designed in detail and reviewed before proceeding with ML algorithm investigation and development. Using this process allowed for an understanding of needed measurements regarding training sets. We also developed and reviewed a graphic user interface (GUI) and how human interaction plays a role in safe AI.

The Sandbox provided an opportunity for experimentation of an integrated hybrid system, combining various AI technologies to represent advanced capabilities (Baum 2011). This hybrid system allowed for "what-if" variations and intentional mistakes to investigate and test various measurements and approaches that could affect accurate forecasts and thereby resolve ML behavioral issues in advance. The following ML algorithms were either coded or design reviewed for implementation in the Sandbox: (1) for the Mission Planner -- Naïve Bayes, Logistic Regression, Random Forest, k Nearest Neighbor and XG-Boost, and (2) for the autonomous vehicles – Deep Neural Network, Deep Reinforced Learning and Convolutional Neural Networks. Algorithm design review included hyper-parameters variations specific to the algorithm under investigation.

The eventual goal of the "Sandbox" is to develop code and analytical measurements for all five different AI-enabled algorithms supporting the mission planner and all the three different AI-enables algorithms used within the autonomous vehicles. Within the sandbox environment, identified AI-enabled systems were analyzed and the measurements described below were identified to address the issue of how quality and quantity of training data might affect the confidence in the behavior of the algorithm in a deployed environment.

This phase of the research has resulted in postulating 14 tips that include best practices and measurements spanning requirements, architecture, design, development and test. All 14 tips focus on how to improve confidence in ML algorithm behavior. This paper presents results associated with two key tips regarding measurements to determine if there is adequate quality and quantity of data within a training set for the ML algorithm to meet operational needs. The measurement approaches described in this paper are to demonstrate the reliability of the training set in establishing confidence in the behavior of the machine learned algorithm.

The paper will highlight insights into both the quality and quantity of the attributes within the training set instances. An instance is a single sample of data used for training the algorithm. The motivation of this research is to include the proposed measurements as part of Objective Quality Evidence (OQE) gathering when submitting recommendations by system safety practitioners for Weapons Systems Explosive Review Board (WSESRB) review in support of justifying ML behavior confidence. In addition to OQE, the research findings will also provide valuable insight to the acquisition community, to include program managers, and test and evaluation engineers.

Training Data Modality

When creating training data, it is important to understand the operational environment being represented in order to ensure adequate development of the ML algorithms. The training data is either found from live events or synthetically created to match the operational scenario that will be provided as input to the ML algorithm. Therefore, the ML algorithm must learn how to perform under these conditions. Three types of modality represent various operational environments that can be encountered during deployment, where the type of modality defines how the ML algorithm needs to be trained.

ML Training Data Modality 1: This modality supports training data sets that are based on an operational environment from multiple data sources, where each source contains one or more attributes as described in Figure 1. In Figure 1, the various sources of separate data attributes is either found from live events or synthetically simulations created to match the deployed operational scenario. Therefore, the input for ML algorithm for training needs to replicate the input that will be received during deployment.

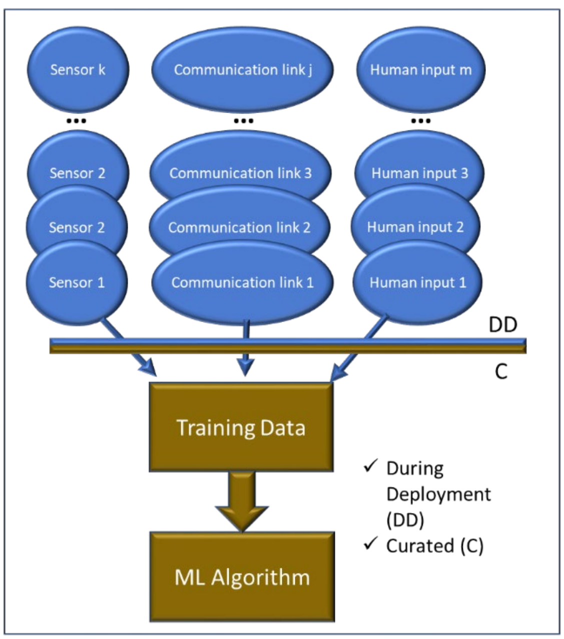


Figure 1. ML Training Data Modality 1

ML Training Data Modality 2: Training data sets that are based on an operational environment from a single data source, where the single data source contains multiple data attributes as described in Figure 2. In Figure 2, the one stream set of aggregated attributes is either found from live events or synthetically simulations created to match the deployed operational scenario. Therefore, the input for ML algorithm for training needs to replicate the input during deployment.

Figure 2 represents several versions of Modality 2, labeled (a), (b), (c) and (d). Version (a) describes the simple case where a sensor is capturing an image (in some frequency spectrum) that contains all the attributes needed to train the ML algorithm. As in all versions, Version (a) contains all the attributes needed to train the ML algorithm based on how the algorithm will be operationally deployed. Version (b) describes how one sensor might create a string of images causing channels for the ML algorithm to learn. Each channel might require one algorithm or a unique set of algorithms for processing. Version (c) describes a series of images, similar to Version (b), but in this case, as in an attempt to capture a 3-D image, where the combination of each slice of the image may constitute a single attribute that is part of the training. Finally, Version (d) describes how multiple attributes sources might be fused/combined into one source that will be used for training the ML algorithm. Again, the selection of Modality 2 Versions is based on the operational need/requirements associated with its deployment.

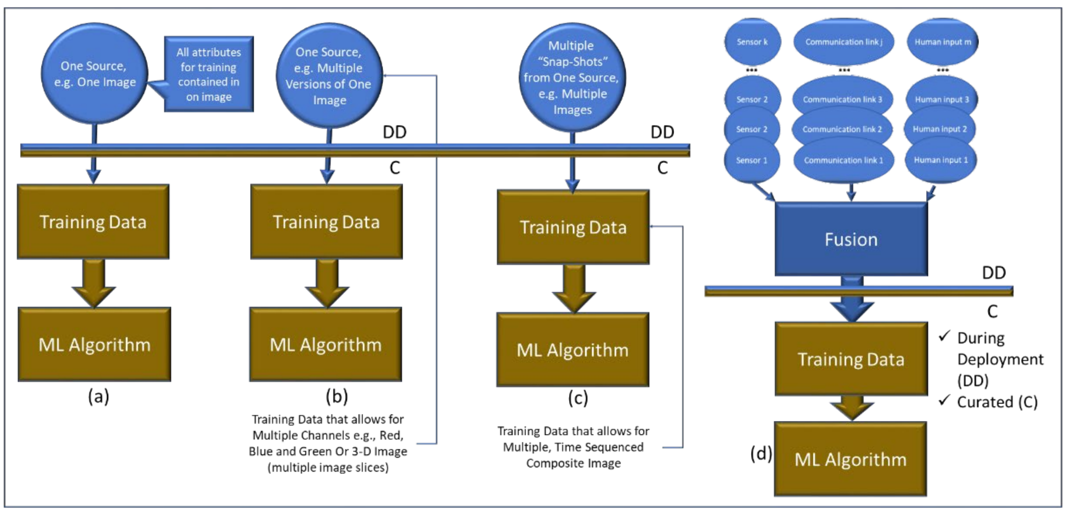


Figure 2. ML Training Data Modality 2

ML Training Data Modality 3: Training data sets that are based on an operational environment from a combination of multiple data sources where each source contains one or more attributes form various sources and from a single source containing multiple aggregated data attributes.

Modality 3 is the most challenging data set to replicate or find that can adequately represent "realistic" operational environments. In all three modalities, the primary challenge with using adequate data sets for ML training is to ensure the training set accurately represents "realistic" operational environments. The more complex in the composition of data sources that the ML algorithm needs for training in order to adequately perform its function, the more challenging it is to replicate a "realistic" training set that includes issues, such as communication failures over data links, unintentional human input error or sensor malfunctions. Additional challenges stem from adequately replicating noise that blur, surround or somehow challenges the data source feeding the ML algorithm. For example, synthetic replication of a single attribute over various slices of an image may be difficult to create with the adequate noise background, e.g., the blur needs to be consistent. The difficulty increases when that attribute needs to train the ML algorithm using hundreds of slight variations. Complications increase when dozens of attributes need to be included within the slices of images that will constitute a training set will realistically represent the required operational environment.

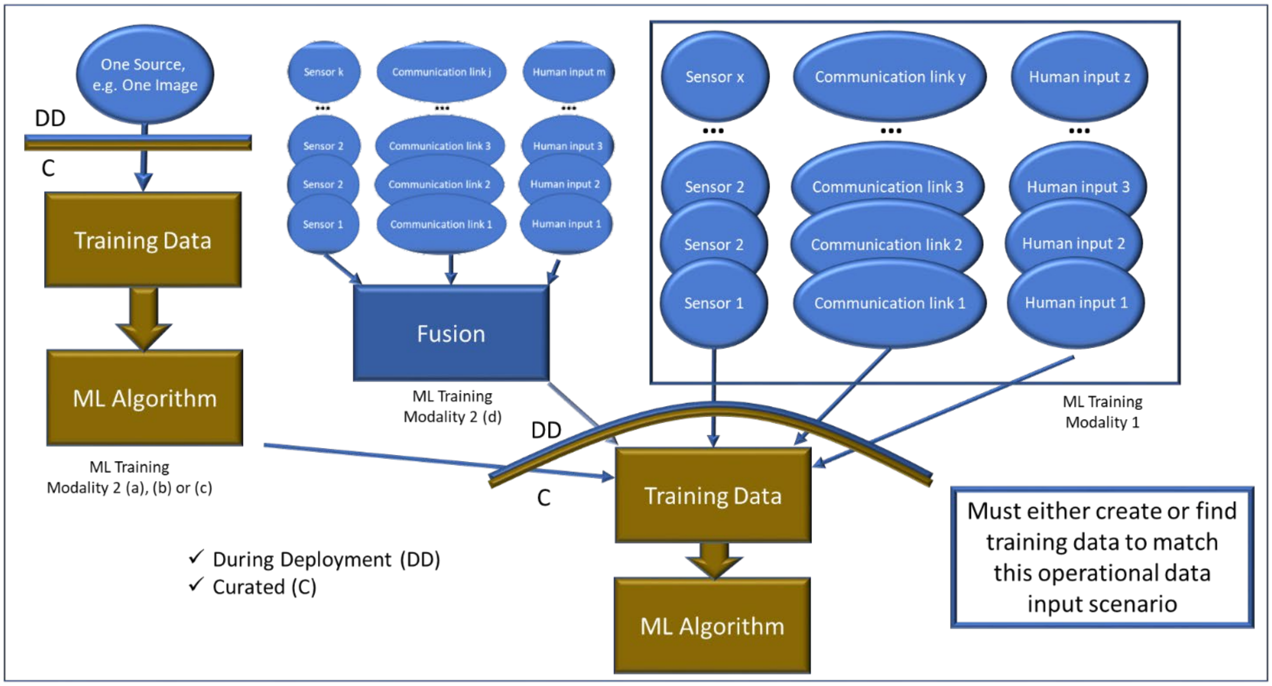


Figure 3. ML Training Data Modality 3

Given these potential challenges, a need to measure the quality and quantity of the data set to meet the operational needs of a “realistic,” noisy environment become essential to ensure confidence in the behavior of the ML algorithm during deployment.

Missing and Sparse Data Effects on Modality

For Modality 1, missing data can be represented as a sensor, communication link or human input issue. The sandbox software filtered the faulty data as being out of performance bounds and therefore provided no values. Sparse data occurred when the sensor, communication link or human input was properly working but data was unavailable for the ML algorithm to use. In this case, the sandbox implementation design provided zero values for those data sources. The sandbox software handled the zero values as no input to the ML algorithm. The challenge for Modality 1 is that missing and sparse issues can occur at the same time from different data sources. Either the data source fails, causing missing data or the data source does not register any input. In either case, the sandbox software filtered the sparse or missing data and therefore provided no values in those situations. Therefore, when developing AI functions, the training data needs to represent these occurrences and the developer needs an approach to handle the occurrences of sparse or missing attribute data. In the case of the sandbox, the training set consisted of secondary attributes when the primary attributes were not present. Primary attributes represent nominal input expectations, whereas secondary attributes are a back up to the unexpected.

For Modality 2, we still used our previous definition of missing or sparse data, when either occurs from a single sensor. Because a single source was replicated, only one could occur but not both. From our sources, this was an accurate representation of a realistic operational environment. For Modality 3, the combination of missing and sparse data could occur causing significant replication issues with regard to the training data set supporting a "realistic" operational environment. It is important to note that in Modality 2 and 3, missing and sparse data issues become even more challenging because filter techniques, like used in our sandbox, are harder to apply. For example, if missing data occurs, then the sensor may be malfunctioning causing blurs in the image, which would require the developer to train on secondary attributes that compensate for this type of blur in the picture. Once the first algorithm failed to categorize above threshold, this compensation approach may require a second algorithm trained on primary and secondary combination of attributes within the image. If the image contains no attributes, potentially from a sparse data issue, then categorization is impossible, there would be no secondary attributes to use. Again, complexity of how the training data is composed becomes more challenging, but needs to be addressed as part of oversight.

Training Set Composition Measurements

In developing our measurement approach to better ensure a training set represented a "realistic" operational environment, it was important to measure how well the training data represented both in quality and quantity missing and sparse issues with the data sources. Table 1, rows (a), (b), (c) and (d), represents questions that need to be addressed based on Modality, first by adequately defining the operation environment the training set represents based on modularity and then by ensuring the quality and quantity of data is adequate for the ML algorithm training.

	Investigation Topic	(Modality 1) multiple data sources, where each source contains one or more attributes	(Modality 2) single data source containing multiple data attributes, e.g., CNN	(Modality 3) combination of multiple data streams, where each stream contains one or more attributes and from a single data stream containing multiple aggregated data attributes, e.g., Naïve Bayes aggregated with CNN
(a)	Data Source Precedent for Improving Success Rate (ranking of primary, secondary tertiary… n attributes)	Which sensor, communication link or human input content elements take precedent over others for improving success rate when training the ML algorithm under normal to stressed operational conditions?	Which attributes within the single data source take precedent over others for improving success rate when training the ML algorithm under normal to stressed operational conditions?	What data source content is more significant with regard to normal to stressed operational conditions? When dealing with separate streams, which sensor, communication link or human input content elements take precedent for improving success rate when training the ML algorithm under normal to stressed operational conditions? When dealing with combined streams, which attributes within the single data source are identified as primary, secondary and tertiary regarding importance for ML algorithm to improve success rate under normal to stressed operational conditions?
(b)	Missing and sparse data issues modeled	How is sensor malfunction, message corruption and human input errors on the higher precedent attributes forcing lower level attribute mixes of training data to ensure algorithm can deal with "real" operational issues?	Corruption in parts of image, especially containing higher precedent attributes forcing secondary and tertiary attribute mixes of training data to ensure algorithm can deal with "real" operational issues.	Combinations on modalities 1 and 2 regarding training of algorithm to deal with "real" operational issues.
(c)	Quality of Training Data Characterized	What is the precedent list (from highest to lowest) of attributes being used for training.	Same as Modality 1 for this row.	Same as Modality 1 for this row.
(d)	Quantity of Training Data Characterized	How much more emphasis Is placed on quantify of training data variations that have higher precedent than lower?	Same as Modality 1 for this row.	Same as Modality 1 for this row.

Table 1. ML Training Data Investigation Topics by Modality Types

In Table 1, each row represents a series of questions associated with the modality of the ML Training set. Row (a) introduces the need to group attributes in terms of precedence/significance with regard to an expected operational norm and potential source failures (causing Missing and Sparse data issues) that the ML algorithm needs to learn in terms of data inputs. What attributes are primary to consider for training the ML algorithm? What attributes are secondary? Depending on the operational environment, there may be "n" number of groupings. Row (b) focuses on missing and sparse data modeling of attributes for training. As described in the previous section, missing and sparse data issues can direct attribute precedence. For example, when one primary attribute is not available, can another attribute in the secondary group be used to increase behavior confidence? Is the ML Algorithm being trained to use primary and secondary combinations of attributes? Row (c) defines an approach to analyze quality based on a precedence list. Within each group, what is of highest precedence for the ML algorithm so it can be adequately trained? The answer to this question ensures that the developer understands the relationship between attributes and the operational environment those attributes will support. Finally, Row (d) focuses on the need to understand if the quantity of training data is sufficient. Although quantity may be analyzed using overfitting and underfitting techniques specific to the algorithm being trained, this quantity analysis is based on how much more emphasis is placed on training data with higher precedence vs lower precedence. For example, if higher precedence/significant attributes are based on nominal operational conditions, then by definition of precedence/significance, slight variations of higher precedence attributes should have a greater or at least equal number of instances as compared to lower precedence attributes. If this is not the case, then why is one attribute group more significant over the other? Row (d) topic of investigation asks the questions, "Is there sufficient training data based on precedence grouping?" Measurements described in this paper provide answers to the topic investigation questions shown in Table 1.

Training Data Measurements

Using the sandbox, we created and examined two types of measurements that support answers to the questions posed in Table 1. The focus of both measurements is on attributes/features within each sample/instance.

If synthetic data is created, then a Design of Experiments (DOE) review needs to be performed during the requirements and architecture stages, e.g., somewhere during preliminary design review (PDR) and critical design review (CDR) timeframes. This become obvious while working within our sandbox development environment. In our sandbox, we used a modeling and simulation (M&S) approach to create training data. We developed a DOE that ensured primary and secondary data sources were created to support five ML classes using various combinations of seventeen attributes. Figure 4 represents the sandbox data sources, real-time and synthetic, and includes the operational environmental variables that are translated into 17 attributes supporting five classes/categories.

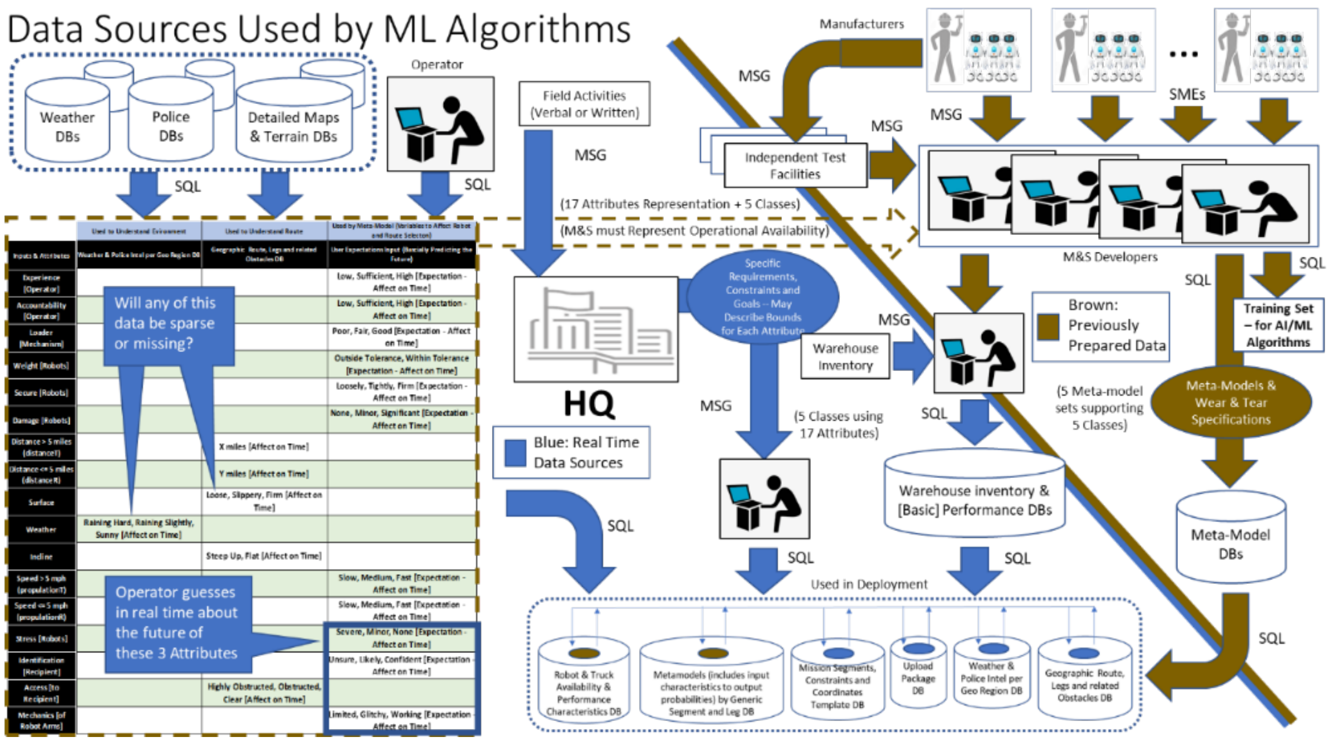


Figure 4. Operational Environment being Represented using M&S Synthetic Data

Training Set Alignment Test (TSAT)

Training Set Alignment Test (TSAT) is an approach we recommend using during requirements definition, architecture review and finally during algorithm code analysis. When creating synthetic training set data via modeling and simulations, attributes from highest to lowest significance should be identified in the Design of Experiments (DOE) to ensure proper emphasis is placed on primary, secondary to n-levels of precedence.

In our sandbox analysis using synthetic training set creation from simulations, we were able to develop the TSAT measurement process. Figure 5 diagrams the steps discussed below when taking a TSAT measurement:

- At Requirements stage and checked during Architecture review
  - First Step: Determine what attributes are most significant as compared to others in terms of the function the ML algorithm must perform. Note: this is based on the part the algorithm plays in the mission. What functions must it perform so the other subsystems can achieve their goals? For example, a common ML algorithm function is computer vision. What are the most significant attributes it should use to perform its image recognition function?
  - Second Step: Group the most significant attributes and consider them as primary attributes to the algorithm's learning process
  - Third Step: Group the other algorithms in terms of secondary and tertiary significance in terms of what the algorithm needs to learn

- When training set is produced, conduct analysis (Note: we've included the next three steps as part of Algorithm Code review because training data creates the weights and structures that constitute the deployed code).
    - Fourth Step: When the training data set is generated/gathered, use the statistics of how often an attribute occurred to determine the ranking.
    - Fifth Step: Perform a weighted calculation (similar to a discrete match filter in signal processing.)
    - Sixth Step: Determine if this grade, meaning the determination of how well the DOE goal matches the generated/created data. In this approach, the grades range up to 100%, where 100%i s a perfect alignment between operational needs and training data, where below 25% is extremely poor. Even with the most tolerant requirements, it is recommended that anything below fifty percent should not be accepted.

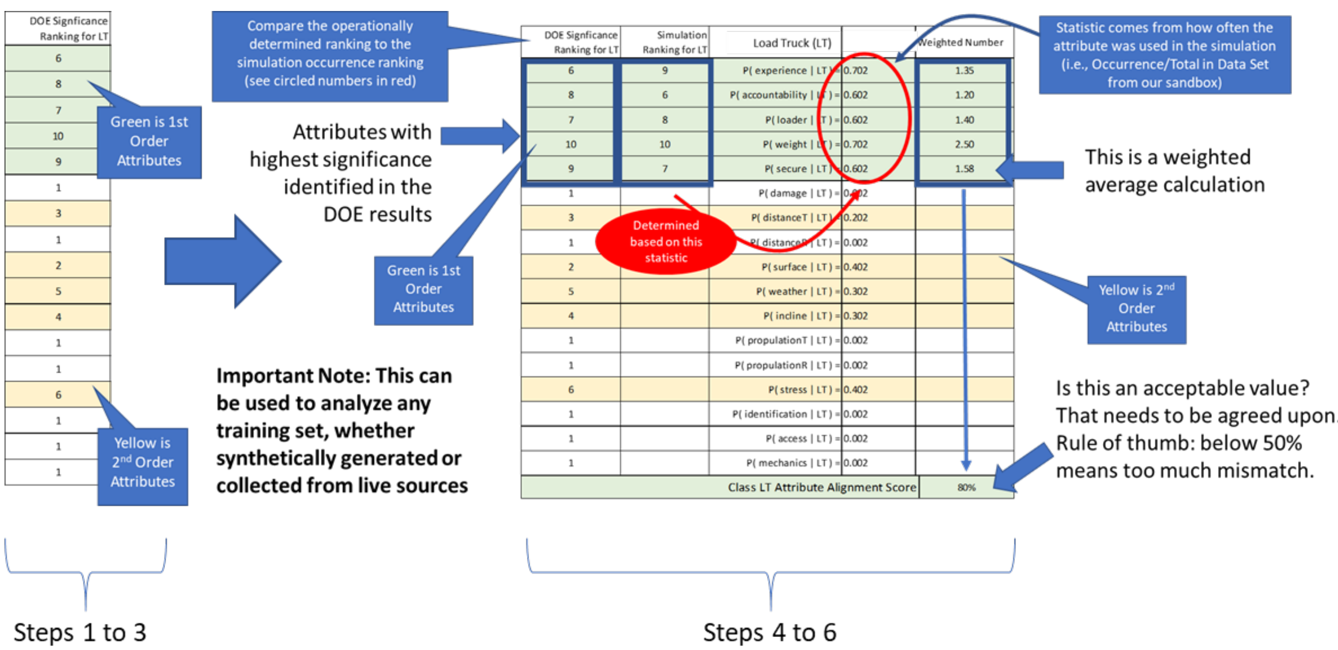


Figure 5. TSAT Diagram of Steps

TSAT ensures that the developer verifies that the attribute priority and ML training set modality is congruent between the deployed architecture and the training data set generation process based on the precedence/rating of attributes defined in TSAT. This will also document compliance to requirements for review.

Procedure for calculation:

1. Determine a scale for grading from 1 to "m," where "m" means greatest attribute priority/significance based on operational deployed needs.
2. Identify attributes a_1 to a_n to grade, such that "n" is the number of attributes being graded out of r total attributes available. Therefore $n \leq r$ and $n \leq m$, where grading a_i with grade "m" indicates $a_i(m)$ is the most important attribute based on operational needs. Additionally, attribute grading range is (m-n+1) to m, consecutively, where lowest grade indicates least operationally important (possibly DOE analysis and/or SME determination).

3. Identify the n attributes that occur the most times in the training data. Using the same scale "m," grade attributes b_1 to b_n based which attribute occurred the most often within the training set (this can be a statistical number, e.g., 70% of the time b_i attribute was used in simulations or 70% of the samples/instances were collected, e.g., images, that contained attribute b_i). Again, grade "m" indicates b_i occurred the most and (m-n+1) indicates b_j occurred the least within the training set.
4. Perform $k =)$ and $\beta =* bi(grade) \leq m$
5. Perform $* 100 = \alpha\% \geq 50\%$ as a constraint

Source to Attribute Ratios – nth Order Grouping (StAR-n)

Source to Attribute Ratios – nth Order Grouping (StAR-n) is an approach that can be used during requirements definition, architecture reviews and finally during code analysis. The basic premise is that attributes (e.g., primary, secondary or tertiary groupings) with the highest significance (precedence/rating) identified in the DOE (defined in TSAT) should be occur in greater numbers of instances within the Training Set than lower significance attributes. The comparison of numbers can be analyzed as ratios.

The reason why developers should verify that primary instances have greater numbers than secondary, and so on, is because: (1) With live data collection, there is a difficulty with finding or creating realistic training data that includes noisy environments representing missing and sparse data issues; and (2) With synthetic data creation, there is a physical limitation with how much simulation can be performed within the timeframe allotted? (Remember that most likely there is an infinite number of possibilities in terms of training data variations.) What should be the priority in your DOE?

In our sandbox analysis, we were able to develop the StAR-n measurement process. Figure 6 diagrams the steps discussed below when taking a StAR-n measurement:

- At Requirements stage and checked during Architecture review:
  - First Step: Create a ten by ten matrix, labeling each axis from zero to 1.
  - Second Step: Label the horizontal axis "% Number of Primary Attributes vs Total Attributes for Class" and the vertical axis "% Number of Primary Attribute Instances vs All Instances for Class"
  - Third Step: Determine a three-color zone scheme (see Figure 6 as an example), where green indicates that the ratio fell within acceptable limits, yellow indicates ratio is boarder line acceptable, and red color zone indicated ration is outside expected limits. Color of the zone should how well training data reflects operational environment. Based on color zone, determine evidence justification. Examples (used for guidance only) are described below:
    - Zone Green: Evidence of data by showing appropriate n-th order groups of training sets collected or generated by the simulations, including success rates as well as the TSAT results.
    - Zone Yellow: Zone Green evidence plus justification on why n-th group precedence can still handle the unexpected and provide acceptable success rates.
    - Zone Red: Zone Green and Yellow evidence as to how this algorithm is going to be supervised or monitored when operationally unexpected events occur.
- When training set is produced during Algorithm code review:

- Fourth Step: Calculate the σ and δ (see Figure 6 as an example) ratios. Each ratio should be less than 1. The example below is for primary attributes, but can be done for any n-th order attributes:
    - σ (by Class) = (Number of Primary Attributes / Number of All Attributes) ≤ 1.
    - δ (by Class) = (Number of all Primary Instances / Number of All Instances) ≤ 1.

- Fifth Step: Plot (x, y) using (σ, δ) pair of numbers and assess where the pair fall within the color zones to determine support action. An example is provided in Figure 6.
    - Zone Green: Evidence of data by showing appropriate n-th order groups of training sets collected or generated by the simulations, including success rates as well as the TSAT results.
    - Zone Yellow: Zone Green evidence plus justification on why n-th group precedence can still handle the unexpected and provide acceptable success rates.
    - Zone Red: Zone Green and Yellow evidence as to how this algorithm is going to be supervised or monitored when operationally unexpected events occur.

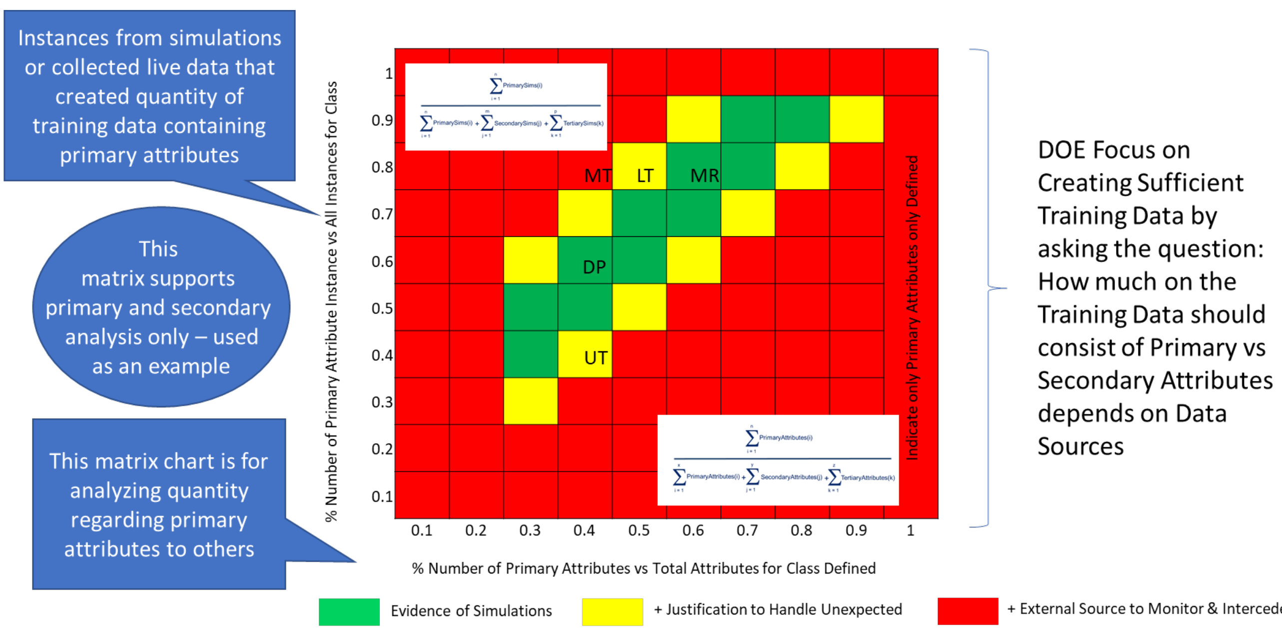


Figure 6. StAR-n Diagram of Steps

Notice that in Figure 6, the five classes have been plotted based on our sandbox results, provided as an example. Given the plot, there are two classes in the green, two in the yellow and one in the red. The color scheme relates to the type of justification needed in support of using the training data for those classes to adequately develop the algorithm to support the required operational environment. Since there is a significant mismatch when a (σ, δ) pair a plotted in the red zone, as in our example in Figure 6, we need to make sure that this algorithm has supervision when the expected occurs. Justification and the relationship to which boxes are colored needs to be described during requirements and checked during the architecture review. Again, the periods can be around PDR and CDR.

Matrices can be created for Primary, Secondary and Tertiary attributes, not just Primary. The StAR-n Grouping Matrix for this sandbox was only 2nd order. A StAR-3 looks at ratios of primary, secondary and tertiary attributes, as they are defined through requirements. As stated, training data is key to the development and the question becomes how much of the training data consist of primary vs secondary vs tertiary attributes as dependent on data sources that will be available in the field. Again, the issue becomes missing and sparse data during deployed operations.

StAR-n provides confidence to the system safety practitioner or test and evaluation engineer when the training data is generated synthetically and an attribute random generator is used. StAR-n ensures that justification is provided as Level of Rigor evidence based on primary, secondary, … n-th order attribute ratios to training data content ratios as part of the assessment of operational needs compared to what the training data contains.

In the Sandbox, an attribute random generator was used to create 15,000 simulations supporting 5 classes and 17 attributes. The analysis focused on determining the "Simulation to Attribute Ratios" for 1, 2 or 3 (nth) Order and graphed in a matrix to determine what type of rigor is needed to justify the ratio involved with each class being modeled via selected attributes. Consideration included how the attribute random generator creating the training data simulated an operational environment of sparse and missing data for the targeted algorithm to learn. The matrix using StAR-n identifies the need for the three types of justification, Zone Green, Yellow and Red, as described above.

StAR-n measures data source requirements, architecture and data set generation process specific to the categorized ratios of attributes defined. This measurement helps ensure that the developer is reflecting reality during algorithm development.

Combining TSAT and StAR-n ensures congruency between the operational environment and the training data set generation process. Figure 7 graphically describes the congruency using our sandbox classes between the blue operational deployment of the algorithms and the brown development of training data incorporating the attributes that will be available to the algorithm during operations.

It should be noted that labeling attributes/features, especially when live data is being collected, can be challenging. Labeling each instance within a Modularity 2 training set means looking at each sample and ranking, grouping and counting various n-order attributes. The challenge increases when evaluating effective sample size and the correlation between the attributes. Effective sample size affects the total number of instances used for training and therefore the classifier's performance/confidence interval (Figueroa 2012). Feature correlation affects total number of attributes used for training. Correlation can be measured and observed. For example, a smile affects two sides of the mouth. It would be inappropriate to consider both ends of the mouth as two different attributes. They are not independent events. Statistical independence of attribute within an instance affects algorithm training. A smile, a single attribute, can have many variations that affect both sizes of the mouth. Effective sample size is related to the randomness of each created or observed instance. Again, the samples should support statistical independence.

Instances and features within the instances need scrutiny to know if an algorithm is taught properly. Without this focus, the training would be uncontrolled. It would be like instructing children math and not knowing if they are being taught the "right" mix of problems or just the same problem with different letters for the variables. In training algorithms, it is important to ensure the training content is specific to the feature level of the education. This rigor is practiced in life sciences (Toloşi 2011) where wrong conclusions might lead to fatality. This same level of rigor should apply to any function performing operations that could cause lives to be at risk. Therefore, as applied to DoD, effective sample size and attribute/feature correlation of each instance needs to both be assessed as statistically independent (or at an acceptable low correlation) when applying TSAT and StAR-n measurements to training sets of algorithms performing operations that could cause lives to be at risk.

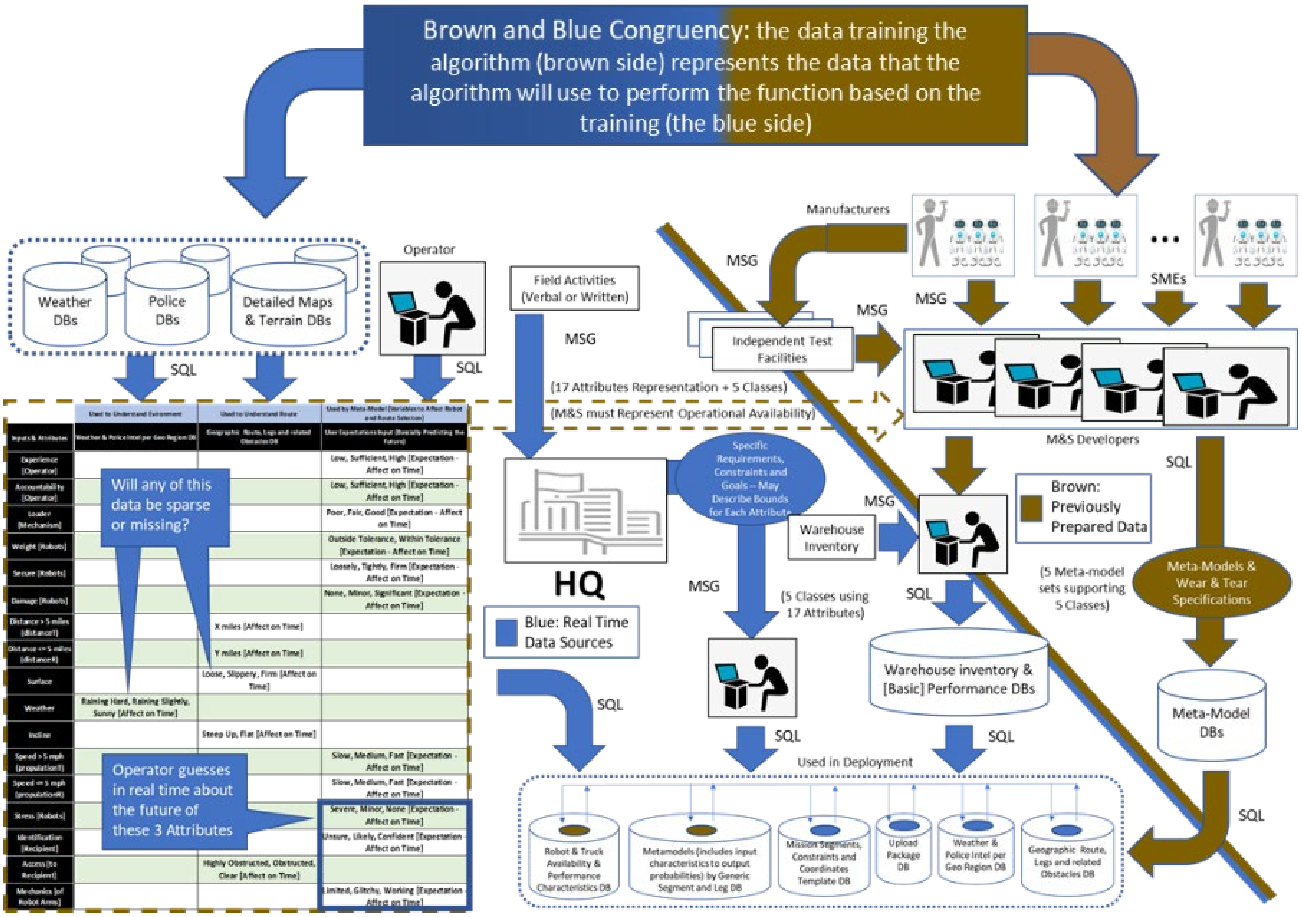


Figure 7. TSAT and StAR-n ensure congruency between what will be operationally deployed and what will be synthetically developed or collected from live data sources.

Numerical and Graphical Interpretation of Measurements

Using sandbox generated training sets, both TSAT and StAR-n were applied. Figure 8 and Table 2 represent the TSAT analysis for 17 attributes used by five classes from our sandbox.

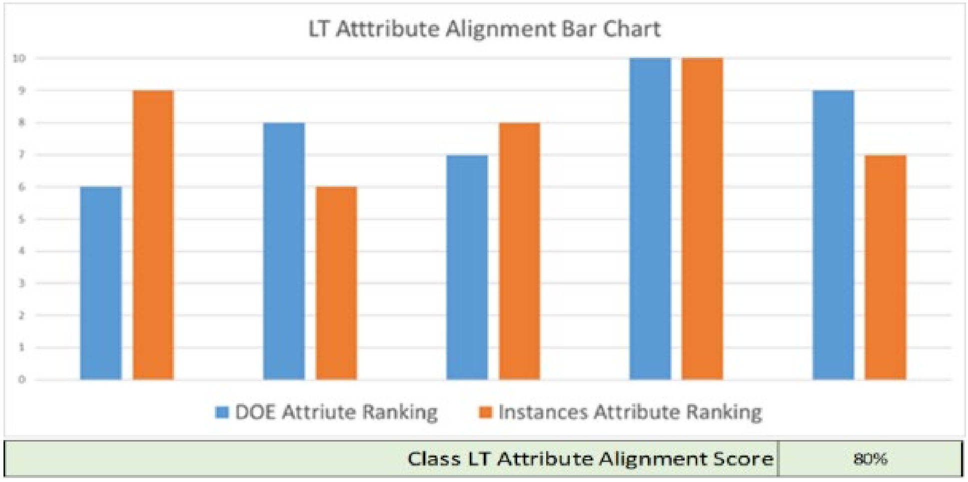


Figure 8. Sandbox Results from Applying TSAT Measurements for the LT class.

In Figure 8 (one of the five classes), you can notice that the blue and red bars are fairly equal in height causing a high score of 80%. Visually, if blue and red bars have significantly different heights, then a lower score will occur.

Class LT Attribute Alignment Score	80%		
	Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance	3.518333333	70%	50%
Secondary Data Source (2nd order significance	1.485	30%	50%
Tertiary Data Source (3rd order significance	0.002333333	0%	0%
TOTAL	5.005666667	100%	100%

Class MT Attribute Alignment Score	87%		
	Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance	4.085333333	68%	46%
Secondary Data Source (2nd order significance	1.919	32%	54%
Tertiary Data Source (3rd order significance	0.001333333	0%	0%
TOTAL	6.005666667	100%	100%

Class UT Attribute Alignment Score	79%		
	Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance	0.785	32%	45%
Secondary Data Source (2nd order significance	1.685333333	68%	55%
Tertiary Data Source (3rd order significance	0.001666667	0%	0%
TOTAL	2.472	100%	100%

Class MR Attribute Alignment Score	88%		
	Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance	4.468666667	74%	60%
Secondary Data Source (2nd order significance	1.534666667	26%	40%
Tertiary Data Source (3rd order significance	0.002333333	0%	0%
TOTAL	6.005666667	100%	100%

Class DP Attribute Alignment Score	85%		
	Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance	4.618666667	41%	60%
Secondary Data Source (2nd order significance	6.679077963	59%	40%
Tertiary Data Source (3rd order significance	0.002333333	0%	0%
TOTAL	11.30007796	100%	100%

Table 2. Attribute Alignment Scores for Each of the Five Classes

When looking a StAR-n ratios, Figure 9 describes a visual inspection of the two axis in the matrix. The "Instances Ratio" represents the vertical axis, "% Number of Primary Attribute Sims vs All Sims for Class." The "Attribute Ratio" represents the horizontal axis, "% Number of Primary Attributes vs Total Attributes for Class." In Figure 9, there are an equal number of primary and secondary attributes, as seen in the "Attribute Ratio" graph. In the "Instances Ratio" graph, although equal in number, there are more primary attributes in the training set.

Figure 9. Visual Review of Instances Ratio to Attribute Ratio in StAR-n Matrix

To better understand the significance of the ratios, consider the combinations of training instances based on use of the sandbox. In Figure 10, just focusing on whether an attribute, i.e., data source, will be present or not in the operational environment. Class combinations range from 821 to 2026, totaling 9580 different combinations. If we decided to train our ML algorithm to recognize a class based on attribute presences and numerical integer value, the combinations become extremely large. If we decide to have values in the real number domain, the combinations become infinite.

How much training data can you generate or collect to support 9580 combinations, in the simple case, or an infinite number of values in the extreme case? Therefore, it is necessary to prioritize attributes in terms of the number and type of instances within the training set. That is why TSAT and StAR-n are vital measurements.

Design of Experiments	1st Order Significant Attributes	2nd Order Significant Attributes	Total 1st and 2nd Order	Keep 2 1st Order	Keep 3 1st Order	Keep 4 1st Order	Keep 5 1st Order	Keep 6 1st Order	Total Configurations
LT	5	5	10	560	210	30	1	0	821
MT	6	5	11	1890	1120	315	36	1	3384
UT	*5*	*6*	11	840	280	35	1	0	1178
MR	6	4	10	1050	700	225	30	1	2026
DP	6	4	10	1050	700	225	30	1	2026
									9580

Figure 10. Attribute Occurrence Combinatorial Variations of Primary and Secondary Attribute Types per Class within Sandbox

From this research, three guidelines when using StAR-n to analyze ratios consistently surfaced.

Guideline 1: Order of precedence/significance should also describe ratio structure. Primary should have more instances than Secondary, Secondary should have more instances than Tertiary, etc. Describing the obvious, you would not want the ratios to be inverted, meaning the secondary would have more secondary attributes than the primary.

Guideline 2: Depending on the n-th order grouping of significance, there should be instances, therefore ratio values, for all n-th order combinations. Figure 11 describes when a tertiary attribute group, equal in number to the primary and secondary attributes, was omitted from the modeling and simulation. This means that the training data will not support the operational need associated with its deployment.

Guideline 3: If one of the n-th order grouping attributes is less than 5% (conservatively) in the attribute ratio graph, consider including it in other attribute groupings. Remember that a lower order attribute is likely to have less simulated or live data instances collected. This means that the instances will be lower than 5%, and likely be between 1 and 2% if Rule 1 and 2 are followed. Therefore, it may make more sense to include this n-th order attribute into another attribute group.

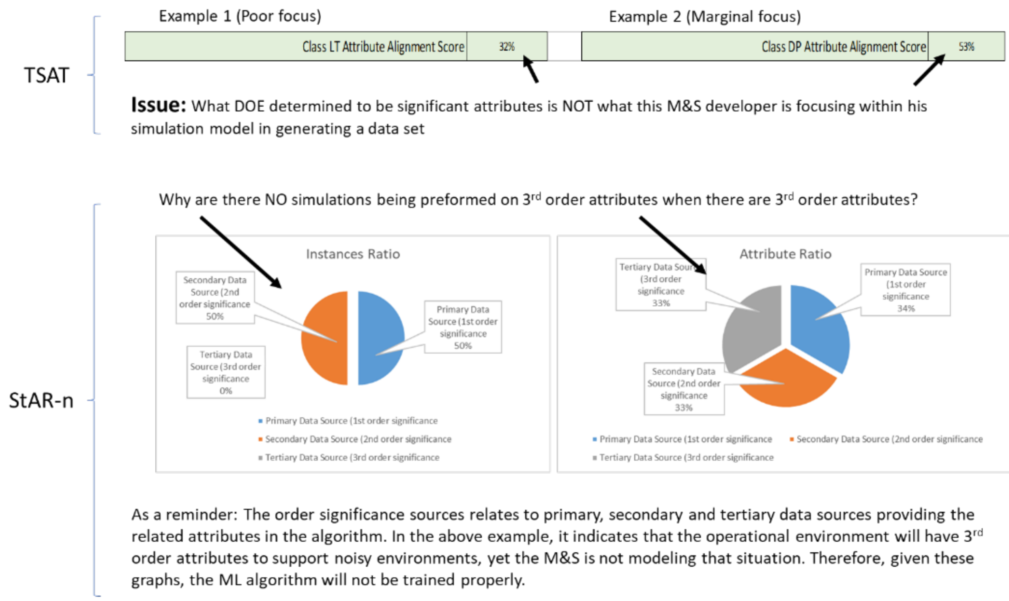


Figure 11. TSAT and StAR-n Identified Issues.

By following the three guidelines, valuable discussions can occur with the developer. As a reminder, the order significance sources relates to primary, secondary and tertiary data sources providing the related attributes in the algorithm. In Figure 11, tertiary data source creating attributes is not included, like our sandbox example, which means tertiary attributes i "not significant" from DOE viewpoint. It could be that there is less than 5%, which is causing the tertiary attributes to not be considered for ML algorithm training. In our sandbox, there were no tertiary attributes.

The Figure 11 example provides discussion points as follows:

- In Attributes Ratio, notice that 1st and 2nd attribute grouping have about the same number of attributes. Yet, if primary and secondary attributes are equally produced in the simulation, why is one group considered more significant than the other? This may not be wrong but definitely a discussion point for the developer. In this example, maybe 2nd order plays important role acting as a noisy environment or non-ideal environment. If of equal importance, then maybe there are only primary attributes. If so, is a noisy environment still being modeled? No right or wrong answers, just discussions that should be had on the operational environment and how the ML algorithm is being trained.
- In this example, there is a potential issue: If 1st and 2nd order are about the same number of simulations and this needs to be understood. What also needs to be discussed is why the fraction of attribute occurrence in the training set is so disproportionate to the fraction of total attributes? (Again, 2nd order acts as noisy or none ideal environment)

- Should the 2nd order and 1st order be a different ratio given the attribute numbers are about equal?
- Consider a better ratio, possibly 33 % to 66%, meaning I've run twice as many simulations on the 1st order vs the 2nd order.

Findings

From a sandbox approach, Table 3 became evident in terms of how much AI and traditional software code differ with respect to the acquisition process:

Category	Traditional Software (designed to provide a "certain" outcome)	ML/AI "Likelihood" Software (designed to provide a "likely" outcome)
Architecture	Architecture can vary.	Architecture needs to be portable and modular, e.g., micro-service and DEVOPS**.
Design	Design focuses on the actual development of the deployed code – the code is designed to provide a certain, specific outcome	Design focuses on the training data development that will create the deployed code – the code is designed to provide a likely outcome
Development	Development of code is done directly – coder determines the logic and math to use and codes it	Development of code is done indirectly – coder determines the logic and math used in the training (machine learning techniques), which results in the code
Test	Test is done on the code directly developed by the developer and to be deployed to prove certainty of an outcome	Test is done on the code (see below) developed from the math and manipulation of training data, i.e. use of machine learning techniques to prove likelihood of an outcome. If limited truth data, only use for testing.
Code Debugging	Can debug code that will be deployed	Cannot debug code that will be deployed – must retrain to create new weights and/or statistics for deployment
Transparency	The logic and math of the developer is shown directly in the code	The logic and math of the developer is not shown directly in the code
Critical Function Analysis	If a function is determined critical, the developer can have the math and logic challenged – focusing on the deployed code	If a function is determined critical, the developer can defend how the training data was created and manipulated – focusing away from the deployed code

Table 3. Traditional Logic vs Likelihood Software– Why treat ML/AI Algorithms differently in Acquisition.

A key question asked in this paper was, "Can the safety question with regard to weapon deployment regarding autonomy/AI ever be answered?" This paper answers this question in terms of rigor with regard to the training data. The measurements focused on improving confidence to an acceptable standard defined in requirements, checked during architecture and validated when reviewing the algorithm coding practices. To improve confidence of ML/AI behavior within the sandbox, TSAT and StAR-n measurements focus on n-th order grouping of attributes based on nominal operations for primary grouping, and non-nominal operations for lower level grouping. The cause of non-nominal operations is noise or faults in the deployed system. As described in previous sections, noise or faults result in missing and sparse data. How missing and sparse data affect the training data is based on the type of modality, as was discussed. TSAT and StAR-n measurements allows for ML algorithm training that ensures a match between the training data set and reality in the operational environment.

In the Sandbox, an attribute random generator was used to create 15,000 simulations supporting 5 classes and 17 attributes. The analysis focused on determining the "Simulation to Attribute Ratios" for 2nd Order analysis and graphed in a matrix to determine what type of OQE rigor was needed to justify the ratio involved with each class being modeled via selected attributes. Consideration included how the attribute random generator creating the training data simulated an operational environment of sparse and missing data for the targeted algorithm to learn. The matrix, using StAR-n, identifies the need for various types of rigor described previously based on where it is located in the matrix.

TSAT and StAR-n demonstrated that these measurement can support quality and quantity factual analysis that can be used by the acquisition community, including system safety and test and evaluation groups, to improve the confidence of the behavior of the algorithm to support a realistic deployment operations. From these measurement processes, issues associated with training, i.e., "garbage in," can be identified and resolved in advance and thereby increase ML functional confidence.

In our analysis, we were able to successfully use TSAT to ensure synthetic data for each of the five classes and 17 attributes had adequate quality and quantity of training data. The basic premise is that attributes (primary, secondary and tertiary) with the highest significance identified in the Design of Experiments (DOE) should be simulated more than attributes with lower significance. TSAT can effectively analyze Primary, Secondary to an n-th order data sources to determine if the training data is adequately aligned the Operational Needs defined in the DOE (note that the DOE must match operational use cases associated with mission parameter and environment).

1) We were also able to successfully use StAR-n. The Star-n can effectively use ratios involving primary, secondary to an n-th order, as they are defined by requirements and described in the architecture. As stated, training data is key to the development and the question becomes how much of the training data consist of primary vs secondary vs tertiary attributes, etc., as dependent on data sources that will be available in the field. StAR-n provides confidence to the system safety practitioner or test and evaluation engineer whether the training data is generated synthetically or collected from live events. StAR-n ensures that a Level of Rigor is provided based on primary, secondary and tertiary attribute ratios being with expected values.

By using both StAR-n and TSAT, the sandbox proved that quality and quantity of training data can be assessed. For TSAT, quality assessment meant the correct ranking of attributes (including primary, secondary, etc. mixes) that represented real world deployment issues associated with data source availability, which included noise factors. For StAR-n, quantity assessment meant the appropriate amount of samples/instances used for training based on operational priorities, which considered the mix of n-th order attribute ratios. Combining both measurements provides the Acquisition community, from project managers to test and evaluation engineers, the ability to maintain positive control over knowing that an AI/ML algorithm have been rigorously developed to support the expected behavior during deployment, even worst case environments. Why? Because using these measurements ensured that those environments were captured using the adequate quality and quantity of samples/instances needed to train the ML algorithm to deal with those issues.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J., & Mané, D. (2016). *Concrete problems in AI safety*. arXiv. https://arxiv.org/abs/1606.06565

Armstrong, S., Bostrom, N., & Shulman, C. (2016). Racing to the precipice: A model of artificial intelligence development. *AI & Soc*, *31*, 201–206. https://doi.org/10.1007/s00146-015-0590-y

Baum, S. (2017). On the promotion of safe and socially beneficial artificial intelligence. *AI & Soc*, *32*, 543–551. https://doi.org/10.1007/s00146-016-0677-0

Baum, S. D., Goertzel, B., & Goertzel, T. G. (2011). How long until human-level AI? Results from an expert assessment. *Technol Forecast Soc Change*, *78*(1), 185–195.

Bohannon, J. (2015). Fears of an AI pioneer. *Science*, *349*(6245), 252.

Borrie, J. (2014). Humanitarian reframing of nuclear weapons and the logic of a ban. *Int Aff*, *90*(3), 625–646.

Chauhan, G. (2018). *AI safety*. Towards Data Science. https://towardsdatascience.com/ai-safety-9aeb9ca42907#:~:text=AI%20Safety%20is%20collective%20termed,of%20real%2Dworld%20AI%20systems

Cooter, R. D. (2000). Three effects of social norms on law: Expression, deterrence, and internalization. *Oregon Law Rev*, *79*, 1–22.

Deci, E. L. (1971). Effects of externally mediated rewards on intrinsic motivation. *J Pers Soc Psychol*, *18*, 105–115.

DoD. (2012, May 11). *Department of Defense standard practice: System safety* (MIL-STD 882E). Pentagon.

Eden, A. H., Moor, J. H., Soraker, J. H., Steinhart, E. (2013). *Singularity hypotheses: A scientific and philosophical assessment*. Springer.

Etherington, D. (2012). Elon Musk says all advanced AI development should be regulated, including at Tesla, Tech Crunch. https://techcrunch.com/2020/02/18/elon-musk-says-all-advanced-ai-development-should-be-regulated-including-at-tesla/

Everitt, T. (2018). *Towards safe artificial general intelligence* [Doctoral thesis, Australian National University]. https://www.tomeveritt.se/papers/2018-thesis.pdf

Figueroa, R., Zeng-Treitler, Q., Kandula, S. & Ngo, L. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, *12*(8).

Foody, G., McCulloch, M., & Yates, W. (1995). The effect of training set size and composition on artificial neural network classification. *International Journal of Remote Sensing*, *16*(9), 1707–1723. https://doi.org/10.1080/01431169508954507

Griffith, E. (2016). Who will build the next great car company? *Fortune*. http://fortune.com/self-driving-cars-silicon-valley-detroit

Gurney, J. K. (2013). Sue my car not me: Products liability and accidents involving autonomous vehicles. *Univ Ill J Law Technol Policy*, *2013*(2), 247–277.

Hamilton, I. (2020) Elon Musk slams Microsoft exclusively licensing OpenAI's text-generating software, Business Insider. https://www.businessinsider.co.za/elon-musk-criticizes-microsoft-exclusively-licensing-gpt-3-2020-9

Joint Software Systems Safety Engineering Workgroup. (2010). *Joint software systems safety engineering handbook* (SSSEH v1.0). Pentagon.

Joint Software Systems Safety Engineering Workgroup. (2017). *Software system safety implementing process and tasks supporting MIL-STD-882E* (JS-SSA-IF Rev. A). Pentagon.

Jones, M. (2019). Artificial intelligence (AI): The need for new safety standards and methodologies (ISSC37-21). *37th International System Safety Conference*.

Joy, B. (2000). Why the future doesn't need us. *Wired*, *8*(4), 238–263.

Kim, Y., Sidney, J., Buus, S., Sette, A., Nielsen, M., & Peters, B. (2014). Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC Bioinformatics*, *15*(241).

Krantz, D. H., Peterson, N., Arora, P., Milch, K., & Orlove, B. (2008). Individual values and social goals in environmental decision making. In T. Kugler, J. C. Smith, T. Connolly, & Y. J. Son (Eds.), *Decision modeling and behavior in complex and uncertain environments* (pp. 165–198). Springer.

Kunda, Z. (1990). The case for motivated reasoning. *Psychol Bull*, *108*(3), 480–498.

Lapinski, M. K., & Rimal, R. N. (2005). An explication of social norms. *Commun Theory*, *15*(2), 127–147.

Lewandowsky, S., Cook, J., Oberauer, K., Brophy, S., Lloyd E. A., & Marriott, M. (2015). Recurrent fury: conspiratorial discourse in the blogosphere triggered by research on the role of conspiracist ideation in climate denial. *J Soc Political Psychol*, *3*(1), 142–178.

Lin, P., Abney, K., & Bekey, G. A. (Eds.). (2011). *Robot ethics: The ethical and social implications of robotics*. MIT Press.

McGinnis, J. O. (2010). Accelerating AI. *Northwest Univ Law Rev*, *104*, 366–381.

Moses, L. B. (2007). Recurring dilemmas: The law's race to keep up with technological change. *Univ Ill J Law Technol Policy*, *2007*(2), 239–285.

National Institute of Standards and Technology. (2019). *U.S. leadership in AI: A plan for federal engagement in developing technical standards and related tools*. U.S. Department of Commerce. https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf

Ozlati, S., & Yampolskiy, R. (2017). *The formalization of AI risk management and safety standards* (WS-17-02). AAAI Workshop on AI, Ethics, and Society. https://www.aaai.org/ocs/index.php/WS/AAAIW17/paper/viewFile/15175/14655

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). DeepXplore: Automated whitebox testing of deep learning systems. *Proceedings of the 26th Symposium on Operating Systems Principles*, 1–18. https://doi.org/10.1145/3132747.3132785

Porter, D., McAnally, M., Bieber, C., Wojton, H., & Medlin, R. (2020). *Trustworthy autonomy: A roadmap to assurance: Part 1. System effectiveness* (IDA Document P-10768-NS). Institute for Defense Analyses. https://www.ida.org/-/media/feature/publications/t/tr/trustworthy-autonomy-a-roadmap-to-assurance/p-10768.ashx

Radio Technical Commission for Aeronautics. (2012). *Software considerations in airborne systems and equipment certification* (DO-178C). Federal Aviation Administration.
Rodríguez-Pérez, R., Vogt, M., & Bajorath, J. (2017). Influence of varying training set composition and size on support vector machine-based prediction of active compounds. *Journal of Chemical Information and Modeling*, *57*(4), 710–716. https://doi.org/10.1021/acs.jcim.7b00088
Russell, S., Dewey, D., & Tegmark, M. (2015). Research priorities for robust and beneficial artificial intelligence. *AI Mag*, *36*(4),105–114.
Schmelzer, R. (2019). What happens when self-driving cars kill people? *Forbes*. https://www.forbes.com/sites/cognitiveworld/2019/09/26/what-happens-with-self-driving-cars-kill-people/?sh=7632fa27405c
Shneiderman, B. (2016). Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(48), 13538–13540.
Shneiderman, B. (2020). Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems. *ACM Transactions on Interactive Intelligent Systems*, *10*(4). https://doi.org/10.1145/3419764
Toloşi, L., & Lengauer, T. (2011). Classification with correlated features: Unreliability of feature ranking and solutions. *Bioinformatics*, *27*(14), 1986–1994. https://doi.org/10.1093/bioinformatics/btr300
Wiggers, K. (2019). *11 companies propose guiding principles for self-driving vehicles*. VentureBeat. https://venturebeat.com/2019/07/02/self-driving-car-report-safety-first-for-automated-driving/
Wiggers, K. (2020) Waymo's driverless cars were involved in 18 accidents over 20 months VentureBeat https://venturebeat.com/2020/10/30/waymos-driverless-cars-were-involved-in-18-accidents-over-20-month/
Wiggers, K. (2019) 11 companies propose guiding principles for self-driving vehicles VentureBeat https://venturebeat.com/2019/07/02/self-driving-car-report-safety-first-for-automated-driving/
Moses, LB (2007) Recurring dilemmas: the law's race to keep up with technological change. Univ Ill J Law Technol Policy 2007(2):239–285
Cooter, RD (2000) Three effects of social norms on law: expression, deterrence, and internalization. Oregon Law Rev 79:1–22
McGinnis, JO (2010) Accelerating AI. Northwest Univ Law Rev 104:366–381
Russell, S, Dewey, D, Tegmark, M (2015) Research priorities for robust and beneficial artificial intelligence. AI Mag 36(4):105–114
Baum, S. (2017) On the promotion of safe and socially beneficial artificial intelligence. AI & Soc 32, 543–551 (2017). https://doi.org/10.1007/s00146-016-0677-0
Kunda, Z (1990) The case for motivated reasoning. Psychol Bull 108(3):480–498
Yampolskiy, R. (2020) On Controllability of AI. arXiv:2008.04071v1. https://arxiv.org/ftp/arxiv/papers/2008/2008.04071.pdf
Chauhan, G. (2018) AI Safety. .
Deci, EL (1971) Effects of externally mediated rewards on intrinsic motivation. J Pers Soc Psychol 18:105–115
Krantz, DH, Peterson, N, Arora, P, Milch, K, Orlove, B (2008) Individual values and social goals in environmental decision making. In: Smith JC, Connolly T, Son YJ (eds) Kugler T. Decision modeling and behavior in complex and uncertain environments New York, Springer, pp 165–198
Armstrong, S., Bostrom, N. & Shulman, C. Racing to the precipice: a model of artificial intelligence development. AI & Soc 31, 201–206 (2016). https://doi.org/10.1007/s00146-015-0590-y
Borrie, J (2014) Humanitarian reframing of nuclear weapons and the logic of a ban. Int Aff 90(3):625–646
Bohannon, J (2015) Fears of an AI pioneer. Science 349(6245):252\ Joy B (2000) Why the future doesn't need us. Wired 8(4):238–263
Lin, P, Abney, K, and Bekey, GA (eds) (2011) Robot ethics: the ethical and social implications of robotics. MIT Press, Cambridge
Ozlati, S., & Yampolskiy, R. (2017). The Formalization of AI Risk Management and Safety Standards. AAAI Workshop, AI, Ethics, and Society WS-17-02.
National Institute of Standards and Technology. (2019). U.S. LEADERSHIP IN AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools (Response to Executive Order 13859). Washington D.C.: US Department of Commerce.

Defense Standardization Program Office. (2012) System Safety (MIL-STD 882E). Washington, D.C.: Pentagon.
Radio Technical Commission for Aeronautics. (2012). Software Considerations in Airborne Systems and Equipment Certification. (DO-178C). Washington D.C.: Federal Aviation Administration.
Joint Software Systems Safety Engineering Workgroup. (2010) Joint Software Systems Safety Engineering Handbook (SSSEH v1.0). Washington, D.C.: Pentagon.
Joint Software Systems Safety Engineering Workgroup. (2017) Software System Safety Implementing Process and Tasks Supporting MIL-STD-882E (JS-SSA-IF Rev. A). Washington, D.C.: Pentagon.
Lapinski, MK, Rimal, RN (2005) An explication of social norms. Commun Theory 15(2):127–147
Everitt, T. (2018) Towards Safe Artificial General Intelligence. PhD thesis, Australian National University, pages 20-21.
Lewandowsky, S, Cook, J, Oberauer, K, Brophy, S, Lloyd EA, Marriott M (2015) Recurrent fury: conspiratorial discourse in the blogosphere triggered by research on the role of conspiracist ideation in climate denial. J Soc Political Psychol 3(1):142–178
Shneiderman, B., (2020). Bridging the Gap between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems, ACM Transactions on Interactive Intelligent Systems 10, 4
Shneiderman, B., (2016) Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight, Proceedings of the National Academy of Sciences of the United States of America. https://doi.org/10.1073/pnas.1618211113
Porter, D., McAnally M., Bieber, C., Wojton, H., Medlin, R. (2020) Trustworkthy Autonomy: A Roadmap to Assurance Part 1: System Effectiveness. IDA Dcoument P-10768-NS, Log: H 2019-000369.
Jones, M. (2019) Artificial Intelligence (AI) - the Need for New Safety Standards and Methodologies. 37th International System Safety Conference, ISSC37-21.
Eden AH, Moor JH, Soraker JH, Steinhart E (2013) Singularity hypotheses: a scientific and philosophical assessment. Springer, Berlin
Amodei, D., Olah, C., Steinhardt, J., Christiano, P.F., Schulman, J., & Mané, D. (2016). Concrete Problems in AI Safety. ArXiv, abs/1606.06565.
Foody, G., McCulloch, M. & Yates, W. (1995) The effect of training set size and composition on artificial neural network classification, International Journal of Remote Sensing, 16:9, 1707-1723, DOI: 10.1080/01431169508954507.
Pei K., Cao Y., Yang J., Jana S. (2017) DeepXplore: Automated Whitebox Testing of Deep Learning Systems, ACM ISBN 978-1-4503-5085-3/17/10.
Rodríguez-Pérez R., Vogt M., and Bajorath J. (2017) Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds, Journal of Chemical Information and Modeling 2017 57 (4), 710-716 DOI: 10.1021/acs.jcim.7b00088
Kim, Y., Sidney, J, Buus, S., Sette1 A., Nielsen M., and Peters B. (2014) Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions, BMC Bioinformatics 2014, 15:241.
Baum, SD, Goertzel, B, Goertzel, TG (2011) How long until human-level AI? Results from an expert assessment. Technol Forecast Soc Change 78(1):185–195
Figueroa, R., Zeng-Treitler, Q., Kandula, S. and Ngo, L. (2012) Predicting sample size required for classification performance. BMC Medical Informatics and Decision Making 12: 8
Toloşi, L., Lengauer, T. (2011) Classification with correlated features: unreliability of feature ranking and solutions, Bioinformatics, Volume 27, Issue 14, Pages 1986–1994, https://doi.org/10.1093/bioinformatics/btr300

Artificial Intelligence Systems: Unique Challenges for Defense Applications

Bonnie Johnson—has over 25 years of leadership and systems engineering experience in naval engineering research and development. She was a Senior Systems Engineer at SAIC and Northrop Grumman, researching automated decision aids for naval combat systems and missile defense capabilities. She joined the Naval Postgraduate School (NPS) Systems Engineering Department in 2011. She has a PhD in systems engineering from NPS, an MS in systems engineering from Johns Hopkins, and a BS in physics from Virginia Tech. [bwjohnson@nps.edu]

Abstract

Today's warfighters are bombarded with information and faced with challenging decision spaces as technology exponentially expands and threat environments become more complex. Artificial intelligence (AI) and machine learning (ML) are advancements that can lessen the burden on the warfighter. AI systems offer far-reaching benefits—improving situational awareness and detection and understanding of threats and adversary capabilities and intents; identifying and evaluating possible tactical courses of action; and offering methods to predict outcomes and effects of course of action decisions. AI systems are the key to understanding and addressing highly complex tactical situations.

AI systems offer advantages to the warfighter, but only if these systems are engineered and implemented correctly and in a manner that lessens the warfighter's cognitive load. Implementing AI systems for defense applications presents unique challenges. This paper identifies four unique challenges and describes how they affect the tactical warfighter, the engineering design community, and national defense. This paper offers solution ideas for addressing these unique challenges through defense acquisition and systems engineering initiatives.

Keywords: Artificial intelligence, machine learning, complexity, tactical decision aids, systems engineering, trust, human–machine teaming

Introduction

AI is a field that includes many different approaches with the objective of creating machines with intelligence (Mitchell, 2019). Figure 1 shows a simple Venn diagram with machine learning (ML) as a subset of AI, and with AI as a subset of the broader category of automation. Automated systems function with minimal human input and often perform repetitive tasks based on commands and rules. AI systems perform functions that mimic human intelligence. They incorporate learning from past experiences with new information received to make decisions and reach conclusions.

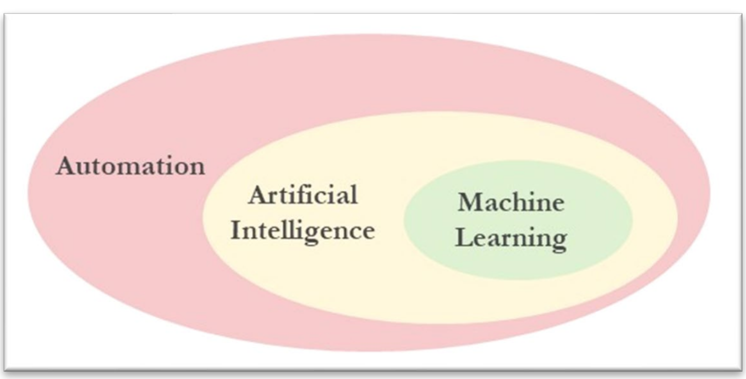


Figure 1. Venn Diagram of Automation, Artificial Intelligence, and Machine Learning

There are two primary types of AI systems, as described in Figure 2. The first type, which are explicitly programmed, are also known as Handcrafted Knowledge Systems. Allen (2020) described Handcrafted Knowledge Systems as "AI that use traditional, rules-based software to codify subject matter knowledge of human experts into a long series of programmed 'if given x input, then provide y output' rules" (p. 3). These systems use traditional, or normal, programming languages. The second type are ML systems that are trained from large sets of data. The ML systems "learn" from the trained data sets, and the "trained" system is then used operationally to produce predicted outcomes given new operational data.

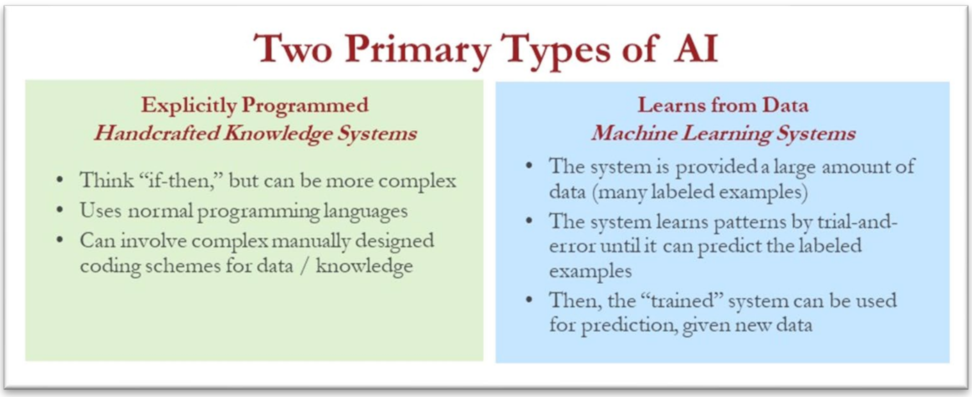


Figure 2. Two Types of Artificial Intelligence: Explicitly Programmed and Learning Systems

Automation, AI, and ML systems—both handcrafted knowledge systems and learning systems—offer great potential for the Department of Defense (DoD) with diverse applications in most mission domains. These intelligent systems can extend the DoD's abilities to make sense of complex and uncertain situations, to develop and weigh options, to predict the success of actions, and to assess the consequences. They offer the potential to support the DoD in strategic, planning, and tactical domains. AI systems can lessen the burden on the warfighter, but only if these systems are engineered and implemented correctly and in a manner that lessens the warfighter's cognitive load. Implementing AI systems for defense applications presents unique challenges. This paper identifies four unique challenges and describes how they affect the tactical warfighter, the engineering design community, and national defense.

The first unique challenge in implementing AI systems for defense applications is that tactical warfare presents highly complex situations. Tactical complexity can involve information overload, multiple concurrent missions that need to be addressed, time-critical decisions with dire consequences, unknowns/inaccuracies/incompleteness in situational awareness, and engineering challenges arising from the interoperability required among a diverse set of distributed warfare capabilities. Adding AI systems into this already-complex environment is a necessary but highly challenging endeavor.

The second unique challenge is that AI systems require large amounts of data for the systems to be trained. The quality of the resulting AI systems that are developed depends largely on the quality and quantity of the training data sets. Data in the military domain can be especially hard to come by. Military data may involve classification issues, cyber vulnerabilities, data validation challenges, and may simply be very costly and time-consuming to gather based on the need for fleet exercises and war games.

The third unique challenge is that the engineering of AI systems presents a new frontier for systems engineering. In traditional systems, behavior is set and is therefore predictable: given an input and conditions, the system will produce a predictable output. Some AI solutions may involve systems that are complex in their own right—adapting and learning—and therefore producing unforeseen outputs and behaviors. In fact, the intent of some AI systems is to do just that—team with a human decision-maker by taking on some of the cognitive load and producing intelligent recommendations. Systems engineering methods are needed to engineer intelligent systems and ensure that they are explainable, trustable, and safe to human operators.

The fourth unique challenge is that for defense applications there is always a potential adversary that needs to be considered. In terms of AI systems, the acquisition community must be mindful that peer competitor nations are making their own strides in AI advancements. U.S. defense systems must also advance in this AI race. Cyberattacks are always a possibility in defense systems. As defense capabilities increase reliance on automation and AI systems, this may be creating more cyber vulnerabilities. Finally, technology is rapidly evolving, and the adversarial threat space is changing. The defense acquisition and systems engineering communities must ensure that AI systems evolve and adapt to address changes in the threat environment and do this in a trustable and safe manner.

Challenge: Complex Decision Spaces

The first unique challenge is that many defense domains present a complex decision space. Therefore, engineering and implementing appropriate AI systems to address this complexity will be highly challenging. Figure 3 highlights some of the many factors that contribute to decision complexity in the tactical domain. Naval strike force operations, as an example, can quickly change from a peaceful state to one of great peril-- requiring alertness to the threat and appropriate response actions—all within a highly compressed decision time line. Tactical threats may arise from underwater, on the surface, in the air, from the land, from space, or even virtually, resulting in the need to address multiple time-critical missions. With naval and defense assets on ships, submarines, aircraft, land, and in space; the tactical decision space must address the optimal collaborative use of these dispersed and diverse resources. Developing effective tactical courses of action must also occur in highly dynamic operational environments with only partial and uncertain situational knowledge. The decision space must also consider constraints imposed by command authority, rules of engagement, and tactical doctrine. The role of humans as tactical decision-makers adds to the complexity of the decision space—with the challenges of information overload, operator error, AI trust, and AI ambiguity and explainability issues. Finally, the stakes can be very high for tactical decisions and their possible consequences.

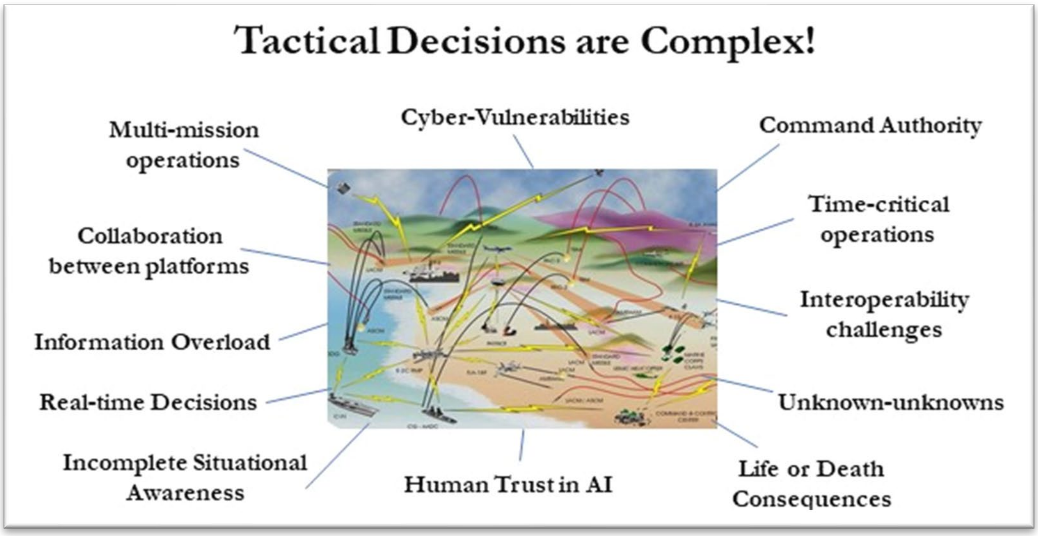


Figure 3. Factors that Lead to Tactical Decision Space Complexity

Addressing highly complex decision spaces is a challenge for the DoD. AI offers a potential solution to addressing this complexity—by handing large amounts of data, dealing with uncertainty, making sense of complex situations, developing and evaluating decision alternatives, and understanding risk levels and decision consequences. AI solutions can apply at the DoD strategic, planning, and tactical levels. The Naval Postgraduate School (NPS) has developed an engineering framework and theory for addressing highly complex problem spaces that require the use of intelligent and distributed AI systems for gaining situational awareness and making collaborative course of action decisions that adapt to a dynamic situation (Johnson, 2019). A complex tactical scenario was modeled to demonstrate the use of AI to validate the approach (Johnson, 2020a). NPS has developed a conceptual design for a predictive analytics capability to be implemented as an automated real-time war-gaming system that explores different possible tactical courses of action and their predicted effects and red force responses (Johnson, 2020b). NPS studies have identified the need to characterize the level of complexity during tactical operations and to implement an adaptive human–machine teaming arrangement to make tactical decisions where the level of automation adapts according to the level of situational complexity. Ongoing NPS research is studying the application of these conceptual engineered approaches in a variety of defense use case applications, including air and missile defense, over-the-horizon strike, ship self-defense, UAV operations, and laser weapon systems.

Complex decision spaces create challenging problems for AI systems to try and solve. Table 1 compares different AI application domains based on the complexity of their decision space. The table contains 10 factors that characterize the complexity of a decision space: epistemic uncertainty (the amount of uncertainty in the knowledge of the situation), situational dynamics, the decision time line (amount of time to make the decision), the complexity of the human interaction in the decision process, the resource complexity (the number, types, distance between them, and how dynamic they are), whether there are multiple missions involved, the existence of adversaries (competitors, hackers, or outright enemies that intend to destroy or overtake), the margin of allowable error (how much decision error is acceptable), and the severity of decision consequences.

Table 1. Comparison of Decision Complexity for Different AI Applications

	Epistemic Uncertainty	Situational Dynamics	Decision Time Line	Human Interaction in Decision Process	Resource Complexity (Number, Diversity, Geographical Dispersion, Dynamics)	Multi-Mission (Complexity in the Mission)	Training Data Sets (Ease of Obtaining, Data Rich vs. Data Poor)	Existence of Adversaries	Margin of Allowable Error	Decision Consequences
Loan Approval	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Advertising	Green	Green	Green	Green	Green	Green	Green	Yellow	Green	Green
Medical Treatment	Green	Green	Green	Green	Green	Green	Yellow	Green	Yellow	Red
Shipping Routes	Green	Red	Yellow	Green	Yellow	Green	Green	Green	Yellow	Green
Self-Driving Cars	Red	Red	Red	Yellow	Green	Yellow	Green	Green	Red	Red
Military Tactical Decisions	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
Key	Green	= little or no contribution to decision complexity		Yellow	= medium amount of contribution to decision complexity		Red	= high contribution to decision complexity		

The decision spaces involved in AI applications for advertising (determining which ads to stream to specific users based on their buying habits or Internet searches), loan approvals (determining loan eligibility based on loan amounts and credit scores), and medical treatments (determining diagnoses based on patient symptoms) are relatively straightforward. Large amounts of training data exist, calculations and human interaction in the decision process are straightforward, and the situations are relatively stable. The consequences of poor advertising are minimal. A bad loan approval decision can be audited. Poor medical diagnoses can have more serious consequences, but there is often enough time to seek more evaluation and opinions before treatment. Determining optimum shipping routes and engineering AI systems for self-driving cars are more complicated endeavors. These applications are dynamically changing and require shorter amounts of time to make decisions. Shipping routes will have complexity in the numbers of possible routes—which can result in many possible options. However, there is room for shipping errors, and the consequences are usually not too severe. The margin for decision error is very small for self-driving cars. Poor decisions in this application can cause serious accidents.

However, the military tactical domain presents extreme complexity in all areas of the decision space: uncertainty and limited knowledge/awareness, highly dynamic situations, very limited time lines, complicated human interaction, large numbers and types of resources, multiple missions, costly and hard-to-obtain training data sets, extremely small margins of allowable errors, and life-or-death consequences of actions (or inaction).

Challenge: Data Can Be Hard to Acquire

The second unique challenge is that AI/ML systems require large amounts of relevant and high-quality data for training and development, and these data can be hard to come by in the military domain. Handcrafted knowledge systems that are explicitly programmed need data during the development process for evaluation and validation. ML systems have an even greater dependence on data during development. As shown in Figure 4, ML systems "learn" from data sets that represent what the operational conditions and events will be. The process of ML system learning is also called being trained, and the data used during the development phase are called training data sets. There are several types of ML learning or training—these are supervised, unsupervised, and reinforcement. All three types of ML learning require training data sets. The ML systems continue to need data during the post-deployment or operational phase. Figure 4 shows that during operations, the ML system, or "model," receives operational real-time data and determines predictions or decision outcomes by processing the operational data with its "trained" algorithms. Thus, throughout the systems engineering and acquisition life cycles, the ML system is intimately connected to data. The ML system "emerges" from the process of learning from the training data sets. ML systems are a product of the quality, sufficiency, and representativeness of the data. They are wholly dependent on their training data sets.

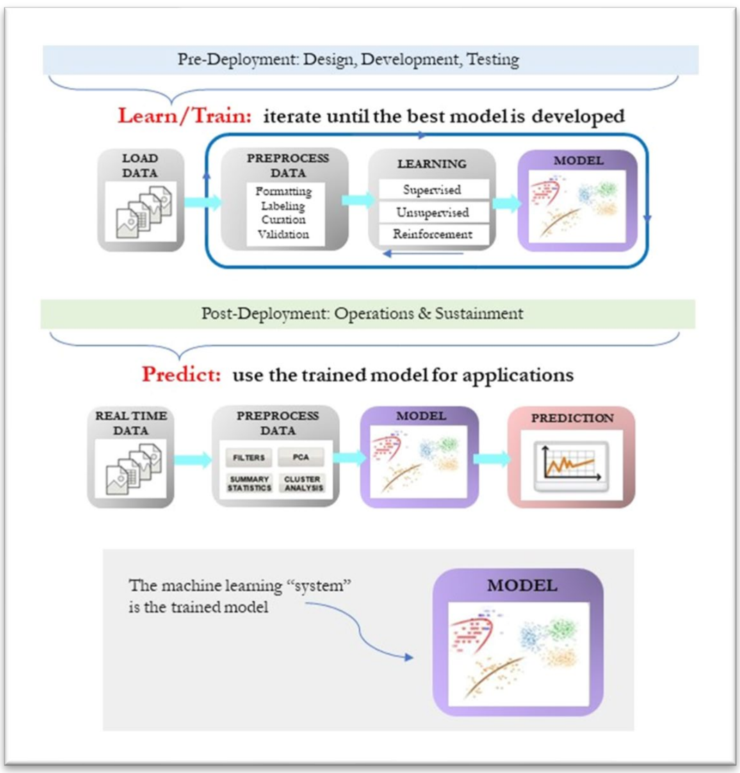


Figure 4. Developing and Implementing Machine Learning Systems

The DoD is beginning to recognize the need for these data sets as more AI developers in many domains (warfare, supply chain, security, logistics, etc.) are understanding the potential benefit of AI solutions and are embarking on AI system development. In some cases, the data exists and is ready to support AI system development. In other cases, the data exists but is not saved and stored. Finally, in other cases, the data does not exist and either needs to be simulated or gathered in fleet exercises or war games. Figure 5 illustrates a process of considerations that need to be made to gather, obtain, and in some cases develop data for use in developing and training AI and ML systems.

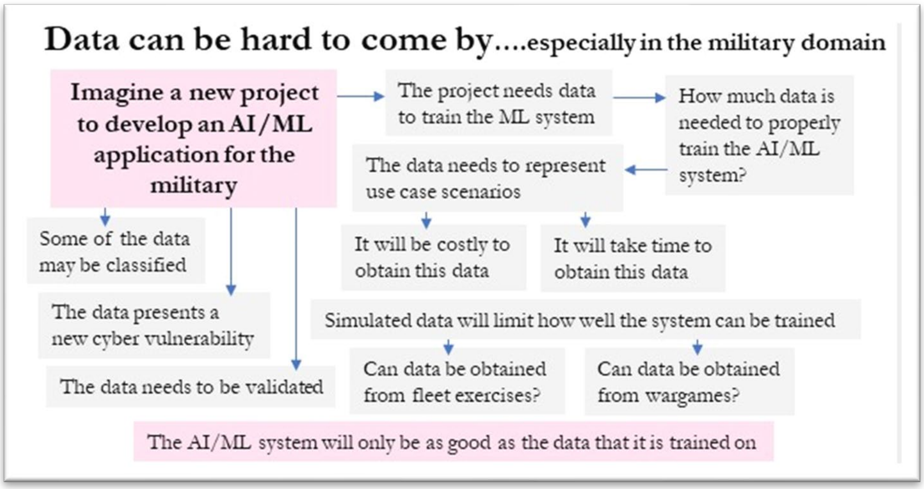


Figure 5. Development of Data Sets for Artificial Intelligence and Machine Learning System Training

The military domain presents some unique challenges for developing training data sets—the data may be classified, the data may present a cyber vulnerability (it could be hacked and purposely corrupted by an adversary), and if the data doesn't exist, it may need to be obtained from military/fleet exercises or war games. Data validation is a challenging endeavor as well.

NPS is performing a needs analysis and conceptual design for a data management system for the Navy that will collect and provide data to many disparate organizations within the Navy that are developing AI/ML systems (French et al., 2021). Figure 6 is a context diagram of the Navy Central Artificial Intelligence Library (CAIL) that is envisioned as a data management system and process for identifying data sets and providing indexing, validation, auditing, and secure access to data that can be used by AI/ML developers working on naval applications. The CAIL would not be a data repository or database, but instead, a central organization that enables AI/ML developers to access validated and secure naval data—to help identify the existence of data sets, enable the authorized access, and help support developers when data that is needed does not yet exist and needs to be obtained—possibly through fleet exercises or war games.

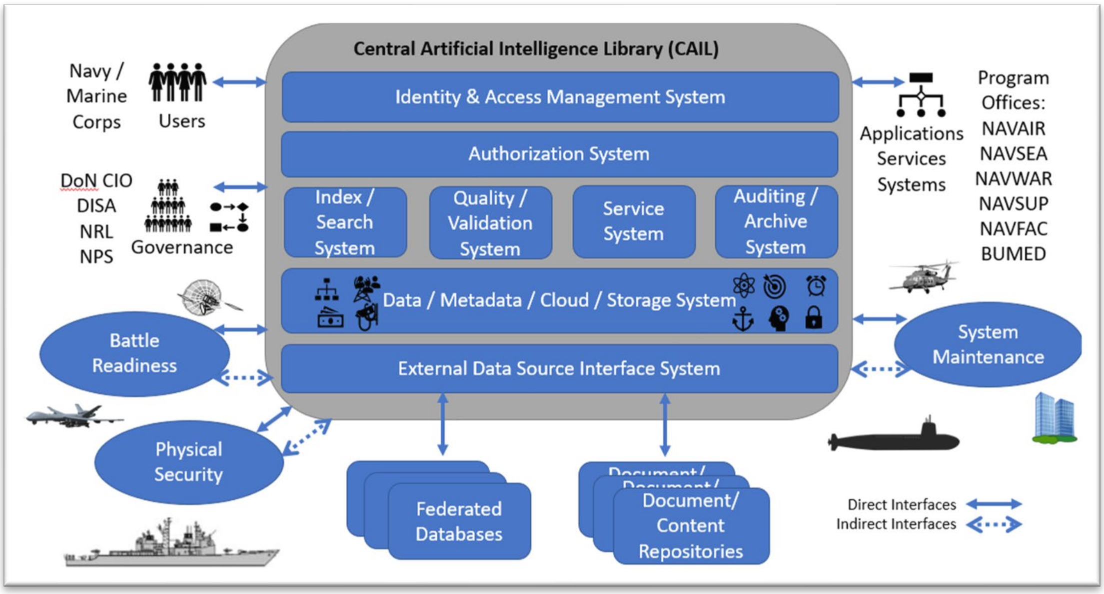


Figure 6. The Conceptual Central Artificial Intelligence Library (CAIL; French et al., 2021)

Challenge: AI Presents a New Frontier for Systems Engineering

The third unique challenge is that developing AI systems is presenting a new frontier for systems engineering. Systems engineering methods have been developed for engineering traditional systems that can be highly complicated but also deterministic (Calvano & John, 2004). Traditional systems have predictable behavior: for a given input and conditions they will produce a predictable output. Figure 7 illustrates the need for changes to traditional SE methods, like the SE Vee process, in order to engineer AI systems that are complex and nondeterministic. In particular, new methods will be needed to define requirements for a learning system that adapts over time, and the process of system validation may need to evolve and continue during operations to ensure safe and desired behavior. For military systems with high stakes consequences, there is very little room for error, so implementing a systems engineering process that can ensure safe and desired operations for AI systems is a requirement.

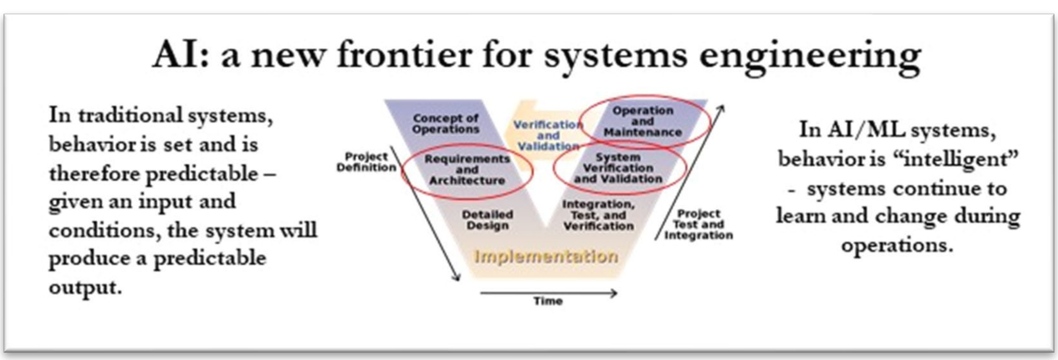


Figure 7. Artificial Intelligence: A New Frontier for Systems Engineering

A recent initiative by the International Council of Systems Engineers (INCOSE) has begun to explore what changes need to be made to systems engineering methods to effectively develop AI systems. Figure 8 was created as part of this initiative to highlight five aspects of AI systems that need to be considered during the SE process. In addition to nondeterministic and evolving behavior, AI systems may present new types of failure modes that are unanticipated, may occur suddenly, and whose root causes may be difficult to discern. Robust design—or ensuring that AI systems can handle and adapt to future scenarios—is another systems engineering design consideration. Finally, for AI systems with more involved human–machine interactions, careful attention must be paid to designing systems so they are trustworthy, explainable, and ultimately useful to the human decision-makers.

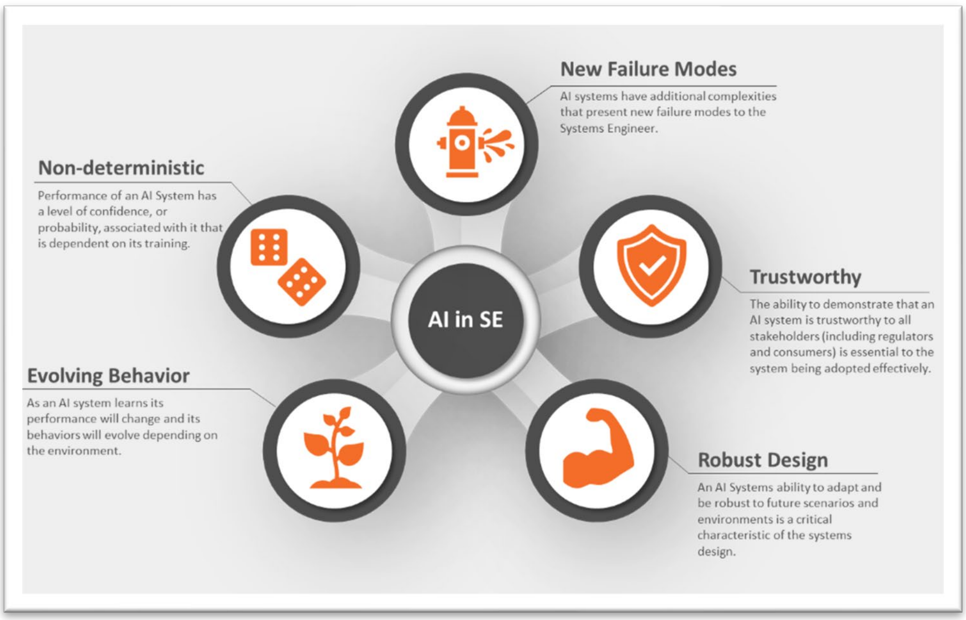


Figure 8. Challenges in the Engineering of Artificial Intelligence Systems (Robinson, 2021)

NPS is studying systems engineering methods that can support the design and development of complex, adaptive, and intelligent AI systems. A systems engineering framework and methodology has been developed to engineer complex adaptive systems of systems solutions (Johnson, 2019). The methodology supports the development of systems of systems that, through the use of AI, can collaborate to produce desired emergent behavior. A current research project is studying safety measures that can be engineered into AI systems during the design process to ensure safety during operations (Cruz et al., 2021). NPS is studying a design solution called *metacognition* as an approach for an AI system to identify internal errors (Johnson, 2021). Another current NPS thesis project is studying how to engineer "trust" into AI systems to ensure effective human–machine teaming arrangements (Hui, 2021). Several NPS research projects have studied the use of an SE design approach called *coactive design* to determine interdependences between human operators and AI systems (Blickley et al., 2021; Sanchez, 2021).

Challenge: Adversaries

The fourth unique challenge is the presence and role of the adversary in defense applications. The DoD must keep up in the race with adversaries to advance AI capabilities, AI systems must be protected from cyberattacks, and AI systems must adapt to the ever-changing evolution of the threat environment. Figure 9 highlights this unique set of challenges that the existence of adversaries presents for AI systems being developed for the DoD.

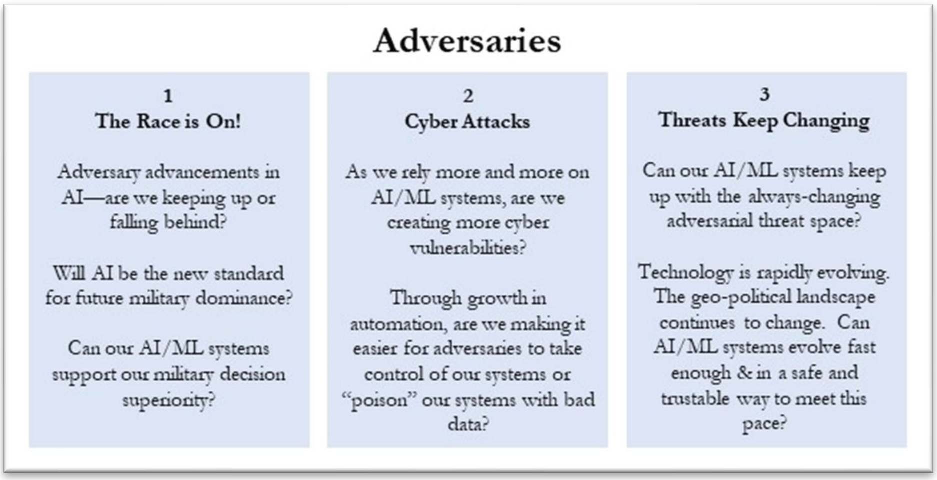


Figure 9. Adversarial Challenges

The race among peer competitor nations to develop AI capabilities is ultimately about getting inside the adversary's decision cycle to decide and act faster than the adversary can (Rosenberg, 2010). AI systems offer the potential to improve the quality and speed of decisions and are therefore critical to gaining decision superiority. As the DoD explores AI solutions, peer competitor nations are doing the same. Ultimately, realizing the goal of using AI for the DoD depends on more than AI research. It requires proper data gathering and management, effective systems engineering and acquisition methods, and careful consideration of the human interaction with AI systems. The DoD must ensure that it meets all the challenges involved with implementing AI systems in order to win the race. An NPS research initiative is studying how to apply AI and game theory to get inside an adversary's tactical decision cycle (Johnson, 2020b). This project is developing a concept for creating models of the tactical situation, the adversaries' location and capabilities, and a prediction of what the adversaries know about the situation. The conceptual system would then play a real-time "war game" to analyze tactical decision options based on predicted adversarial responses and second- and third-order effects. This is an example of studying what future tactical warfare might be like with enhanced knowledge and decision aids for both blue and red forces. Other NPS initiatives to prepare the DoD for the AI race include studying new SE methods and acquisition practices for developing AI capabilities, studying the data management needs of the Navy and the DoD (French et al., 2021), and studying AI system safety risks to develop engineering practices that ensure safe AI capabilities (Cruz et al., 2021; Johnson, 2021).

Cyber warfare is another race that the DoD must successfully compete in to stay ahead of the constant onslaught of hacking attempts. As the DoD implements more automation, it naturally results in more cyber vulnerabilities. The use of AI systems that are intrinsically dependent on both trained and operational data, opens up opportunities for hackers to poison the systems with corrupt data during the development phase and also during the operational phase. If an adversary gains control of an operational AI system, the possible harm they can inflict will depend on the application domain. For automation that supports weapon control decisions, the consequences can be deadly. In a recent study on automotive cybersecurity, a car company posted a fake vehicle electronic control unit online, and in under 3 days, 25,000 breach attempts were made (Taub, 2021). The DoD must be mindful of the particular cyber vulnerabilities presented as AI systems are developed. Careful cyber risk analysis and cyber defense strategies must be implemented for each new AI system. NPS is studying data security requirements for ensuring that ML training data sets are safe from hacking and will require secure authorization to access (French et al., 2021). NPS is studying the use of metacognition as a method for AI systems to perform self-evaluation as a means to identify cyber intrusions, tampering, or any unusual behavior (Johnson, 2020b). NPS is also studying the use of ML to identify malicious spoofing and tampering with the Global Positioning System (GPS; Kennedy, 2020).

The evolution of the threat environment is the third adversarial race for the DoD as it develops AI systems. As the adversarial threat space is constantly changing over time with faster and more lethal weapons, more autonomy, greater surveillance assets, more advanced countermeasures, and more stealth, this poses a challenge for the DoD to be able to anticipate and identify new threats and cope with unknowns in the battlespace. NPS research is focused on engineering systems that continue to adapt and learn during operations to detect and identify unknown unknowns in the battlespace and quickly respond to new threats through innovative courses of action (Grooms, 2019; Jones et al., 2020; Wood, 2019). NPS is studying ML methods for identifying anomalies in patterns of life by studying data over time for a given region to identify unusual changes (Zhao et al., 2016). An example is the study of commercial aircraft flight patterns and identifying suspicious aircraft based on unusual flight patterns. Ground-based operations can be surveilled over time to identify new and unusual construction projects that could signify military operations.

Conclusions

AI systems offer the DoD significant advances in achieving and maintaining knowledge and decision superiority. However, implementing AI systems for defense applications presents unique challenges. The military tactical domain presents extreme complexity in all areas of the decision space: uncertainty and limited knowledge, highly dynamic situations, very limited time lines, complicated human interaction, large numbers and types of resources, multiple missions, costly and hard-to-obtain training data sets, extremely small margins of allowable errors, and life-or-death consequences of actions (or inaction). AI systems, and ML systems in particular, require representative, sufficient, secure, and validated data sets for their development. Gathering suitable data for defense application has the additional challenges of handling classified data sets and ensuring that data is secure and protected from cyberattacks; it will also be a major endeavor to gather real-world data that represents tactical operations. New systems engineering methods will be required to effectively specify, design, and evaluate AI systems that present new levels of complexity through their non-determinism, new types of human–machine teaming challenges, and new safety failure modes that are hard to anticipate and prevent. Finally, the existence of adversaries in the military domain presents an AI race in three forms: a race to develop AI systems as quickly as adversaries, a race to stay ahead of possible cyberattacks, and a race to train AI/ML systems that can cope with the ever-advancing adversarial threat space.

NPS is addressing the four unique challenge areas through a series of ongoing research initiatives. NPS researchers are studying the implementation of AI systems in the naval tactical warfighting domain, conducting needs analysis and requirements development for military data sets, studying systems engineering methods for developing complex AI systems, and developing methods to engineer AI systems that are safe, trustable, and mindful of the role of potential adversaries. NPS is providing AI research and educational opportunities for military officers and civilian students. NPS welcomes collaboration with the DoD and naval organizations to continue studying AI systems for defense applications and to continue exploring solution strategies and methods for overcoming the challenges of developing and implementing AI capabilities.

References

Allen, G. (2020). *Understanding AI technology*. Joint Artificial Intelligence Center, U.S. Department of Defense.

Blickley, W., Carlson, J., Mariana, M., Pacheco, A., & Roscher, J. (2021). *The cognitive laser weapon system: An understanding, architecture design, and integration of an automated decision aid within a laser weapon system*. Naval Postgraduate School.

Calvano, C., & John, P. (2004). Systems engineering in an age of complexity. *Systems Engineering*, *7*(1).

Cruz, L., Hoopes, A., Pappa, R., Shilt, S., & Wuornos, S. (2021). *Evaluation of the safety risks in developing and implementing automated battle management aids for air and missile defense*. Naval Postgraduate School.

French, R., Fukumae, W., Hun, K., Matuga, O., & O'Shaughnessy, C. (2021, March). *Data management for artificial intelligence/machine learning implementation* [Presentation]. Naval Applications for Machine Learning (NAML) Conference.

Grooms, G. (2019). *Artificial intelligence applications for automated battle management aids in future military endeavors* [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. https://calhoun.nps.edu/handle/10945/62722

Hui, P. (2021). *Proposing a strategy to develop a trustworthy artificial intelligence enabled air and missile defense system* [Thesis proposal]. Naval Postgraduate School.

Johnson, B. (2019). *A framework for engineering complex adaptive systems of systems* [Doctoral dissertation, Naval Postgraduate School]. NPS Archive: Calhoun. https://calhoun.nps.edu/handle/10945/63463

Johnson, B. (2020a). Modeling and simulation analysis of a complex adaptive systems of systems approach to naval tactical warfare. In D. Braha et al. (Eds.), *Unifying Themes in Complex Systems X: International Conference on Complex Systems (ICCS)*. Springer.

Johnson, B. (2020b). Predictive analytics in the naval maritime domain. In Y. Zhao (Ed.), *Proceedings of the AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks 2020*, co-located with the Association for the Advancement of Artificial Intelligence 2020 Spring Symposium Series (AAAI-SSS 2020).

Johnson, B. (2021, April). Metacognition for artificial intelligence system safety [Manuscript submitted for publication]. *Proceedings of the 16th International Conference on Systems (ICONS).*

Jones, J., Kress, R., Newmeyer, W., & Rahman, A. (2020). *Automated battle management aids for air and missile defense: Leveraging artificial intelligence* [Systems engineering capstone report, Naval Postgraduate School]. NPS Archive: Calhoun. https://calhoun.nps.edu/handle/10945/66088

Kennedy, R. (2020). *Applying artificial intelligence to identify cyber spoofing attacks against the global positioning system* [Thesis proposal]. Naval Postgraduate School.

Mitchell, M. (2019). *Artificial intelligence: A guide for thinking humans*. Picador.

Robinson, K. (2021, January 31). *Engineering artificial intelligence with systems engineering* [Presentation]. INCOSE International Workshop, Artificial Intelligence-Systems Engineering Primer Workshop.

Rosenberg, B. (2010, April 21). Military seeks to disrupt the enemy's decision cycle. *Defense Systems*. https://defensesystems.com/articles/2010/04/26/cover-story-getting-inside-the-enemy-decision-cycle.aspx#:~:text=Success%20in%20battle%20is%20increasingly,quickly%20than%20your%20adversary%20can.&text=As%20a%20result%2C%20military%20forces,known%20as%20the%20OODA%20loop.

Sanchez, K. (2021). *A study of human interactions with artificial intelligence systems for naval tactical applications* [Unpublished manuscript].

Taub, E. (2021, March 18). Carmakers strive to stay ahead of hackers. *New York Times*.

Wood, S. (2019). *Leveraging artificial intelligence in support of decision superiority: Enabling artificial intelligence, a system of systems approach* (NPS-19-N157A). NPS Archive: Calhoun. https://calhoun.nps.edu/handle/10945/65039

Zhao, Y., Kendall, W., & Johnson, B. (2016). Big data and deep analytics applied to the common tactical air picture (CTAP) and combat identification (CID). In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management* (Vol. 1, pp. 443–449).

Acquiring Artificial Intelligence Systems: Development Challenges, Implementation Risks, and Cost/Benefits Opportunities[1]

Thomas J. Housel, PhD—is a tenured, full professor for the Information Sciences (Systems) Department of the U.S. Naval Postgraduate School (NPS). He received his PhD from the University of Utah in 1980. He specializes in valuing intellectual capital, knowledge management, acquisitions research, telecommunications, information technology, value-based business process reengineering, and knowledge value measurement in profit and nonprofit organizations. He has managed a $5+ million portfolio of field studies, educational initiatives, and industry relationships. His current research focuses on determining the most promising methodologies for improving the acquisitions life cycle and developing a physics-based model to predict revenue growth in knowledge-intensive companies.

Johnathan Mun, PhD—is a research professor at NPS and teaches master's and doctoral courses as well as executive seminars in quantitative risk analysis, decision sciences, real options, simulation, portfolio optimization, and other related concepts. Mun received his PhD in Finance and Economics from Lehigh University. He also has an MBA in business administration, an MS in management science, and a BS in Biology and Physics. He has taught at universities all over the world, is considered a leading world expert on risk analysis/real options analysis with more than 23 patents, and has authored more than 22 books.

Raymond D. Jones—is a Professor of the Practice in Acquisition Management at the Graduate School of Defense Management at NPS. He retired as a Colonel from the U.S. Army in 2012 and his last assignment in the Army was as the Deputy Program Executive Officer for the Joint Tactical Radio System (JTRS). He has a BS in Aerospace Engineering, an MS in Aeronautical Engineering from the Naval Postgraduate School, an MBA from Regis University, and an MA in National Resource Strategy from the Industrial College of the Armed Forces. He is currently an IS PhD candidate at NPS.

Timothy R. Shives, EdD—is a Department of Defense civil servant in the field of Information Technology (IT) management serving at the Defense Manpower Data Center. He is also an officer in the United States Marine Corps and has served more than 15 years in both the active duty and reserve components. He holds a Doctor of Education (EdD) and an Education Specialist (EdS) postgraduate degree, an MS in Information Technology Management and an MBA from NPS, and an MA in National Security and Strategic Studies from the U.S. Naval War College, and is currently an IS PhD student at NPS.

Benjamin Carlton, Major, USMC—is an active-duty Marine Officer currently serving as a project officer for the Global Combat Support Systems-Marine Corps (GCSS-MC) at the Marine Corps Systems Command. Carlton contributed to this research while completing his studies for an MS in Information Technology Management at NPS, which he completed in 2020.

Vladislav Skots, Major, USA—is an active-duty Army officer. Major Skots contributed to this research while completing his studies for an MBA in Information Systems Technology at NPS, which he completed in 2019.

Abstract

The acquisition of artificial intelligence (AI) systems is a relatively new challenge for the U.S. Department of Defense (DoD). Given the potential for high-risk failures of AI system acquisitions, it is critical for the acquisition community to examine new analytical and decision-making approaches to managing the acquisition of these systems in addition to the existing approaches (i.e., Earned Value Management). In addition, many of these systems reside in small start-up or relatively immature system development companies, further clouding the acquisition process due to their unique business processes when compared to the large defense contractors. This can lead

[1] This ARP Conference Paper is an updated and abridged version of the full ARP Report NPS-PM-21-014, which is available at https://dair.nps.edu/handle/123456789/4313

to limited access to data, information, and processes that are required in the standard DoD acquisition approach. The well-known recurring problems in acquiring information technology automation within the DoD will likely be exacerbated in acquiring complex and risky AI systems. Therefore, more robust, agile, and analytically driven acquisition methodologies will be required to help avoid costly disasters in acquiring AI systems. This research provides a set of analytical tools for acquiring organically developed AI systems through a comparison and contrast of the proposed methodologies that will demonstrate when and how each method can be applied to improve the acquisitions life cycle for AI systems.

Introduction

This report seeks to address the emerging field that falls under the umbrella term "Artificial Intelligence" or AI, and the Defense Acquisition System (DAS) that is not designed to develop and procure such state-of-the-art, rapidly evolving AI technologies. Thus, the research examines the challenges of acquiring an AI-based system within the typical acquisitions framework in business and in the Department of Defense (DoD) by conducting an analysis on a set of recommended quantitative tools for use in analyzing the processes in the acquisition of organically developed DoD AI systems. Thus, this research introduces tried and true quantitative methods (EVM, KVA, IRM) for their application to the DoD Acquisition of AI systems. Therefore, this research proposes the re-examination of acquisition methods, strategies, and methodologies based on the category of AI being acquired.

Acquisitions of AI systems is a relatively new challenge for the DoD. Given the high risk of failure for such system acquisitions, it is critical that the acquisition community examines potential new approaches to help manage the AI acquisition life cycle. The well-documented recurring problems in acquiring information technology within the DoD will likely be exacerbated in acquiring these leading edge, complex, and risky systems. The identification, review, and recommendation for the optimal use of new acquisition methodologies, to supplement or replace existing methodologies, should help avoid costly disasters in AI system acquisitions. In addition, the use of these methodologies should also create a more flexible acquisition scheme that allows for incorporation of unanticipated, value added components of future AI systems. Please note that this conference paper is an abridged version of the research conducted by Housel et al. (2020). For a more thorough discussion, see the full report of this research that provides a robust literature review, more detailed explanations, as well as the step-by-step process for each of the three methodologies please see the full report.[2]

AI has been in use in various commercial and governmental domains to address a variety of decision support problems. However, existing DoD acquisition frameworks may not be adequate to address the unique nature of AI systems life-cycle investments. AI systems are qualitatively different than standard automation systems that focus on routine, repeatable tasks. To develop acquisition frameworks for AI systems, it is first necessary to examine how AI systems will be used to support, or supplant, decision-makers. The purpose of this research project is to provide a set of quantitative and analytically robust decision-making methodologies for acquiring AI systems that address the inadequacies of the current standard investment acquisition life-cycle framework.

To better understand the potential contributions of this research, it is important to recognize the recent drive towards using innovation in improving Defense Acquisition outcomes. During the Cold War, the United States and particularly the Department of Defense enjoyed a position of prominence in the realm of military technological development. Therefore, the use of

[2] This ARP Conference Paper is an updated and abridged version of the full ARP Report NPS-PM-21-014 which is available at https://dair.nps.edu/handle/123456789/4313

the common Defense Acquisition System allowed the DoD to develop, test, and field large-scale weapons systems through a slow, labor intensive development process. However, the rapid growth of technological developments has diminished the DoD's technological advantage over many of the United States' near peer global competitors. As a result of the current challenges, the DoD launched several key initiatives such as "Better Buying Power 3.0," which is "aimed at accelerating acquisition reform and incentivizing innovation within government," as well as the Defense Innovation Initiative, and the "Third Offset Strategy" (Voelz, 2016, p. 180).

The underlying premise behind these recent efforts is that the traditional methods of acquisition are less than optimal in achieving the desired outcomes in DoD weapons and business systems. Therefore, the goal of implementing innovation in the DoD acquisition process has been designed to provide the means of achieving these better acquisition outcomes. Some of the key attributes in the recent initiatives include adopting best practices in DoD labs (Sullivan, 2018) and increasing the use of rapid development cycles through prototyping (DiNapoli, 2019). In addition, the DoD also seeks to take advantage of the commercial sector's rapid development of innovative technologies by partnering with the commercial development of dual use technology—technology that has application for both the private and defense sectors (Kendall, 2017; Voelz, 2016). It is this urgent context that provides a push for innovation in the DoD acquisition processes and provided the impetus for the current research study.

The underlying challenge is best understood through a discussion of programmatic risk management requires a foundational understanding of the effect of risk on human behavior. Primarily, this is the effect of risk on the PM and other program leaders. The goal is to gauge how these principal acquisition professionals respond to risk and their aversion to risk with regard to acquisitions decision-making in terms of cost, schedule, and performance. Bhatt et al. (2005) note that a fundamental understanding of risk management addresses the question, "How much risk is acceptable?" (p. 64). As noted by Housel et al. (2019), "A recurring issue at the U.S. Department of Defense (DoD) is that acquisitions of information technology (IT) have been fraught with schedule and cost overruns. The problem is the risk and project management tools the DoD currently use inadequately address the fiscal and temporal overruns (p. 3)." This premise is supported by numerous studies on DoD acquisitions, particularly those that involve complex IT systems. A prime example that illustrates this issue is the multitude of reports on the development of space and satellite systems (Chaplain, 2017, 2019; Ludwigson, 2019). The issue of concern is that PMs and other managers of DoD acquisitions, particularly in the case of advancing cutting-edge technological systems, are increasingly becoming either overly risk tolerant or increasingly risk adverse.

Thus, the motivation to be risk averse promotes a cautious culture among the DoD Acquisition community. As the DoD continues to develop systems that are increasingly complex, the risk tolerance of PMs and other Acquisition leaders is diminished in direct proportion to their inability to meet the program requirements for cost, schedule, and performance. It follows that the result of this is an organizational culture in the DoD Acquisition community that is unwilling to assume risk. In addition, there is tremendous pressure to push unrealistic schedule and cost goals and results in unforced errors in the acquisition program's baseline. As a result of these unintended consequences, programs are most often more expensive, late, and/or not able to perform to the standards specified.

Research and Problem Statement

The current problem at the DoD is that the complexity and speed of decision-making is increasing exponentially with the advent of intelligent systems that support or that actually make decisions in time-critical, high-impact problem spaces. The current process management and control tools that a program manager (PM) might use to support acquisitions do not provide adequate warning of, or provide sufficient information about the root causes of, fiscal budgetary overruns and time schedule delays. This is a problem because PMs are, as a result, unable to respond to issues in a timely manner, delaying the delivery of promised capabilities to the services. Additionally, the money and resources spent in excess of the original budget could be used in other acquisition programs. To better understand the possible causes and solutions to the AI acquisition problem, this study will examine the strengths and weaknesses of three performance and project management methodologies.

These methodologies, Earned Value Management (EVM), Knowledge Value Added (KVA), and Integrated Risk Management (IRM), are used to strategically and tactically plan, monitor in real time, measure, and preemptively forecast the value and progress of AI acquisitions. A review of these recommended project analysis and control methodologies will offer insights into the strengths and weaknesses each approach could offer acquisition professionals within the general phases of the Defense Acquisition System. This research offers potential solutions to improve early warnings of cost and schedule overruns, and value opportunities foregone in the acquisition process. As such, this research focuses on the review of these methodologies and their applications to the acquisition process of AI systems.

Research Objective

The current research examines the challenges of acquiring an AI-based system within the common acquisitions framework in business and in the DoD. The primary objective of the current research is to evaluate three quantitative analysis tools for improving the acquisition of organically developed DoD AI systems. A comparison and contrast of the proposed methodologies will identify when and how each method can be applied to improve the acquisitions life cycle for AI systems.

Research Questions

The questions examined in this research are as follows:

1. How should each proposed methodology be used throughout the AI systems acquisition life cycle?
2. Will the combination of methodologies reduce the risks associated with acquiring an organically developed AI system?
3. When should the methodologies be used in the acquisition life cycle to ensure successful acquisition of AI systems?
4. What are the risks inherent in following a 5000 series acquisition framework when acquiring an organically developed AI system?

Technical Approaches and Outcomes of the Research

This research provides an in-depth review of each of the three methodologies (IRM, KVA, and EVM). While each acquisition project is unique, all must pass a series of common hurdles to succeed. A successful AI acquisition approach requires the support of methodologies that are designed to identify and value system options and forecast the future value of systems while assessing and mitigating investment risks. The dominant methodology for managing DoD acquisitions that exceed $20 million is EVM. The current structure of EVM may be enhanced with the addition of the IRM and KVA methodologies due to the unique needs of an AI system acquisition. This research examines how these three methodologies might be incorporated within an acquisition life-cycle framework assessing the benefits and risks of this potential extension of the standard framework. These methodologies have been used extensively in the past in acquisition research performed for the Acquisition Research Program (ARP) at the Naval Postgraduate School (NPS). This current study will build on the key learnings from this prior research to enhance the acquisition life-cycle framework with a focus on the unique characteristics of AI system acquisitions. The anticipated outcome of this research will be a set of guidelines for how and when to use the three methodologies to improve the potential success of acquisition of AI within the acquisition's life cycle. The history of organic complex information technology (IT) has been characterized by cost and schedule overruns creating havoc for acquisition professionals as well as system designers and future users who expect to receive valuable new capabilities (Housel et al., 2019; Oakley, 2020). The DoD's standard 5000 series acquisition life cycle will provide the context for reviewing the ways the methodologies can be used to enhance the acquisition life-cycle approach in managing the acquisition of AI systems.

Artificial Intelligence (AI)

This next section will briefly discuss Artificial intelligence (AI) in context of its current impact in the Defense Acquisition System through a discussion on how the three Acquisition methodologies (EVM, KVA, and IRM) can be utilized to assess an AI program based on its stage in the Acquisition life cycle. AI does not refer to a specific system. It is a broad nomenclature for a collection of related inorganic computer science methods used to simulate human intelligence. The term AI typically conjures up the general concept of machine learning, which is a type of AI where a computer system is programmed to identify and categorize external real-world stimuli via a "learning" process. The DoD's AI strategy defines AI as "the ability of machines to perform tasks that normally require human intelligence—for example, recognizing patters, learning from experience, drawing conclusions, making predictions, or taking action—whether digitally or as the smart software behind autonomous physical systems" (DoD, 2019). This capability of enhanced automation is of great interest to the DoD as potential future near-peer adversaries such as Russia and China, are investing heavily in this field for military purposes (DoD, 2019).

The Growth of AI Literature from Inception to Industry 4.0

Utilizing the Web of Science comprehensive academic search engine, the researchers found 316,009 scholarly publications, including 188,275 academic journal articles on the topic of AI. The period of these AI publications covers the entire timeline from the inception of term "artificial intelligence" in the early 1960s until present day. Using Van Eck and Waltman's (2020) VOSviewer tool for visualizing scientific landscapes, the researchers created a network map that showed the illustrates how AI ("intelligence" in the diagram) relates to other key topics. A visualization of the major key terms in AI research is depicted below in Figure 1.

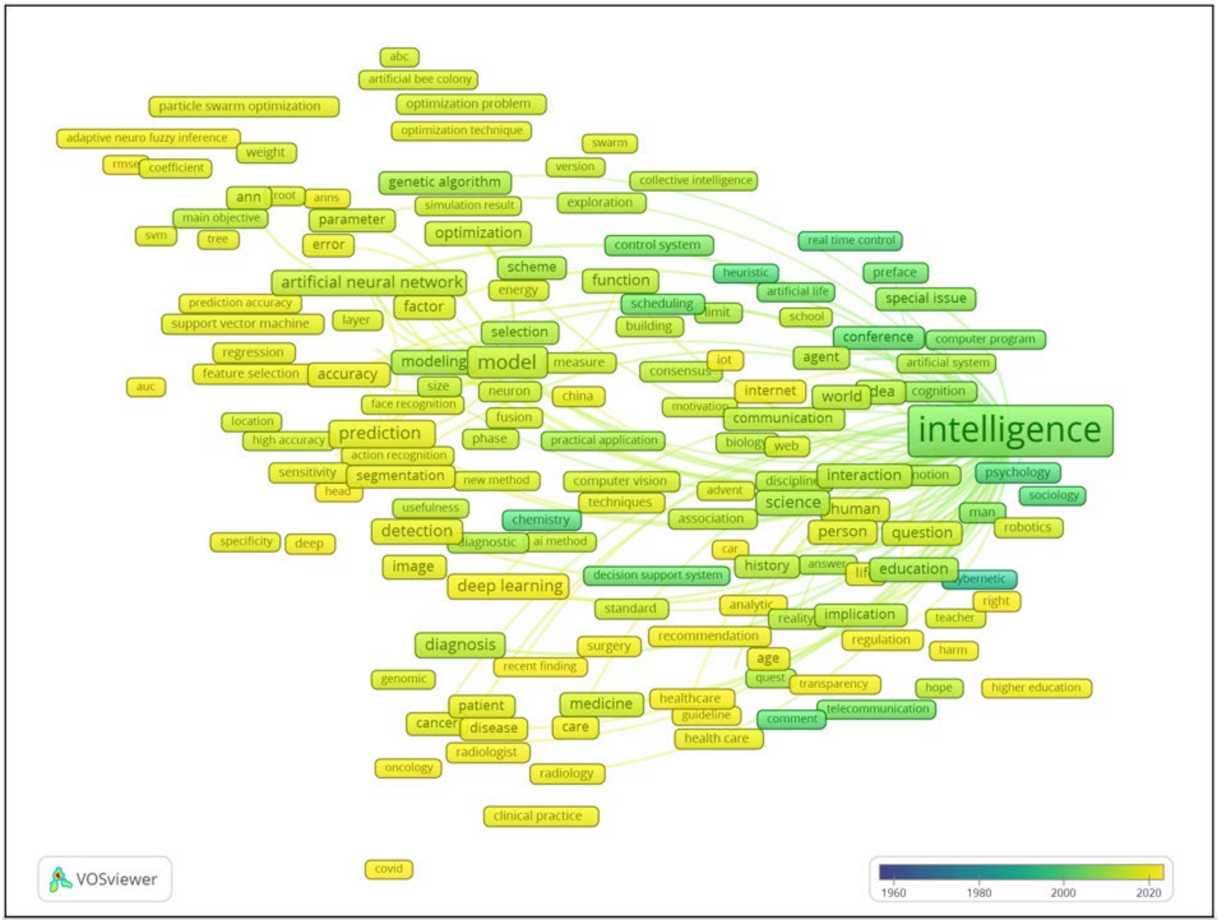


Figure 1. The Key Topics in AI from 1960 to Present Day

Figure 1 depicts how AI relates to numerous terms that have more than 50 publications related to AI since it branched out of cybernetics in the 1960s. As Figure 1 depicts, AI topics have branched out of the general computer science and automation fields into other fields such as healthcare, as noted in medical terms in Figure 1. As the timeline illustrated in Figure 1, much of the expansion in AI topics has occurred in the past two decades. A major cause of the recent interest in AI in the past decade is the phenomenon known as “The Fourth Industrial Revolution” or “Industry 4.0” (Lee, 2013; Schwab, 2015). In noting how steam power impacted the agrarian society in the First Industrial Revolution, the assembly line in the Second Industrial Revolution, the computers in the Third Industrial Revolution, Schwab (2015) argues that society is on the cusp of a “technological revolution that will fundamentally alter the way we live, work, and relate to one another.” Schwab (2017) and his colleagues at the 2011 Hanover Conference on Technology made the case that impact of this new revolution is quite extensive,

> We have yet to grasp fully the speed and breadth of this new revolution. Consider the unlimited possibilities of having billions of people connected by mobile devices, giving rise to unprecedented processing power, storage capabilities, and knowledge access. Or think about the staggering confluence of emerging technology breakthroughs … such as artificial intelligence. (p. 9)

While much of this research focuses on the DoD Acquisition of AI, the tools this research proposes for the project management in developing and procure AI (KVA and IRM) have a broader application.

AI in National Security and Defense Applications

As the capabilities of AI has expanded in the Fourth Industrial Revolution, there has been a growing concern that the international arena, particularly the three Great Powers of the United States, China, and Russia, may already be in the throes of an "AI Arms Race" (Geist, 2016). One of prevalent fears among scholars in the age of expanding AI, is the risk of developing "autonomous weapons" that can no longer be controlled (Geist, 2016). In essence, by developing AI weapon systems, humanity may be sowing the seeds of its own destruction. As former Secretary of State Kissinger noted on the rise of AI:

> The scientific world is impelled to explore the technical possibilities of its achievements, and the technological world is preoccupied with commercial vistas of fabulous scale. The incentive of both these worlds is to push the limits of discoveries rather than to comprehend them. And governance, insofar as it deals with the subject, is more likely to investigate AI's applications for security and intelligence than to explore the transformation of the human condition that it has begun to produce. (Kissinger, 2019)

Kissinger's (2019) words strike at the heart of the growing mistrust of this new technology that has exploded with the recent rise of Big Data and more powerful computing. This fear may also grow as AI can potentially dominate international relations with a new race to develop and weaponize AI (Geist, 2016).

AI and The DoD's Third Offset Strategy

The discussion will now shift towards the DoD's recent drive towards AI. As the potential capabilities of AI became more evident, then Secretary of Defense Chuck Hagel (2014) launched the "Third Offset Strategy." The DoD's Third Offset Strategy hearkens back to the "First Offset" launched by President Eisenhower where the United States developed its strategic arms to reduce the need for standing conventional forces during the early Cold War and the "Second Offset" where after the Vietnam War, the DoD, through programs such as DARPA, increased the capabilities of its conventional forces to counter the Warsaw Pact (Hillner, 2019). Then Deputy Secretary Work noted that it is the challenges of the current geopolitical environment, artificial intelligence can help reinstate the U.S. previous technological overmatch, "Learning machines are an example of technology that can help turn AI and autonomy into an offset advantage" (Pellerin, 2015). It was the Third Offset Strategy that pushed for subsets initiatives such as the Defense Innovation Initiative and Frank Kendall's (2017), then Under Secretary of Defense for Acquisition, Technology and Logistics, "Better Buying Power 3.0" which were focused on bringing back the competitive edge that the U.S. military once had over its geopolitical competitors.

The theme that arises from the Third Offset, Defense Innovation Initiative, and Better Buying Power 3.0 is that the Defense Acquisition System is unable to meet the requirements of fielding software intensive systems such as AI systems. Kendall (2017) discussed this problem by illustrating the challenging that PMs with software intensive systems such as AI face due to the extensive cycle of developing, testing, fielding "several builds of software in various stages of maturity" simultaneously while dealing with the organizational bureaucracy that slows the PM down (p. 50).

DoD Acquisition of New Software-Intensive Technology

The fielding of new and advanced technologies such as AI is a challenge for the DoD and all federal government. The current methodologies have proven unsuccessful in meeting the task of providing the requirements to the warfighter to face the challenges of the modern battlefield (Kendall, 2017). However, with the release of recent strategic changes such as implementing innovation practices and advanced prototyping, the DoD may prove up to the task of fielding the materiel and equipment to support the Department and the Services (Kendall, 2017; Voelz, 2016). Some of the practices that have proven success is the adopting best practices in DoD labs and increasing the use of rapid development cycles through prototyping (DiNapoli, 2019; Sullivan, 2018).

A recent case study on the two-decade process of developing biometrics for the use in the Services provides the lessons learned on comes to acquisition of new and advanced technology. According to Voelz (2016), “The case study of biometrics demonstrates that effective military innovation can only occur through an integrated approach that takes into account the interdependent elements of technology development, acquisition planning, doctrinal design, and warfighting strategy” (p. 180). While not exactly the same, the lessons learned of adapting biometrics in the Services is an example of how the DoD can adopt AI throughout the department. This is because a key attribute that biometric systems and AI systems share are that they are both heavily software intensive development process.

Earned Value Management

EVM provides cost and schedule metrics to track performance in accordance with an acquisition project plan. EVM is required for large DoD acquisition programs that use incentive contracts valued at or greater than $20 million (DoD, 2015a). EVM methodology uses a WBS to try to ensure that an acquisition project is on schedule and within the estimated cost for each work package. It is used to measure work progress and any deficiencies using cost and schedule metrics that also can be used to measure program performance trend analysis with a focus on identifying any budget and schedule deviations from plan. However, the analysis is done after each process or stage in the WBS. In other words, the actual cost and time spent to execute a particular phase is compared against the initially projected budgeted plan.

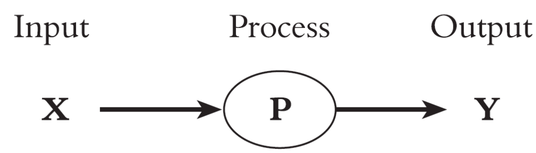
Given the propensity of IT and AI acquisitions to be over budget and behind schedule, EVM metrics help PMs identify and attempt to avoid overruns and schedule deviations. Recognized plan deficiencies can help program managers identify waste and chokepoints that require immediate correction. When deficiencies in cost or schedule occur, EVM analysis can be used to reforecast the budget and schedule with the focus of providing PMs with up-to-date accurate performance information. EVM analysis uses schedule and cost estimates to find the Planned Value (PV) of a given acquisition project. Cumulative PV provides the total value that should be achieved by a specified date. The specific label for PV within the DoD acquisitions community is Budgeted Cost for Work Scheduled (BCWS). Actual Cost (AC) is the accumulated accrued costs of labor and materials at any point in time during a project. The label for AC within the DoD acquisitions community is Actual Cost of Work Performed (ACWP). Earned Value (EV) measures the progress for a given plan. The DoD acquisitions label for EV is Budgeted Cost of Work Performed (BCWP).

In sum, EVM exists to provide an assessment of the actual physical work a project has completed compared to a baseline plan (Fleming & Koppelman, 2010). EVM integrates the actual cost spent on the project to date with the work that has been performed on the project, allowing managers to compare the progress of the project with their planned budget and schedule (Fleming & Koppelman, 2010). It provides managers the ability to compare cost performance with work completion rather than simply cost performance and planned cost, as is done in traditional cost management (Fleming & Koppelman, 2010). When properly employed, EVM provides a reliable prediction of the total cost and schedule requirements for a project through three distinct dimensions: the planned value, earned value, and actual cost (Fleming & Koppelman, 2010). It is important to note the term *value* in EVM does not have the same meaning as in other methodologies, such as KVA. Within the context of EVM, *value* is defined as the work accomplished towards completion of the project. There is no reference to the quality of the completed work or additional (or missing) benefits the work might provide to a system. The value is assumed because the specifications were defined in the project requirements. EVM has proven to be a reliable system to manage cost and schedule performance for manufacturing in both defense and commercial industries. However, as systems become more complicated and information technology (IT) and AI gain a more prominent place within even traditional manufacturing projects, EVM may need additional information from additional methodologies to improve its capabilities. Better incorporating the strategic guidance associated with a program, the value gained from subcomponents and subprocesses, the risk associated with developing subcomponents of a system, and incrementally improving a process may help improve the Defense Acquisition System as a whole.

Knowledge Value Added

KVA is an empirical model that focuses on the practical application and implementation of knowledge management (Tsai, 2014). Originally developed to assist in business process reengineering, KVA creates an objective, quantifiable method to measure the value of a process or service (Housel & Kanevsky, 1995). Typical financial approaches to business process reengineering use the dollar amount of a final product to determine the value of an object, failing to account for the knowledge required in the various subprocesses involved in making the product (Housel & Kanevsky, 1995). In its essence, KVA performs a single function: describing all process outputs in common units. KVA accounts for the value of all components, processes, and support systems necessary to complete a task or create a product or service by describing all outputs in common units. It allows managers to compare the efficiency of the various steps across all processes within a common value reference point.

Value has a different meaning in KVA than it does in other methodologies, such as EVM or IRM. KVA bases its definition of *value* on complexity theory and views organizational processes by their ability to change their input (raw material, information, energy, etc.) into common units of output, as shown in Figure 2 (Housel & Kanevsky, 1995). Per Figure 2, process P changes the input in some manner, creating a different product or service at the output, adding value to the system based on the number of common unit changes from input to output (Housel & Kanevsky, 1995). If process P did not change input X, then output Y is the same as input X, indicating no value was added by the system (Housel & Kanevsky, 1995). While the change from X to Y may be minute or large depending on the process, KVA converts all changes into common units, and these changes indicate the amount of value added by process P to produce the final product. The value generated through the process is proportional to the change in the state from X to Y, denoting the amount of knowledge required to make the changes (Yu et al., 2009). Thus, the contribution to a process is equivalent to the sum of all knowledge necessary to produce a product and/or interpret meaning from an input (Housel & Kanevsky, 2006). This is true for all processes within a system, from production to service to management.



$P(X) = Y$

Fundamental assumptions:

1. If X = Y no value has been added.
2. “value” ∝ “change”
3. “change” can be measured by the amount of knowledge required to make the change.

So “value”∝ “change” ∝“amount of knowledge required to make the change”

Figure 2. Value Added Process (Housel & Bell, 2001, p. 94)

The KVA methodology is best completed by following the seven-step process shown in Figure 3. Practitioners can use several methods to describe the units of change, such as tasks, Haye knowledge points, Shannon bits, units of knowledge, and so on (Housel & Bell, 2001). For ease of measurement, three measures are typically used within KVA to estimate the embedded knowledge within a process (Housel & Bell, 2001). Learning time, column two in Figure 3, measures the length of time it takes an average user to learn a process and correctly complete it (Housel & Bell, 2001). Process description, column three, is the number of process instructions used to transform the given input into the desired output (Housel & Bell, 2001). Each instruction must require an approximately equal amount of knowledge to complete a task (Housel & Bell, 2001). The binary query method uses the number of binary questions (i.e., bits) necessary to accomplish the process, roughly equivalent to the lines of code within a computer program (Housel & Bell, 2001). However, any measure that satisfies the basic concepts of KVA can be used to create a common-units measure (Housel & Bell, 2001).

Steps	Learning time	Process description	Binary query method
1.	Identify core process and its subprocesses.		
2.	Establish common units to measure learning time.	Describe the products in terms of the instructions required to reproduce them and select unit of process description.	Create a set of binary yes/no questions such that all possible outputs are represented as a sequence of yes/no answers.
3.	Calculate learning time to execute each subprocess.	Calculate number of process instructions pertaining to each subprocess.	Calculate length of sequence of yes/no answers for each subprocess.
4.	Designate sampling time period long enough to capture a representative sample of the core process's final product/service output.		
5.	Multiply the learning time for each subprocess by the number of times the subprocess executes during sample period.	Multiply the number of process instructions used to describe each subprocess by the number of times the subprocess executes during sample period.	Multiply the length of the yes/no string for each subprocess by the number of times this subprocess executes during sample period.
6.	Allocate revenue to subprocesses in proportion to the quantities generated by step 5 and calculate costs for each subprocess.		
7.	Calculate ROK, and interpret the results.		

Figure 3. The KVA Approach (Housel & Bell, 2001)

KVA identifies the actual cost and value of an organization's assets (human and technological), standard functional areas, or core processes. KVA identifies every process required to produce an output, and the historical costs of those processes, the unit costs, and unit values of products, processes, functions, or services can be measured. By describing all process and subprocesses (down to the detailed level of WBS) outputs in common units, the methodology also permits market-comparable data to be generated; this ability is particularly important for nonprofits like the military and government organizations. When market comparables from industry are used, value is quantified in two key productivity metrics: Return on Knowledge (ROK) and Return on Investment (ROI). Following these steps yields a defensible estimate of the productivity (i.e., ROK, ROI) of a given process or set of subprocesses. These estimates can then be used to track progress in an EVM framework in terms of cost, schedule, and importantly, the value produced. The KVA estimates can also be used to track the volatility of a set of processes and this metric can be used in the IRM processes that forecast future value from, for example, an AI system.

KVA is potentially useful tool for inclusion in the Defense Acquisition System. Since the DoD is not a for-profit company, it does not have revenue to judge the effectiveness of its programs in a monetized form. Instead, it relies on various metrics and evaluations that are not comparable from system to system. If the DoD implements the KVA methodology more widely, PMs may have a more objective measure to compare various technological solutions to fulfill evolving requirements. Understanding the value that a system or process provides in direct comparison with the value of other systems, whether they are similar or unrelated processes, could provide beneficial information in the decision-making, budgeting, and planning processes.

Integrated Risk Management

IRM is a system developed by Dr. Johnathan Mun designed to provide management the ability to analyze risk associated with the development of a new project or initiative. IRM combines several commonly accepted analytical procedures, such as predictive modeling, Monte Carlo simulation, real options analysis, and portfolio optimization, into a single, comprehensive methodology. The methodology uses existing techniques and metrics such as discounted cash flow, return on investment (ROI), and other metrics within the analytical processes to improve the traditional manner of evaluating potential projects within a company or the DoD. In contrast to the other methodologies, IRM focuses on the risk involved with a decision. It seeks to mitigate negative effects from risk while maximizing rewards from potential outcomes. At its core, IRM is a technique to provide managers the best analytic information available to use during the real options process. Figure 4 illustrates the comprehensive IRM process.

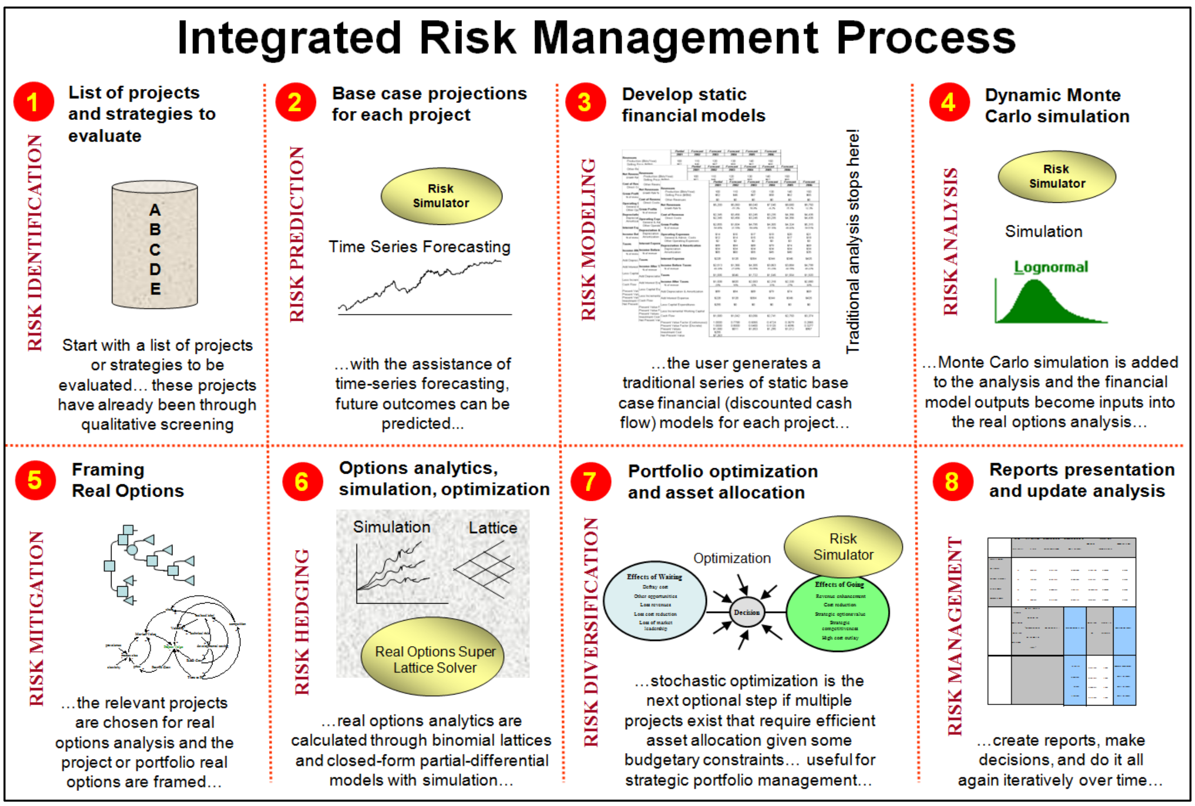


Figure 4. Integrated Risk Management Process (Mun & Housel, 2010)

As depicted in Figure 4, there are eight steps within the IRM methodology:

1. Qualitative management screening
2. Forecast predictive modeling
3. Base case static modeling
4. Monte Carlo risk simulation
5. Real options problem framing
6. Real options valuation and modeling
7. Portfolio and resource optimization
8. Reporting, presenting, and updating analysis

While each of the individual steps provides value to a project manager, incorporating all of them in a contiguous approach will allow decision-makers the most effective use of the IRM process.

IRM is a comprehensive methodology that is a forward-looking risk-based decision support system incorporating various methods such as Monte Carlo Risk Simulation, Stochastic Forecasting, Portfolio Optimization, Strategic Flexibility Options, and Economic Business Case Modeling. Economic business cases using standard financial cash flows and cost estimates, as well as non-economic variables such as expected military value, strategic value, and other domain-specific subject matter expert (SME) metrics (e.g., Innovation Index, Conversion Capability, Ability to Meet Future Threats, Force Structure, Modernization and Technical Sophistication, Combat Readiness, Sustainability, Future Readiness to Meet Threats) can be incorporated (Mun, 2016). These metrics can provide robust forecasts as well as mitigating risk via simulations that account for program uncertainties. The tools set also uses modeling to determine potential program benefits compared to program costs (e.g., return on investment for innovation or return on sustainability). Capital investment and acquisition decisions within AI program investment portfolios can then be made based on the resulting rigorous quantitative analysis (considering budgetary, manpower, and schedule constraints). Projects can be broken down into their detailed work breakout structure (WBS) and tasks, where these tasks can be combined in complex systems dynamic structures or implementation paths. The cost and schedule elements for each task can be modeled and risk-simulated within the system to estimate the resulting total cost and schedule risk of a given AI acquisition program. Portfolio management is often integrated with IRM methods to provide a more holistic view in terms of acquisitions of IT and AI acquisition programs.

The IRM methodology is a systematic technique to determine the best possible projects to pursue based on the statistical likelihood of their success. Using historical knowledge of defense acquisition programs and AI systems in both the government and commercial realms could improve the budgeting and scheduling processes. Determining the likely range of outcomes through dynamic statistical modeling may improve the program's performance. By better understanding the risk associated with various components, a more appropriate schedule and budget could be developed. IRM may also help determine which real options should be included in acquisition contracts. A high-risk program may need more options, such as the options to abandon, delay, or expand, based on its actual performance. Finally, IRM could prove useful in portfolio management, helping decision-makers determine which programs to initiate when viewing the portfolio of other programs in progress and used operationally.

Acquisition Life Cycle & AI

Housel et al. (2019) noted that the DoD 5000 series Acquisition Life Cycle (see Figure 5) can be aligned to the generic technology investment life cycle. As depicted in Figure 6, while terminology differs between the DoD 5000 and generic technology life cycle phases, the sequence of activities in these respective life cycles are congruent.

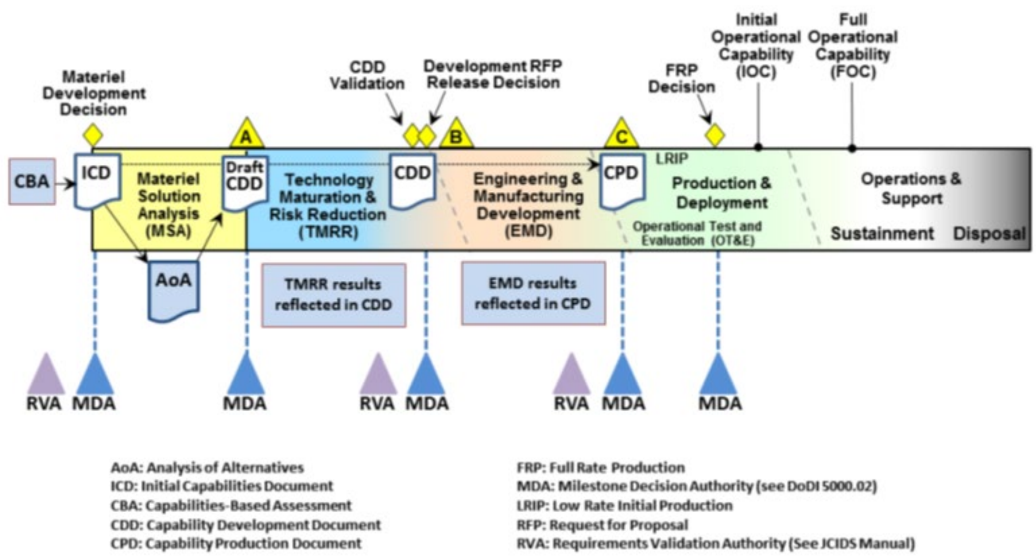


Figure 5. The 5000 Series Acquisition Life Cycle (DoD, 2017)

Pre-Materiel Solutions Analysis	Materiel Solutions Analysis	Technology Maturation and Risk Reduction	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
-Strategic goal alignment -Pre-investment	Pre-Investment	Pre-investment	Implementation	Implementation	Post-implementation

Figure 6. Aligning the Generic and 5000 Series Life Cycles (Housel et al., 2019)

Housel et al. (2019) noted that the acquisition methodologies, to include EVM, KVA, IRM, and others may be used "concert," however, "certain tools are more appropriate for a particular phase than others" (p. 48). Thus, the PM and should "use the tools appropriately in that they provide more information for a complex environment" (Housel et al., 2019, p. 48). Therefore, during the beginning of the life cycle such as the Pre-Materiel Solution Analysis, the Materiel Solution Analysis (MSA), or the Technology Maturation and Risk Reduction (TMRR) phases, methodologies that such as IRM and KVA may be of more use to the PM as they provide quantitative value metrics such as a common unit assessment of the technology and risk, as opposed to EVM that only measures cost. Meanwhile, during the main implementation phases of Engineering and Manufacturing Development (EMD) and Production and Development (PD), all three metrics—EVM, KVA, and IRM—can provide useful data to assess the program. However, during the post-implementation or Operations and Support (OS) phase, KVA is more likely to provide useful data over EVM.

Comparison of Acquisition Methodologies in AI Development

In a recent Congressional Research Service (CRS) on AI it was noted that "standing DoD processes—including those related to standards of safety and performance, acquisitions, and intellectual property and data rights—present another challenge to the integration of military AI" (Sayler, 2020, p. 16). As discussed earlier in the report, the difficulty of traditional Acquisition methods in relation to emerging technology such as AI is the speed the development and deployment of these technologies often outpaces the Defense Acquisition System. When compared to the commercial sector, the DoD process for developing and fielding its AI systems has a mismatch between when compared to the timelines for its requirements. This is likely because there is a stark contrast with "the pace of commercial innovation and the DoD's acquisition process" (Ilachinski, 2017, p. 1). This is because it takes an average of 91 months, or 7 and a half years, to go from the Analysis of Alternatives (AoA) to the Initial Operational Capability (IOC; Ilachinski, 2017). The Defense Science Board (2018) found the DoD's timeline for fielding systems is multiple times longer than the commercial sector which uses an iterative process to field AI systems in approximately 6 months. As noted in the RAND Report that assessed the DoD's posture for AI:

> Our starting point at the onset of our study was the DoD model of technology development, procurement, fielding, and sustainment, giving rise to two dimensions of posture assessment related to technologies: advancement and adoption. However, as we carried out our study, it became clear that this model is not valid for AI, owing to the spiral nature of AI technology development. (Tarraf et al., 2019, p. 51)

Because of the challenges of procuring and fielding AI systems, the CRS Report reached the conclusion that the "DoD may need to continue to adjust its acquisitions process to account for rapidly evolving technologies such as AI (Sayler, 2020, p. 17). Similarly, the RAND Report also noted that the DoD should utilize and adapt acquisition approaches that are "appropriate for the technology" (Tarraf et al., 2019).

One method that the DoD can adjust or adapt its acquisition process and/or approach is to adapt and utilize the Acquisition methodologies based on the complexity of the development. As noted by Ilachinski (2017) and the Defense Science Board (2018), AI systems development processes tend to be iterative in their approach. Thus, the need for rapid development and prototyping is likely to be utilized more than in traditional Acquisition Life Cycles (Tarraf et al., 2019). However, despite being iterative in nature, the techniques recommend by Housel et al. (2019) of using EVM, KVA, IRM, or a combination of using various methodologies in concert can also be applied to the Acquisition of AI systems as shown in Table 1.

Table 1. Selection of Acquisition Methodologies Based on Complexity of Development

Complexity of Development	EVM	KVA	IRM	
Mature Technologies	X	X		
Iterative Development		X	X	
Complex AI Systems		X	X	X
Non-Complex AI Systems	X	X		

Note. Iterative Development is for AI systems that are either prototypes or have rapid generational development cycles (i.e., versions).

For the mature technologies and/or non-complex AI systems, because there is typically less risk in these life cycles, the PM can follow the normal of the EVM methodologies to monitor the progress of the development. However, the PM can also choose to utilize KVA to assess the AI systems with its common unit of value.

Comparison of Key Attributes

Choosing a methodology should depend on the nature of the project under consideration, specifically, the commitment needed from the organization, the organization's desire to align strategic goals with the project, the predictive capability of the methodology, the flexibility required, and the time available. While others in the organization need to understand the concepts to comprehend status reports, EVM only needs the management team to track the cost and schedule of the project compared to the baseline as there is no determined goal alignment with the organization. While the CPI and SPI can help estimate the final cost and schedule, there is no true predictive ability associated with EVM since the assumption is that the schedule will proceed according to the baseline, regardless of previous performance volatility. Adherence to the baseline is essential in EVM, and changing requirements can drastically alter a baseline, reducing the effectiveness of the methodology. Setting up, monitoring, and reporting the cost/schedule performance of each work package within the WBS can be a time-consuming and expensive task for an AI project with its many unknowable components and capabilities a priori.

KVA needs only the KVA analyst and the process owner, as the SME, to determine the value of a process or component output, supporting the need to align the project with an organization's productivity goals. Using this analysis, they can model the current baseline as-is process ROK and compare it with the proposed to-be process model ROK, thus offering a simple prediction of the improvement between the models. Since KVA can be used with any language of description that defines the process outputs in common units, analysts can choose whichever method is most beneficial for the particular system in question, providing flexibility. This analysis can be completed quickly, potentially providing a rough-cut assessment within a few days.

IRM requires the organizational leadership, portfolio and project managers, and the analyst to determine how a project fits within an organization's portfolio, the present value (PV) of the project, and potential real options. By analyzing and simulating various scenarios, IRM provides a prediction of a project's likely performance, which allows managers to build in flexibility via real options at the appropriate locations within the project. Assuming the data necessary for the analysis is available, the process can be completed in a relatively quick manner.

Methodologies in AI Acquisition

As previously discussed, the methodologies all have strengths and weaknesses, making them more suitable in certain applications than others. The biggest challenge in using EVM when acquiring AI is the iterative nature of software development. EVM needs clearly stated, detailed requirements for intermediate steps to be most effective. While the outputs of software programs are defined well, the steps required to build the software are not, leading to problems when developing cost and schedule estimates. If the software is not complex or consists of known processes, EVM can sufficiently monitor the progress. Integrating software and hardware is also complicated with EVM since there are numerous pieces of the program that must be combined to meet the goals, resulting in additional debugging and recoding. EVM is more efficient when used to manage the physical creation of systems or infrastructure. It can monitor the cost/schedule progress of software work packages but is not as useful at estimating the value of those programs.

KVA can provide an objective, ratio-scale measure of value and cost for each core process and its subprocesses or components within any IS system. Using the two parameters, managers can then analyze productivity ratios information, such as ROK and ROI, to determine the efficiency of a process compared to the resources used to achieve the output. This can help the manager decide how to use resources to update systems or estimate the future value of a system being acquired. Combining the KVA results with IRM allows managers to iterate the value of system real options analysis through simulation and other techniques. IRM can also quantify risks and forecast performance probabilities for measures of the potential success for programs and components of programs using historical data. It is a tool to assist with the investment strategy, making it useful when acquiring all types of AI. However, it is not designed to help manage the actual acquisition of an AI program or determine how to meet its detailed requirements.

Examining the benefits and challenges of the proposed methodologies demonstrates the scope, capabilities, and limitations of various AI systems. It also helps inform in which areas and phases of the Defense Acquisition System life cycle it may be appropriate to include the methodologies or components of the methodologies within the system. The main research question of this study was, simply, how can certain advanced analytical decision-making methodologies be used in the acquisition life cycle to complement existing methods to ensure a successful acquisition of AI technologies?

As discussed, EVM remains the only program management methodology required by the U.S. government for all DoD acquisition programs with a contract value exceeding $20 million. Regardless of this requirement, EVM is a methodology that provides a structured approach to the acquisition of IT via program management processes that can help ensure an acquisition program stays on schedule and within budgeted cost estimates. However, there are significant limitations when using EVM for AI acquisitions, the major weakness being that it was not designed for managing AI acquisitions that follow a very iterative and highly volatile pathway. Organic AI acquisitions require a high level of flexibility to deal with the unknowns that arise during the development process as well as value adding possibilities not in the original plan. In addition, EVM does not provide a common unit of value metric to enable standard productivity metrics, such as ROI. When value is inferred by how consistent a program is with original baseline cost and schedule estimates, the performance of the program may be sacrificed in terms of the quality of the outputs when planned program activities become iterative, as in the development of many AI programs. If an AI acquisition program is trending toward cost and schedule overruns, but the resulting value added of the modifications to the original requirements provides disproportionate increases in value, EVM is not designed to recognize this increase in value.

To remedy these shortcomings of EVM in AI acquisitions, the methodology should be combined with KVA and IRM, which can be useful during the requirements and monitoring phases of EVM by ensuring that a given AI acquisition is aligned with organizational strategy and that a baseline process model has been developed for establishing current performance before acquisition of an AI system. A future process model that estimates the value added of the incorporation of the AI can also set expectations that can be measured against the baseline model after the AI has been acquired. IRM can be used to forecast the value of strategic real options flexibility that an acquired AI may provide so that leadership can select the options that best fit their desired goals for the AI in defense core processes.

Because it provides an objective, quantifiable measure of value in common units, KVA should also be used in AI acquisitions to allow decision-makers to better understand and compare different strategic options based on their value and the cost. Obtaining a return on investment of AI systems can only be done when using KVA to determine the value embedded in the system. This information provides insight to PMs as well as a more complete perspective regarding the performance of both the current and the to-be systems.

Likewise, using IRM is recommended when acquiring AI through the Defense Acquisition System. Applying dynamic and stochastic uncertainty and risk-based modeling techniques to predict likely and probabilistic outcomes can improve the risk estimates associated with the components and subcomponents of a program, in terms of their potential cost overruns, value variabilities, and schedule delays. Analyzing various real options within the context of the models' outputs will help PMs make the most advantageous choices when determining a program's future.

PMs should use EVM only in the Engineering and Manufacturing Development (EMD) phase, as is currently done. That said, EVM will work best in hardware manufacturing solutions with technology that is fully mature prior to the program starting. Since many AI acquisition programs consist of advancing the current technology and developing new software solutions to meet requirements, EVM is not perfectly suited for AI development. Nevertheless, PMs can use various agile EVM techniques to complete projects on cost/schedule baselines provided the appropriate steps are taken when establishing the baseline. Requirements must be broken into small, easily definable tasks with suitable risk and uncertainty factors accounted for within the schedule. Other methodologies, such as KVA and IRM, should be used with EVM to ensure these factors are based on defendable metrics rather than simply guessing how much additional time, money, and value may be necessary to complete complex tasks.

During the Materiel Solution Analysis (MSA) phase, KVA and IRM will help determine the value of the different options considered in the analysis of alternative (AoA) process. KVA can objectively measure the value of the current, as-is system and the potential to-be systems under consideration. Then IRM can use additional factors to value the alternatives in terms of their relative parameter values such as cost, value, complexity, timeline. As the chosen solutions mature during the Technology Maturation and Risk Reduction (TMRR) phase, an updated KVA analysis will reassess initial estimates and provide a projected ROI that can be incorporated in an IRM risk and real options analysis for the AI solution prior to entering the EMD phase as appropriate.

Limitations and Future Research

This research examined whether the various methodologies—EVM, KVA, and IRM—could be used within the Defense Acquisition System to improve the acquisition of AI. Future research should examine how these methodologies may interact with or improve other components of the acquisition system. This includes the Joint Capabilities Integration and Development System (JCIDS) and Planning, Programming, Budgeting, and Execution (PPBE) components as individual processes and the interaction of JCIDS, PPBE, and the Defense Acquisition System as a whole. Certain methodologies, specifically IRM, may be more beneficial when used throughout the entire acquisition process instead of within a portion of the system. Additionally, future research could examine how these different methods may be used in the acquisition of products outside the AI or IT realm.

The research conducted looked at AI as a whole and not specific types of AI. Future studies should examine if acquisition methods, strategies, and methodologies should change based on the category of AI being acquired. This is of specific interest when considering artificial intelligence and its subsets. Machine learning, intelligence with a specific focus or field of expertise, and general or universal intelligence would likely have different methods used in the acquisition process based on their complexity, complicated nature, undeveloped technology, and level of risk.

The applicability of these methodologies within commercial acquisition of AI is another area of potential research. This research focused exclusively on the application of the respective techniques within the DoD acquisition process. However, commercial entities also struggle when acquiring complex or complicated AI and IT systems, particularly when the systems operate at the enterprise level. Further research may indicate if these same methodologies could provide value to decision-makers in the private sector during the creation, adoption, or customization of commercial AI. As noted earlier, the hype cycle for AI and automation is on the rise and the demand to procure such technologies is as relevant for the commercial sector as it is for the DoD. Furthermore, the recent pandemic caused by the Coronavirus Disease 2019 (COVID-19) has forced a permanent shift in society towards an increased trend towards a permanent remote workforce. As these trends are likely to continue in the foreseeable future, an increased in automation tools will be required to support this workforce. These trends could be explored for their implications as part of the Fourth Industrial Revolution and Industry 4.0.

Finally, this research only examined the most promising methodologies out of numerous different possibilities. Future research could examine other program management tools, management philosophies, analytic tools, or other methodologies and their benefit when acquiring AI. While the examined methodologies were chosen because they would likely benefit the process and support improvements in EVM, other systems may be more appropriate in certain phases or may offer additional benefits not seen in this research.

References

Bhatt, U. S., Newman, D. E., Carreras, B. A., & Dobson, I. (2005). Understanding the effect of risk aversion on risk [Conference Paper]. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 64b+. https://doi.org/10.1109/HICSS.2005.649

Chaplain, C. (2017). *Space acquisitions: DOD continues to face challenges of delayed delivery of critical space capabilities and fragmented leadership* (GAO-17-619T). GAO.

Chaplain, C. (2019*). DOD faces significant challenges as it seeks to address threats and accelerate space programs* (GAO-19-482T). GAO.

Defense Science Board. (2018, February). *Design and acquisition of software for defense systems*. DoD. https://apps.dtic.mil/dtic/tr/fulltext/u2/1048883.pdf

DoD. (2015a, January 7). *Operation of the defense acquisition system* (change 4) (DoD Instruction 5000.02). https://www.esd.whs.mil/Directives/issuances/DoDi/

DoD. (2017, February 26). *Defense acquisition guidebook*. https://www.dau.mil/tools/dag

DoD. (2019, February 12). *Summary of the 2018 Department of Defense artificial intelligence strategy: Harnessing AI to advance our security and prosperity.* https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF

DiNapoli, T. J. (2019) *DOD's use of other transactions for prototype projects has increased* (GAO-20-84). GAO.

Fleming, Q. W., & Koppelman, J. M. (2010). *Earned value project management* (4th ed.). Project Management Institute, Inc. https://app.knovel.com/hotlink/toc/id:kpEVPME001/earned-value-project/earned-value-project

Geist, E. (2016). It's already too late to stop the AI arms race—we must manage it instead. *Bulletin of the Atomic Scientists*, *72*(5), 318–21.

Hillner, E. P. (2019, May). *The third offset strategy and the Army modernization priorities*. Center for Army Lessons Learned Director's Action Group Report. https://usacac.army.mil/sites/default/files/publications/17855.pdf

Housel, T., & Bell, A. H. (2001). *Measuring and managing knowledge*. McGraw Hill.

Housel, T., & Kanevsky, V. A. (1995). Reengineering business processes: A complexity theory approach to value added. *Infor, 33*(4), 248. http://libproxy.nps.edu/login?url=https://search-proquest-com.libproxy.nps.edu/docview/228436959?accountid=12702

Housel, T. J., Mun J., Jones, R., & Carlton, B. (2019). *A comparative analysis of advanced methodologies to improve the acquisition of information technology in the Department of Defense for optimal risk mitigation and decision support systems to avoid cost and schedule overruns (*Acquisition Research Program Report No. NPS-AM-20-002). Naval Postgraduate School. https://dair.nps.edu/handle/123456789/2774

Housel, T., Mun, J., Jones, R., Shives, T., Carlton, B., & Skots, V. (2020). *Acquiring artificial intelligence systems: Development challenges, implementation risks, and cost/benefits opportunities* (Acquisition Research Report NPS-PM-21-014). https://dair.nps.edu/handle/123456789/4313

Ilachinski, A. (2017, October). *AI, robots, and swarms: Issues, questions, and recommended studies*. Center for Naval Analyses. https://apps.dtic.mil/dtic/tr/fulltext/u2/1041749.pdf

Kendall, F. (2017). *Getting defense acquisition right*. DAU Press. https://DoD.defense.gov/Portals/1/Documents/pubs/Getting-Acquisition-Right-Jan2017.pdf

Kissinger, H. A. (2019, August 30). How the Enlightenment ends. *The Atlantic.* https://www.theatlantic.com/magazine/archive/2018/06/henry-kissinger-ai-could-mean-the-end-of-human-history/559124/

Lee, J. (2013). Industry 4.0 in big data Environment. *German Harting Magazine, Technology Newsletter*, 26.

Ludwigson, J. (2019). *DOD space acquisitions: Including users early and often in software development could benefit programs* (GAO-19-136). GAO.

Mun, J. (2016). *Real options analysis* (3rd ed.). Thomson-Shore and ROV Press.

Mun, J., & Housel, T. (2010). *A primer on applying Monte Carlo simulation, real options analysis, knowledge value added, forecasting, and portfolio optimization.* Calhoun.

Oakley, S. S. (2020). *Defense acquisitions annual assessment: Drive to deliver capabilities faster increases importance of program knowledge and consistent data for oversight* (GAO-20-439). GAO.

Pellerin, C. (2015, November 8). *Work: Human-machine teaming represents defense technology future*. https://www.defense.gov/Explore/News/Article/Article/628154/workhu/

Sayler, K. M. (2020). *Artificial intelligence and national security* (CRS Report No. R45178). https://fas.org/sgp/crs/natsec/R45178.pdf

Schwab, K. (2015, December 12). The fourth industrial revolution: What it means and how to respond. *Foreign Affairs*. https://www.foreignaffairs.com/articles/2015-12-12/fourth-industrial-revolution

Schwab, K. (2017). *The fourth industrial revolution*. Crown Business.

Sullivan, M. J. (2018) *Defense science and technology: Actions needed to enhance use of laboratory initiated research authority* (GAO-19-64). GAO.

Tarraf, D. C., Shelton, W., Parker, E., Alkire, B., Gehlhaus, D., Grana, J., Levedahl, A., Leveille, J., Mondschein, J., Ryseff, J., Wyne, A., Elinoff, D., Geist, E., Harris, B. N., Hui, E., Kenney, C., Newberry, S., Sachs, C., Schirmer, P., … Warren, K. (2019). *The Department of Defense posture for artificial intelligence: Assessment and recommendations*. RAND Corporation. https://www.rand.org/pubs/research_reports/RR4229.html

Voelz, G. (2016) Catalysts of military innovation: A case study of defense biometrics. *Defense Acquisition Research*, *23*(2), 178–201.

Leverage Artificial Intelligence to Learn, Optimize, and Wargame (LAILOW) for Navy Ships

Ying Zhao—is a research professor at the Naval Postgraduate School (NPS). Her research focused on data sciences, machine learning, and artificial intelligence methods for defense military applications such as semantic and social networks, common tactical air pictures, combat identification, logistics, and mission planning. Since joining the NPS, Zhao has been a principal investigator (PI) on many projects awarded for DoD research projects. She received her PhD in mathematics from MIT and is the co-founder of Quantum Intelligence, Inc. [yzhao@nps.edu]

Maj. Gabe Mata—is with Marine Operations Analysis Program Office, USMC, responsible for various PMW programs including Tactical Logistics Systems, Logistics Integrated Information Solutions-MC, and their modernizations. Mata is also a long-distance student at the NPS, working on his master's degree on data sciences and operation research. [gabino.mata@usmc.mil]

Erik Hemberg—is a research scientist in the AnyScale Learning For All (ALFA) group at Massachusetts Institute of Technology Computer Science and Artificial Intelligence Lab (MIT-CSAIL). He has a PhD in computer science from University College Dublin, Ireland, and undergraduate degree from Chalmers University of Technology, Sweden. His work focuses on ALFA methods to cyber security on novel ways to cyber hunt, make defenses more robust, understand what makes attacks effective, and model attack-defense coevolution. He also works in domains such as program synthesis, tax compliance, online learning, and neuronal development. [hembergerik@csail.mit.edu]

Una-May O'Reilly—is a Principal Research Scientist at the MIT-CSAIL. O'Reilly is the leader of ALFA Group at MIT-CSAIL, which conducts wide range of research applying ALFA methods, especially genetic and evolutionary computation to cyber security, software analysis, MOOC technology, and medical technology. She has served as Vice-Chair of the ACM Special Interest Group on Genetic and Evolutionary Computation (SIG-EVO). O'Reilly is a Fellow of International Society for Genetic and Evolutionary Computation (ISGEC). [unamay@csail.mit.edu]

Nate Derbinsky—is an Associate Teaching Professor, as well as the Assistant Dean for Teaching Faculty, in the Khoury College of Computer Sciences at Northeastern University. His research focuses on artificial intelligence, drawing inspiration and techniques from cognitive science, database systems, and optimization. Derbinsky earned his PhD in computer science and engineering from the University of Michigan and was a Postdoctoral Associate at Disney Research. [n.derbinsky@northeastern.edu]

Bruce Cormany—is the Integrated Product Support/Logistics Lead, Fleet Readiness Directorate (FRD) at the NAVWARSYSCOM. [bruce.cormany@navy.mil]

Joy Allen—is the C4I Systems RM&A Branch Head, Acquisition Readiness Department at the U.S. Navy NSWC Corona. [joy.b.allen@navy.mil]

LCDR Andrew Haley, USN—is a U.S. Navy Supply Officer, currently serving at Naval Supply Systems Command (NAVSUP). He has served aboard USS *Springfield* (SSN 761) and USS *Ronald Reagan* (CVN 76) and has experienced the full range of afloat operations, from deployments to shipyard availabilities. He received his Master of Science degree in Operations Research from the Naval Postgraduate School in Monterey, CA. The views expressed in this paper do not necessarily represent the views of the Navy or the United States. [andrew.r.haley1@navy.mil]

LCDR Adam Hilliard, USN—is a U.S. Navy Supply Officer, currently serving in the Office of the Chief of Naval Operations (N412). He has served multiple tours onboard ships undergoing shipyard maintenance. He received his Master of Science degree in Operations Research from the Naval Postgraduate School in Monterey, CA. The views expressed in this paper do not necessarily represent the views of the Navy or the United States. [adam.hilliard@navy.mil]

Abstract

Navy ships are complex enterprises comprised of multiple organizations that must interact smoothly and interface externally without threats to efficiency and combat-readiness. As logistical challenges increase and technology pushes response times, it is critical to introduce state of the art computational methods for analyzing the interlocked systems and training for different events. To address these challenges in this context, we introduce a framework called LAILOW: learn, optimize, and wargame. LAILOW exploits data arising from multiple sources in a complex enterprise by offering data mining, machine learning, and predictive algorithms that can be used for analysis and discovery of patterns, rules, and anomalies. LAILOW's output can then be used to optimize business processes and course of actions.

We show three use cases of using the of LAILOW framework. We show the whole LAILOW framework to search for vulnerability of a major Marine equipment's maintenance and supply system for difficult tests and evolve resilience and novel solutions accordingly. We show using of lexical link analysis (LLA) as part of LAILOW to improve the prediction accuracy of probability of failure of critical Navy Ship parts, related to C4I systems, for NAVWARSYSCOM's Predictive Risk Sparing Matrix (PRiSM) product. We also show the comparison of LLA prioritizing items in the Financially Restricted Work Que (FRWQ) with the baseline calculation.

Introduction

Leveraging deep analysis such as LAILOW for the U.S. Navy Ships is motivated by current challenges and needs. The Navy Ships conduct their activities based on the concepts of operations (CONOPS). For example, Distributed Maritime Operation (DMO) is a CONOP for the Navy; Expeditionary Advanced Base Operation (EABO) is a CONOPS for the U.S. Marine Corps (USMC). These CONOPS require capabilities, manpower, maintenance, and supply among other resources to be carefully analyzed, planned, and executed to complete missions successfully. Meanwhile, military sensors have been constantly collecting big data in many readiness components to facilitate decision-making and improve courses of actions. All the activities require data analytics. Deep analytics for big data and business intelligence including machine learning (ML), and artificial intelligence (AI) algorithms such as deep learning algorithms (LeCun et al., 2015), and game theory (Brown & Sandholm, 2017; Silver et al., 2017) have performed benchmark tasks and demonstrated the superb performance to human. It is imperative to adopt these analytics and tools to understand the entire spectrum of the Navy Ships related to complex enterprises including capabilities, manpower, maintenance, supply, transportation, health services, general engineering, and finance. The paper addresses the needs on the Navy and Marine logistics value chain, where there is a need to consider uncertainty, disruption, and perturbation that can impact the logistics plans as a whole. For example, uncertainty factors related to the environment in wide geographic areas, such as weather change, mission change from a peace time to a conflict time, or a sudden event can cause a perturbation and disruption for previous logistics and supply plans. Previously high-impact but low-failure parts may suddenly become in high demand.

The LAILOW framework focuses on three pillars of deep analytics, that is, machine learning, optimization, and wargame as shown in Figure 1. When there are data from various sources, data mining, machine learning, and predictive algorithms are often used to analyze data and discover patterns, rules, and anomalies that can later be used to optimize business processes and course of actions. New requirements have emerged in recent years that emphasize greater complexity in uncertainty, unknowns, and unexpected situations for Navy Ships. More importantly is the risk of adversarial novelty: adversaries might work on the scenarios and situations that Navy Ships might never encounter before or there are no data are available for decision-making. These requirements and concerns motivate wargame simulations that could generate synthetic data, perform what-if analyses, and explore analyses of alternatives (AoAs). The ultimate goal is to enhance total force readiness and project combat power across the whole range of military operations and spectrum of conflict at any time.

LAILOW Framework

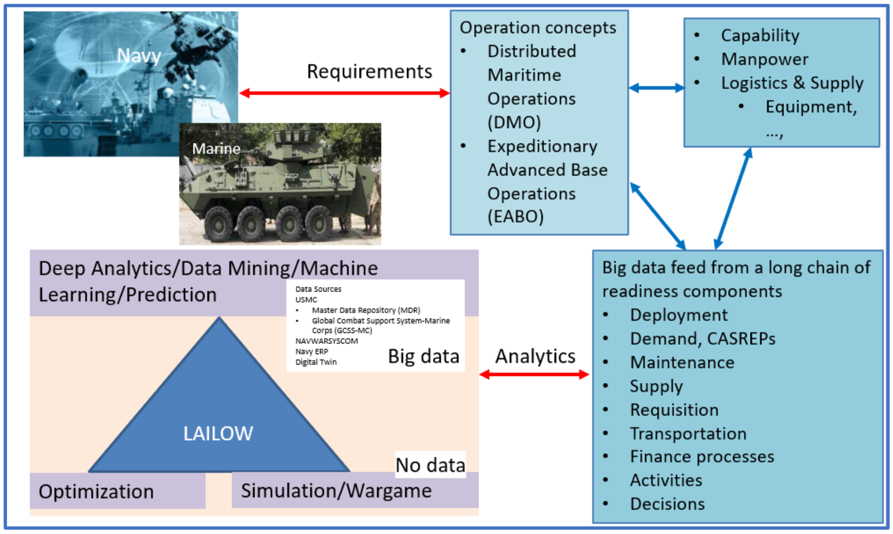


Figure 1. The LAILOW Framework Applied to U.S. Ships

The technical concept of this paper is to leverage artificial Intelligence (AI) to learn, optimize, and wargame (LAILOW) for a complex enterprise, customized to Navy Ships, especially to the logistics value chain and readiness of Navy and Marine enterprises as shown in Figure 1. The components of LAILOW are as follows:

- Component 1—Learn: When there is certain amount of data available, LAILOW performs data mining, machine learning, and learns predictive, associative, and sequential patterns from historical data.

The motivation for Component 1 is the great value of prediction. Once specific way prediction helps is to guide anticipatory preparations. For example, predictive maintenance reduces downtime and indicates which spare parts should be proactively prepared. Predictions of maintenance-related parameters (e.g., MTBF—mean time between failures, probability of failure, probability of demand) enable forecasts of parts' lifetimes under potential circumstances and scenarios. Predications of customer waiting time—the days between a trouble ticket opened and closed—reveal potential bottlenecks or ensure accurate customer expectations. This general ML capability indirectly helps extend asset product life and reduces total ownership costs.

In our use cases, we apply open source ML and data mining toolkit (i.e., Orange software by University of Ljubljana, 1996–2021) to data sets. Orange features a visual programming front-end for explorative rapid qualitative data analysis and interactive data visualization. It contains supervised ML algorithms of logistic regression, decision trees, naïve Bayes, random forest, k-nearest neighbors, and neural networks; and unsupervised ML algorithms such as principal component analysis (PCA) and k-means algorithms among others. Orange algorithms are wrapped from the python machine learning library scikit-learn (Pedregosa et al., 2011).

We also apply Soar (Laird, 2012; Laird et al., 2012), a cognitive architecture that scalably integrates a rule-based AI system and reinforcement learning (RL; Sutton & Barto, 1998, 2014). Soar-RL has advantages for defense applications over other ML/AI algorithms because it is rule-based and explainable, providing reasons for prediction, classification, and anomaly detection results. Rules can include existing tactical knowledge and rules of engagement. New rules can also be discovered from big data via online, on-policy, and continual learning. Soar has been used in modeling large-scale complex cognitive functions for warfighting processes in kill chain applications such as Combat Identification (Zhao et al., 2018) and Battle Readiness Engagement Management (BREM) wargame (Zhao et al., 2020)

The machine learning component also includes unsupervised ML algorithms. We often perform clustering, unsupervised neural network, or lexical link analysis (LLA) from the LAILOW framework to improve prediction, detect anomalies, and sort/rank important information. LLA is an unsupervised ML method and describes the characteristics of a complex system using a list of attributes or features, or specific vocabularies or lexical terms. Because the potentially vast number of lexical terms from big data, the model can be viewed as a deep model for big data. For example, we can describe a system using word pairs or bi-grams as lexical terms extracted from text data. LLA automatically discovers word pairs, and displays them as word pair networks. This innovative configuration of LLA allows us to use it to discover and rank high-value information such as attributes and factors that correlate to the measures of performance of a complex enterprise from both unstructured and structured data. Bi-grams allow LLA to be extended to numerical or categorical data. For example, using structured data, such as attributes from maintenance and supply chain databases, we discretize numeric attributes and categorize their values to word-like features. The word pair model can further be extended to a context-concept-cluster model (Zhao & Zhou, 2014). A context can represent a location, a time point, or an object shared across data sources. Figure 2 shows an output of word networks from LLA for an unstructured data of a ship corrosion patent. Each node represents a word and each link represents how likely (the strength of the link) two words are next to each other as a bi-gram phrase. For example, “polystyrene dish” has a strength 242.5. An example of LLA for unstructured data is shown in Figure 2.

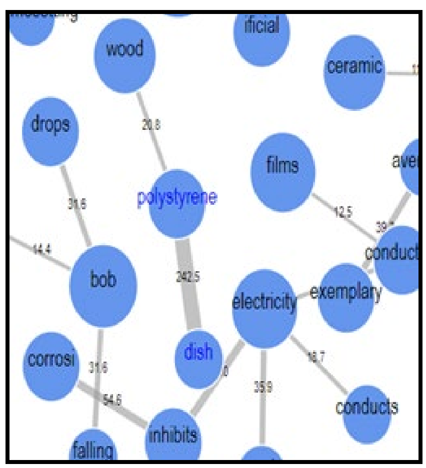


Figure 2. Examples of LLA from Unstructured Data

In this paper, we use LLA for the structured data. With the bi-gram representation and context-concept-cluster models, LLA can be used as a “market basket analysis,” where items appear together in the same context are considered associated or linked. Such association and link patterns can be used to improve prediction (Use Case 1 and 2) and rank items (Use Case 3).

Causal learning is also important for process improvement and quality control of a complex enterprise such as Navy Ships. The common consensus is that data-driven analysis or data mining can discover initial statistical correlations and associations from big data. Decision-makers and engineers often need to validate causes behind any observable effects. This calls a systematic approach of causal machine learning. The key factors for causal learning include the three layers of a causal hierarchy—association, intervention, and counterfactuals (Mackenzie & Pearl, 2018). A typical causal machine learning needs to select a cause (*C)* that maximizes the counterfactual difference *Probability (E|C) – Probability(E|Not C),* where the effect *E* is observable data and cause *C* is actionable and controllable variable*.* If deep analytics can reason and detect the cause for good or bad events (effects), engineers and decision-makers can fix the cause, avoid bad events/effects, and achieve desired effects. LLA potentially allows such a causality analysis. Causality analysis is related to the counterfactual regret minimization (CFR) that becomes important for many ML/AI applications.

- Component 2—Optimize: Based on the patterns, LAILOW optimizes the measures of effectiveness (MOEs) or the measures of performances (MOPs), defined by business decision-makers, by searching through all possible courses of actions to improve performance. The MOPs can be probability of failure (POF), probability of demand (POD), cost, time, and total readiness *(*e.g., a system uptime).

After machine learning algorithms in LAILOW discover associations, patterns, and rules, optimization algorithms can use them to search for decisions, course of actions, configurations, and combinations to optimize predicted MOEs and MOPs. LAILOW draws upon evolutionary algorithms for optimization. Evolutionary algorithms are genetic algorithms, which integrate the metaphor of genetic reproduction of selection, mutation, and crossover where the objective function's derivatives are not easy to compute.

- Component 3—Wargame: The LAILOW framework can be set up as a wargame with two players in order to test the quality or performance capabilities of a complex enterprise. One player, "self," represents the complex enterprise. The "opponent" of "self" evaluates the complex enterprise for robustness and resilience under stress, for example, when environmental factors, such as mission requirements, weather, emergency events, natural disasters, or adversaries, (who may deliberately generate disruption and exploit the vulnerability of the self-player), come into consideration or suddenly emerge.

The wargame serves the purpose of improving a real-time and dynamic operational environment through adaptative modeling. The opponent generates new operation conditions and events that might challenge the whole value chain and readiness measures in an intent to either improve the complex enterprise or disrupt the complex enterprise. The self-player adapts to optimize the actions and solutions to counter the opponent's actions/decisions. The whole process iterates and escalates due to each player adapting to the other.

To create a wargame environment, LAILOW uses coevolutionary algorithms (O'Reilly & Hemberg, 2018; Popovici et al., 2012). These are related to evolutionary algorithms and genetic algorithms (Back, 1996; Goldberg, 1989) and provide search, adaptation, and optimization mechanisms for two populations that engage to test and solve problems respectively. Coevolutionary algorithms explore domains in which the quality of a candidate solution (e.g., an action combination) is determined by its ability to successfully pass some set of tests (attackers), for example, solutions (defenders) in a logistics chain need to pass the known difficult or adversarial tests (attacks). Competitive coevolutionary algorithms are used to solve minmax problems, similar to those encountered by generative adversarial networks (GANs; Arora et al., 2017; Goodfellow et al., 2014), where adversarial engagements of opponents can be computationally modeled. Competitive coevolutionary algorithms take a population-based (parallel) approach to iterative adversarial engagement. In this competitive setting, the test (attacker) and solution (defender) strategies can lead to an arms race between the players, both adapting or evolving while pursuing conflicting objectives.

In summary, Navy Ships including USMC need to constantly perform a type of what-if and AoA wargame simulations in order to get ready for the unknown situations and perform in a contested environment. There are many challenges for Navy Ships and global materiel distribution that often require such wargame simulations. Forward deployed Navy Ships, particularly in the high operating tempo (OPTEMPO) areas such as the Seventh and Fifth Fleets, have challenges that arise in receiving logistical support when parts failures occur. These failures manifest as either a demand on the supply system, a casualty report (CASREP), or a request for technical assistance, which can cause a “redline,” or a failure that stops the unit from being able to complete the whole mission until the problem can be resolved. Limited manpower, funding, storage space, and resources for repair are all in high demand (Stevens & Zhao, 2021). A good system needs to be in place to determine the most efficient and effective method of stocking, forward staging, or contracting for the materials that have the highest likelihood of demand and balance with the potential impact of failure. LAILOW can support these system requirements because it exploits algorithms for learning, optimization and wargaming.

Use Case 1: Marine Maintenance and Supply System

As part of Navy Ships, the USMC maintenance and supply chain is a complex enterprise and exemplifies socio-technological infrastructures that require continuous learning, optimizing, and wargaming. To show the feasibility of the whole LAILOW framework, we first fuse and synthesize seven years of maintenance and supply time series data for a Marine equipment, namely, Land Armored Vehicles (LAV), including maintenance, supply, and equipment usage from the database Global Combat Support System-Marine Corps (GCSS-MC). We then aggregate the data for each maintenance and supply ticket as shown in Figure 3(a). There are about 500 aggregated variables representing states and actions for both the self-player and opponent when applying LAILOW. The sample data set contains ~11% tickets that have the days between deadlined (i.e., the Marine term for “redlined”) and closed date more than 32 days (32 days is the mean of the days between the deadlined and closed dates for the data set).

As shown in Figure 3(b), we first apply Orange’s predictive algorithms to predict the target variable “days between deadlined and closed” for each ticket. We add LLA to improve predictive models. We also add Soar-RL as another predictive algorithm outside Orange to predict the same target variable which result in comparable predictive accuracy. Finally, we divide all the variables into two groups: Attackers and Defenders, shown in Figure 3(d), and apply the coevolutionary algorithm using the predictive rules generated using Soar-RL. The predictive rules are generated for both Attacker variables and Defender variables to predict the target variable or fitness function in opposite directions. During the wargame phase, the Attacker variables change their values to increase the Attackers’ fitness, or increase the days between deadlined and closed; while the Defender variables change their values to increase the Defenders’ fitness or decrease the days between deadlined and closed.

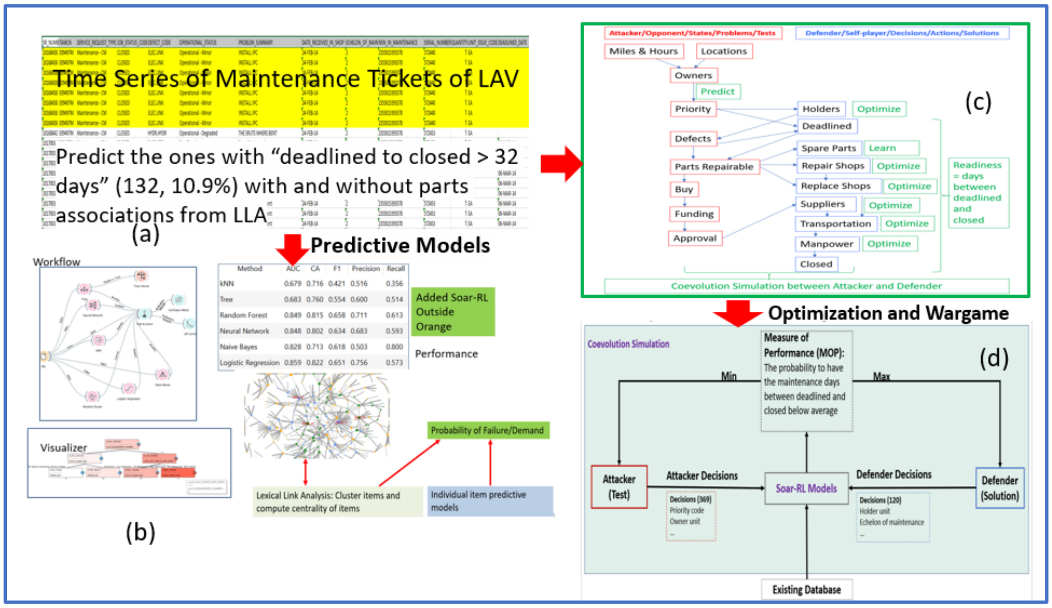


Figure 3. The LAILOW Framework Applied to the Marine Use Case

The Soar-RL and coevolutionary algorithms thus systematically simulate and discover possible new tests or “vulnerabilities” for a complex system and evolve solutions accordingly. For example, the evolved Attacker “d284e4” in Figure 4(c), which is a specific combination of Attacker variables, has an improved fitness -0.204 from where it starts from the database, i.e., -0.34 for “10fe75,” in Figure 4(a), against the best Defender “b642cf.” Such an attacker can potentially present a challenge or vulnerability to the current logistics solution system, because it is difficult for the defender to come up with a better solution than “b642cf.” Of course, the feasibility of such an Attacker configuration needs to be considered as well.

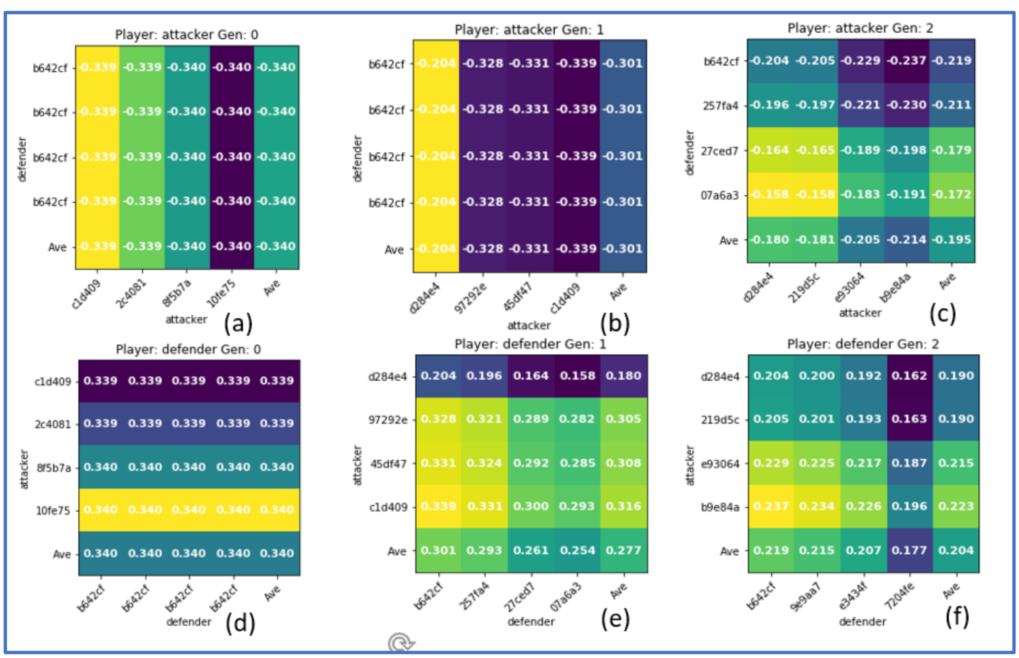


Figure 4. The Evolution Process for Attackers and Defenders.

The evolved Attacker “d284e4” in Figure 4(c), which is a specific combination of Attacker variables, has an improved fitness -0.204 from where it starts from the database, i.e., -0.34 for “10fe75,” in Figure 4(a), against the best Defender “b642cf.” Such an attacker can potentially present a challenge or vulnerability to the current logistics solution system, because it is difficult for the defender to come up with a better solution than “b642cf.”

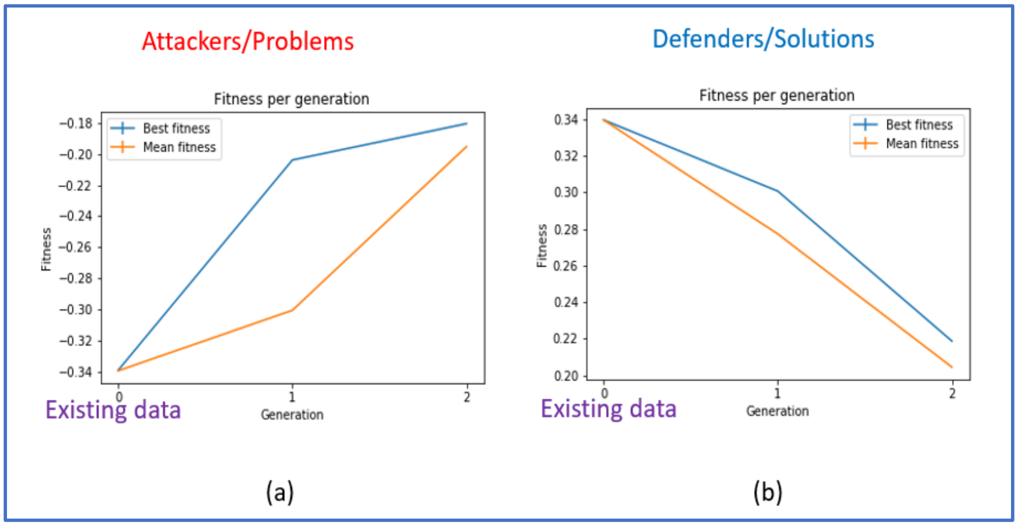


Figure 5. The Evolution Process for Attackers and Defenders

Figures 5(a) and 5(b) show the Attackers’ and Defenders’ mean and best fitness values changing for three generations in the coevolutionary algorithms, respectively. The trends validate the results and analyses that the self-player or Defender, representing the logistics solutions, gets worse on average while the opponent or Attacker, representing logistics tests, gets better on average in the coevolution simulation.

In summary, we show a use case of LAILOW which is capable of evolving, searching, simulating, and performing what-if analyses that reveal the new tests and solutions, possible vulnerability of the logistics system. The simulation can also suggest novel and more powerful solution (defender) configurations to handle new tests (attacker) that are never seen before.

Use Case 2: Predictive Risk Sparing Matrix (PRiSM)

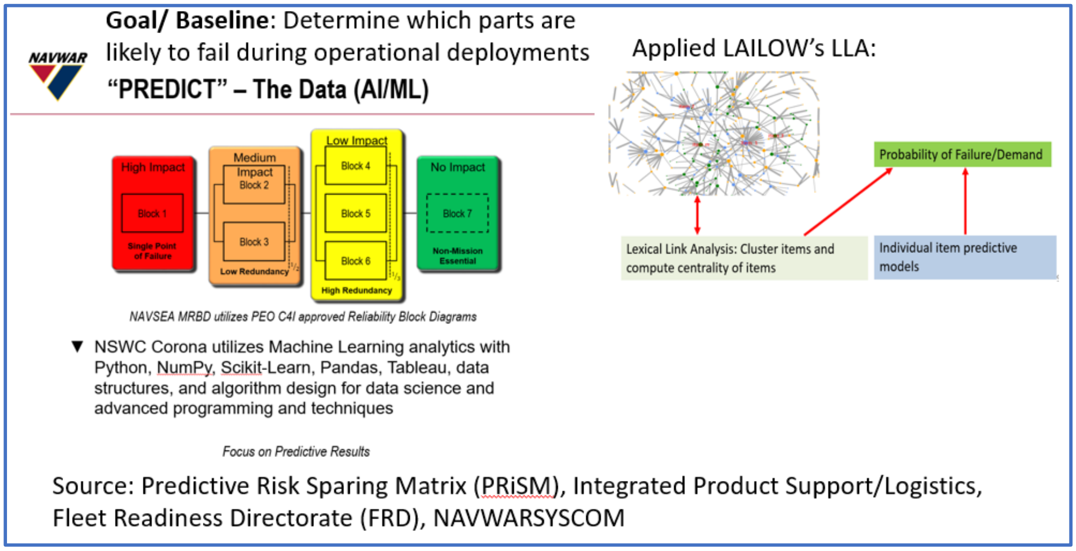


Figure 6. Predictive Risk Sparing Matrix (PRiSM)

NAVWARSYSCOM's Predictive Risk Sparing Matrix (PRiSM) as shown in Figure 6 is a product to accurately predict which critical Navy ship parts, related to Command, Control, Communications, Computers, and Intelligence (C4I) systems, are likely to fail during an operational deployment. If the ship parts' failure can be correctly predicted, the parts can be proactively staged aboard Carrier Strike Group (CSG) and Amphibious Ready Group (ARG) platforms. Advanced analytics is needed for decision-making to replace parts with low remaining service life or pre-position parts afloat or ashore or accept risk with no parts support. The decisions are then to use for course of actions, for example, to build and fill AT-5 TYCOM Allowances onboard CSG/ARG ships, transfer parts to areas of responsibility (AOR) DD site, and monitor results during the deployments. This provides support from their advanced training phase (COMTUEX) to end of deployment in increasing mission readiness by reducing Mean Logistics Delay Time (MLDT) and relative Mean Down Time (MDT). Using various key data points, PRiSM utilizes ML analytics with Python, NumPy, Scikit-Learn, Pandas, Tableau, data structures, algorithm design for data science, and advanced programming and techniques to build a baseline of prediction performance. It has been previously funded by Commander, U.S. Pacific Fleet and developed by NAVSEA, NSWC Corona. In 2020, we tested the LLA algorithm with the PRiSM product. LLA uses the parts association patterns (e.g., what parts are likely to fail with what other parts) to improve the probability of failure. For the test using real-time USS *Boxer* (BOXER ARG) and USS *Theodore Roosevelt* Carrier Strike Group (TR CSG) deployment data, the predictive accuracy improved from ~60% to ~80% by adding LLA. PRiSM and LLA are complementary and use different information to pick up different types of failure, which made the improvement possible.

Data Set

The following detail of LLA to PRiSM is showing using the data from the TR CSG. Like many ML algorithms, LLA first sifts through a so-called train data set to extract patterns (i.e., failure association patterns for patterns), network models, and visualizations, and then apply the patterns to a test/validation data set as shown in Figure 7. Both data sets were extracted from the current PRiSM application, reflecting the real-time event of the TR CSG deployment.

- Train data: Failure data 2 years prior to the TR CSG deployment

- Test/validation data: Failure data during the TR CSG deployment

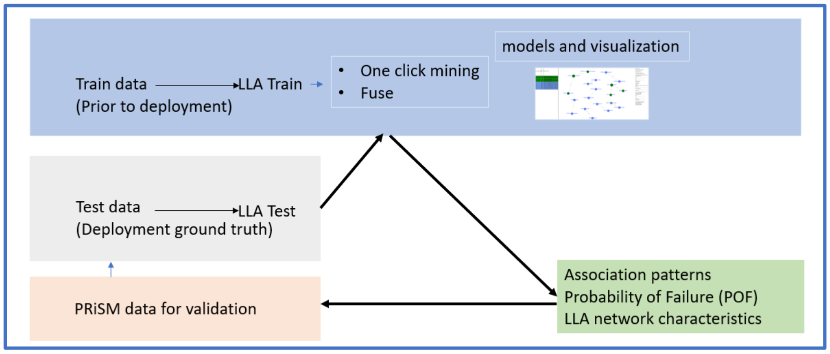


Figure 7. LLA Process Overview

LLA Application

We first use LLA to compute associations to see if two items or parts (used interchangeably below) are linked in the sense if one item fails, the other item also fails. The steps are shown below:

1. In order to compute statistically significant association patterns, we first group the failed parts into "baskets." Each basket is identified as a combination of the location of a failed part in the system (e.g., "object index," "unit id") and timestamp of the failure event (e.g., "year and month of event start date time") as shown in Figure 8. We choose the failure event time "month" for the basket combination for associated failed items which fail in a period of month sequentially. Figure 8 also shows an example of basket and item pairs.

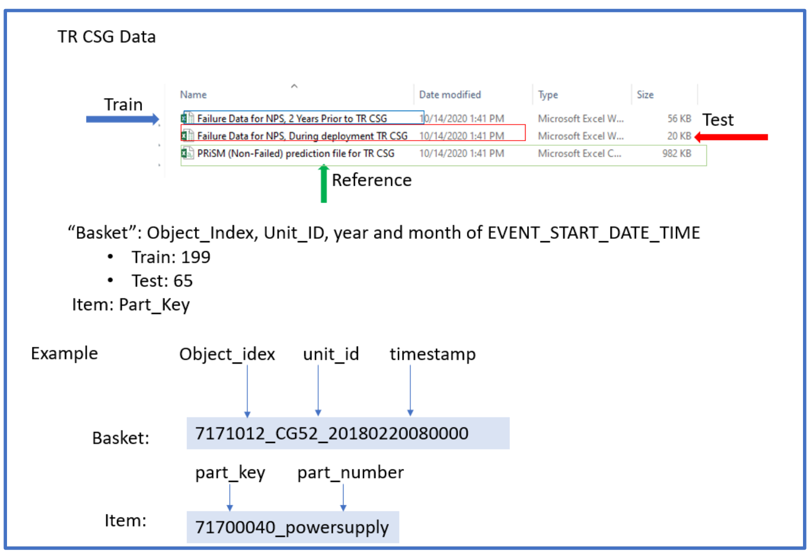


Figure 8. Basket and Item Mapping for the TR CSG Sata

2. LLA applies causal learning and compute counterfactual proportion difference, i.e.,

   cf= ***[P(B|A) – P(B|Not A)]*(pooled sample size)*** *(1)*

   to compute the strength of the association of two parts as *cf*, where P(B|A) is the probability of part B fails within the same basket (i.e., fails at the same location and time frame) if part A fails. The pooled sample size is a pooled number of historical failure of item A and B based on their pooled historical failure probabilities. *cf* is a z-score (PSU, 2021) and we use *cf >1.96* for *p-value < 0.05* as the statistical significance for the associations. In Figure 9, item A could be more important than item B for causality, because A's failure might cause B's failure although A has fewer failure than B in total. The step generates an item network for all historical failed parts for the train data set.

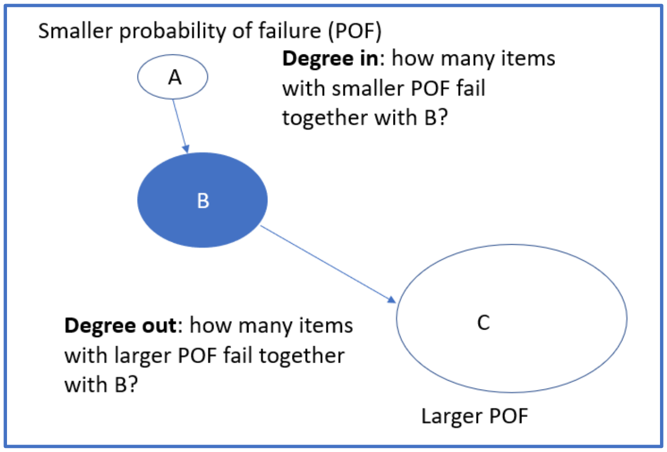


Figure 9. Causal Learning in LLA

3. Once a network of items is generated using the measure defined in Equation (1) , a community finding algorithm (Girvan & Newman, 2002) is performed to cluster items into groups or communities, and then compute centralities such as degree in, degree out, and betweenness scores.
4. Outputs and centralities from LLA
    a. The probability of failure (POF): the percentage of baskets containing an item (e.g., item B in Figure 8)
    b. Degree in: the number of items with smaller POF (e.g., item A in Figure 9) fail together (i.e., in the same basket) with item B
    c. Degree in weight: degree in/average (cf), total estimated causal impact from other items
    d. Degree out: the number of items with bigger POF (e.g., item C in Figure 9) fail together (i.e., in the same basket) with item B
    e. Degree out weight: degree out/average (cf), total estimated causal impact to other items if item B fails
    f. Betweenness: the number of items to which item B links are in the different groups from the one of item B based on the community finding algorithm output.

Results

The outputs and centralities from LLA are used to improve the prediction for PRiSM. Fifty-one high impact C4I parts that were predicted failed and had actually, matched with either the LLA predictions or PRiSM predictions, improved from 36 matched from the PRiSM predictions alone. The total failed number of items failed is 64. PRiSM and LLA are complementary in terms of predicting failed items in this use case.

Use Case 3: Readiness Impacts of Underfunding Spares Backlogs

Navy Ships' aviation and maritime units order spares from their general funds to fill modeled allowances. If there is not enough funding to buy all modeled allowances, spares requirements accumulate in a Financially Restricted Work Que (FRWQ) awaiting resourcing. In the meantime, the systems with these parts support are still fielded, and the Fleet still generates requirements to replace these parts.

The goal of this use case is to conduct a comparison of Fleet Demands against requirements in a FRWQ and assess these requirements with high priority demands linked to maritime units' CASREP and aviation unit's casualties, i.e., Non-Mission Capable Supply (NMCS). The result will help improve and determine the efficient and effective method of prioritizing materials that have the highest likelihood of demand balanced with their impact to readiness as a whole.

Baseline

The current tool and methodology of scoring and prioritizing the items are based on the DoD Manual 4140.01-V2 (DoD, 2018) in a FRWQ, with respect to their impact to the weapon system and aviation readiness data from CASREP and NMCS. The DoD Manual 4140.01-V2 describes that two categories of measures, i.e., the weapons system criticality and fleet demand, are needed to prioritize items as shown in Figure 10. The weapons system criticality is measured by the Item Mission Essentiality Code (IMEC) or Weapon System Group (WSG) code. We use IMEC in this paper as follows:

- IMEC Points: IMEC=5, 100 points awarded; IMEC=4, 80 points awarded; IMEC=3, 60 points awarded; IMEC=2, 40 points awarded; and IMEC=1, 20 points awarded. Figure 11 shows the detail of IMEC points calculation.

The fleet demand is based on the two criteria of intermittency and correlation variance (CV) for scoring and prioritizing items

- CV Points: calculated as the ratio of the standard deviation of the demand to the average demand and normally expressed as a percentage. A "low variance" is less than 75%; "median variance" is between 75% and 125%; and "high variance" is greater than 125%.
- Intermittency Points: calculated as the percentage of total historical demand periods (e.g., months in a year) that have non-zero demand. A "continuous" intermittency for an item means it is needed greater than 85% of 12 months (i.e., at least 11 of 12 months), while "limited" is less than 10%; "uneven" is between 10% and 60%; "erratic" is between 60% and 85%. Figure 12 show the details of CV and Intermittency calculations.
- Platform/Type Points: In addition to the IMEC, CV, and Intermittence points, platform and hull type are also used in the baseline calculation for total points. The overarching idea is that afloat units are given higher priority than shore units. Within afloat, we can prioritize further by leaning on U.S. Fleet Forces Command, which releases a fleet priority list on a semi-annual basis. Figure 13(a) shows the detail of the points calculation. This is a very insightful list, as it shows which units are highest priority (deploying soon) vs. low priority units (those in extended shipyard avails). But even the lowest priority afloat unit would still receive slightly higher points than a shore unit. The platform/type ranking shown in Figure 13(b) is just for CVNs. Other platforms and types receive their own rankings.

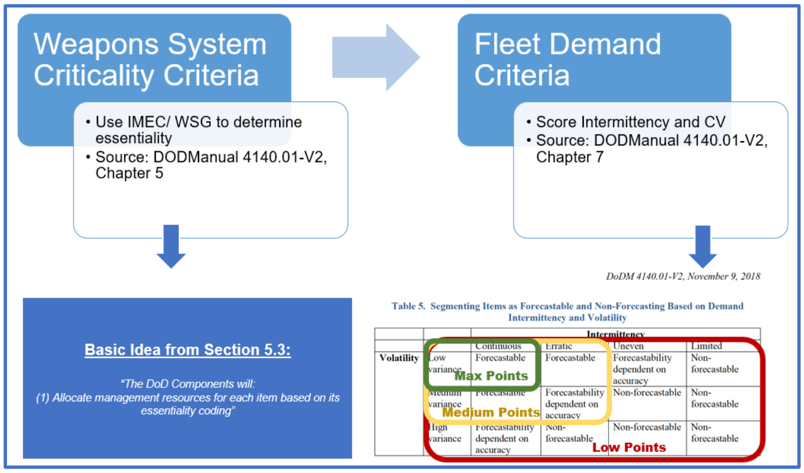


Figure 10. Current Methods of Prioritizing Items for a FRWQ

IMEC Score Legend

IMEC_CATEGORY	IMEC Meaning	POINTS AWARDED
5	Critical	100
4	UNK	80
3	UNK	60
2	UNK	40
1	Non-Critical	20
Unknown/Missing	N/A	60

Figure 11. IMEC Points Calculation (DoD, 2018)

Platform Type Score Legend

TYPE_NAME	TYPE_CATEGORY	POINTS AWARDED
CVN/LDECK	AFLOAT [AVIATION]	55-100 (USFF Sliding Scale)
All Maritime Units	AFLOAT [MARITIME]	51-100 (USFF Sliding Scale)
MALS	SHORE	50
NAS	SHORE	50
Unknown/Missing	N/A	50

CVN Name	Type Points
DWIGHT D. EISENHOWER	100
THEODORE ROOSEVELT	95
RONALD REAGAN	90
CARL VINSON	85
HARRY S. TRUMAN	80
NIMITZ	75
ABRAHAM LINCOLN	70
GEORGE H.W. BUSH	65
GEORGE WASHINGTON	60
JOHN C. STENNIS	55

Figure 13. Platform/Type Points Calculation

In the base line calculation, total points are the sum of CV points, Intermittency points, IMEC points, and Platform/Type Points. The maximum possible score is 400 points, and the minimum possible score is 115 points.

Data Sets

There are two data sets used in the use case:

- Data set 1: Historical raw demand for items related to aviation readiness and NMCS
- Data set 2: Historical raw demand for items related to maritime parts and CASREPs

The baseline points calculation examples:

1. An aviation example for NIIN 015761575 associated with R2010393330312 (LHA 6 Document)
   - CV Points: low variance (100 points awarded)
   - Intermittency Points: continuous (100 points awarded)
   - IMEC Points: 4 (critical system; 80 points awarded)
   - Platform/Type Points: LDECK (afloat type, highest LDECK Priority for USFF; 100 points awarded)
   - Grand Total: 380 points (highest priority for FRWQ Investment)
2. A maritime example: NIIN 005181789 associated with N2194510161535 (DDG 71 Document)
   - CV Points: low variance (100 points awarded)
   - Intermittency Category: continuous (100 points awarded)
   - IMEC Category: 5 (critical system; 100 points awarded)
   - Platform/Type Points: DDG (afloat type, 4th highest CG/DDG priority for USFF; 98.5 points awarded)
   - Grand Total: 398.5 Points (highest priority for FRWQ Investment)

LLA Application

LLA can be applied to the maritime data set 2 more meaningfully since it contains a data attribute JCN that is used to group the items into a same requisition time or "basket." The items are the National Item Identification Numbers (NIIN) and hull type. There are 611,335 unique baskets and 280,762 unique items in this data set; 2,093,633 statistically significant associations are found. The hypothesis is that items that appear together in the same baskets in the raw data, historical data might they be associated with a same cause so they are demanded together.

In general, LLA and network theory are potentially to provide a network and centrality view of the items/parts generated from raw demand data, which is related to the network analysis applications, for example, ranking people in a social network or ranking an object such as a biological gene in an environment where such a baseline ranking is not available.

However, application of LLA in this use case do not conclude better and more meaningful rankings than the existing methods. The correlations of LLA scores and baseline points are shown in Table 1. LLA suggests using "Degree out weight" scores as the total estimated impact to other items as the scores for the item's importance, which has the least correlation of the total points. One of association patterns discovered is meaningful as shown in Figure 14. POD has a correlation 0.34 with the total points. This indicates LLA's centrality measure "Degree out weight" does not use demand as signals for deciding the importance of an item. This may indicate the argument of causality learning that one item's demand might cause another item's demand may not fit to this problem, and the low-demand and high-impact items may not exist in the FRWQ data.

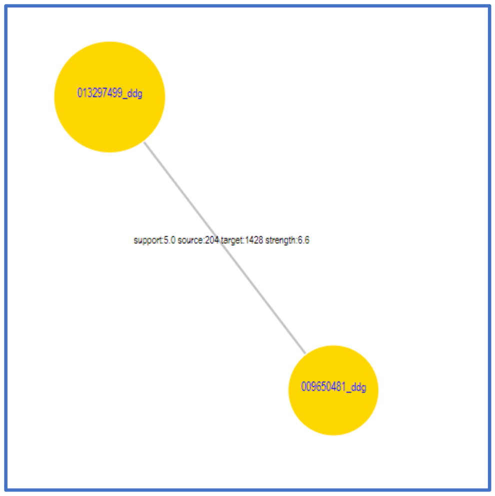


Figure 14. Example of LLA Associations Between Two Items

This calls for further research of LLA, possibly applications of other LAILOW methods such as supervised ML methods. For example, feedback data needs to be collected for consequences to understand how item prioritizing decisions and resource allocation decisions' impact future readiness. Future work also includes a review of business processes at a holistic level and consideration to plan for a whole class of ships or a whole fleet for a period of time (e.g., the CVN-74, USS *John C. Stennis* group for last a few years).

Table 1. Correlations of LLA and Baseline Points

	Total Points	CV	Intermittency	IMEC	Platform/Type
POD	0.34	0.34	0.36	0.09	-0.09
Degree in weight	0.20	0.10	0.25	0.16	-0.11
Degree out weight	0.08	0.06	0.19	-0.019	-0.06
Degree	0.07	0.03	0.15	0.012	-0.04
Betweenness	0.17	0.07	0.19	0.18	-0.13

Conclusions

In this paper, we show the LAILOW framework provides a holistic predictive and simulation platform to improve the readiness of Navy Ships. The Soar-RL, comparable to other predictive machine learning algorithms, rule-based, and explainable, can be integrated with the coevolutionary algorithm to conduct a wargame for a Navy complex enterprise. The wargame simulates and discovers possible new tests or "vulnerabilities" of a value chain for U.S. Ships and related complex enterprises, and evolve solutions or "resiliency" accordingly for uncertain and new conditions.

Recommendations and Future Work for Navy Ships

In some use cases, LAILOW methods may require to collect more right data for deep analytics such as feedback data to be collected for consequences to understand how item prioritizing decisions and resource allocation decisions' impact future readiness. Navy Ships need to adopt more deep analytics, machine learning and AI algorithms for big data or no data and focus on the entire spectrum or end-to-end (E2E) logistic planning.

References

Arora, S., Ge, R., Liang, Y., Ma, T., & Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (GANs). *Proceedings of the 34th International Conference on Machine Learning*, 70, 224–232. http://proceedings.mlr.press/v70/arora17a.html

Brown, N., & Sandholm, T. (2017). Safe and nested endgame solving for imperfect information games. *Proceedings of the AAAI workshop on Computer Poker and Imperfect Information Games*. https://arxiv.org/pdf/1705.02955.pdf

DoD. (2018). *DoD supply chain materiel management procedures: Demand and supply planning* (DoD Manual 4140.01-V2). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/414001m/414001m_vol02.pdf

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*(12),7821–7826.

Goldberg, G. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680. https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT Press.

Laird, J. E., Derbinsky, N., & Tinkerhess, M. (2012). Online determination of value-function structure and action-value estimates for reinforcement learning in a cognitive architecture. *Advances in Cognitive Systems*, 2, 221–238.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521 (7553), 436–444. https://doi.org/10.1038/nature14539

Loch, J., & Satinder, S. (1998). Using eligibility traces to find the best memoryless policy in partially observable Markov decision processes. *Proceedings of the 15th International Conference on Machine Learning*, 323–331. Morgan Kaufmann Publishers.

Mackenzie, D., & Pearl, J. (2018). *The book of why: The new science of cause and effect*. Penguin.

O'Reilly, U., & Hemberg E. (2018). An artificial coevolutionary frame work for adversarial AI. *Proceedings of AAAI Fall Symposium: Adversary-aware Learning Techniques and Trends in Cybersecurity (ALEC)*. http://ceur-ws.org/Vol-2269/FSS-18_paper_37.pdf

Pearl, J. (2019). The seven pillars of causal reasoning with reflections on machine learning. *Communications of the ACM.* https://ftp.cs.ucla.edu/pub/stat_ser/r481.pdf

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830. http://scikit-learn.org/

Popovici, E., Bucci, A., Wiegand, R. P., & De Jong, E. D. (2012). *Coevolutionary principles.* In G. Rozenberg, T. Back, & J. N. Kok (Eds.), *Handbook of natural computing* (pp. 987–1033). Springer.

Silver, D., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359. https://doi.org/10.1038/nature24270

Stevens, E., & Zhao, Y. (2020). Using lexical link analysis (LLA) as a tool to analyze a complex system and improve sustainment. *Proceedings of AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks*. http://ceur-ws.org/Vol-2819/session1paper3.pdf

Sutton, R. S., & Barto, A. G. (1998, 2014). *Reinforcement learning: An introduction*. MIT Press.

University of Ljubljana. (1996–2020). *Orange software.* https://orange.biolab.si/

Penn State University (PSU). (2021). *Online statistics: Normal approximation method formulas*. https://online.stat.psu.edu/stat200/lesson/9/9.1/9.1.2/9.1.2.1

Zhao, Y., & Zhou, C. (2014). System and method for knowledge pattern search from networked agents (U.S. Patent No. 8,903,756). U.S. Patent and Trademark Office. https://www.google.com/patents/US8903756

Zhao, Y., Derbinsky, N., Wong, L., Sonnenshein, J., & Kendall, T. (2018, December 2). Continual and real-time learning for modeling combat identification in a tactical environment. *Proceedings of the NIPS 2018 Workshop on Continual Learning.* Montreal, Canada. https://marcpickett.com/cl2018/CL-2018_paper_77.pdf

Zhao, Y., Nagy, B., Kendall, T., & Schwamm, R. (2020, November). Causal learning in modeling multi-segment war game leveraging machine intelligence with EVE structures. *Proceedings of AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks* [Virtual]. http://ceur-ws.org/Vol-2819/session3paper3.pdf

PAPERS ONLY

	Schedule Risks Associated with Middle Tier Acquisition Amir Etemadi, George Washington University John Kamp, George Washington University ***Two-Stage, Dynamic Data Envelopment Analysis of Technology Transition Performance in the U.S. Defense Sector*** Toshiyuki Sueyoshi, New Mexico Tech Youngbok Ryu, Northeastern University ***Developing a Threat and Capability Coevolutionary Matrix: Application to Shaping Flexible Command and Control Organizational Structure for Distributed Maritime Operations*** Ying Zhao, Naval Postgraduate School

Schedule Risks Associated with Middle Tier Acquisition

Amir Etemadi—received a Doctor of Philosophy degree in electrical engineering from the University of Toronto, Toronto, Canada, a Master of Science degree in electrical engineering from Sharif University of Technology, Tehran, Iran, and a Bachelor of Science degree in electrical engineering from the University of Tehran, Tehran, Iran. He joined the faculty at the George Washington University in 2013, where he is currently an Assistant Professor in the School of Engineering and Applied Science in the Department of Electrical and Computer Engineering and the Department of Engineering Management and Systems Engineering. He teaches masters- and doctorate-level courses in electrical engineering and engineering management.

Etemadi's research interests include distributed generation, power system dynamics and control, quantitative risk analysis, and engineering management. He is the Principal Investigator on several current research projects, including a National Science Foundation project on geomagnetic disturbance impacts on power system operation and an Acquisition Research Program grant on defense acquisition strategies. In addition, he is also a Faculty Graduate Research Advisor in the School of Engineering and Applied Sciences and has supervised over 50 masters and doctoral candidates for degrees in both electrical engineering and engineering management. Etemadi is a member of the Institute of Electrical and Electronics Engineers (IEEE). [etemadi@gwu.edu]

John Kamp—received a Doctor of Engineering in engineering management from the George Washington University in 2019, a Master of Engineering in nuclear engineering from Iowa State University, and a Bachelor of Arts in mathematics and French from the University of Nebraska-Lincoln. Kamp joined the George Washington University research staff in 2019 and is supporting Etemadi's acquisition strategy research project. He is a retired naval submarine officer with extensive experience in research and development and program management. His research interests include engineering management, maritime systems, and acquisition system research. Kamp is a Fellow in the Royal Institution of Naval Architects and a member of several professional associations. [jckamp2018@gwu.edu]

Abstract

Major defense acquisition programs take about 8 years to proceed from program initiation to an initial operational capability. Recent changes created Middle Tier Acquisition programs intended to deliver capabilities and products in less than 5 years.

This research defines schedule risk as the likelihood of exceeding a duration. We developed quantitative models to identify significant factors and relationships. This report summarizes our approach and presents modeled results associated with Middle Tier Acquisitions.

Program offices must concurrently adapt to both emergent guidance and programmatic realities. This research contributes to the understanding of the risks and opportunities associated with recent Middle Tier Acquisitions. The research results will be useful to program offices and acquisition leadership in executing current and future rapid acquisition programs.

Introduction

This research explored schedule-related risks and opportunities associated with implementing new rapid prototyping and fielding program authorities, modular open system architectures, and Agile development. We assessed schedule risks and opportunities associated with these innovations and identified programmatic modifications and measures to manage schedule growth.

Research Scope

Our research focused on quantifying the programmatic and system engineering–related schedule risks attributable to using rapid prototyping and fielding pathways, modular open systems architectures, and Agile acquisition practices. We developed quantitative models to identify significant factors and relationships. This report summarizes our approach and presents results associated with Middle Tier Acquisitions (MTAs).

The research applies to rapid prototyping and rapid weapon system MTA programs and specifically excludes programs intended to acquire services or defense business systems. It includes acquisition policy and management changes enacted in the 2016, 2017, and 2018 National Defense Authorization Acts (NDAAs) and the Department of Defense (DoD) and service guidance, governance, and execution strategies implementing these changes. The research findings may not be valid for programs beyond these innovations.

Research Questions and Objectives

1. What types of programs have delivered prototypes or fielded systems within 5 years?
2. What characterized innovative technologies and systems fielded within 5 years?
3. How do acquisition process innovations such as Agile development and modular open systems affect program schedule performance?

Research Objectives

1. To develop a program database from publicly available sources suitable for research.
2. To identify and quantify significant factors for rapid acquisition strategies and significant predictors of and risk factors associated with achieving schedule objectives.

This paper continues with a review of recent literature in the Literature Review. A methodology overview in Methodology describes datasets developed from publicly available sources and the quantitative methods used. Results and Analysis presents the results of quantitative analysis, and Conclusions and Future Work summarizes research results and suggests future opportunities.

Literature Review

Schedule is an outcome. Successful past approaches include incremental or evolutionary acquisition strategies, exemplified by Mortlock's (2019) case study, adopting or reusing existing technologies, and updating or modifying existing systems (Tate, 2016). We briefly review policy and highlight recent innovations and research, including open systems architectures, modularity, Agile development, and Middle Tier Acquisitions.

Rapid Acquisition Policies

Fox (2011) produced a comprehensive summary of defense acquisition reform efforts between 1960 and 2009 and chronicles the interplay between Congress, the DoD, and the defense industry. Significant changes included

- McNamara centralized acquisition authorities and introduced budgeting, programming, and requirements processes within the DoD.
- Laird and Packard instituted policies related to management by objective, decentralized execution, cost reforms, prototyping, identifying, and managing technical risks, and formalizing acquisition training.
- Increased accountability for results with congressional legislative initiatives such as the Nunn-McCurdy Amendment, Federal Acquisition Streamlining Act, and Clinger-Cohen Act.

Fox (2011) argued that DoD personnel did not have the expertise to implement and execute these reforms but emphasized the importance of congressional, DoD and industry leadership in creating these process and policy changes. Recent reforms starting with the 2016 National Defense Authorization Act increasingly emphasized speed of development and delivery. We will evaluate these reforms' effectiveness using publicly available data.

Middle Tier Acquisitions

Congress enacted Middle Tier Acquisition (MTA) processes in 2016 (NDAA, 2015, § 804). In 2019, the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD[A&S]) issued a new policy directive, *Operation of the Middle Tier of Acquisition (MTA)* (Lord, 2019). This policy introduced two new acquisition paths, rapid prototyping and rapid fielding, which are structured for rapid start, including setting requirements or starting production within 6 months, and delivery of a prototype residual capability or completed fielding within 5 years of start (Lord, 2019). The services released their own middle acquisition references concurrent with DoD issuance.

The DoD introduced the Agile Acquisition Framework, bringing traditional acquisition, urgent acquisition, Middle Tier Acquisitions, software, business and services acquisitions into a common framework (Lord, 2020b). The DoD issued extensive acquisition policy revisions in 2020, including *The Defense Acquisition System* (Lord, 2020b), and *Operation of the Adaptive Acquisition Framework* (Lord, 2020b). The MITRE Corporation created a comprehensive website collecting the DoD and service acquisition executive policy and guidance (MITRE, 2019).

Contracting Innovations

Contracting strategies are critical to rapid acquisitions. The time from program start to contract award is in series with development and production. Most government acquisitions use contracts conforming to the Federal Acquisition Regulation (FAR). Several innovations exist to reduce procurement acquisition lead time, including

- *Modular* contracting, introduced by the Clinger–Cohen Act (Clinger–Cohen Act, 1996). The intent was reducing investment risk, product delivery times, and barriers to introducing new information technology (Office of Management and Budget [OMB], 2012). Modular contracting has expanded beyond information technology acquisitions to include software and hardware development and procurements (OUSD[A&S], 2019).
- *Agile* contracting adapts contract management to support *agile acquisition* processes. Pennington (2018) notes that inherently Agile attributes such as incomplete requirements, incremental deliverables, and acceptance criteria make for challenging procurements. Contracting officers predominantly use fixed-price type contracts to manage Agile procurements; key issues include setting quality standards, definitions of done, and appropriate risk sharing (Ellis et al., 2019).

Statutory alternatives to FAR contracts also reduce lead times, including Procurements for Experimentation (Procurement for Experimental Purposes, 1993), and Commercial Solutions Opening (NDAA, 2016, sec. 879). These simplify commercial item procurements for research and development. Other Transaction Agreements (Research Projects: Transactions other than Contracts and Grants, 1993) are legally binding agreements where generally contract- and grant-related Federal laws and regulations do not apply. There are three common types of other transactions – other transactions for prototypes, other transactions for research, and other transactions for production (Research Projects: Transactions other than Contracts and Grants, 1993).

Business Innovations

Several advisory panels provided specific recommendations for business process innovations. A recent example was the Section 809 Panel, which provided extensive recommendations intended to accelerate acquisition processes by leveraging commercial marketplaces and processes, simplifying acquisition regulations, changing resource allocation processes, and improving the acquisition workforce (Drabkin et al., 2016). The DoD implemented less than half of their recommendations as of January 2021.

The DoD and Congress created several funding processes designed to accelerate technology transitions from non-traditional performers (Office of the Under Secretary of Defense for Research and Engineering [OUSD(R&E)], 2020). For example, the DoD Rapid Innovation Fund was created by Congress in 2011 and expanded in 2018 to accelerate small business technology transition to the DoD (NDAA, 2011). It is structured to move small business technology into operational use or to an acquisition program within 24 months (OUSD[R&E], 2020). Congress did not appropriate funding for this activity in 2020.

The Defense Innovation Unit is a different effort, embedded in Silicon Valley and reporting to the Under Secretary of Defense for Research and Engineering (USD[R&E]). It is focused on transitioning commercial advanced technologies to the DoD and uses an extension of other transaction agreement authorities (NDAA, 2015, § 815) to fund development and transition. It recently expanded to other locations and provides market access and non-dilutive capital for non-traditional defense contractors (Defense Innovation Unit [DIU], 2020).

Rapid System Acquisitions

Arellano et al. (2015) analyzed two successful rapid acquisition programs and noted the importance of direct senior leadership involvement to successful rapid acquisitions. This support mandates schedule adherence, requires programs to accept more risk, and creates agility to bypass financial and bureaucratic obstacles.

Wong (2016) identified three long-term (replacement, expedited, or traditional) and three opportunistic (missed, new, or alternative) acquisition categories. In his analysis, rapid acquisition processes depend upon budget reprogramming for initial action, but quantities depend upon capability adoption and use proliferation (Wong, 2016). The recently introduced acquisition pathways or strategies emphasize accelerated demonstration of a prototype or fielding of a new capability and are consistent with Wong's (2016) opportunistic categories. Of note, Congress provided statutory relief allowing transfer of procurement funds to rapid fielding accounts (NDAA, 2016), further supporting Wong's (2016) analysis.

Following Wong (2016), rapid acquisitions are a response to an emergent need—an immediate investment. Van Atta et al. (2016) defined "accelerated acquisitions" as those with requirements urgency, requirements specificity, and technology availability. They noted that relatively few (18 of about 330 Major Defense Acquisition Programs [MDAPs] reviewed) programs met these criteria and resulted from emergent urgent needs. Nine of these 18 programs delivered a prototype or claimed initial operational capability (IOC) within 5 years of program start (Van Atta et al., 2016). Dougherty (2018) examined programmatic and objective differences between six current rapid acquisition offices, noting attributes reflecting flexibility in contracting, transition, and programmatic objectives.

Open Architectures

Initiatives such as *open system architectures* intend to minimize change costs[1] by encapsulating functions in modules, using interfaces conforming to consensus standards, and

---

[1] Congress directed DoD use of open systems, called Modular Open Systems Architectures (MOSA; 10 U.S.C. § 2446a, 2016). MOSA development strategies emphasize module-level competition, development, testing and deployment, with multiple competing open standards (Engebretson & Frey, 2017).

establishing processes to ensure architectural compliance (Firesmith, 2015). Watson et al. (2016) related design margin (what they called "excess capacity") to the ability to evolve a military ground vehicle design over time. They found, given future requirements uncertainty, the optimal design in terms of cost and benefits of excess capacity was related to the expected design service life; also, excess capacity *was not cost-effective* when expected service lifetimes were *below a certain value* (Watson et al., 2016).

The implications are that system designs face a choice: produce systems designed for changeability over a long service life or in favor of sustained production of incrementally evolving systems with shorter system design lifetimes. We see examples of both choices today, such as the aircraft carrier (changing over a long life) and commercial computers.

Modularity

Modularity, like open systems architectures, is a design choice to reduce complexity or change system function or performance without creating a new system. It partitions product knowledge into distinct but related processes and products. Modularity allows companies to develop products faster and either protects or exposes company intellectual property (Baldwin & Henkel, 2015, p. 1641).

van Gent and Kassapoglou (2014) examined modularizing composite airframes and showed the effects on direct operating costs and fuselage weight with increasing modularity. They derived cost and weight values for specific flight load conditions and optimized structural designs. While cost and weight savings were achievable, they were reduced or lost at high modularity levels as modules became heavier (and more expensive; van Gent & Kassapoglou, 2014). Chadha et al. (2018) examined redesign of a representative missile using additive manufacturing processes to simplify system interfaces and expand the design space. Applying their redesign process to selected modules and components, they presented design changes improving reliability and manufacturability by reducing module part count, eliminating internal module interfaces, and optimizing module design. While improved, the redesigned missile still required flight certification (Chadha et al., 2018).

Agility and Agile Development

Agility has multiple meanings within the DoD; it may refer to software development, acquisition processes, or any changeable process or product. In 2017, the DoD had few programs using Agile software development methods.

Rosa et al. (2017) developed cost models for traditional ("waterfall") and Agile software processes within the DoD. Notwithstanding a small Agile process dataset, they found that *product size* (source lines of code) is a valid measure of required effort and Agile methods were more productive than traditional (non-Agile) software development methods (Rosa et al., 2017, p. 36).

Nidiffer et al. (2014) described Agile programs as "implementation-driven," meaning requirements are dependent on interactions and direct communications to establish short-term requirements, while traditional approaches focus on documented requirements. While Agile helps requirements validation, there still are specific issues such as contract modifications and managing non-functional requirements (Inayat et al., 2015). Adams (2017) identified DoD and non-DoD related factors affecting DoD Agile software development adoption, including contracting, requirements management, training and team organization. Schoeni (2015) found similar cultural barriers and identified regulatory constraints.

Rework is a significant shortcoming of Agile processes for physical systems, as it uses incremental function[2] delivery (OUSD[A&S], 2019). Cooper and Sommer (2016) proposed a hybrid development process, called Agile Stage-Gate, where Agile methods are applied within selected stages, such as studies and technology development, and gated with clear exit or "done sprint" criteria.

Haberfellner and Weck (2005) provide an excellent overview of Agile systems engineering in a series of illustrative case studies highlighting the systems engineering challenges of designing agility (speed of change) into real systems. They show that agility is valuable for long-lived systems when "significant switching costs exist coupled with substantial uncertainty[3] in the environment" (Haberfellner & Weck, 2005, p. 1463).

Islam and Storer (2020) developed a case study examining how safety-critical systems development conflicts with Agile development. While qualitative and from a single case, they identified three broad grounds of challenges: the influence of "waterfall-like" systems engineering processes on Agile teams, complex customer interactions, and conflicts between Agile process and regulatory standards, such as upfront design requirements for hazard analysis conflicting with incremental Agile design (Islam & Storer, 2020).

Production

DoD production (inventory) quantities for traditional acquisition programs are defined by requirements (Wicecarver, 2017), reducing incentives to produce more than contract requirements. Desai et al. (2007) considered the problem for commercial durable goods production and found inventory holding costs and durability incentivize lower inventory. Davis and Tate (2020) provide several examples of how acquisition quantities change over time and that systems change over time such that later production versions may be quite different than initial deliveries.

Physical system production at large scale requires extensive facilities. For example, in December 2019, Boeing and Airbus delivered 29 and 138 large commercial aircraft, respectively (Oestergaard, 2020). Boeing's Everett production facility covers nearly 100 acres (Boeing, 2020), and Airbus has five final assembly lines world-wide (Airbus, 2020). Changing production demand may exceed a contractor's capacity. In such cases, leader–follower production strategies may be useful.

Reconfigurable manufacturing systems reduce short-run production overhead and retooling costs by modularizing production processes for an intended parts family. Commercial modular production firms use mechanisms such as cost-sharing agreements, hedged delivery dates,[4] and premiums for early deliveries[5] to incentivize rapid acquisitions (Zhai et al., 2016), and spot and future markets can be created for premium demand purchases (Cai et al., 2020). Asghar et al. (2018) developed a multi-objective algorithm to optimize module (machine) sequencing and usage (scheduling) as production demands change (p. 4397). The research was specific for a part production line using programmable multi-axis milling machines. Efficient production sequencing minimized production downtime.

---

[2] Function, also called story or story points.

[3] Examples include requirements or demand uncertainty.

[4] In this case, hedging consists of setting module delivery dates earlier than need dates, thus covering the module production process time uncertainty.

[5] Zhai et al. (2016) call these premiums "crashing money."

The U.S. Air Force recently announced flight testing of a Next Generation Air Dominance prototype, developed using a “digital engineering[6]” based development process, asserting this to be a faster path to prototype demonstration than prior methods (Reim, 2020). The development time is not stated, but the program office was activated in October 2019, suggesting a short development cycle (Waldron, 2019).

Schedule Estimating–Related Research

Schedule risk has different definitions in the literature, ranging from the likelihood to achieve a predicted duration (Dubos et al., 2007) to an estimate of likelihood and consequence (Tao et al., 2017). Browning (1998) used causal loop representations to identify likely sources and consequences of schedule delays and showed how uncertainty drives risk. Thomas et al. (2014) extended earned value methods to estimate schedule risk within a detailed cost and schedule Monte Carlo simulation. Similarly, Wauters and Vanhoucke (2017) used machine learning techniques to simulate project schedule duration within an earned value methodology. Such simulations require detailed work project schedules and duration uncertainty distributions as inputs.

Jaifer et al. (2020) examined effort and time drivers for aerospace new product development and grouped them into complexity[7] and proficiency[8] categories, later adding uncertainty as a separate category following a subject matter expert survey. Jahr (2014) tried to quantify the effect of Agile project management on schedule relative to traditional program management processes. He developed a hybrid management process for software development using constrained activity-on-node graphs and ran an experiment comparing performance to scrum processes for modification and new product development. The teams using the modified process were able to outperform scrum teams in terms of schedule and cost growth for both new and modified software development (Jahr, 2014). This suggests that applying planning constraints and management can benefit Agile processes such as scrum-type software development.

Ingold (2014) noted that schedule durations for small software development efforts are approximately the cube-root of the planned effort in person-months, and the square root of planned effort for large efforts. He argued that Agile efforts tend to follow square root relationships and that while reducing schedule leads to cost growth, Agile processes are able to achieve schedules shorter than predicted by standard software cost estimating models (Ingold, 2014). He developed and calibrated an Agile schedule systems dynamic model with 12 qualitative schedule acceleration sub-factors that predicted schedule durations within a few percent (Ingold, 2014). His schedule-accelerating subfactors map to previously identified schedule factors (Riposo et al., 2014), and his people-related factors map to Jahr’s (2014) proficiency group.

Discussion and Summary

The literature provides an overview of program better practices and decisions associated with shorter schedules. These include

- Reducing requirements to meet capability and deliver something sooner
- Having a competent team and bounding the system by what is known and in use, including interfaces and standards

---

[6] Also called a digital thread (Bone et al., 2019).

[7] Such as size, technical difficulty, and uncertainty.

[8] Examples include experience, communications, and process management competency.

- Starting with an existing or proven system to reduce development delays
- Adjusting work to retire schedule risk
- Having sponsorship from the top and using the resulting flexibility to overcome inevitable obstacles
- Segmenting integration risk

There is extensive literature on specific manifestations of rapid acquisitions, notably modularity and Agile software development; however, these are focused cases and provide lessons learned for program management offices. The literature shows extensive development in policies and process supporting rapid acquisitions.

Congress and the DoD instituted several process and statutory changes in recent years, such as Modularity, Agility, and Middle Tier Acquisitions. The DoD adopted these innovations to reduce program cycle times. There are few quantitative articles on the effects of modularity and Agility on schedule performance, and Middle Tier Acquisition programs as relatively new, [9] representing a significant research gap. We focused this paper on Middle Tier Acquisitions schedule risk and the following research hypothesis:

- Middle Tier Acquisition programs have lower schedule risk than traditional MDAPs.

Methodology

We defined the likelihood of exceeding a specified schedule duration as the *schedule risk*. We did not assess risk context, severity, or treatment, as these are program-dependent. In Figure 1, the *vertical red dashed line* indicates a schedule risk of about 0.9 (89.2%) that the schedule of a program in this dataset will exceed 60 months.

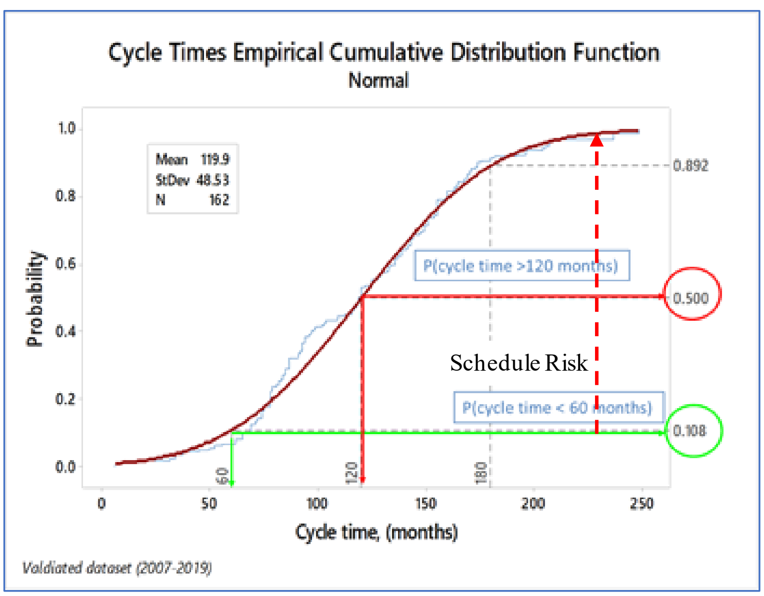


Figure 1. Estimation of schedule risk

[9] Some quantitative information is in recent budget documentation. The Fiscal Year 2022 budget data was not released as of March 31, 2021.

The key assumption is that similar program types have similar schedule durations and risks—in this case, MDAPs. If true, then there is a 50% chance of a similar program's schedule exceeding 120 months, and about an 11% chance of exceeding 180 months.

Research Design Overview

We relied on several publicly available data sources for this research: the Government Accountability Office (GAO) annual weapon system assessments, the Director, Operational Test and Evaluation (DOT&E) Annual Reports, FPDS.gov, and usaspending.gov. We included the Fiscal Year 2019 through 2021 budget documentation to identify rapid acquisition programs. We compiled data into comma-separated variable files that are available upon request. Contract data was substantial; programmatic data was sparse. We manually validated the smaller datasets.

Budget data text searches were critical to labeling programs as modular, open systems, Agile development, Middle Tier, or Section 804 acquisitions, and claiming rapid fielding or rapid prototyping activities.

We used Microsoft and Adobe text search engines, R, Minitab, and SPSS to identify relevant programs and significant factors. We developed simplified schedule models in Minitab and ran Monte Carlo simulations to estimate schedule risk for the previously-labeled strategies. We performed additional modeling on Air and Missile system commodity types to identify and characterize influential variables and test model predictive performance.

Research Terms and Definitions

GAO and DOT&E 2020 reports and DoD budget documentation provided most of the research data. We searched for programs with the following text strings:

- *"Agile development."* This is commonly a software-dominated development and delivery process with incremental requirements elaboration, schedule-driven product delivery and acceptance. Searches included "Agile," "Agility," and "Agile software." Searching using "Agile development" provided good specificity to specific budget documents. An example of "Agile development" is the previously mentioned Air Force Air Operations Center Weapon System 10.2 (AOC-WS 10.2) replacement program.
- "*Modular system."* A modular system is often the product of a program with multiple components in development or production, with system function determined by the types of modules used to compose systems. Searches included "modular" and "modularity." An example of a modular system is the Army Field Medical Equipment.
- *"Middle Tier."* This is a program or project using Section 804 (Middle Tier Acquisition) pathways. Searches were also conducted using "Section 804," "MTA," "rapid fielding," and "rapid prototyping." Prior to the 2016 National Defense Authorization Act, the DoD used "Agile," "rapid fielding," and "rapid prototyping" within budget documentation descriptions of program plans and strategies. Examples include the Air Force Air-Launched Rapid Response Weapon (ARRW) and AOC-WS 10.2
- *"Open System."* Also known as Open System Architecture & Modular Open Systems Architecture Explicit Interfaces, standards, composition rules (structure) Often includes configuration management. Examples include the Air Force F-16 Modification of In-Service Aircraft and AOC-WS 10.2.

Results and Analysis

Text searches of budget documents between Fiscal Years 2010 and 2021 inclusive showed that Agile development and Modular system are more common in procurement documents, and Middle Tier and Open System are more common in research and development documents. Also, the Air Force has more activity related to Agile development and Middle Tier Acquisitions that the other services. Figure 2 shows these trends.

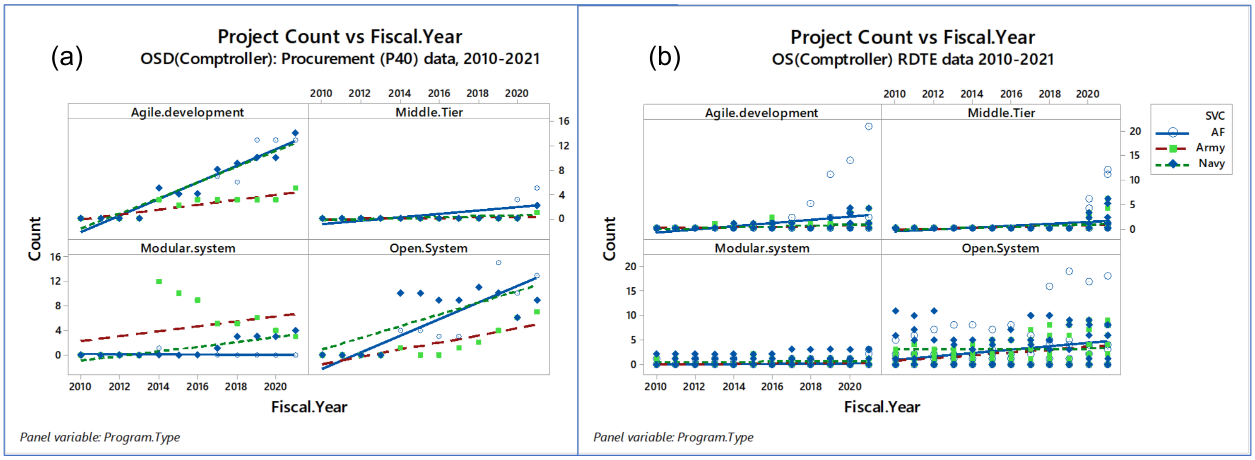


Figure 2. Term usage count summary by fiscal year

Figure 2 shows increasing overall usage trends for both procurement (Figure 2[a]) and research and development (Figure 2[b]). Agile development and open system activities are growing in all services. Open systems are a sustained emphasis in research and development, and Middle Tier Acquisition programs are a recent development.

Modeling Middle Tier Acquisition Program Schedule Duration

We used data from the GAO 2020 annual weapon systems assessment and added Fiscal Year 2020 and 2021 budget documents to create a dataset with program cycle times or schedules with programmatic factors and classification using the previous labels.

We found that modifications of existing systems were commonly occurring with modular systems and considered “modification” as a proxy for modular and open systems in the dataset.

Agile development programs did not always have a clear end date or planned initial operational capability date. The challenge is part of determining the “definition of done” (OUSD[A&S], 2020), essentially when the accumulated product value meets the customer requirement. This is normally a software development issue and may be in-service use, date of authority to operate, or another defined state. This definition problem exists in Agile development and Middle Tier Acquisition programs. We chose either the latest specified product delivery date or the last date in the budget submission.

We subsetted the data to consider Air (AIR) and Missile (MSL) commodity type programs. Figure 3 summarizes the cycle times by Agile (AGILE), MDAP, modification (MOD) and Middle Tier Acquisition (MTA.RP).[10]

---

[10] All Middle Tier Acquisition programs in the dataset were rapid prototyping projects.

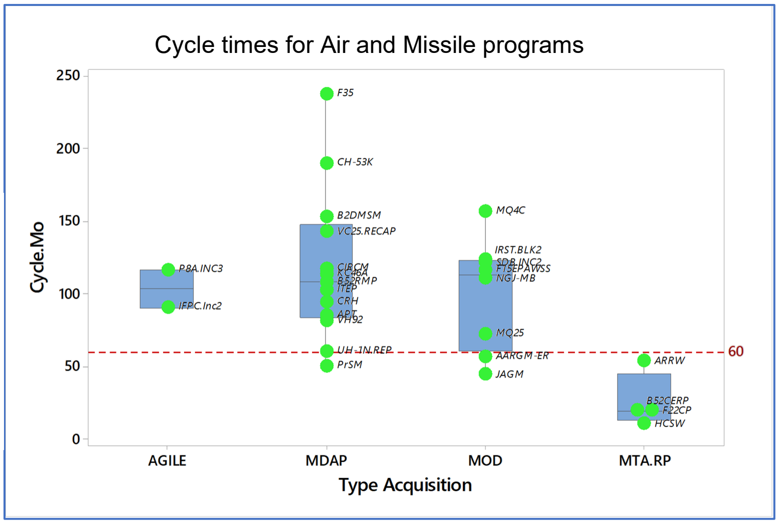


Figure 3. Cycle Times for AIR and MSL Programs (GAO, 2020)

Figure 3 highlights that Middle Tier Acquisition programs are so new that none have reported schedule growth resulting in durations greater than 5 years (60 months). It also shows that modifications of existing systems have a smaller range and lower median cycle time than an MDAP does. Random forest modeling identified the most important predictor variables: time since program start, change in the research and development budget, the natural log transforms of the research and development budget, and estimated unit cost.[11] We developed a multivariate regression, leaving out time since program start.[12] Table 1 summarizes the multivariate regression model that excluded time since program start.

Table 1. Cycle Time (Schedule) Regression Model Summary

Factor	Coefficient	Contribution	p-value	VIF
Intercept	1.6			
LN(R&D budget)	12.66	25.87%	0.000	1.13
PCT change (R&D)	37.74	20.32%	0.000	1.17
LN(Unit cost estimate)	4.97	6.75%	0.013	1.03
MTA = TRUE	-39.28	11.94%	0.000	1.03
S	**R-sq**	**R-sq(adj)**		**R-sq(pred)**
34.74	64.88%	61.90%		51.23%

We used a reduced version of this model as one estimator of Middle Tier Acquisition program schedule risk.

Predicting Schedule Risk for Middle Tier Acquisition Programs

We fitted polynomials to GAO cycle time (schedule) cumulative distributions for the four different program types.[13] These provide a simple estimate of schedule risk following the Figure

[11] If not specified, then unit cost was total budget divided by planned buy.

[12] This was evaluated as reflecting the schedule growth of older programs in the dataset.

[13] The cumulative distributions could be modeled by a 3 parameter Weibull distribution. We developed algebraic models to allow schedule risk estimation using simpler tools.

1 approach. Most polynomials were quadratic or cubic, with R-squared values above 95%. The Middle Tier Acquisition distribution was best fit by a logarithmic equation:

$$P(>D) = 1 - \{0.4284 * \ln(D) - 0.8496\}, \quad (1)$$

where D is the schedule duration in months.

We ran three Monte Carlo simulations to estimate the likelihood of Middle Tier Acquisition programs exceeding 60 months. The first simulation (*normal data fit*) assumed normally distributed schedule durations with estimated upper (53 months) and lower (10 months) bounds and an estimated standard deviation. The second simulation (*regression model*) started with a regression model predicting schedule distributions from program budgets.[14] The third simulation (*Weibull data fit*) assumed schedules followed a Weibull distribution, with scale (49.7), shape (2.705), and a 0 threshold. Table 2 shows Middle Tier Acquisition program duration simulation schedule risk results at 5, 6, 7, and 8 years (60–96 months).

Table 2. Middle Tier Acquisition Schedule Risk Simulation Results

Simulation/Model	P(>60 months)	P(>72 months)	P(>84 months)	P(>96 months)
Normal data fit	2.7%	0.4%	0.19%	0%
Regression model	46.6%	25.4%	11.5%	4.3%
Weibull data fit	19.8%	7.2%	1.6%	0.2%
poly fit	9.6%	1.7%	0%	0%

The last row (*poly fit*) is the estimate for the GAO cumulative schedule distribution for comparison with the simulations. The normal data fit simulation estimated a duration less than zero 2.3% of the time; 2.7% is the schedule risk of duration exceeding 60 months. The other models predict some risk of exceeding 60 months. Regression model results suggest that larger budget (greater than $1 billion) programs will be more likely to exceed 60 months. No Middle Tier model or simulation predicted significant schedule risk beyond 96 months.

We modeled MDAP schedule risk as above. Table 3 summarizes MDAP schedule risk for the GAO data and a polynomial model.

Table 3. MDAP Schedule Risk Simulation Results

Simulation/Model	P(>60 months)	P(>120 months)
Monte Carlo simulation	88.7%	46.7%
Polynomial model	88%	46%

This shows that Middle Tier Acquisition (MTA) programs have less schedule risk than MDAPs at the same *absolute* schedule duration. However, the *schedule risk* of an MTA and an MDAP at the *same relative percent schedule completion are comparable.*

Conclusions and Future Work

The literature review provided an overview of program better practices and decisions associated with shorter schedules, including

- Reducing requirements to meet capability and deliver something sooner

---

[14] The regression model had a predicted R-squared of about 30%.

- Starting with proven technologies, interfaces, and standards
- Having a competent team and capable suppliers
- Adjusting work to retire schedule risk and segmenting integration risk
- Having a plan to get to contract award and production sooner

Middle Tier Acquisition programs can benefit from these practices and add policy limits on schedule durations, oversight, and stakeholder involvement to incentivize schedule adherence and lower schedule risk.

This research provides insight into the schedule risk of MTA programs. Simulations showed the likelihood that an MTA program will exceed its planned schedule is less than 20%. Simulations also predict that the schedule risk for MTA programs grows with larger total research and development budgets (MTAs with budgets greater than $1 billion are more likely to exceed 60 months).

Middle Tier Acquisition programs have less schedule risk than traditional MDAPs at the same absolute duration. For example, at 60 months after program start, the schedule risk for an MTA program should be less than 20% and over 80% for an MDAP, as the MDAP has years of schedule left, while the MTA is (should be) nearly completed.

The schedule risk estimation process is extensible to other rapid acquisition innovations. Future work should include replicating this effort using restricted datasets and program-level data, validating schedule risk predictions and predictor significance with observed program performance, and developing context and severity estimators.

References

Adams, L. (2017). *Agile software development in the Department of Defense environment* (Senior Service College Fellowship Report, p. 61). Defense Acquisition University. https://www.dau.edu/training/career-development/sscf/Documents/Adams_SRP%20Final%20Draft_29%20Mar%2017--Edited%20and%20Corrected%20Final.pdf

Airbus. (2020). *How is an aircraft built*. https://www.airbus.com/aircraft/how-is-an-aircraft-built.html

Arellano, R. L., Pringle, R. G., & Sowell, K. L. (2015). *Analysis of rapid acquisition processes to fulfill future urgent needs* [Master's thesis, Naval Postgraduate School]. https://apps.dtic.mil/dtic/tr/fulltext/u2/1009069.pdf

Asghar, E., u. Zaman, U. K., Baqai, A. A., & Homri, L. (2018). Optimum machine capabilities for reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *95*(9–12), 4397–4417. https://doi.org/10.1007/s00170-017-1560-y

Baldwin, C. Y., & Henkel, J. (2015). Modularity and intellectual property protection. *Strategic Management Journal*, *36*(11), 1637–1655. https://doi.org/10.1002/smj.2303

Boeing. (2020). *Everett production facility* [Corporate]. https://www.boeing.com/company/about-bca/everett-production-facility.page

Bone, M. A., Blackburn, M. R., Rhodes, D. H., Cohen, D. N., & Guerrero, J. A. (2019). Transforming systems engineering through digital engineering. *The Journal of Defense Modeling and Simulation*, *16*(4). https://doi.org/10.1177/1548512917751873

Browning, T. R. (1998). 2.2.2 Sources of Schedule Risk in Complex System Development. *INCOSE International Symposium*, *8*(1), 213–220. https://doi.org/10.1002/j.2334-5837.1998.tb00031.x

Cai, D., Agarwal, A., & Wierman, A. (2020). On the inefficiency of forward markets in leader–follower competition. *Operations Research*, *68*(1), 35–52. https://doi.org/10.1287/opre.2019.1863
Chadha, C., Crowe, K. A., Carmen, C. L., & Patterson, A. E. (2018). Exploring an AM-enabled combination-of-functions approach for modular product design. *Designs*, *2*(4), 37. https://doi.org/10.3390/designs2040037
Cooper, R. G., & Sommer, A. F. (2016). The Agile–Stage-Gate hybrid model: A promising new approach and a new research opportunity. *The Journal of Product Innovation Management*, *33*(5), 513–526. https://doi.org/10.1111/jpim.12314
Davis, G. A., & Tate, D. M. (2020). Complexity in an unexpected place: Quantities in selected acquisition reports. *Defense Acquisition Research Journal*, *27*(1), 28–59. https://doi.org/10.22594/dau.19-832.27.01
Defense Innovation Unit. (2020). *Defense Innovation Unit*. https://www.diu.mil
Desai, P. S., Koenigsberg, O., & Purohit, D. (2007). Research note—The role of production lead time and demand uncertainty in marketing durable goods. *Management Science*, *53*(1), 150–158. https://doi.org/10.1287/mnsc.1060.0599
Dougherty, G. M. (2018). Promoting disruptive military innovation: Best practices for DoD experimentation and prototyping programs. *Defense Acquisition Research Journal*, *25*(1), 2–29.
Drabkin, D., Ahern, D., Branch, E., Dyer, J., Hoskin, M., Merchant, K. D., Trowel, L., Raney, T. L., Thompson, N. R., Blake, C. D., Burman, A. V., Garman, C. D., LaPlante, W. A., Metzger, D. P., & Scott, D. A. (2016). *Section 809 panel*. Defense Technical Information Center. https://discover.dtic.mil/section-809-panel/
Dubos, G., Saleh, J., & Braun, R. (2007, September 18). *Technology readiness level, schedule risk and slippage in spacecraft design: Data analysis and modeling*. https://doi.org/10.2514/6.2007-6020
Ellis, W., Molloy, M., Cohen, J., & Crozier, P. (2019). *Contracting for Agile*. KPMG. https://assets.kpmg/content/dam/kpmg/uk/pdf/2019/08/contracting-for-agile.pdf
Fox, J. R. (2011). *Defense acquisition reform 1960–2009: An elusive goal* (GOVDOC: D 114.2:AC 7). Center of Military History, U.S. Army. https://permanent.fdlp.gov/gpo27025/CMH_Pub_51-3-1.pdf
Haberfellner, R., & Weck, O. (2005). 10.1.3 Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering. *INCOSE International Symposium*, *15*(1), 1449–1465. https://doi.org/10.1002/j.2334-5837.2005.tb00762.x
Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, *51*(Part B), 915–929. https://doi.org/10.1016/j.chb.2014.10.046
Ingold, D. (2014). *A model for estimating schedule acceleration in agile software development projects* [Doctoral dissertation, University of Southern California]. https://search.proquest.com/docview/2066832925?pq-origsite=summon
Islam, G., & Storer, T. (2020). A case study of agile software development for safety-critical systems projects. *Reliability Engineering & System Safety*, *200*, 106954–18. https://doi.org/10.1016/j.ress.2020.106954
Jahr, M. (2014). A hybrid approach to quantitative software project scheduling within agile frameworks. *Project Management Journal*, *45*(3), 35–45. https://doi.org/10.1002/pmj.21411
Jaifer, R., Beauregard, Y., & Bhuiyan, N. (2020). New framework for effort and time drivers in aerospace product development projects. *Engineering Management Journal*, 1–20. https://doi.org/10.1080/10429247.2020.1772950
Lord, E. M. (2019). *Operation of the middle tier of acquisition (MTA)* (DoD Instruction 5000.80). DoD.

https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500080p.PDF?ver=2019-12-30-095246-043

Lord, E. M. (2020a). *The defense acquisition system* (DoD Directive 5000.01). DoD. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodd/500001p.pdf?ver=2020-09-09-160307-310

Lord, E. M. (2020b). *Operation of the adaptive acquisition framework* (DoD Instruction 5000.02). DoD. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500002p.pdf?ver=2019-10-25-134150-283

MITRE. (2019). *AiDA- MITRE acquisition in the digital age*. https://aida.mitre.org/middle-tier/

Mortlock, R. F. (2019). *Acquisition strategy formulation: Evolutionary/incremental development approach* (NPS-PM-19-179). Naval Postgraduate School.

National Defense Authorization Act for Fiscal Year 1996, Pub. L. No. 104–106, 110 Stat. 186, Division E (1996). https://www.govinfo.gov/content/pkg/PLAW-104publ106/html/PLAW-104publ106.htm

National Defense Authorization Act for Fiscal Year 2011, Pub. L. No. 111–383, 124 Stat. 4137, 2359a 10 (2011). https://www.law.cornell.edu/uscode/text/10/2359a

National Defense Authorization Act for Fiscal Year 2016, Pub. L. No. 114–92, 129 Stat. 726 (2015), S.1356. https://www.congress.gov/114/plaws/publ92/PLAW-114publ92.pdf

National Defense Authorization Act for Fiscal Year 2017, Pub. L. No. 114–328, 130 Stat. 2000, 10 970 (2016). https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title10-section2446a&num=0&edition=prelim

Nidiffer, K. E., Miller, S. M., & Carney, D. (2014). *Potential use of agile methods in selected DoD acquisitions: Requirements development and management* (Technical Note CMU/SEI-2013-TN-006). Carnegie Mellon University. https://apps.dtic.mil/dtic/tr/fulltext/u2/a609864.pdf

Oestergaard, J. K. (2020, September 29). *Airbus and Boeing report December and full-year 2019 commercial aircraft orders and deliveries*. Defense & Security Monitor. https://dsm.forecastinternational.com/wordpress/2020/01/21/airbus-and-boeing-report-december-and-full-year-2019-commercial-aircraft-orders-and-deliveries/

Office of Management and Budget. (2012). *Contracting guidance to support modular development.* https://obamawhitehouse.archives.gov/sites/default/files/omb/procurement/guidance/modular-approaches-for-information-technology.pdf

Office of the Under Secretary of Defense for Acquisition and Sustainment. (2019). *Contracting considerations for Agile solutions: Key agile concepts and sample work statement language* (Ver. 1.0). Department of Defense. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Contracting%20Considerations%20for%20Agile%20Solutions%20v1.0.pdf

Office of the Under Secretary of Defense for Acquisition and Sustainment. (2020, February 27). *Agile software acquisition guidebook: Best practices & lessons learned from the FY18 NDAA Section 873/874 Agile pilot program* (Ver. 1.0). DoD. https://www.dau.edu/cop/it/_layouts/15/WopiFrame.aspx?sourcedoc=/cop/it/DAU%20Sponsored%20Documents/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf&action=default

Office of the Under Secretary of Defense for Research and Engineering. (2020, August). *Defense innovation marketplace: Connecting industry and the Department of Defense*. Defense Innovation Marketplace. https://defenseinnovationmarketplace.dtic.mil/business-opportunities/rapid-innovation-fund/

Pennington, R. (2018). Agile: A new frontier for contract management. *Contract Management*, *58*(08), 28–37.

Procurement for Experimental Purposes, 10 U.S.C. § 2373, Pub. L. No. 113-160, 2372 10 (1993).
Reim, G. (2020, September 15). *USAF secretly builds and flies next-generation fighter demonstrator*. FlightGlobal. https://www.flightglobal.com/fixed-wing/usaf-secretly-builds-and-flies-next-generation-fighter-demonstrator/140183.article
Research Projects: Transactions Other Than Contracts and Grants, 10 U.S.C. § 2371, Pub. L. No. 103-160, 2371 10 (1993).
Riposo, J., McKernan, M., & Duran, C. K. (2014). *Prolonged cycle times and schedule growth in defense acquisition: A literature review*. RAND.
Rosa, W., Madachy, R., Clark, B., & Boehm, B. (2017). Early phase cost models for agile software processes in the US DoD. *ESEM*, 30–37. https://doi.org/10.1109/ESEM.2017.10
Schoeni, D. E. (2015). Long on rhetoric, short on results: Agile methods and cyber acquisitions in the Department of Defense. *Santa Clara High-Technology Law Journal*, *31*(3), 385.
Tao, L., Wu, D., Liu, S., & Lambert, J. H. (2017). Schedule risk analysis for new-product development: The GERT method extended by a characteristic function. *Reliability Engineering & System Safety*, *167*, 464–473. https://doi.org/10.1016/j.ress.2017.06.010
Tate, D. M. (2016). *Acquisition cycle time: Defining the problem* (Revised; IDA Document NS D-5762). Institute for Defense Analyses. https://www.jstor.org/stable/resrep22689?seq=1#metadata_info_tab_contents
Thomas, D., Kunc, W., Barth, S., Smart, C., Watern, K., & Comstock, D. (2014). *Joint agency cost schedule risk and uncertainty handbook*. Department of the Navy.
Van Atta, R. H., Kneece, R. R., & Lippitz, M. J. (2016). *Assessment of accelerated acquisition of defense programs* (IDA Paper P-8161). Institute for Defense Analyses. https://apps.dtic.mil/dtic/tr/fulltext/u2/1028525.pdf
van Gent, I., & Kassapoglou, C. (2014). Cost-weight trades for modular composite structures. *Structural and Multidisciplinary Optimization*, *49*(6), 931–952. https://doi.org/10.1007/s00158-013-1019-1
Waldron, G. (2019, October 4). *USAF sets up advanced fighter technology office*. FlightGlobal. https://www.flightglobal.com/defence/usaf-sets-up-advanced-fighter-technology-office/134672.article
Watson, J. D., Allen, J. D., Mattson, C. A., & Ferguson, S. M. (2016). Optimization of excess system capability for increased evolvability. *Structural and Multidisciplinary Optimization*, *53*, 177–1294. https://doi.org/10.1007/s00158-015-1378-x
Wauters, M., & Vanhoucke, M. (2017). A nearest neighbour extension to project duration forecasting with artificial intelligence. *European Journal of Operational Research*, *259*(3), 1097–1111. https://doi.org/10.1016/j.ejor.2016.11.018
Wicecarver, J. M. (2017). *Joint Requirements Oversight Council procurement quantity validation process for major defense acquisition programs* (DODIG-2017-117). Department of Defense. https://media.defense.gov/2017/Sep/11/2001806280/-1/-1/1/DODIG-2017-117.PDF
Wong, J. P. (2016). *Balancing immediate and long-term defense investments.* RAND. https://www.rand.org/pubs/rgs_dissertations/RGSD378.html
Zhai, Y., Zhong, R. Y., Li, Z., & Huang, G. (2016). Production lead-time hedging and coordination in prefabricated construction supply chain management. *International Journal of Production Research*, *55*(14), 3984–4002. https://doi.org/10.1080/00207543.2016.1231432

Two-Stage, Dynamic Data Envelopment Analysis of Technology Transition Performance in the U.S. Defense Sector

Dr. Toshiyuki Sueyoshi—is a Full Professor at New Mexico Tech and is an internationally well-known scholar in the field of data envelopment analysis. He has published more than 300 articles in the top-tier journals such as *European Journal of Operational Research*, *Decision Support Systems*, *Energy Policy*, *Energy Economics*, and *Applied Energy*. Before joining New Mexico Tech, he was a Full Professor at Tokyo Institute of Technology and an Assistant and Associate Professor at the Ohio State University. [toshiyuki.sueyoshi@nmt.edu]

Dr. Youngbok Ryu—is an Assistant Teaching Professor at Northeastern University. Before joining Northeastern University, he worked with academic and research institutions in the United States and South Korea. His research has appeared in journals such as *Energies*, *STI Policy Review*, *Journal of Conservation Planning*, and *RAND Health Quarterly*, as well as in research reports published by RAND Corporation and the National Academies of Sciences, Engineering, and Medicine. [y.ryu@northeastern.edu]

Abstract

This study aims to measure the technology transition performance of 252 small firms that won the U.S. Department of Defense (DoD) Small Business Innovation Research (SBIR) Phase II awards from 2001 to 2010 and filed more than 15 patents ("elite DoD SBIR awardees") and to explore how social, industrial, and geospatial contexts influence the performance. For the purpose, we first employ two-stage, dynamic Data Envelopment Analysis (DEA) that incorporates network building sub-process as well as R&D and commercialization sub-processes and then utilize Tobit regression analysis. We find two implications. One of them is that more than a quarter of the elite DoD SBIR awardees are efficient and their efficiency scores of about a half are higher than 60%. The other is that their strong networks with big-sized funders and their high-tech concentration are positively associated with the technology transition performance whereas locational factors are not significantly related with the performance.

Introduction

As a complex system, regional, industrial, or national innovation system involves many players who are interdependent with each other (Dougherty, 2017; Katz, 2016). Generally, the public sector (e.g., federal agencies with a substantial amount of extramural R&D budget) and non-profit organizations (e.g., private foundations) provide R&D funding to knowledge producers or technology developers. Universities and national and corporate research laboratories produce knowledge or develop technology depending on the funding. The private sector (e.g., small businesses) capitalizes on the produced and transferred knowledge and makes revenues and profits that are sources of investment and incentives to the players. While the segmentation of role responsibilities worked well in the era of the public sector-dominant R&D (particularly, in the wartime), the boundaries that were drawn for each player have been blurred (Kaufmann & Tödtling, 2001; Lundberg, 2013). For instance, the Department of Defense (DoD) develops technologies in house through Air Force, Army, and Naval research laboratories, outsources high-risk, high-return R&D projects to universities or corporations through the Defense Advanced Research Projects Agency (DARPA), and acquires state-of-the-art technologies from the private sector through the Defense Innovation Unit (DIU) and Small Business Innovation Research (SBIR) and Small Business Technology Transfer (STTR) programs. It stresses not only technology transfer (e.g., a knowledge flow from the DoD to small businesses) but also technology transition (e.g., a knowledge flow from small businesses to the DoD). It is especially true at the current times when the private sector is leading in many high-tech areas such as information and communications technology and biotechnology (Ryu, 2017). Nonetheless, most literature has paid attention to the former (particularly, technology transfer from universities or national laboratories to the private sector) whereas a paucity of literature has studied the latter (particularly, technology transition from small businesses to the public sector).

Innovation itself is also complex in that it entails various processes and sub-processes at multiple levels (Dias et al., 2014; Pelz, 1985). At the product level, for instance, it should pass along a series of product development processes from ideation to prototyping to testing in order to create a new product. To sell the new product, it should get through a chain of value-added processes from inbound logistics to operations to outbound logistics to marketing and sales to service. Innovation is affected by other upper-level characteristics (Autio et al., 2014; Tidd, 2001). For instance, organizational culture and institutions (e.g., incentive system) at the firm level, state government's business environment (e.g., tax credits and support for the public universities) at the regional level, and federal government's policy programs (e.g., R&D spending and procurement) at the national level can influence innovation. Given that innovation is affected by various contexts, policy-makers are responsible to reconstruct the contexts to attract more innovators to their jurisdictions and to grow their economies. Innovators (particularly, technology-based small businesses), on the other hand, tend to search better locations to operate their businesses considering not only government's support but also other factors (e.g., access to market and benefits resulted from agglomeration). In this vein, decision-makers who seek technological innovation in their organizations or jurisdictions may need to evaluate their current performance and find some entities to benchmark in order to improve their performance (Guan et al., 2006; Sun, 2011). That is why Data Envelopment Analysis (DEA) has been widely used to measure the innovation performance and identify leading entities at the firm-, regional-, and national levels.

With this background, this study seeks to (a) assess technology transition performance (as a special case of innovation) of technology-based small businesses that received the DoD SBIR awards, and (b) examine how social, industrial, and geospatial contexts influence the performance. For the purpose, we first employ two-stage, dynamic DEA to deal with the sub-processes of innovation in measuring technology transition performance and then Tobit regression analysis to determine what factors drive the performance.

The remaining sections of this research are organized as follows. The Literature Review section discusses a literature study. The Methodology section describes the methodology used in this study. The Empirical Study section summarizes empirical results. The last section concludes this research along with summary and future research.

Literature Review

DEA Applications to Measuring Innovation Efficiency

DEA, to date, has been utilized to measure innovation efficiency at multiple levels such as organization (e.g., firm and university), region, and country. At the microscopic level, for instance, Sueyoshi & Goto (2013) examined the firm-level efficiency by associating R&D expenditure with Tobin's q as an indicator for the corporate value. Kuah & Wong (2011) measured the university-level R&D efficiency by using multiple inputs (e.g., research grants and the numbers of research staff and students) and outputs (e.g., the numbers of publications, awards, and intellectual property rights). At the mesoscopic level, Zabala-Iturriagagoitia (2007) looked at the performance of regional innovation systems based on the European Innovation Scoreboard (EIS) data. At the macroscopic level, Sharma & Thomas (2008) explored the cross-national R&D efficiency of 22 countries. While they offer useful information about which entities lead in terms of the innovation performance and which entities can serve as benchmarks, they have some limitations: (a) they tend to regard the innovation process as a single big black box (and thus it is hard to understand the specifics of the innovation and to incorporate the time-consuming nature of the innovation), (b) although they seek to shed light on the specifics, they tend to focus on one specific segment of the whole value chain of innovation (and thus it is difficult to understand the interdependence between the innovation sub-processes), (c) they tend to stress only traditional production factors such as labor and capital materialized by financial resources (R&D personnel and expenditure in our case) ignoring the aspects of social capital, and (d) they tend to be general (particularly in the regional and country-level studies) and thus their frameworks need to be tailored to solving specific program-related performance measurement issues.

To address the aforementioned limitations (a) and (b), multi-stage and/or dynamic network DEA approaches have been recently developed and applied to measure the innovation efficiency. At the firm-level, for instance, Chun et al. (2015) and Wang et al. (2016) used two-stage DEA models by decomposing the innovation process into R&D and marketing/commercialization sub-processes and analyzed the innovation efficiency of Korean and Chinese companies. At the regional level, Chen and Guan (2012) and Chen et al. (2018) looked at the innovation efficiency of Chinese regions based on two-stage network DEA models. At the national level, Carayannis et al. (2016) and Kou et al. (2016) measured the innovation efficiency of European or OECD countries. All of these studies incorporate the systematic and dynamic aspects of the complex and non-linear innovation process. Along with the emerging concepts of Regional Innovation System (RIS) as well as the National Innovation System (NIS), those studies can better inform regional- and national-level policy-makers who desire to invigorate the economies of their jurisdictions through technological innovation. While achieving the desired result in the limitations (a) and (b), they are not still sufficient to address the remaining limitations (c) and (d).

As the position of this study is to fill this gap in the existing literature, we seek to incorporate the social capital dimensions into a two-stage, dynamic DEA model. Moreover, we seek to customize a general DEA model into addressing the technology transition issue, a special case of the innovation process but one of the objectives that the DoD SBIR program desires to achieve. Through the proposed DEA model, thus, we can better inform technology transition-related policy-makers and DoD SBIR program managers.

Role of Social Capital in Small Businesses' Innovation Context

There is a great body of studies on the corporate size of businesses. Most of them argued over the advantages and disadvantages of business size (e.g., MacMillan, 1975; Moen, 1999). For instance, large businesses can be price-competitive by reducing average cost through the economies of scale and also be technology- and market-competitive by investing more in R&D and marketing whereas they can suffer from bureaucracy, ineffective communication, and concerns about cannibalization or creative destruction. On the other hand, small businesses can take advantage of their flexibility, agility, and risk-taking innovation while they have to face many challenges such as lack of well-educated workers and well-secured financial resources and limited access to valuable information. One of the solutions to those challenges may be developing social networks (e.g., Miller et al., 2007; Lee, 2015). To hire qualified employees or to obtain useful information, for instance, small firms or their founders may be able to utilize their social networks (from strong and informal ties such as family and friends to weak and formal ties such as professional communities). They can also build social networks with potential funders to secure external financial sources. Particularly for startups and early-stage small businesses, securing funds through public venture programs such as federal-level Small Business Innovation Research (SBIR) grants/contracts and state-level SBIR matching grants and private equity such as investments from corporate venture capitals and angels are critical for their survival and growth.

Particularly, the SBIR program has contributed to facilitating the innovation of small businesses. Since the 1980s, the program has successfully attained its four objectives: 1) stimulating technological innovation, 2) using small businesses to meet federal R&D needs, 3) fostering and encouraging participation by minority and disadvantaged people in technological innovation, and 4) increasing private-sector commercialization (Small Business Administration [SBA], 2014). Of them, the second objective is especially important to the DoD that accounts for almost half of the total SBIR budget. Unlike the past where the public sector (e.g., national laboratories) dominated technological innovation, the private sector is currently leading in the high-tech innovation (e.g., information and communications technology and biotechnology). It is essential for the DoD to acquire cutting-edge technologies from the private sector in order to maintain its military and technological leadership. In this regard, we define technology transition as a knowledge/technology flow from the private sector to the public sector (when compared to technology transfer indicating a knowledge/technology flow from the public sector to the private sector) following Dobbins's (2004) definition about technology transition: "the process by which technology deemed to be of significant use to the operational military community is transitioned from the science and technology environment to a military operational field unit for evaluation and then incorporated into an existing acquisition program or identified as the subject matter for a new acquisition program" (p. 14).

Considering both the importance of social networks among small business communities and the technology acquisition purpose of the DoD SBIR program, we include the technological distance from the DoD as one of the input variables and the number of small firms' connections to funders (captured by their eigenvector centrality in the SBIR funding network) as one of the intermediate variables. The rationale is that the DoD may seek novel technologies that are different from ones in its technology portfolio through the SBIR program. The technological distance measures the degree of dissimilarity between small firms' and the DoD's technological portfolios based on their patent distributions across the patent classification codes (e.g., Bar & Leiponen, 2012; Benner & Waldfogel, 2008). Our underlying concept is that the higher technological distance may lead to the more (or stronger) connections to the SBIR funders. To better reflect the reality of the SBIR budget allocation (where three services such as Air Force, Army, and Navy take a lion's share of the DoD SBIR budget), we use the eigenvector centrality (which counts the number of connections differently by placing more weights on the connections to big-sized funders and less weights on the connections to small-sized funders) instead of degree centrality (which counts the number of connections equally by placing the same weights on all connections). See Bonacich (2007) and Faulk et al. (2017).

Methodology

Primary

Nomenclatures used in this study are summarized as follows:

x_{ijt} is the observed *i* th input of the *j* th DMU (*i* = 1, …, *m* & *j* = 1, ..., *n*) at the *t* th stage, g_{rjt} is the observed *r* th output of the *j* th DMU (*r* = 1, ..., *s* & *j* = 1, ..., *n*) at the *t+1* th stage, y_{hjt} is the observed *h* th intermediate output of the *j* th DMU (*h* = 1, …, *z* & *j* = 1, ..., *n*) at the *t* th stage, y_{rjt+1} is the observed *h* th intermediate input of the *j* th DMU (*r* = 1, ..., *s* & *j* = 1, ..., *n*) at the *t+1* th stage, ξ is an inefficiency measure, d_{it}^{x} is an unknown slack variable of the *i* th input at the *t* th stage, d_{rt}^{g} is an unknown slack variable of the *r* th output at the *t* th stage, d_{ht}^{y} is an unknown slack variable of the *h* th intermediate output/input at the *t* and *t+1* th stages, λ_{jt} is an unknown intensity (or structural) variable of the *j* th DMU at the *t* th stage, ε_s is a prescribed very small number and J_t is a set of all DMUs at the *t* th stage. This study considers the first and second stage, so *t* = 1 and 2.

Before applying the proposed formulations, we need to specify the following data ranges on *X* (inputs) and *G* (outputs):

R_i^x is a data range on the *i* th input which is specified as

$$R_i^x = (m+s)^{-1}\left(\max_{jt}\{x_{ijt}\mid all\ j\ \&\ all\ t\} - \min_{jt}\{x_{ijt}\mid all\ j\ \&\ all\ t\}\right)^{-1} \quad (1)$$

R_r^g is a data range on the *r* th desirable output which is specified as

$$R_r^g = (m+s)^{-1}\left(\max_{jt}\{g_{rjt}\mid all\ j\ \&\ all\ t\} - \min_{jt}\{g_{rjt}\mid all\ j\ \&\ all\ t\}\right)^{-1}. \quad (2)$$

The data ranges are applied to the all DMUs (*j = 1, .., n*) in all periods (*t = 1, .., z*) in the proposed DEA models. The purpose of these data ranges is that DEA results can avoid an occurrence of zero in dual variables (i.e., multipliers). Such an occurrence implies that corresponding production factors (*X* and *G*) are not fully utilized in our DEA applications. Such an occurrence is problematic. To avoid the difficulty, this study incorporates the data ranges, (1) and (2), into the proposed formulations so that we can fully utilize available information on the two production factors.

Operational Efficiency Measurement

This research considers the operational performance of various entities. Each entity is considered as a DMU. In every DMU, the production technology transforms an input vector with m components ($X \in R_+^m$) into a desirable output vector with s components ($G \in R_+^s$).

The axiomatic form of Production Technology (PT) on a production possibility set (P) is expressed at the specific *t* th period as follows:

$$P_t = \{P_t(X) : X_t \text{ can produce } G_t\} \in \mathrm{R}_+^m \quad (3)$$

within the framework of (3), the production possibility set at the *t* th period can be expressed as follows:

$$P_t^v(X_t) = \left\{G_t \le \sum_{j=1}^{n} G_{jt}\lambda_{jt}, X_t \ge \sum_{j=1}^{n} X_{jt}\lambda_{jt}, \sum_{j=1}^{n}\lambda_{jt} = 1\ \&\ \lambda_{jt} \ge 0\ (j=1,\ldots,n)\right\} \quad (4)$$

The expressions on *P* incorporate the assumption on variable Returns to Scale (RTS). Equation 4 incorporates the variable (v) RTS because Equation 4 has the side constraint (i.e., $\sum_{j=1}^{n}\lambda_{jt} = 1$). See Sueyoshi & Goto (2018) for a detailed mathematical description on RTS.

Figure 1 depicts the contract assessment used in this study. The contract consists of two staged processes. There is an intermediate stage that connects between them by considering a "time lag." The first stage uses (a) the amount of SBIR awards, (b) the number of employees, and (c) the level of technology distance. There are two outputs: (a) the number of patents and the number of connections at the intermediate stage. After 5 years, the two outputs from the first stage serve as inputs at the second stage. The final output is the amount of federal procurement contract. Since the Empirical Study section provides a detailed description on inputs and outputs, this section focuses upon a description on the methodology.

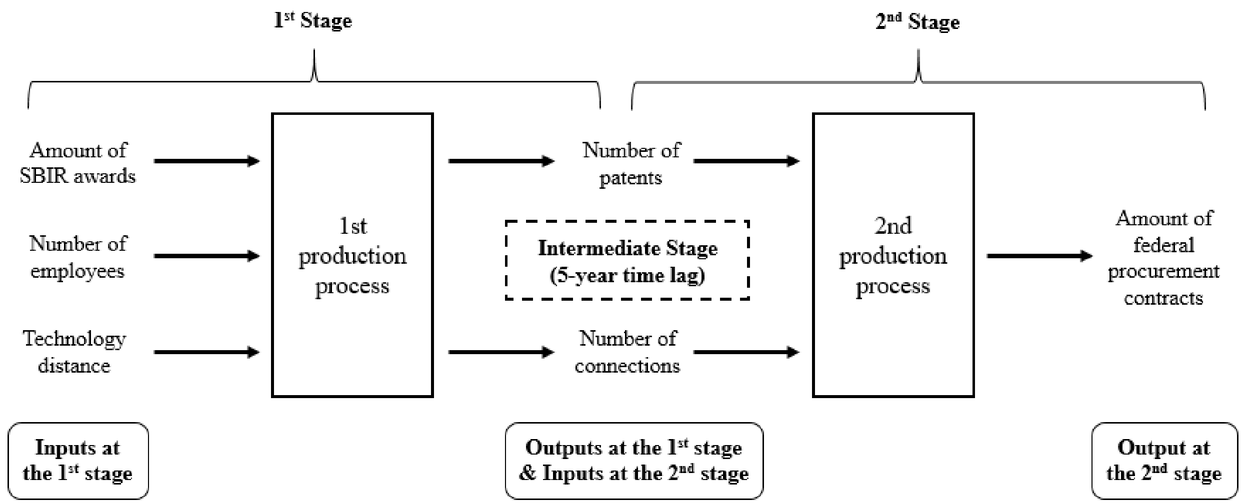


Figure 1. Two-Stage Analytic Framework

Note: (a) we collected data from various sources: (i) SBIR awards data from the Small Business Administration's (SBA) SBIR database, (ii) employment data from the System for Award Management (SAM), (iii) patent data from the Korea Intellectual Property Rights Information Service (KIPRIS, original data imported from the U.S. Patent and Trademark Office), and (ix) federal procurement contracts data from the Federal Procurement Data System-Next Generation (FPDS-NG). (b) There are three inputs at the first stage: the amount of SBIR awards (*ASA*), the number of employees (*EMP*), and technological distance from the DoD (*TDD*). Both *ASA* and *EMP* are measured at US$ million and full-time equivalent (*FTE*), respectively, while *TDD* is valued between 0 and 1. There are two variables at the intermediate stage (which act as stage-1 outputs and stage-2 inputs simultaneously): the number of patents (*PAT*) and the number of connections (*SEC*). The former is measured as the number of patent applications and the latter is valued between 0 and 1 and measured as the eigenvector centrality.[1] There is also one output at the second stage: the amount of federal procurement contracts (*FPC*). FPC is measured at US$ million. (c) It is also worth noting that there is a 5-year time lag between inputs and outputs. Given that technology transition takes a substantial amount of time.

To analyze the procurement process specified in Figure 1, this study uses a DEA-based radial approach to determine the level of Operational Efficiency (OE) on the specific *k* th DMU at the *t* th period. Given X_{kt} and G_{kt+1}, we evaluate the performance of the *k* th DMU to be examined. The subscription (j_t) is used to express each DMU ($j = 1, .., n$) in the total set (J_t).

Based upon the framework of Figure 1, this study proposes the following formulation to measure the level of Operational Efficiency (OE_{kt}^{v}) on the *k* th DMU at the *t* th period:

[1] There are several ways to calculate the number of connections in a network. For instance, degree centrality focuses on the absolute number of links while closeness centrality pays attention to the distance between nodes (that is why closeness centrality is often used for the analysis of information spread) and betweenness centrality stresses the path of pairs of nodes (that is why betweenness centrality is used for identifying brokers or intermediaries who can connect two different groups). In this study, we use eigenvector centrality because we believe the connections of small firms to big-sized funders are different from those to small-sized funders. We put more weight on the former connections. From a firm's perspective, for instance, connections to the Air Force, Army, or Navy may be more valuable than those to the Defense Logistics Agency or Missile Defense Agency.

$$
\begin{aligned}
&Maximize\ \ \xi+\varepsilon_s\Big(\sum_{i=1}^{m} R_i^x d_{it}^x+\sum_{r=1}^{s} R_r^g d_{rt+1}^g\Big)-\sum_{h=1}^{z}\big(d_{ht}^y+d_{ht+1}^y\big)\\
&s.t.\quad \sum_{j=1}^{n} x_{ijt}\lambda_{jt}+d_{it}^x+\xi x_{ikt}=x_{ikt}\quad (i=1,..,m),\\
&\sum_{j=1}^{n} y_{hjt}\lambda_{jt}-d_{ht}^y=y_{hkt}\quad (h=1,..,z),\\
&\sum_{j=1}^{n} y_{hjt+1}\lambda_{jt+1}+d_{ht+1}^y=y_{hkt+1}\quad (h=1,..,z),\\
&\sum_{j=1}^{n} g_{rjt+1}\lambda_{jt+1}-d_{rt+1}^g-\xi g_{rkt+1}=g_{rkt+1}\quad (r=1,..,s),\\
&\sum_{j=1}^{n}\lambda_{jt}=1\ (j=1,..,n),\\
&\sum_{j=1}^{n}\lambda_{jt+1}=1\ (j=1,..,n),\\
&\lambda_{jt}\ge 0\ (j=1,..,n), \lambda_{jt+1}\ge 0\ (j=1,..,n),\ \xi: URS,\ d_{it}^x\ge 0\ (i=1,..,m),\\
&d_{ht}^y\ge 0\ (h=1,..,z),\ d_{ht+1}^y\ge 0\ (h=1,..,z)\ \&\ d_{rt+1}^g\ge 0\ (r=1,..,s).
\end{aligned}
\tag{5}
$$

The superscript (v) of OE_{kt}^v indicates variable (v) RTS.

The degree of OE_{kt}^v is measured by

$$
OE_{kt}^v=1-\Big[\xi^*+\varepsilon_s\Big(\sum_{i=1}^{m} R_i^x d_{it}^{x*}+\sum_{r=1}^{s} R_{rt}^g d_{rt}^{g*}\Big)\Big],
\tag{6}
$$

where the inefficiency score and all slack variables are determined on the optimality (*) of Model 5. Thus, the equation within the parenthesis is obtained from the optimality of the objective value of Model 5. The OE_{kt}^v is obtained by subtracting the level of inefficiency from unity. If the degree of OE_{kt}^v is unity, then it indicates the status of "full efficiency." On the other hand, the degree is less than unity, it includes some level of "inefficiency." If the degree is zero, it indicates "full inefficiency."

Here, it is important to note four concerns related to Model 5. First, as formulated in Equation 6, the degree of OE_{kt}^v is measured by the first groups of constraints (i.e., $\sum_{j=1}^{n} x_{ijt}\lambda_{jt}+d_{it}^x+\xi x_{ikt}=x_{ikt}$) on inputs at the *t* th period and the fourth groups of constraints (i.e., $\sum_{j=1}^{n} g_{rjt+1}\lambda_{jt+1}-d_{rt+1}^g-\xi g_{rkt+1}=g_{rkt+1}$) on outputs at the *t+1* th period. Both are used to determine a degree of the inefficiency measure in the whole process for the two (*t* and *t+1*) periods. Next, the second group of constraints (i.e., $\sum_{j=1}^{n} y_{hjt}\lambda_{jt}-d_{ht}^y=y_{hkt}$) indicates that intermediate factors functions as outputs at the *t* th period so that the frontier (i.e., $\sum_{j=1}^{n} y_{hjt}\lambda_{jt}$) locates above or on their observed values (y_{hkt}) because of $y_{hkt}+d_{ht}^y$. The slacks (d_{ht}^y) is minimized in Model 5. Meanwhile, the third group of constraints (i.e.,

$\sum_{j=1}^{n} y_{hjt+1}\lambda_{jt+1} + d_{ht+1}^{y} = y_{hkt+1}$) indicates that intermediate factors functions as inputs at the *t+1* th period so that the frontier (i.e., $\sum_{j=1}^{n} y_{hjt+1}\lambda_{jt+1}$) locates below or on their observed values (y_{hkt+1}) because of $y_{hkt+1} - d_{ht+1}^{y}$. The slacks (d_{ht+1}^{y}) is minimized in Model 5. Third, the fifth and sixth constraints ($\sum_{j=1}^{n} \lambda_{jt} = 1$ and $\sum_{j=1}^{n} \lambda_{jt+1} = 1$) indicate that the sum of these intensities (weights) is unit at the *t* th and *t+1* th periods, respectively. Such constraints imply that the degree of OE_{kt}^{v} is measured under variable RTS. Finally, it is necessary to describe that we are interested in the performance between initial inputs (X_t) and final outputs (G_{t+1}). So, Equation 6 measures $OE_{kt}^{v} = 1 - [\xi^{*} + \varepsilon_{s}(\sum_{i=1}^{m} R_{i}^{x} d_{it}^{x*} + \sum_{r=1}^{s} R_{rt}^{g} d_{rt}^{g*})],$ The intermediate factors (Y_t and Y_{t+1}) make a linkage between the two stages. Figure 1 visually describes such relationship among the three groups of factors.

Empirical Study

Data

This study uses a data set on 252 small firms that meet two criteria: (a) filed more than 15 patents, and (b) awarded the SBIR Phase II funding from the DoD over the decade (from 2001 to 2010). The criteria evidence their R&D and network building capacities because of the three rationales. First, Hicks & Hegde (2005) have defined firms with more than 15 patents as "serial innovators." Second, the SBIR Phase II is followed by the successful completion of Phase I that focuses on the assessment of technical feasibility. The SBIR funding usually entails close relationships between funders and awardees. Finally, the DoD SBIR program selects and announces very specific SBIR topics that require technical fit with awardees. While 2,889 firms meet the second criterion, only 252 firms meet both criteria. Hereafter, we call the 252 firms "elite DoD SBIR awardees."

For the analysis in the frame of multiple inputs and outputs across two stages, we collected data from various sources: (a) SBIR awards data from the Small Business Administration's (SBA) SBIR database, (b) employment data from the System for Award Management (SAM), (c) patent data from the Korea Intellectual Property Rights Information Service (KIPRIS, original data imported from the U.S. Patent and Trademark Office), and (d) federal procurement contracts data from the Federal Procurement Data System-Next Generation (FPDS-NG).

There are three inputs at the first stage: the amount of SBIR awards (*ASA*), the number of employees (*EMP*), and technological distance from the DoD (*TDD*). Both *ASA* and *EMP* are measured at US$ million and full-time equivalent (*FTE*), respectively, while *TDD* is valued between 0 and 1. There are two variables at the intermediate stage (which act as stage-1 outputs and stage-2 inputs simultaneously): the number of patents (*PAT*) and the number of connections (*SEC*). See Figure 1. The former is measured as the number of patent applications and the latter is valued between 0 and 1 and measured as the eigenvector centrality.[2] There is

[2] There are several ways to calculate the number of connections in a network. For instance, degree centrality focuses on the absolute number of links while closeness centrality pays attention to the distance between nodes (that is why closeness centrality is often used for the analysis of information spread) and betweenness centrality stresses the path of pairs of nodes (that is why betweenness centrality is used for identifying brokers or intermediaries who can connect two different groups). In this study, we use eigenvector centrality because we believe the connections of small firms to big-sized funders are different from those to small-sized funders. We put more weight on the former connections. From a firm's perspective, for instance, connections to the Air Force, Army, or Navy may be more valuable than those to the Defense Logistics Agency or Missile Defense Agency.

also one output at the second stage: the amount of federal procurement contracts (*FPC*). FPC is measured at US$ million.

It is also worth noting that there is a 5-year time lag between inputs and outputs. Given that technology transition takes a substantial amount of time, we collected input-related data as of 2010 and output-related data as of 2015. This approach has two advantages: (a) reflect more realistic conditions, and (b) avoid the endogeneity issue in the analysis. There are some studies supporting this. NASEM (2009, p. 230), for instance, showed a table describing the time elapsed between SBIR awards (R&D) and actual sales (commercialization), which tends to be 5–7 years. Xue & Klein (2010) also used a 5-year time lag between independent and dependent variables related to entrepreneurial activities. Seegopaul (2016) explored the time required for development/commercialization by industry (e.g., 0–2 years for software and 5–15 years for advanced materials).

Table 1 presents a summary of data descriptive statistics for DEA. The table shows detailed data and descriptive statistics of inputs and outputs at the first, intermediate, and second stages. Companies are listed in the alphabetical order of their names. On average, elite DoD SBIR awardees made approximately US$ 100 million worth of federal procurement contracts and filed approximately 50 patents while they have received approximately US$ 6 million of SBIR awards and employed 90 people.

Table 1. Descriptive Statistics of Data for DEA

Category	Variable	Descriptive Statistics				
		Obs	Mean	Max	Min	SD
Stage-1 inputs	ASA	252	5.78	103.27	0.29	11.21
	EMP	252	86.17	480.00	2.00	109.11
	TDD	252	0.38	0.88	0.00	0.21
Stage-1 outputs &	PAT	252	49.61	1,251.00	15.00	94.77
Stage-2 inputs	SEC	252	0.02	0.05	0.00	0.01
Stage-2 output	FPC	252	98.14	2,433.14	0.19	292.76

Note: *ASA*: amount of SBIR awards; *EMP*: number of employees; *TDD*: technological distance from DoD; *PAT*: number of patents; *SEC*: eigenvector centrality in the SBIR funding network; *FPC*: federal procurement contract

For the Tobit regression as a subsequent analysis, we also collected firms' demographic data, such as age (*AGE*), location (*HUB*: Historically Under-utilized Business Zones; *RUR*: rural area with less than 50,000 people; *LOC*: leading states such as California and Massachusetts; and *STE*: states), technological concentration (*HTC*: high-tech focus; and *IPC*: technical areas expressed by the international patent classification codes), and ownership (*MOW*: minority-owned firms; and *WOW*: women-owned firms), and calculated technological distance from prime contractors (*TDP*) and closeness centrality in the SBIR funding network (*SCC*). While *AGE*, *TDP*, and *SCC* are continuous variables, *HTC, RUR, LOC, HUB, MOW*, and *WOW* are binary (or dummy) variables. *STE* and *IPC* are categorical variables (but transformed into binary variables for the analysis). The former includes 33 states in which 252 elite DoD SBIR awardees are located while the latter includes 8 sections of technical fields (A: human necessities; B: performing operations and transporting; C: chemistry and metallurgy; D: textiles and paper; E: fixed constructions; F: mechanical engineering, lighting, heating, weapons, and blasting; G: physics; and H: electricity) in which the awardees are situated.

Table 2 presents a summary of Tobit regression data descriptive statistics. On average, elite DoD SBIR awardees are approximately 22 years old. A majority of them are nested in high-tech industries and situated in urban areas of leading states. Few of them are owned by minorities or women.

Table 2. Descriptive Statistics of Data for Tobit Regression

Category	Variable	Descriptive Statistics				
		Obs	Mean	Max	Min	SD
Continuous	AGE	252	22.17	122.00	2.00	14.93
	TDP	252	0.46	0.97	0.00	0.26
	SCC	252	0.88	1.00	0.51	0.12
Binary	HTC	252	0.86	1.00	0.00	0.35
	RUR	252	0.38	1.00	0.00	0.49
	LOC	252	0.72	1.00	0.00	0.45
	HUB	252	0.01	1.00	0.00	0.11
	MOW	252	0.04	1.00	0.00	0.20
	WOW	252	0.04	1.00	0.00	0.20

Note: *AGE*: age of firms; *TDP*: technological distance from prime contractors; *SCC*: closeness centrality in the SBIR funding network; *HTC*: high-tech concentration; *RUR*: location in the rural area; *LOC*: location in the leading states; *HUB*: location in Historically Under-utilized Business Zones (HUBZones); *MOW*: firms owned by minority; *WOW*: firms owned by woman

Results

Table 3 summarizes the OE measure of some companies measured by Model 3. Instead of listing all 252 companies' OE scores, we present those of 24 companies as an example. The first company (1st Detect Corp) of the table shows the status of full efficiency (OE = 1.00) while the second company (Aculight) exhibits the status of inefficiency (OE = 0.78). All firms are characterized by their OE measures. To visually summarize all the measures, Figure 2 exhibits the distribution of OE measures of all firms.

Table 3. Operational Efficiency of Some Companies

Company	OE	Company	OE
1st Detect Corp	1.00	Calspan Corporation	0.54
Aculight Corp.	0.78	Cambridge Scientific, Inc.	1.00
Ada Technologies, Inc.	0.83	Cape Cod Research, Inc.	1.00
Adaptive Materials, Inc.	0.20	Cascade Designs	0.28
Adesto Technologies	0.73	Ceradyne, Inc.	1.00
Advanced Ceramics Research, Inc.	0.36	Ceramatec, Inc.	0.43
ADVANCED CIRCULATORY SYSTEMS, INC.	0.50	CFD Research Corp.	1.00
Advanced Energy Systems, Inc.	0.27	CHEMIMAGE CORP.	1.00
Advanced Fuel Research, Inc.	1.00	CIPHERGEN BIOSYSTEMS, INC.	0.54
Advanced Mechanical Technology, Inc.	0.45	Cleveland Medical Devices, Inc.	0.29
Advanced Scientific Concepts, Inc.	0.65	Coherent Logix, Inc.	0.19
AEC-ABLE ENGINEERING CO., INC.	0.52	Coherent Technologies, Inc.	0.22

Note: OE = Operational Efficiency

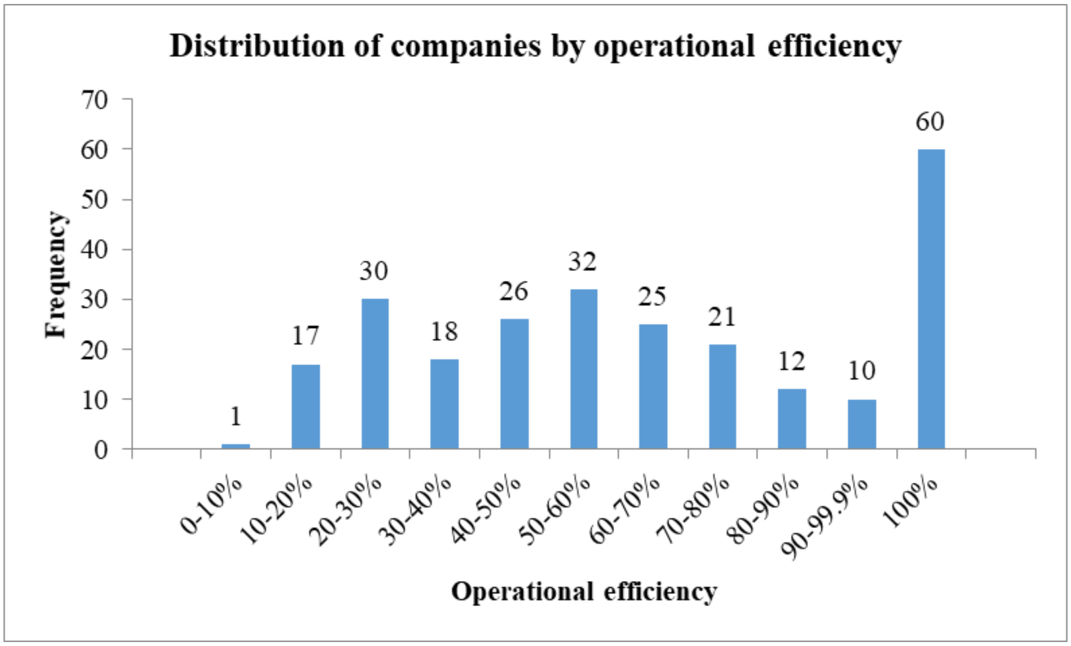


Figure 2. Distribution of Companies by Operational Efficiency

Statistical Analysis

Assuming that the operational performance of elite DoD SBIR awardees may be affected by not only input and output variables used in the DEA investigation but also their demographics, relationships to prime contractors, and agility in seeking external funding, this study constructs the following hypotheses. Since firm-level characteristics related to the transition from *R&D* to commercialization have been studied by other literature, we focus on firm-level traits related to the transition from *network building* to commercialization and contextual factors such as firms' technological concentration (industrial context) and location (geospatial context).

Hypothesis 1: *Small companies' network building capacity (captured by TDD, SEC, and SCC) is positively related to their operational performance.*

Sub-hypothesis 1a (H1a): *Small firms with higher TDD outperform those with lower TDD.* One of the primary objectives of the DoD SBIR program is acquiring R&D outcomes developed by the private sector (technology-based small businesses in this case) but not yet held by the public sector (DoD in this case). To fill the technological gap, the DoD may look for small firms with complementary technical assets so that small firms with different patent portfolios from the DoD's may be more advantageous in developing networks with the DoD than those with similar patent portfolios to what the DoD's are.

Sub-hypothesis 1b (H1b): *Small firms with higher SEC outperform those with lower SEC.* The amount of the SBIR budget (determined by the percentage of extramural R&D budget of SBIR-participating agencies) and procurement contracts depends highly upon the size of agencies. Given that three services (Air Force, Army, and Navy) are the largest DoD components and other components are relatively small, connections to big-sized components may be more valuable to small firms than those to small-sized components.

Sub-hypothesis 1c (H1c): *Small firms with higher SCC outperform those with lower SCC.* Social closeness often means better access to information that is critical for securing external funding. Thus, social adjacency to funders and small firms' agility in seeking funding sources may lead to better performance.

Hypothesis 2: *Small companies' high-tech concentration (captured by HTC and IPC) is positively associated with their operational performance.*

Sub-hypothesis 2a (H2a): *Small firms operating in the high-tech industries outperform those in the non-high-tech industries.* The R&D- and capital-intensive nature of high-tech industries tends to lead to higher value-added. Nowadays it is especially true since technology plays a pivotal role in firms' sustainable competitiveness. Thus, small firms with high-tech focus may achieve better performance than those with non-high-tech focus.

Sub-hypothesis 2b (H2b): *Small firms operating in the industries indexed by specific IPC codes outperform those in the industries indexed by other IPC codes.* According to the Eurostat indicators on high-tech industry and knowledge,[3] this study includes (a) computer and automated business equipment (indexed by G06C, G06D, etc.), (b) aviation (indexed by B64B, B64C, etc.), (c) micro-organism and genetic engineering (indexed by C40B, C12P, etc.), (d) lasers (indexed by H01S), (e) semiconductors (indexed by H01L), (f) communication technology (indexed by H04B, H04H, etc.), and (g) biotechnology (indexed by A61K, G01N, etc.) in the fields of high technology. In congruence with sub-hypothesis 2a, small firms with high-tech concentration (indexed by the aforementioned IPC codes) may perform better than those with non-high-tech concentration.

Hypothesis 3: *Small companies' location in better places (captured by HUB, RUR, LOC, and STE) is positively associated with their operational performance.*

Sub-hypothesis 3a (H3a): *Small firms located in the HUBZones or rural areas underperform counterparts.* Despite various incentive programs (e.g., tax credits) offered by governments for small firms located in the HUBZones or rural areas, small firms in those areas tend to have many disadvantages in conducting R&D, building networks, and commercializing R&D outcomes because of (a) limited access to a well-trained workforce, financial resources, and valuable information, (b) lack of knowledge spillover and infrastructure (e.g., broadband), and (c) distance to the market or customers. Thus, small firms situated in the HUBZones or rural areas may perform worse than counterparts do.

Sub-hypothesis 3b (H3b): *Small firms located in the economically or technologically leading states or states with business-friendly environment outperform counterparts.* Leading states such as California, New York, and Massachusetts tend to offer a better business environment to small firms than lagging states such as Mississippi, Montana, and Wyoming. Particularly financial resources such as venture capitalists and angels, which are critical for high-risk technology-based small businesses, tend to concentrate in the leading states. Moreover, leading states tend to have more prestigious research universities and large companies that play a role as anchor institutions in the regional innovation system or entrepreneurial ecosystem. Thus, small firms situated in the leading states may perform better than counterparts do.

To empirically test these hypotheses, we employ Tobit regression models that are appropriate for censored data considering the boundary of firms' efficiency scores (between 0 and 1; Bi et al., 2016). We test three different Tobit models (M 1–3) using efficiency scores as a dependent variable. The first model (M 1) includes only DEA input and output variables, the second (M 2) adds one more network building capacity-related variable (i.e., *SCC*), and the third

[3] Eurostat indicators on High-tech industry and Knowledge-intensive services Annex 6 – High-tech aggregation by patents (https://ec.europa.eu/eurostat/cache/metadata/Annexes/htec_esms_an6.pdf)

adds contextual variables (i.e., *HTC*, *IPC*, *LOC*, and *STE*). Because of a long list of variables, we utilized the stepwise function that removes insignificant variables from a full model. We also dropped some variables due to the multicollinearity issue. Table 4 summarizes the Tobit analysis results.

Table 4. Results of Tobit Regressions

Variables		M 1		M 2		M 3	
Controls	ASA	0.0027	(1.48)	-0.0016	(-0.99)	-0.0017	(-1.06)
	EMP	-0.0011***	(-6.18)	-0.0010***	(-6.19)	-0.0010***	(-6.24)
	PAT	0.0009***	(3.66)	0.0009***	(3.86)	0.0009***	(3.97)
	FPC	0.0004***	(5.27)	0.0004***	(6.31)	0.0004***	(6.50)
Network building capacity	TDD	-0.3604***	(-4.48)	-0.3754***	(-5.39)	-0.3952***	(-5.58)
	SEC	1.0694	(0.67)	30.1008***	(8.78)	31.5687***	(9.17)
	SCC			-2.9771***	(-9.19)	-3.0433***	(-9.41)
High-tech focus	HTC					0.0989*	(1.94)
	IPC (Section A)					0.2308***	(2.87)
Location	HUB					0.0202	(0.16)
	RUR					-0.0471	(-1.58)
	LOC					-0.0389	(-1.16)
	STE (North Carolina)					-0.2742**	(-2.08)
Model fit	Pseudo R^2	0.56		1.20		1.32	
	AIC	66.86		-5.51		-7.47	
	BIC	95.10		26.25		45.47	

Note: (a) *ASA*: amount of SBIR awards; *EMP*: number of employees; *PAT*: number of patents; *FPC*: federal procurement contract; *TDD*: technological distance from DoD; *SEC*: eigenvector centrality in the SBIR funding network; *SCC*: closeness centrality in the SBIR funding network; *HTC*: high-tech concentration; *IPC*: international patent classification; *HUB*: location in Historically Under-utilized Business Zones (HUBZones); *RUR*: location in the rural area; *LOC*: location in the leading states; *STE*: state (b) Values in parenthesis are t-statistics. *** significant at 1% level; ** significant at 5% level; * significant at 10% level.

Based on our hypotheses and the model fit, our interpretation follows the third model (M 3). According to the analysis results, operational efficiency of the elite DoD SBIR awardees has statistically significant relationships (a) positively with *PAT*, *FPC*, *SEC*, *HTC*, and *IPC (Section A)*, and (b) negatively with *EMP*, *TDD*, *SCC*, and *STE (North Carolina)*. Figure 3 visually summarizes results on the Tobin analysis and partly supports our three hypotheses. The empirical results are summarized as follows:

First, sub-hypothesis *H1b* is only supported and *H1a* and *H1c* are rebutted. *SEC* is positively related to efficiency, meaning that small firms' funding connections to big-sized DoD components enhance their operational performance. However, *TDD* and *SCC* are negatively related to efficiency, meaning that small firms' technological dissimilarity to the DoD and closer connections to more funding sources hurt their operational performance. Those results imply that (a) technological similarity (low technological distance) is better for small firms in building networks with funders, rather than technological dissimilarity (high technological distance), and (b) building a strong network with one of the big-sized DoD components and sticking to one funding source are better for small firms in terms of operational efficiency, rather than developing weak networks with multiple DoD components.

Second, sub-hypotheses *H2a* and *H2b* both are supported. *HTC* and *IPC (Section A)* are positively associated with efficiency, suggesting that small firms' high-tech concentration, particularly in biotechnology (e.g., IPC A61B: diagnosis and surgery; A61F: prostheses; and A61K preparation for medical and dental purposes), improves their operational performance. Those results confirm that industrial context plays an important role in determining small firms' operational performance.

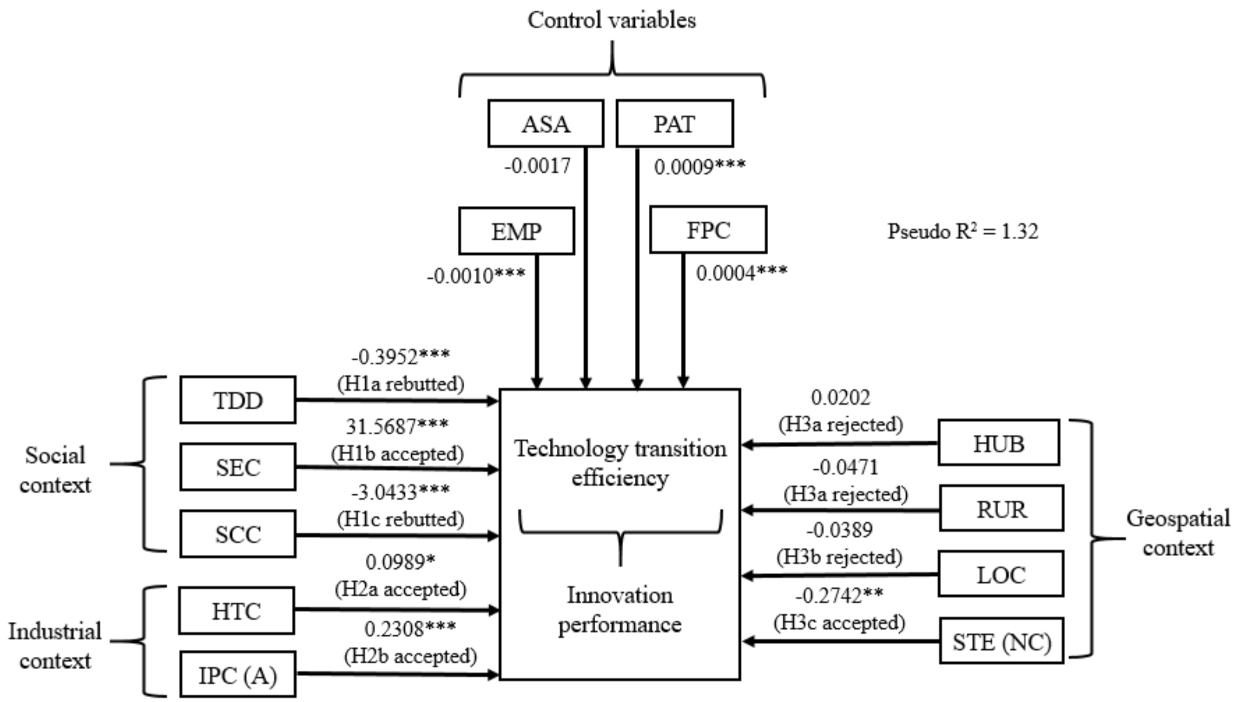


Figure 3. Concept and Result of Tobit Regression

Note: (a) *ASA*: amount of SBIR awards; *EMP*: number of employees; *PAT*: number of patents; *FPC*: federal procurement contract; *TDD*: technological distance from DoD; *SEC*: eigenvector centrality in the SBIR funding network; *SCC*: closeness centrality in the SBIR funding network; *HTC*: high-tech concentration; *IPC (A)*: international patent classification (section A); *HUB*: location in Historically Under-utilized Business Zones (HUBZones); *RUR*: location in the rural area; *LOC*: location in the leading states; *STE (NC)*: state (North Carolina).

(b) *** significant at 1%; ** significant at 5%; * significant at 10%.

Finally, sub-hypotheses *H3a* and *H3b* are not confirmed. *HUB*, *RUR*, and *LOC* do not have statistically significant relationships with efficiency while *STE (North Carolina)* is negatively associated with efficiency. Those results imply that geospatial context does play a substantial role in determining small firms' operational performance. There are some possible explanations: (a) the SBIR funding is geographically distributed on an equity basis or considering assistance for small firms located in disadvantageous areas, (b) lagging states, as well as leading states, are also active in offering business-friendly environment to small firms in their jurisdiction to invigorate their economies (e.g., the State of New Mexico is implementing the SBIR matching fund program), and (c) national laboratories and military bases located in remote areas contribute to R&D, network building, and commercialization (e.g., Sandia and Los Alamos

National Laboratories and Air Force Research Laboratories in New Mexico play a key role in the regional innovation system). On the other hand, it turned out that small firms located in the State of North Carolina underperform those in other states. It is rather contradictory considering its Research Triangle Park (RTP) that consists of three research universities (Duke University, the University of North Carolina at Chapel Hill, and North Carolina State University). We cautiously conjecture that (a) there may be an imbalance between R&D/network building and commercialization (i.e., the former is strong due to the existence of RTP but the latter may be relatively weak), (b) there may be lack of knowledge spillover from RTP or business and policy efforts in promoting entrepreneurial innovation, and (c) the proportion of small firms that specialize in high technology may be relatively small.

Conclusion and Future Extensions

This study first evaluated the innovation performance of 252 elite DoD SBIR awardees in the context of technology transition and then examined the impacts of social, industrial, and geospatial factors on the performance. For the first task, we employed a two-stage, dynamic DEA to reflect more realistic conditions of innovation (as a complex and time-consuming process that requires social capital as well as traditional input factors). According to the DEA result, more than a quarter of companies (60) turned out efficient while three quarters are not fully efficient. About half of companies showed efficiency scores that are higher than 60%. It implies that there is still significant room for improvement for many companies.

For the second task, we used Tobit regression analysis to deal with censored data (the upper limit of the 60 efficient companies in efficiency score is 100%). The statistical analysis demonstrated that our three hypotheses are partly supported. Our first hypothesis was that small companies with higher network building capacity outperform those with lower network building capacity. It turned out that small firms' connections to influential funders contributed to their performance but their heterogeneous technological portfolios and connections to multiple funders did not. It suggests that developing and strengthening networks with big-sized funders (focused networking rather than distracted networking) positively affects the technology transition performance. The second hypothesis was that small companies with high-tech concentration outperform those with low- or medium-tech concentration. It turned out true, particularly for biotech companies. It suggests that industrial context plays a significant role in the technology transition performance. The last hypothesis was that small companies situated in the preferred location outperform those in the unpreferred location. It turned out that locational factors were not critical. It suggests that the geospatial context plays a minor role in the technology transition performance.

While there are many studies on the knowledge generation function (e.g., Antonelli & Colombelli, 2018) and knowledge utilization or revenue/profit generation function (e.g., Lichtenthaler, 2005; Bergman & Usai, 2009), which account for the R&D sub-process at the first stage and commercialization sub-process at the second stage of our DEA framework, respectively, there are a relatively small number of studies on social link (or trust) generation function that represents the network building sub-process at the first stage. This study incorporated the concept of social capital into the DEA-based innovation performance measurement for the first time. In this study, we used technological distance based on the assumption that the DoD seeks small companies with different technological portfolios from its own portfolio as R&D partners (SBIR awardees in this case). Thus, we used technological distance as an input to the network building sub-process at the first stage of our DEA framework and eigenvector centrality as an output in the sub-process. However, there may be other alternative inputs that can better capture the input factors to the network building sub-process.

As the number of studies on social capital increases, we may be able to determine more suitable measures.

Appendix

All abbreviations used in this study are summarized as follows: DARPA: Defense Advanced Research Projects Agency, DEA: Data Envelopment Analysis, DIU: Defense Innovation Unit, DMU: Decision-Making Unit, DoD: Department of Defense, FPDS-NG: Federal Procurement Data System-Next Generation, EIS: European Innovation Scoreboard, KIPRIS: Korea Intellectual Property Rights Information Service, RIS: Regional Innovation System, RTS: Returns to Scale, NIS: National Innovation System, OE: Operational Efficiency, OECD: Organisation for Economic Cooperation and Development, PT: Production Technology, R&D: Research and Development, RIS: Regional Innovation System, RTP: Research Triangle Park, RTS: Returns to Scale, SAM: System for Award Management, SBA: Small Business Administration, SBIR: Small Business Innovation Research, STTR: Small Business Technology Transfer, and URS: Unrestricted.

References

Antonelli, C., & Colombelli, A. (2018). External and internal knowledge in the knowledge generation function. In *The Evolutionary Complexity of Endogenous Innovation*. Edward Elgar Publishing.

Autio, E., Kenney, M., Mustar, P., Siegel, D., & Wright, M. (2014). Entrepreneurial innovation: The importance of context. *Research Policy*, *43*(7), 1097–1108.

Bar, T., & Leiponen, A. (2012). A measure of technological distance. *Economics Letters*, *116*(3), 457–459.

Benner, M., & Waldfogel, J. (2008). Close to you? Bias and precision in patent-based measures of technological proximity. *Research Policy*, *37*(9), 1556–1567.

Bergman, E., & Usai, S. (2009). Knowledge diffusion in European regions. *IAREG Deliverable*, 4.1.

Bi, K., Huang, P., & Wang, X. (2016). Innovation performance and influencing factors of low-carbon technological innovation under the global value chain: A case of Chinese manufacturing industry. *Technological Forecasting and Social Change*, *111*, 275–284.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social Networks*, *29*(4), 555–564.

Carayannis, E. G., Grigoroudis, E., & Goletsis, Y. (2016). A multilevel and multistage efficiency evaluation of innovation systems: A multiobjective DEA approach. *Expert Systems with Applications*, *62*, 63–80.

Chen, K., & Guan, J. (2012). Measuring the efficiency of China's regional innovation systems: Application of network data envelopment analysis (DEA). *Regional Studies*, *46*(3), 355–377.

Chen, X., Liu, Z., & Zhu, Q. (2018). Performance evaluation of China's high-tech innovation process: Analysis based on the innovation value chain. *Technovation*, 74–75, 42–53.

Chun, D., Chung, Y., & Bang, S. (2015). Impact of firm size and industry type on R&D efficiency throughout innovation and commercialisation stages: Evidence from Korean manufacturing firms. *Technology Analysis & Strategic Management*, *27*(8), 895–909.

Dias, M. F. P., Pedrozo, E. A., & da Silva, T. N. (2014). The innovation process as a complex structure with multilevel rules. *Journal of Evolutionary Economics*, *24*(5), 1067–1084.

Dobbins, J. H. (2004). Planning for technology transition. *Defense AT&L*, *33*(2), 14–17.

Dougherty, D. (2017). Organizing for innovation in complex innovation systems. *Innovation*, *19*(1), 11–15.

Faulk, L., McGinnis Johnson, J., & Lecy, J. D. (2017). Competitive advantage in nonprofit grant markets: Implications of network embeddedness and status. *International Public Management Journal*, *20*(2), 261–293.

Guan, J. C., Yam, R. C. M., Mok, C. K., & Ma, N. (2006). A study of the relationship between competitiveness and technological innovation capability based on DEA models. *European Journal of Operational Research*, *170*(3), 971–986.

Hicks, D., & Hegde, D. (2005). Highly innovative small firms in the markets for technology. *Research Policy*, *34*(5), 703–716.

Katz, J. S. (2016). What is a complex innovation system? *PloS one*, *11*(6).

Kaufmann, A., & Tödtling, F. (2001). Science–industry interaction in the process of innovation: The importance of boundary-crossing between systems. *Research Policy*, *30*(5), 791–804.

Kou, M., Chen, K., Wang, S., & Shao, Y. (2016). Measuring efficiencies of multi-period and multi-division systems associated with DEA: An application to OECD countries' national innovation systems. *Expert Systems with Applications*, *46*, 494–510.

Kuah, C. T., & Wong, K. Y. (2011). Efficiency assessment of universities through data envelopment analysis. *Procedia Computer Science*, *3*, 499–506.

Lee, W. J. (2015). Social capital as a source of business advantages for a woman entrepreneur in the context of small-size business. *Asian Social Science*, *11*(12), 155.

Lichtenthaler, U. (2005). External commercialization of knowledge: Review and research agenda. *International Journal of Management Reviews*, *7*(4), 231–255.

Lundberg, H. (2013). Triple helix in practice: The key role of boundary spanners. *European Journal of Innovation Management*, *16*(2), 211–226.

MacMillan, I. C. (1975). Strategy and flexibility in the smaller business. *Long Range Planning*, *8*(3), 62–63.

Miller, N. J., Besser, T., & Malshe, A. (2007). Strategic networking among small businesses in small U.S. communities. *International Small Business Journal*, *25*(6), 631–665.

Moen, Ø. (1999). The relationship between firm size, competitive advantages and export performance revisited. *International Small Business Journal*, *18*(1), 53–72.

Pelz, D. C. (1985). Innovation complexity and the sequence of innovating stages. *Knowledge*, *6*(3), 261–291.

Ryu, Y. (2017). *Accelerating the transition of technologies created through the U.S. Department of Defense small business innovation research program*. Pardee RAND Graduate School, Santa Monica, CA.

Sharma, S., & Thomas, V. (2008). Inter-country R&D efficiency analysis: An application of data envelopment analysis. *Scientometrics*, *76*(3), 483–501.

Small Business Administration. (2014). *Small business innovation research (SBIR) program: Policy directive*.

Sueyoshi, T., & Goto, M. (2013). A use of DEA–DA to measure importance of R&D expenditure in Japanese information technology industry. *Decision Support Systems*, *54*(2), 941–952.

Sueyoshi, T., & Goto, M. (2018) *Environmental assessment on energy and sustainability by data envelopment analysis.* John Wiley & Sons.

Sun, C. C. (2011). Evaluating and benchmarking productive performances of six industries in Taiwan Hsin Chu Industrial Science Park. *Expert Systems with Applications*, *38*(3), 2195–2205.

Tidd, J. (2001). Innovation management in context: Environment, organization, and performance. *International Journal of Management Reviews*, *3*(3), 169–183.

Wang, Q., Hang, Y., Sun, L., & Zhao, Z. (2016). Two-stage innovation efficiency of new energy enterprises in China: A non-radial DEA approach. *Technological Forecasting and Social Change*, *112*, 254–261.

Zabala-Iturriagagoitia, J. M., Voigt, P., Gutiérrez-Gracia, A., & Jiménez-Sáez, F. (2007). Regional innovation systems: How to assess performance. *Regional Studies*, *41*(5), 661–672.

Developing a Threat and Capability Coevolutionary Matrix: Application to Shaping Flexible Command and Control Organizational Structure for Distributed Maritime Operations

Dr. Ying Zhao—is a research professor at the Naval Postgraduate School (NPS). Her research is focused on data sciences, machine learning, and artificial intelligence methods for defense military applications such as semantic and social networks, common tactical air pictures, combat identification, logistics, and mission planning. Since joining NPS, Zhao has been a principal investigator (PI) on many projects awarded for DoD research projects. She received her PhD in mathematics from MIT and is the co-founder of Quantum Intelligence, Inc. [yzhao@nps.edu]

Abstract

The mission of the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD A&S) is to quickly and cost effectively deliver and sustain secure and resilient capabilities to warfighters and international partners. There are urgent requirements to develop adaptive acquisition framework (AAF) to speed up software development and acquisition processes that strengthen the concepts of operations (CONOPS) such as distributed maritime operations (DMO). It is imperative for the Department of Defense (DoD) to shape the AAF using data-driven analysis linked to the National Defense Strategy and the nature of global threats, and scale new capabilities to counter new threats. The threat and capability coevolutionary matrix (TCCM) addresses the requirement. A threat is a problem a capability tries to deal with. A capability is the solution to the problem that represents a threat. Coevolutionary algorithms explore domains in which the quality of a capability or combination of capabilities is determined by its ability to successfully defeat a threat or combination of threats. TCCM has the potential to systematically optimize, recommend, and coevolve capabilities and threats in new and contested environments. We show a use case regarding helping a program executive office (PEO) to wargame capabilities and threats against a specific domain DMO using unclassified data compiled from open sources.

Introduction

It is necessary not only for the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD A&S) to shape the acquisition strategy but also for the whole Department of Defense (DoD) to apply data-driven analysis and innovative and adaptive concepts of operations (CONOPS) linked to the National Defense Strategy and the nature of global threats and to scale new capabilities for warfighters.

For example, to enhance total force readiness and project combat power across the wide range of operations and spectrum of conflict at any time, the Navy needs flexible command and control (C2) organizational structures to meet the CONOPS. For example, DMO is a CONOPS for the Navy, and expeditionary advanced base operations (EABO) is a CONOPS for the U.S. Marine Corps (USMC). Both DMO and EABO are emerging operation concepts for modernization of naval warfare. PMW 150, PEO C4I's Program Office for C2 Systems and preeminent provider of C2 solutions, focuses on acquisitions to transform operational needs into effective and affordable operational and tactical C2 capabilities for the Navy, Marine Corps, Joint and coalition warfighters. PMW 150's mission is to "innovatively meet operational requirements with relevant capabilities, enabling the warfighter to maintain C2 superiority" (Colpo, 2016).

On the other hand, U.S. ships' maritime operations, particularly in the littorals, will continue to be contested and dangerous; therefore, it is imperative to develop DMO and EABO towards a unifying operational vision. DMO aims to support national and strategic objectives in contested environments. The DMO concept considers not only offensive strikes as the primary tactic for winning in battle, but also identifies the ability to deceive and confuse the enemy as a critical task to achieve success in a contested environment. The current efforts are focused on integration of existing platforms, systems, and capabilities with the DMO specific tactics to achieve maritime strategic and operational objectives. DMO is defined as the "warfighting capabilities necessary to gain and maintain sea-control through the employment of combat power that may be distributed over vast distances, multiple domains, and a wide array of platforms" (Navy Warfare Development Command [NWDC], 2017).

The development of DMO as a concept for the operations of Navy and Marine assets stems from the Distributed Lethality (DL) model (Popa et al., 2018). The concept of DMO adopts an extended viewpoint of DL, comprised of three pillars: the ability to increase the offensive power of individual warships through networked firing capability, distribution of the offensive capability over a wide geographic area, and the allocation of sufficient resources to the surface platforms in order to enable the enhanced combat capability (Rowden, 2017). DMO also stresses the need for more resilient and sustainable surface platforms in all domains, including air, subsurface, and cyber warfare. The futuristic view of DMO is to be a fleet-centric fighting power, enabled by integration, distribution, and maneuverability that allows simultaneous and synchronized execution of multiple capabilities and tactics across multiple domains (contested-air, land, sea, space, and cyberspace; DoD, 2018) in order to fight and win in complex contested environments (Canfield, 2017). Therefore, DMO not only includes traditional warfare capabilities of sensors, platforms, networks, and weapons, but also extends to other tactics that evolve with new technologies. The DMO concepts use advanced detection and deception involving ISR, machine learning (ML), and artificial intelligence (AI), with the use of unmanned systems particularly for enhanced capabilities in offensive tactical operations; therefore, by potentially leveraging different combinations of platforms, sensors, weapons, networks, and tactics, the combat power of a diverse yet unified force can be amplified across all maritime domains.

The DMO concepts include detailed capabilities such as tactics for counter-measures, counter-targeting, and counter-engagements. Counter-measures are defensive capabilities which aim to divert threats. Counter-targeting may be offensive capabilities, deceptive tactics, and operational maneuvers that divert a threat. Deceptive tactics include swarms of unmanned assets, mechanical and physical counter-measures, electronic jamming, and the limiting of electromagnetic radiation, or emissions control (EMCON). Counter-engaging is to neutralize a threat.

Traditionally, a baseline force structure consists of a fixed set of friendly force ships and aircraft arranged into action groups including a Carrier Strike Group (CSG), Expeditionary Strike Group (ESG), Surface Action Group (SAG), and various independent deployable units such as expeditionary Marine units for EABO.

The DMO operational requirements include capabilities, manpower, maintenance, and supply, among other resources, to be carefully analyzed, planned, and executed, which require the right data strategy, distributed infrastructure, and deep analytics. The technical concept of Threat and Capability Coevolutionary Matrix (TCCM) addresses the requirements of DMO and EABO operations. A threat is a problem that a capability tries to deal with, including the complexity and urgency. A capability is the solution to the problem that represents the threat. Coevolutionary algorithms from the ML/AI community explore domains in which the quality of a capability or combination of capabilities are determined by their ability to successfully defeat a threat or combination of threats. Coevolutionary algorithms used in a wargame simulation are similar to the Monte Carlo simulation widely used in defense applications, except they engage ML/AI like forecast and prediction, optimization, and game (minmax) algorithms. The DMO and

EABO concepts require flexibility and evolution of the capability and resource networks that handle ever changing and evolving threats.

Methodology Review

A TCCM contains three aspects, as follows:

Data strategy

One data strategy for a big organization such as the U.S. Navy is to build a centralized big data store for all the suborganizations. For this strategy, one needs to gather data from across the organizations and enterprises and put them in a centralized location. Building centralized data repositories can be very expensive, in addition to creating security and trust issues. An alternative strategy is distributed data strategy, where a complex enterprise usually includes highly interacting, interrelated, and interdependent sub-systems. For example, data for a complex enterprise might be collected using distributed locations. This data strategy provides convenience, safety, and privacy for the data; however, it presents difficulty and challenges for data fusion and deep analytics. Traditional data sciences, even ML/AI algorithms used in small- or moderate-sized analysis, typically require tight coupling of the computations, where such an algorithm often executes in a single machine or job and reads all the data at once. Making a generic case of parallel and distributed computing across distributed data source proves a difficult task. One requires novel infrastructure such as Collaborative Learning Agents (CLA; Zhao & Zhou, 2014) or federated learning, where data from system of systems can be quickly examined locally, while analytic models from multiple agents can be also fused properly.

Distributed Infrastructure and Collaborative Learning Agents

The data strategy we focus on here is not only relevant to information warfare, but also to physical infrastructure such as force distribution, as well. Distributed force distribution allows avoidance of detection and flexibility of C2 among other innovations; for example, dynamic emergence and self-organization of new global structures can confuse the threats and adversaries. CLAs include distributed, networked, and peer-to-peer agent architecture and analytics. A single agent represents a single system capable of ingesting data, indexing, cataloging information, and performing knowledge and pattern discovery, machine learning from data, and separating patterns and anomalies from data. Multiple agents can work collaboratively in a network in Figure 1. In more detail, a CLA first applies unsupervised machine learning and data mining algorithms, indexes, catalogs, and data-mines structured and unstructured data sources and discovers knowledge patterns, then fuses models from its peer lists and makes them available for search and pattern match used for prediction. A network of CLAs' collaboration is achieved through a peer list defined within each agent initially, through which each agent passes shared information to its peers, and then re-organizes or emerges based on a coevolution wargame with the threats. A CLA network and collaboration mechanism is fault-tolerant, self-organizing, adaptive, and resilient. A CLA is fault-tolerant because if one CLA goes down, it can be locally excluded and does not affect the whole network; it is self-organizing because each CLA can have trusted peers (e.g., friends) based on its own real-time situation awareness and change dynamically. A CLA is adaptive because the top-level search, pattern match, and prediction depend on the real-time self-organized network structure. A CLA is resilient since it can apply the coevolutionary analytics in a wide space and simulate novel threat and capability for new and unknown situations.

CLAs have been used in Navy applications such as building swarm intelligence to health monitoring of systems of systems such as ships, Internet of Things (IoTs; Zhao & Zhou, 2019), and edge computing. CLA also participated in a Naval Trident Warrior exercise (Zhou et al., 2009).

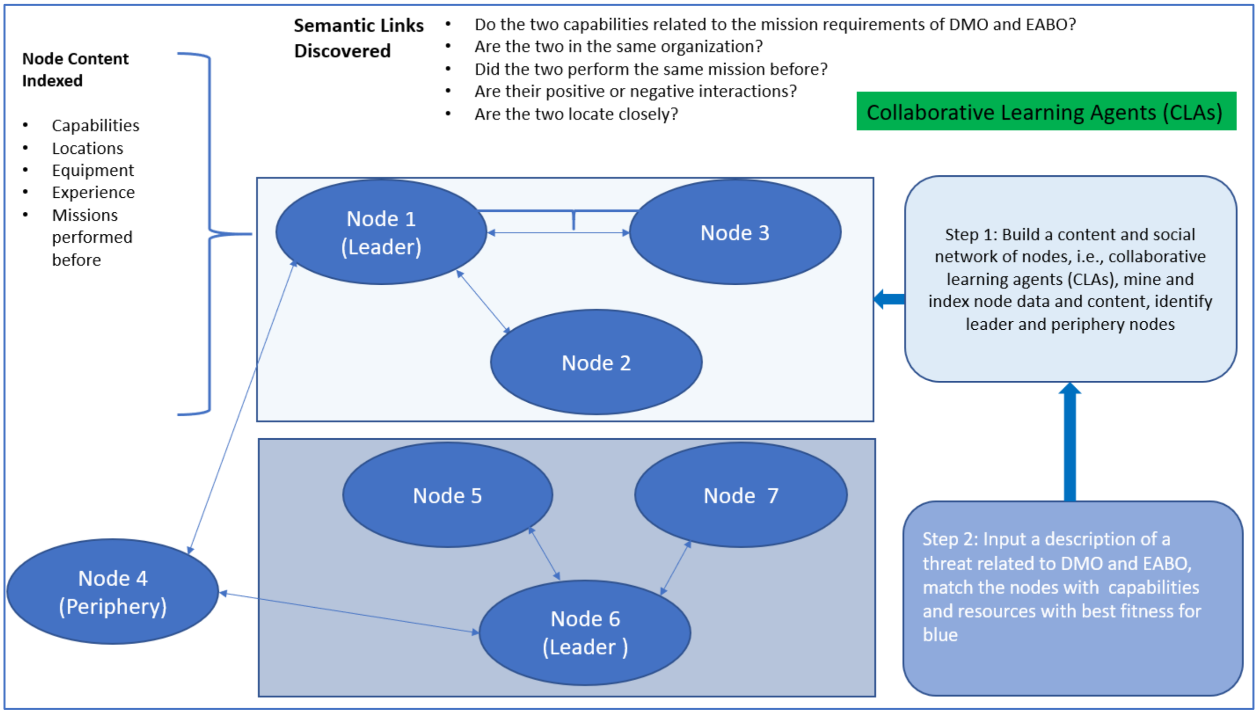


Figure 1. A CLA is Used in Each Node
Note: Each node's content and data may include capabilities; capability needs to be indexed, cataloged, and data-mined first.

Analytics

The core analytics for TCCM is a wargame simulation with two asymmetrical players. The self-player (blue) is the capability holder. The opponent (red) of the self-player is a threat generator. The opponent generates new threats that may challenge the self-player's capability. The self-player tries to predict and optimize capabilities to counter the opponent's threat. The whole process iterates. The self-player uses a wargame simulation to constantly perform what-if analyses in both threat and capability perspectives to defend a complex enterprise and its operations in a distributed and contested environment. Such a wargame simulation allows one to search, simulate, and detect vulnerability of the complex enterprise and evolve countermeasures, solutions, and resilience in a dynamic and flexible fashion.

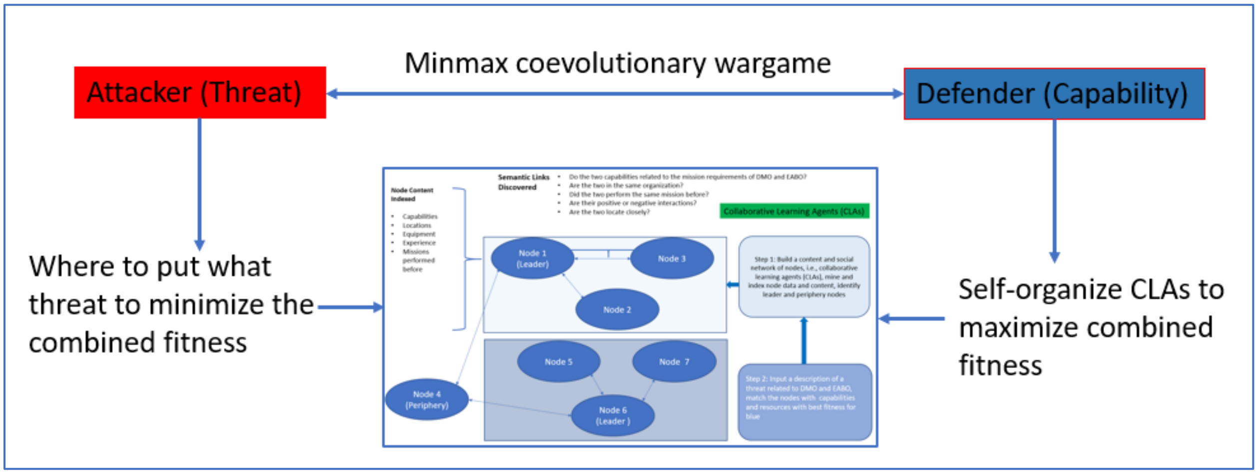


Figure 2. The Concept of TCCM and Wargame Simulation

Coevolutionary algorithms (Goldberg, 1989; O'Reilly & Hemberg, 2018; Popovici et al., 2012) provide search and optimization mechanisms based on evolutionary and genetic principles such as selection, mutation, and crossover. Coevolutionary algorithms explore domains in which the quality of a candidate solution (i.e., capability) is determined by its ability to successfully pass some set of tests (i.e., threats). Reciprocally, a threat's quality is determined by its ability to force errors and inefficiencies from a capability. Such a competitive coevolution is similar to game theory (Brown & Sandholm, 2017); however, it does not require computation of gradient as in ML algorithms and requires less data compared to other ML/AI algorithms. The search can lead to an arms race between threats and capabilities, with both evolving while pursuing opposite objectives. Coevolutionary algorithms are similar to those encountered by generative adversarial networks (GANs; Arora et al., 2017; Goodfellow et al., 2014).

The mutation and crossover evolutionary principles are unsupervised with known trends to produce better solutions. In TCCM, a selection is accomplished by evaluating a fitness function for the capability holder to see how likely it can successfully defeat a threat. A fitness function is typically modeled using supervised or reinforcement machine learning algorithms when a payoff (reward or penalty) can be clearly observed. By using a CLA to represent a self-player as a capability holder, the fitness function in this paper refers to a nearest neighbor lookup for an input threat (see the Use Case Scenario).

Use Case Scenario

The goal of this use case scenario is to help a program executive office (PEO) to wargame capabilities and threats in DMO focused areas. We will use this scenario to show a proof-of-concept of the TCCM.

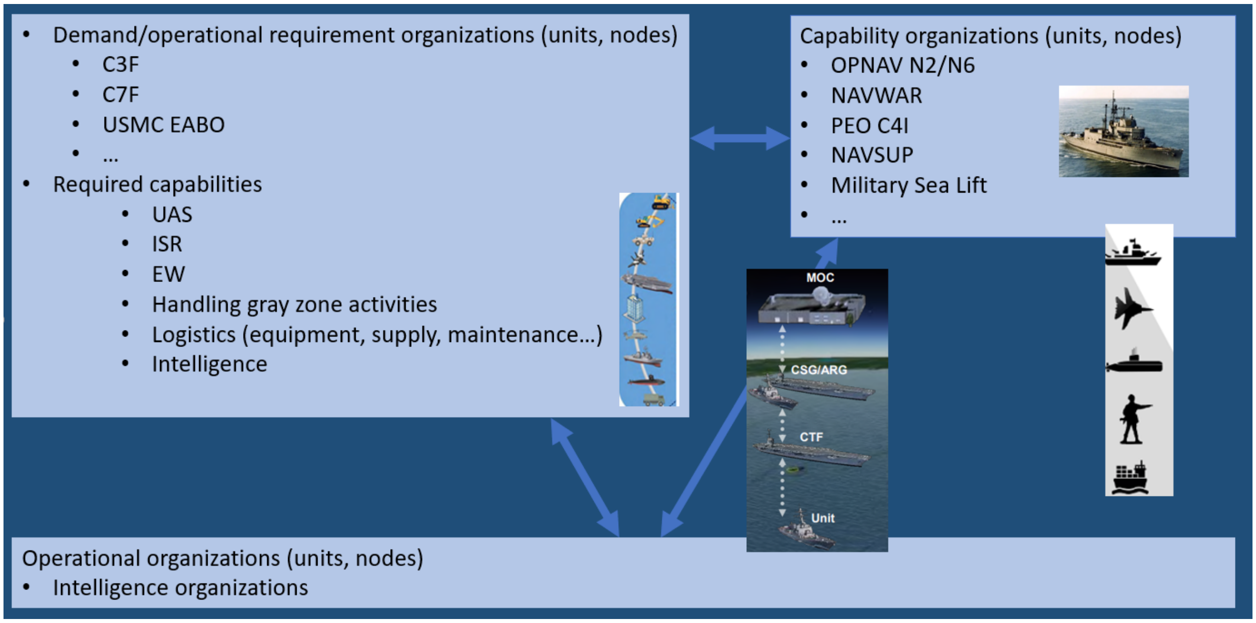


Figure 3. TCCM Use Case Scenario

As shown in Figure 3, the Navy's warfare resource, capability, and assets may reside in different traditional warfare domains such as sensors, platforms, networks, and weapons, as well as the new tactics related to the DMO concept. The goal is to achieve the creation and continuous use of a distributed tactical common operational picture, weapon system network, and the integration of unmanned assets and existing platforms to enhance defensive and offensive capabilities.

An order of battle (OOB) incorporates maritime platforms of surface ships, aircraft, weapons systems and sensors for both friendly and enemy forces. The high-level information of each platform, and asset for this paper, is compiled from open source databases, where a kill chain typically contains three steps: "find, target, and engage" (Joint Chiefs of Staff [JCS], 2013).

Operational domains can also include different areas such as air, surface, subsurface, land, and cyber, as well as detailed tactics that are associated with each domain. In a DMO environment, flexible combinations of these capabilities and tactics can provide powerful swarms of new capabilities that are unpredictable to new threats. A demand or operational node, such as U.S. Fleet Forces C3F or C7F as a node, may encounter a threat that may require capabilities of UAS, ISR, EW, new capabilities of handling gray zone (e.g., South China Sea) activities, logistics, or intelligence. The requirements go to capability organizations (units, nodes). In current C2 structure, a Maritime Operations Center (MOC) may communicate with a carrier strike group (CSG) or an amphibious ready group (ARG), then a carrier task force (CTF), and other units. Can the structure be more flexible, re-organized in a more dynamic fashion, and handle a vast amount and wide range of capabilities, resources, and requests? Future DMO enables a force that is capable of winning a fleet-on-fleet engagement through the integration of manned and unmanned systems, execution of deceptive tactics, and enabling units to conduct offensive strikes (Popa et al., 2018).

A wargame between a threat and capability can be set up as follows using the concept of coevolutionary process:

Self-player (blue, capability): Evolve a solution based on a current threat's characteristics (e.g., location). The solution may mean to find another CLA based on the input of required capability to increase the fitness functions representing measures of effectiveness (MOEs; global and overall measures) and measures of performance (MOPs; metrics for subtasks); for example, the selected CLA that represents an asset needs to contain required capability, and the asset has the highest probability of detection (POD) or probability of kill (POK) as MOP and MOE.

Opponent (red, threat): Evolve the location and other characteristics to reduce the fitness of the self-player.

A detailed simulation is set up based on a matrix of association and fitness values shown in Table 1. A synthetic data of such a table can be formed using the reference (Popa et al., 2018); for example, sensor capabilities and platforms associations for the blue force can be shown as in Table 2.

Table 1. Threat and Capability Association Matrix

	Capability's Sensor *Cs*	Capability's Platform *Cp*	Capability's Network *Cn*	Capability's Weapon *Cw*	Capability's Tactics *Ct*	Threat's Sensor *Ts*	Threat's Platform *Tp*	Threat's Network *Tn*	Threat's Weapon *Tp*	Threat's Tactics *Tt*
Capability's Sensor *Cs*		x	x	x	x	x				
Capability's Platform *Cp*	x		x	x	x					
Capability's Network *Cn*	x	x		x		x				
Capability's Weapon *Cw*		x		x	x	x	x	x	x	x
Capability's Tactics *Ct*	x	x	x	x	x	x	x	x	x	x
Threat's Sensor *Ts*	x	x	x							
Threat's Platform *Tp*	x	x								
Threat's Network *Tn*	x		x	x	x					
Threat's Weapon *Tp*	x	x	x	x	x		x		x	x
Threat's Tactics *Tt*	x	x	x	x	x	x	x	x	x	x

Table 2. Sensors and Platforms for the Blue Force (Popa et al., 2018)

Sensor Capabilities	**Platforms**
Visual	All Surface, All Air, All Unmanned
Infrared	CVN, LHD/LHA, CG, DDG-51, DDG-1000, LCS, LPD, F-35, F/A-18, EA-18, E-2, P-8, MH-60, AH-1, MQ-8 Fire Scout, MQ-4 Triton, TERN
Electronic Support Measures (ESM)	CVN, LHD/LHA, CG, DDG-51, DDG-1000, LCS, LPD, F-35, F/A-18, EA-18, E-2, P-8, MH-60, AH-1, MQ-8 Fire Scout, MQ-4 Triton
Air Search Radar	CVN, LHA/LHD, CG, DDG-51, DDG-1000, LCS, LPD, MH-60, AH-1, TERN
Surface Search Radar	All Surface Platforms, MH-60, AH-1, TERN
Fire Control Radar	CVN, LHD/LHA, CG, DDG-51, DDG-1000, LCS, LPD, MH-60, AH-1, MQ-8 Fire Scout
Navigation Radar	All Surface Platforms
Phased Array Radar	CVN, CG, DDG-51, DDG-1000
AESA (Active Electronic Scanned Array Radar)	F-35, F/A-18, EA-18, E-2, P-8, MQ-4 Triton
Airborne Early Warning Radar	E-2, P-8
Synthetic Aperture Radar—Maritime	MH-60, MQ-8 Fire Scout, MQ-4 Triton

Weapon capabilities and platforms associations for the blue force are sampled in Table 3.

Table 3. Weapon Capabilities and Platforms Associations for the Blue Force (Popa et al., 2018)

Missile	**Designator**	**Type**	**Launching Platform(s)**
Standard Missile-2	RIM-66	Medium Range Surface to Air	CG, DDG-51, DDG-1000
Standard Missile-3	RIM-161	Ballistic Missile Defense	CG, DDG-51, DDG-1000
Standard Missile-6	RIM-174	Extended Range Surface to Air, Anti-Ship Cruise Missile (ASCM)	CG, DDG-51, DDG-1000
LRASM	AGM-158C	Long Range Anti-Ship Missile	CG, DDG-51, DDG-1000, F-35, F/A-18
Maritime Strike Tomahawk	MST	Long Range Anti-Ship Cruise Missile	CG, DDG-51, DDG-1000
Harpoon	AGM/RGM-84	Over the Horizon Anti-Ship Missile	CG, DDG-51, LCS, F-35, F/A-18
ESSM	RIM-162	Evolved Sea Sparrow - Medium Range Surface to Air Missile	CVN, LHA/D, LPD, CG, DDG-51, DDG-1000, LCS
Sidewinder	AIM-9	Short Range Air to Air	F-35, F/A-18, EA-18, AH-1
Hellfire	AGM-114	Short Range Air to Surface	F-35, F/A-18, MH-60, AH-1, MQ-8, TERN
AMRAAM	AIM-120	Advanced Medium Range Air	F-35, F/A-18
HARM	AGM-88	High Speed Anti-Radiation	F-35, F/A-18

The current operations for efficient ships are enabled by multiple systems and multiple weapons systems on a single platform. However, with DMO, "rather than heavily invest in expensive and exquisite capabilities that regional aggressors have optimized their forces to target, naval forces will persist forward with many smaller, low signature, affordable platforms" (Blivas, 2020), for example, integrating the Marine's EABO systems. These smaller platforms will be greatly advantaged by employment from EABOs situated on partner territory in proximity to close and confined seas (Corbett, 2018). Also, even with the current capabilities, weapons and sensors do not have to be in the same platform to collaborate for a kill chain; for example, an Aegis ashore can launch a missile when another DDG detects a threat.

Table 3 lists examples of DMO tactics and counter-measures. As stated by Chung (2015), with the "increasing availability and proliferation of unmanned system technologies, such as unmanned aerial vehicles (UAVs) in civilian and military applications, both opportunities and challenges arise in addressing large numbers of robots capable of collective interactions." Swarm, described as a cooperative system comprised of numerous UAVs that function with limited operator involvement (Lachow, 2017) or as a tactic for deception, including saturation of radar and detection systems by deploying a large number of remotely piloted vehicles, as well as the ability to emulate a larger vessel such as a surface combatant or manned aircraft by radiating active emissions from the unmanned systems.

IR Smoke can be used as a decoy for heat-seeking sensors and weapons. Electronic jamming is a function within the EW subcomponent of electronic attack and serves to overwhelm or deceive a sensor through the controlled and directed propagation of electromagnetic signals. Electronic jamming can also exploit a specific vulnerability such as the reliance on a single frequency. An EMCON employs measures to reduce the electromagnetic, acoustic, heat, and radar cross section signatures from the platform. For example, a ship or aircraft can limit nearly all navigation, communications, propulsion, and weapons systems to the minimum in order to reduce the probability of being detected.

Another example is developing detection capabilities of signature deception for DMO and EABO since emerging ISR capabilities mean the applicability of DMO, and EABO is dependent on a competition of detection. The deliberate use of signature emission to deceive a self-player and use of detection methods to detect deception have the potential.

Table 4. Examples of Blue Force DMO's Tactics (Popa et al., 2018)

Variable	Minimum	Maximum	Type
Swarm	0	1	Discrete
Chaff	0	200	Continuous
Flares	0	50	Continuous
Visual Smoke	0	50	Continuous
IR Smoke	0	50	Continuous
Active Decoys	0	25	Continuous
Passive Decoys	0	300	Continuous
Spot Jamming	0	1	Discrete
Barrage Jamming	0	1	Discrete
Sweep Jamming	0	1	Discrete
DRFM Jamming	0	1	Discrete
GPS Jamming	0	1	Discrete
CG EMCON	0	1	Discrete
DDG-51 EMCON	0	1	Discrete
DDG-1000 EMCON	0	1	Discrete

Application of Threat and Capability Coevolutionary Matrix

One CLA can associate with an asset, a platform, a unit, or a node, and the matrix in Table 1 can translate into the following representation in a CLA's association for a node:

- Capability_Platform_ddg-51 Capability_Sensor_visual 1
- Capability_Platform_ddg-51 Capability_Sensor_infrared 1
- Capability_Platform_ddg-51 Capability_Sensor_ESM 1
- Capability_Platform_ddg-51 Capability_Sensor_fire_control radar 1
- Capability_Platform_ddg-51 Capability_Sensor_phased_array_radar 1

Given a knowledge that the sensor of infrared is able to detect a threat platform x, an association between the threat's characteristics (e.g., platform x) and a capability's feature dimension can be stored in a node (e.g., Sensor 1 in Figure 4) as follows:

- Capability_Sensor_infared Threat_Platform_x 1
- Capability_Weapon_y Threat_Platform_x 1

The primary focus for collaborative assets with respect to DMO is to employ various traditional capabilities as well as DMO specific tactics and counter-measures that enable the disruption of the threat's kill chain to either prevent or lower the probability of the success of the threat. As shown in Figure 4, if a Threat Platform x shows up in a battlefield, the initial phase of a kill chain consists of a sequence of activities of sensor capability to detect and locate the threat. A Platform z equipped with CLA 0 in Figure 4 may send requests as inputs to its peer list of Sensor 1 (CLA 1) and Sensor 2 (CLA 2), which both have the infrared capability. Either of the sensors can potentially detect Threat Platform x and become the solution for Platform y's (CLA

0) request. The following other factors can be integrated into the TCCM's CLA's fitness computation:

- The probability of find or detection (POD), an important dimension for Platform z to make the decision of which solution to select from the sensors, may be different because of distance and range parameters.
- Even for the same sensor, POD can vary due to environmental factors such as weather, clutter, threat employed counter-measures, counter-engagements, and counter-targeting tactics. The environment in the vicinity of contested areas has the potential to impact the ability to perform DMO, specifically with respect to weather conditions and sea states, for example, the heavy presence of neutral commercial air and sea traffic (clutters) that cause significant congestion in sea lanes and air passages. For example, one-third of all global shipping passes through the South China Sea, as it is the one of the most used sea transit lanes in the world (Hoffmann et al., 2016). This may cause significant variation for POD, while an optical or infrared sensor can distinguish the threat as a legitimate target or neutral traffic; however, the fact can be also leveraged as an advantage for deception and decoy operations.
- A factor could be association constraints, for example, a U.S. aircraft carrier is a high value unit that is typically the highest targeting priority for the threat; therefore, the adversary's combat capable platforms will have a non-zero probability of being assigned the CVN for targeting and engagement.
- Another challenge is that adversaries may use maritime militia fishing fleets that serve as non-militarized ISR platforms, and the blue forces include general lack of geographical familiarity with the region as well as considerations for the attempted control and management of the electromagnetic spectrum.

Considering all the factors into the fitness best solution to Platform z via CLA 1 is the highest fitness from both sensors, for example, Sensor 1 (CLA 1) is selected. Since the peer list of the Sensor 1 (CLA 1) only includes Weapon 2 (CLA 4), Weapon 2 (CLA 4) is used for engaging with the Threat in the next step of the kill chain. Should Weapon 2 (CLA 2) be selected, which is a peer for Weapon 1 (CLA 3), Weapon 1 (CLA 3) would be used in the engaging step.

If the threat is successfully engaged and killed, the adversaries may try to learn from the experience (data) and may try to avoid the detection by moving away from the last location of detection and engagement. For example, the Threat Platform x may try to move to a different location that is away from the combination Sensor 1 (CLA 1) and Weapon 2 (CLA 4); because of the distributed and different peer lists and combinations of the self-player's assets, the self-player (blue) may be not predictable from the experience for the Threat, since the Threat can be caught up with Sensor 2 (CLA 2) and Weapon 1 (CLA 3) should the Threat move to a different location.

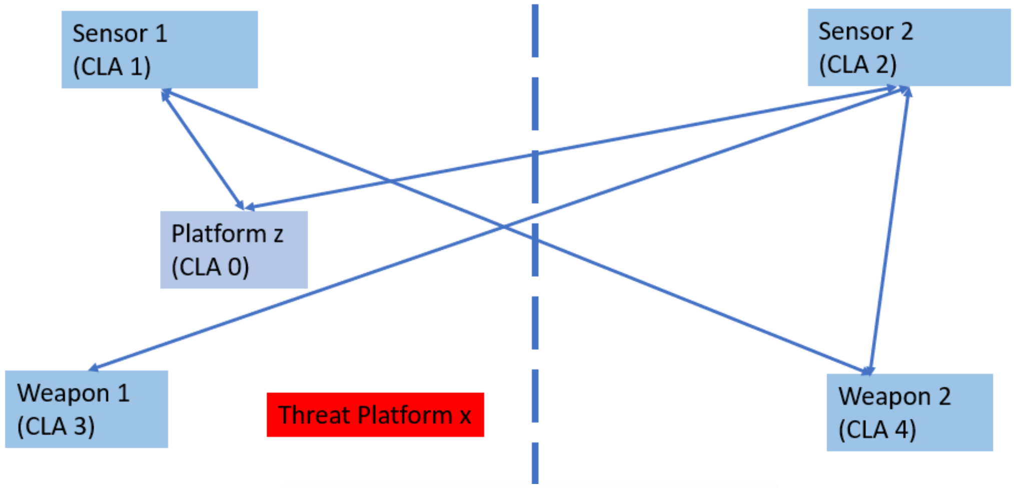


Figure 4. A TCCM Example for Illustration

Therefore, the following list shows the advantages of TCCM:

- TCCM uses the peer lists to combine capabilities at the various stages of a kill chain to disrupt and degrade the threat progress to its target of blue and confuse its learning. Even the self-player (blue force) can become less predictable or unpredictable if the peer lists are purposely altered randomly and periodically.
- TCCM represents pooled resources, and the model represents DMO as a united network of offensive lethality and firepower, and, therefore, all missiles are shared for cooperation and collaboration for all the assets and CLAs.
- Because of the potential dimensions, features, and characteristics of threats and capabilities, and their combinations, peer lists can be extremely large. For example, with the current systems, there are many different types of procedures or action chains used to conduct a detect-to-engage (DTE) series of events where sub-tasks must occur for a weapons system to effectively engage a threat's platform or location. Combinations of detecting or finding the target, establishing a track on the target's location and movement, communication of targeting data between the sensor and weapon system, conducting the engagement with either kinetic or non-kinetic weapons, and evaluating the engagement to determine follow-on actions, can be extremely large. It is necessary to apply a systematic engine like TCCM to manage the large-scale tactical and distributed decisions.
- TCCM is fault tolerant since the network is peer-to-peer. The network can be resilient like swarm intelligence, and CLAs can be re-organized to form emerging patterns which can be effective out of the box, given a wide range of contexts, and adapted to many others through reconfiguration and/or replacement (Goerger et al., 2014).

Conclusions

We show a TCCM and map it to a scenario that can help simulate threats and develop adaptations of capabilities for the benefits of the AAF, DMO, and EABO.

TCCM has the potential to systematically optimize, recommend, and evolve solutions to warfighters' requirements, which are more effective, suitable, survivable, sustainable, and affordable as a network of distributed and shared assets. A CLA network and collaboration mechanism in TCCM makes it fault-tolerant, self-organizing, adaptive, and resilient. TCCM contains a system of data strategy, distributed infrastructure, and deep analytics that can greatly assist reconstructing defense acquisition, improving process effectiveness, and implementing the AAF, DMO, and EABO.

Compared to the current method with less DMO and without CLA, the probabilities of detection and kill or fitness functions for a kill chain are modeled adaptively and are therefore much less predictable by the opponent, threat, or adversaries' point of view, potentially adding to the concept and desired outcome of DMO for offense and defense. Should the opponent also adapt such a strategy, because of the asymmetry of assets and capabilities, the self-player is potentially still more advantageous over the opponent. Future work will include quantitative simulation to implement TCCM.

References

Arora, S., Ge, R., Liang, Y., Ma, T., & Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (GANs). *Proceedings of the 34th International Conference on Machine Learning*, *70*, 224–232. http://proceedings.mlr.press/v70/arora17a.html

Blivas, A. (2020). 6 platforms for Marine expeditionary advanced base operations logistics. *The Diplomat*. https://thediplomat.com/2020/11/6-platforms-for-marine-expeditionary-advanced-base-operations-logistics/

Brown, N., & Sandholm, T. (2017). Safe and nested endgame solving for imperfect-information games. *Proceedings of the AAAI Workshop on Computer Poker and Imperfect Information Games*. https://arxiv.org/pdf/1705.02955.pdf

Canfield, J. (2017). *Fleet design* [Presentation]. NPS CRUSER Warfare Innovation Continuum Workshop, Naval Postgraduate School, Monterey, CA.

Chung, T. H. (2015). Advancing autonomous swarm capabilities: From simulation to experimentation. *Proceedings of the 2015 Winter Simulation Conference*. https://www.informs-sim.org/wsc15papers/236.pdf

Colpo, D. (2016, October 26). *NDIA C41 industry day* [PowerPoint slides]. Program Executive Office Command, Control, Communications, Computers and Intelligence (PEO C41). https://www.ndia-sd.org/wp-content/uploads/2017/02/2.PMW%20150_NDIA%20Fall%20Forum%202016%20Brief_OCT16.pdf

Corbett, A. (2018). *Expeditionary advanced base operations (EABO) handbook: Considerations for force development and employment*. Marine Corps Warfighting Lab, Concepts & Plans Division. https://mca-marines.org/wp-content/uploads/Expeditionary-Advanced-Base-Operations-EABO-handbook-1.1.pdf

DoD. (2018). *Summary of the 2018 National Defense Strategy of the United States of America: Sharpening the American military's competitive edge*. https://www.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf

Goerger, S. R., Madni, A. M., & Eslinger, O. J. (2014). Engineered resilient systems: A DoD perspective. *Procedia Computer Science*, *28*, 865–872.

Goldberg, G. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680. https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

Hoffmann, J., et al. (2016). *Review of maritime transport 2016*. United Nations Conference on Trade and Development. http://unctad.org/en/PublicationsLibrary/rmt2016_en.pdf

Joint Chiefs of Staff. (2013). Simplification of F2T2EA kill chain to FTE process. In *Joint Targeting* (Joint Publication 3-60). https://www.justsecurity.org/wp-content/uploads/2015/06/Joint_Chiefs-Joint_Targeting_20130131.pdf

Lachow, I. (2017). The upside and downside of swarming drones. *Bulletin of Atomic Scientists, 73,* 96–101. https://www.tandfonline.com/doi/full/10.1080/00963402.2017.1290879

Navy Warfare Development Command. (2017). *CNO visits Navy Warfare Development Command*. https://www.nwdc.usff.navy.mil/Press-Room/News-Stories/Article/2406186/cno-visits-navy-warfare-development-command/

O'Reilly, U., & Hemberg, E. (2018). An artificial coevolutionary framework for adversarial AI. *Proceedings of AAAI Fall Symposium: Adversary-Aware Learning Techniques and Trends in Cybersecurity (ALEC)*. http://ceur-ws.org/Vol-2269/FSS-18_paper_37.pdf

Popa, C., et al. (2018). *Distributed maritime operations and unmanned systems tactical development* (NPS capstone project report). https://calhoun.nps.edu/handle/10945/59587

Popovici, E., Bucci, A., Wiegand, R. P., & De Jong, E. D. (2012). *Coevolutionary principles.* In G. Rozenberg, T. Back, and J. N. Kok (Eds.), *Handbook of natural computing* (pp. 987–1033). Springer.

Rowden, T. S. (2017). The U.S. Navy's surface force strategy: Return to sea control. *CHIPS*. https://www.doncio.navy.mil/Chips/ArticleDetails.aspx?ID=8574

Office of the Under Secretary of Defense for Acquisition and Sustainment. (n.d.). https://www.acq.osd.mil

Office of the Under Secretary of Defense for Acquisition and Sustainment. (2020). *Operation of the adaptive acquisition framework* (DoD Instruction 5000.02). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500002p.pdf?ver=2019-05-01-151755-110

Zhao, Y., & Zhou, C. (2014). *System and method for knowledge pattern search from networked agents* (U.S. Patent No. 8,903,756). U.S. Patent and Trademark Office. https://www.google.com/patents/US8903756

Zhao, Y., & Zhou, C. (June, 2019). Collaborative learning agents (CLA) for swarm intelligence and applications to health monitoring of system of systems. In J. Rodrigues et al. (Eds.), *Computational Science – ICCS 2019* (pp. 706–718), *Lecture Notes in Computer Science* (Vol. 11538). Springer. https://link.springer.com/chapter/10.1007/978-3-030-22744-9_55

Zhou, C., Zhao, Y., & Kotak, C. (2009). The collaborative learning agent (CLA) in Trident Warrior 08 exercise. *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR)* (Vol. 1, pp. 323–328). https://doi.org/10.5220/0002332903230328

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net