NPS-PM-23-207

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Case Study: Review and Analysis of Software Development Program Management within the Procurement Integrated Enterprise Environment (PIEE) Suite Platform

June 2023

Jefferson A. Clark, CIV
Jeffrey R. Dunlap, CIV

Thesis Advisors: Dr. Robert F. Mortlock, Professor
Jeffrey R. Dunlap, Lecturer

Department of Defense Management

Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program of the Department 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 the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

ABSTRACT

The Procurement Integrated Enterprise Environment (PIEE) serves as the Department of Defense (DOD) and federal government's centralized platform for procurement capabilities. Within the PIEE, various tools and features are available, including Wide Area Workflow, Electronic Data Access (EDA), MyInvoice, and functions like single sign-on and role-based access, tailored to specific job series. It integrates data from multiple applications, establishes standardized procurement hierarchy for purchase card and procurement processes, and covers pre-award, award, and post-award administration, payment, property management, contract closeout, and rapid application development and deployment. This research aims to examine and analyze the processes of software development program management, focusing on identifying lessons learned, evolutionary changes, and comparisons with similar efforts, while highlighting differences in outcomes. With the DOD's commitment to agile methodologies and robust cybersecurity integration, the PIEE suite embodies a legacy application that fosters extensive collaboration among services, defense agencies, and contractors, aiming to develop functional enterprise software at a reasonable cost to taxpayers. By incorporating these lessons learned and employing cross-functional management, defense leadership can effectively address challenges in software development and application delivery.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

We express our gratitude to the staff at the Naval Postgraduate School. Their expertise and commitment to excellence made this a rewarding experience. Additionally, we extend our sincerest appreciation to our advisors, Dr. Robert Mortlock and Dr. Jeffrey Dunlap, for their guidance and support. Similarly, we thank the Defense Contract Management Agency for this opportunity to obtain Master of Science in Contract Management degrees from the Naval Postgraduate School.

We also thank our leadership, colleagues, friends, and classmates who participated in this journey. Most importantly, we thank our families who supported us behind-the-scenes to enable us to successfully meet the demands of the last two years. Our success is the direct result of their sacrifices and support.

THIS PAGE INTENTIONALLY LEFT BLANK

NPS-PM-23-207

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Case Study: Review and Analysis of Software Development Program Management within the Procurement Integrated Enterprise Environment (PIEE) Suite Platform

June 2023

Jefferson A. Clark, CIV
Jeffrey R. Dunlap, CIV

Thesis Advisors: Dr. Robert F. Mortlock, Professor
Jeffrey R. Dunlap, Lecturer

Department of Defense Management

Naval Postgraduate School

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A&S	Acquisition and Sustainment
ACWS	Army Contract Writing System
ACWS	Army Contract Writing Systems
ADA	Anti-Deficiency Act
AI	Artificial Intelligence
AT-AT	Audit Tracker and Action Tool
BIO	Business Integration Office
CAGE	Commercial and Government Entity
CAP	Capstone Applied Project
CAR	Contract Action Reports
CBAR	Contractor Business Analysis Repository
CCO	Contract Closeout
CFD	Cumulative Flow Diagram
CLS	Clause Logic Service
COE	Common Operation Environment
CON-IT	Contracting-Information Technology
COR	Contracting Officer's Representative
COTS	Commercial Off-The-Shelf
CPFF	Cost-Plus Fixed Fee
CPIF	Cost-Plus Incentive Fee
CSIS	Center for Strategic and International Studies
CSV	Comma-Separated Values
CUI	Controlled Unclassified Information

DCAA	Defense Contract Auditing Agency
DCMA	Defense Contract Management Agency
DFARS	Defense Federal Acquisition Regulations Supplement
DFAS	Defense Finance and Accounting Service
DIB	Defense Innovation Board
DLA	Defense Logistics Agency
DOD	Department of Defense
DODAAC	Department of Defense Activity Address Code
DPAP	Defense Procurement and Acquisition Policy
DPC	Defense Pricing and Contracting
DPC`	Defense Pricing and Contracting
DSM	Delivery Schedule Manager
DWCF	Defense Working Capital Fund
EAF	Enterprise Award File
EBCCB	Electronic Business Configuration Control Board
ECDM	Enterprise Communication & Deliverable Management
ECP	Engineering Change Proposals
ECWM	Enterprise Contract Writing Module
ePS	Electronic Procurement System
FAR	Federal Acquisition Regulation
FFP	Firm Fixed Price
FIFO	First-in-First-out
FPDS-NG	Federal Procurement Data System-Next Generation
FYDP	Future Years Defense Program

GAO	Government Accountability Office
GEX	Global Exchange
GIG	Global Information Grid
GOTS	Government Off The Shelf
GPC	Government Purchase Card
IDIQ	Indefinite Delivery Indefinite Quantity
IGCE	Independent Government Cost Estimates
iRAPT	Invoice Receipt Acceptance and Property Transfer
IWMS	Integrated Workload Management System
JEDI	Joint Enterprise Defense Infrastructure
JWCC	Joint War Cloud Capability
LOA	Line of Accounting
LOE	Line of Effort
MAI	Modernization and Analytics Initiative
MDO	Modifications and Delivery Orders
MIPR	Military Interdepartmental Purchase Requests
ML	Machine Learning
MOCAS	Mechanization of Contract Administration Services
MTTR	Mean Time to Restore
OASDA	Office of the Assistant Secretary of Defense for Acquisition
OASDS	Office of the Assistant Secretary of Defense for Sustainment
ONR	Office of Naval Research
ORC	Operational Requirements Committees
OSS	Open-source Software

OUSD	Office of the Undersecretary of Defense
OUSDC	Under Secretary of Defense Comptroller
P2P	Procure-to-Pay
PDF	Portable Document Format
PDREP	Product Data Reporting and Evaluation Program
PDS	Procurement Data Standard
PGI	Procedures, Guidance, and Information
PIEE	Procurement Integrated Enterprise Environment
PMO	Program Management Office
RIE	Range of Incentive Effectiveness
ROM	Rough Order of Magnitudes
SAM	System for Award Management
SDVOSB	Service-Disabled Veteran-Owned Small Business
SDW	Shared Data Warehouse
SLOA	Standard Line of Accounting
SSO	Single Sign-On
SWAP	Software Acquisition and Practices
UAT	User Acceptance Testing
VOSB	Veteran-Owned Small Business
WAWF	Wide Area Workflow
WIP	Work in Progress
WOSB	Woman-Owned Small Business

I. INTRODUCTION

Software and software development are important mechanisms within the Department of Defense (DOD). Software allows planes to fly, ships to sail, identify targets on the battlefield, transfer data, and many other applications. In recent years, the DOD has spent millions of dollars to procure and develop new applications and interfaces for program offices, technical specialists, contract administrators, and other functional specialists within the federal government to access and manage key programs for the military services. Some applications like eTools and the integrated workspace management system (IWMS) did not do well while others like the Procurement Integrated Enterprise Environment (PIEE) suite work well and are simple to use.

This Capstone Applied Project (CAP) aims to examine and analyze the software development program management processes associated with the PIEE platform. The study seeks to identify lessons learned from the development and evolution of the PIEE software, compare the PIEE software development processes to similar efforts, and explore the differences in outcomes. The research will involve analyzing the development process, assessing the use of agile and traditional software development methodologies, and identifying key factors that have contributed to the success of the PIEE platform. The results of this research will be useful for software development practitioners, project managers, and policymakers involved in the development of large-scale software systems for the DOD and federal government.

A. PROBLEM STATEMENT

Software development efforts, costs, and successes vary across the Department of Defense (DOD). The PIEE suite is growing rapidly, integrating and interfacing with acquisition and payment systems throughout the DOD, using agile development processes, and recently started to develop replacement applications for the Defense Contract Management Agency's eTools suite to provide enterprise capabilities and access to DOD entities. Given the number of players, scale of development, relatively low cost, and overall successful development of applications used throughout DOD acquisition, this program provides many lessons for similar efforts in DOD and federal acquisition.

B. PURPOSE STATEMENT

The purpose of this research is to identify and document lessons learned from the development efforts of the rapidly growing PIEE suite in the DOD. This research aims to provide insights into the successful use of agile development processes, and the integration and interfacing of acquisition and payment systems. Additionally, this program seeks to identify best practices for the development of replacement applications that provide enterprise capabilities and access to DOD entities, based on the relatively low cost and successful development of applications used throughout DOD acquisition. The ultimate goal of this research is to improve the efficiency and effectiveness of software development efforts in the DOD and federal acquisition, by leveraging the lessons learned from the PIEE suite development.

C. RESEARCH QUESTIONS

- What is the past, present, and planned/future state of the PIEE?
- How is PIEE development managed and can DOD leverage lessons to inform future, similar enterprise software development efforts?
- How do PIEE acquisition processes/decisions impact PIEE program management processes?
- What lessons can we take away from PIEE program management of software development efforts?
- How do PIEE efforts compare to the effectiveness, efficiency, and/or feasibility of similar efforts?

D. RESEARCH METHODS

The methodology used for this research is in the procedure of a qualitative case study of program management, implementation, sustainment, and funding specifically to the PIEE dashboard and application tools. This research will highlight how the DOD can leverage iterative software development and implementation to better align traditional (waterfall) and agile methodologies of new software development. The PIEE suite was initially a waterfall software development that has transitioned to an agile framework. The data collected in this research can be used for recommendations for improving development efforts in the DOD and federal acquisition of similar contexts. The research design was crafted to answer open-ended questions about lessons learned, program management of software development efforts, impacts, and effectiveness.

E. PROPOSED DATA, OBSERVATIONS, AND ANALYSIS METHODS

Data will be limited to available government publication on software development and DCMA/DLA PIEE suite development, deployment, and metrics. Information on the development environment, tools, regulations, policies, and tools will be collected and analyzed. The authors expect to correlate the data against existing literature, policies, and practices within the DOD umbrella.

F. POTENTIAL BENEFITS AND LIMITATIONS

The research conducted by the authors benefits the DOD in a variety of ways. First, the Procurement Integrated Enterprise Environment (PIEE) platform and its applications are broadly accessible across the federal government acquisition community, but primarily DOD acquisition, and some of its applications are required for DOD acquisition and its vendors to use. Unless policies change, which seems unlikely, this suite of applications will continue to facilitate or hamper DOD acquisition professionals well into the future. Second, DOD procurement software development and use in the PIEE suite appears remarkably successful, given its relatively short lifespan, its wide use, and the number of applications within the suite. Analyzing this success provides similar application development efforts with insight on how to improve their processes. Third, every program has room for improvement; this program is no different and through this analysis the authors plan to identify specific steps the PIEE program management office (PMO) and its business partners can take to maintain the successes of the current platform while enhancing these efforts and securing additional gains. Finally, analyzing this platform provides additional insight into the benefits and drawbacks of developing in an enterprise environment and collaborating with a variety of partners.

Unfortunately, all research has its limitations. The size of this endeavor produces the most significant constraint on this research. As mentioned previously, this is a DOD enterprise level program with numerous defense agencies and uniformed services playing a role and impacting the program's success. Analyzing all of it is an impossible task for two researchers and limited resources, but the authors plan to cover many significant parts of the program to create a solid foundation for analysis and future research. Another significant challenge related to the size of the program is the accessibility of information. Many players and their respective agencies possess different pieces of the puzzle and it is unlikely the authors can obtain access to all of it or even awareness of all that is available. However, the authors have access to a great deal of publicly available information, as well as data sources internal to their agency, and plan to leverage this these resources as much as possible. Finally, it is likely some data simply is not collected or is in an unusable condition. The authors plan to identify such issues to provide recommendations on how to create and maintain better data sources to inform program managers in the future.

G. ORGANIZATION OF THE THESES

Chapter II is a literature review of topics relevant to this case study. The literature review covers the following topics: The state of DOD's acquisition software, DOD modernization, developing new software, and performance metrics.

Chapter III contains analysis of the PIEE contract, program management, contractor performance, and user interfaces. The analysis covers the background of the PIEE suite and ownership, sustainment, and enhancements. This chapter will also discuss cost comparisons to other DOD acquisition software, capabilities, and collaboration of efforts.

Chapter IV provides a summary of conclusions, research synopsis, and recommendations. The chapter concludes with recommendations for future research.

II. LITERATURE REVIEW

The Department of Defense (DOD) has been facing challenges in keeping up with the rapidly evolving software development industry, as other great powers such as China and Russia develop software capabilities more rapidly (Vergun, 2021). The DOD is constantly requiring new software, enhancements, replacements, and modernization of legacy systems, and developing exponentially more lines of code for newly developed weapons systems than even the most recently developed systems (Defense Science Board, 2000). Private industry, on the other hand, develops new software incrementally and rolls it out to users in as little as every two weeks (Government Accountability Office, 2021b). To address these challenges, the DOD has refocused its attention on software development, working toward developing it "at the speed of relevance" (DOD, 2021c, p. 1). This effort has led to the development of a new acquisition framework consisting of streamlined acquisition pathways, including one specifically for software acquisition (DODI, 2022).

This new acquisition framework stresses the use of agile software development methodologies, which enable iterative and incremental development, collaboration between developers and users, and a focus on delivering software that meets user needs quicker than traditional waterfall development (DOD, 2021c). The agile approach also highlights the use of development, security, and operations (DevSecOps), which integrates security into the software development process and ensures that security is integrated throughout the software development life cycle (DOD, 2021c). The acquisition framework represents a significant shift in how the DOD approaches software development and acquisition, enabling it to develop and acquire software more rapidly, reduce costs, and improve the quality and security of the software it deploys. (GAO, 2021b).

However, the DOD is reliant on legacy software applications and often updates or integrates applications instead of developing or purchasing new software (DOD, 2021c). This chapter describes several of the DOD's past and present software methodologies, legacy systems, advantages, disadvantages, acquisition of software, costs, and metrics.

A. STATE OF DOD'S ACQUISITION SOFTWARE

The DOD acquisition of software has been a recurring topic of concern and criticism for the last several years. Despite efforts to improve the acquisition process using technology, the DOD has struggled to modernize its legacy software and keep pace with advances in the industry. In a report by the Government Accountability Office (GAO) (2020), they found that the DOD's acquisition software was outdated and did not satisfy the current or future needs of the organization. They also recommended the DOD take steps to invest in new technologies to improve the acquisition process (GAO, 2020).

The DOD's acquisition software is often overly complex and difficult to use, leading to delays and increased costs. In a report by the Defense Innovation Board (DIB) (2018), recommended that the DOD adopt more user-friendly software and invest in training to improve usability. In a research report about the future of the Army by David Barno and Nora Bensahel, found the DOD's legacy acquisition software was often fragmented and lacked integration with other systems, leading to inefficiencies and errors (Barno & Bensahel, 2016).

1. Legacy Systems

The DOD relies heavily on legacy systems software to support its various operations like finance, contract management, and security investigations. These legacy systems have been in use for several years and are often outdated, making them difficult to maintain and costly to operate. For example, the mechanization of contract administration services, better known as MOCAS, has been in use since 1958 (Fossbytes, 2017). The system is known for having inaccurate data, lacking integration with other systems, and relies heavily on user inputs to update contracting payment information (GAO, 1998).

The invoice receipt acceptance and property transfer (iRAPT) better known as wide area workflow (WAWF), is another DOD legacy system used for electronic invoicing, receipts, and acceptance for government vendors. WAWF is part of the procurement integrated enterprise environment (PIEE) suite along with MOCAS and other applications used to track and monitor contracts and payments (DLA, n.d.-b). WAWF provides inspectors, acceptors, auditors, and pay officials with a single point application within the PIEE suite to approve vendor products and services (DLA, n.d.-a).

Although legacy applications like MOCAS and WAWF are easily accessible within PIEE, the applications are overly complex and difficult to use, leading to delays and increased costs (DIB, 2019). The DIB (2019) recommended the DOD adopt more user-friendly software applications and interfaces and invest in training to improve usability and develop a more integrated software architecture to streamline operations and better meet the needs of the organization and key stakeholders. Figure 1 provides a visual of the WAWF inspector process.

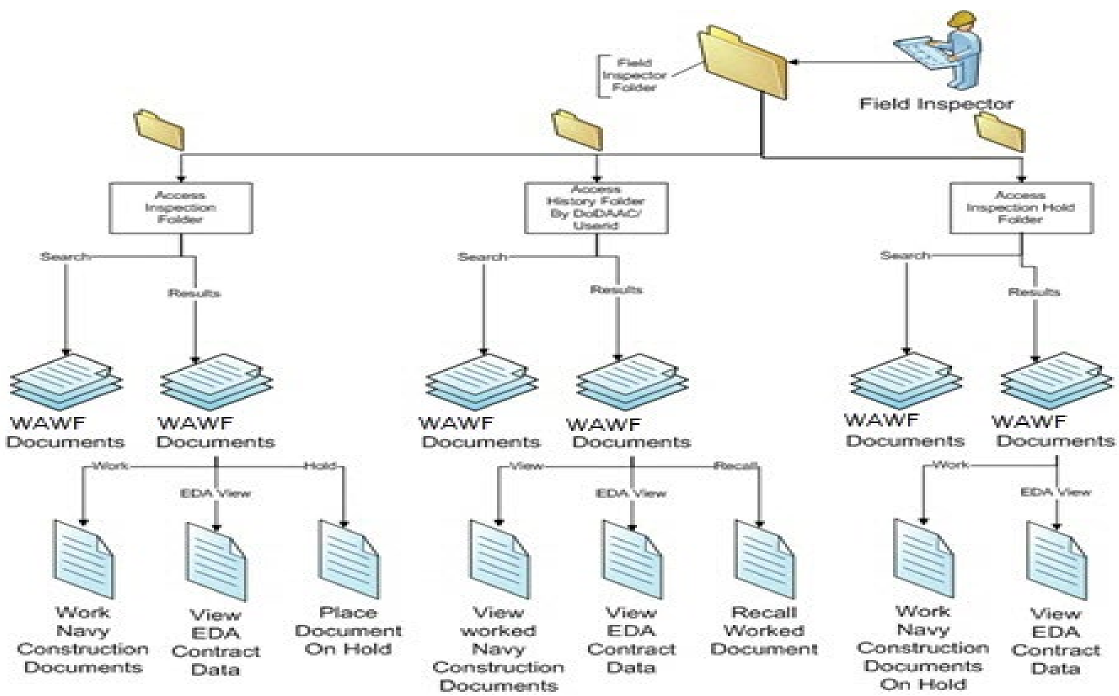

Figure 1. WAWF Inspector Process. Source: DLA (n.d.).

B. DOD MODERNIZATION

The DOD continues to pursue modernization efforts to enhance its capabilities and technological infrastructure through transaction services. These efforts aim to improve the DOD's agility, lethality, and readiness to meet emerging threats as well as modernizing its legacy systems to improve its efficiency and effectiveness in the rapidly evolving technological landscape (DOD, 2021c). In 2018, the DOD approved its software modernization strategy building on its Cloud Strategy identifying goals to (DOD, n.d.):

- Accelerate the DOD Enterprise Cloud Environment
- Establish Department-Wide Software Factory Ecosystem
- Transform Process to Enable Resilience and Speed DOD (n.d.).

The modernization strategy will continue to leverage the DOD's competitive advantage in the global military power struggle. Whether it's an aircraft, a ship, or a management tool application, software enables the U.S. military to maintain an edge over its enemies.

1. Transaction Services

The DOD has been implementing various efforts to modernize its acquisition software transaction services in recent years. These efforts include the adoption of agile methodologies, the use of cloud computing, and the implementation of DevOps practices. One of the significant initiatives in this regard is the DIB's Software Acquisition and Practices (SWAP) Study where it aims to provide recommendations and improvements to the acquisition of software-intensive systems within the DOD (DIB, 2020). Figure 2 shows the SWAP lines of effort to improve the acquisition process. The study emphasizes the need for the DOD to adopt agile development methodologies and DevOps practices, as well as to increase its use of open-source software (OSS) and commercial off-the-shelf (COTS) products (DIB, 2020).

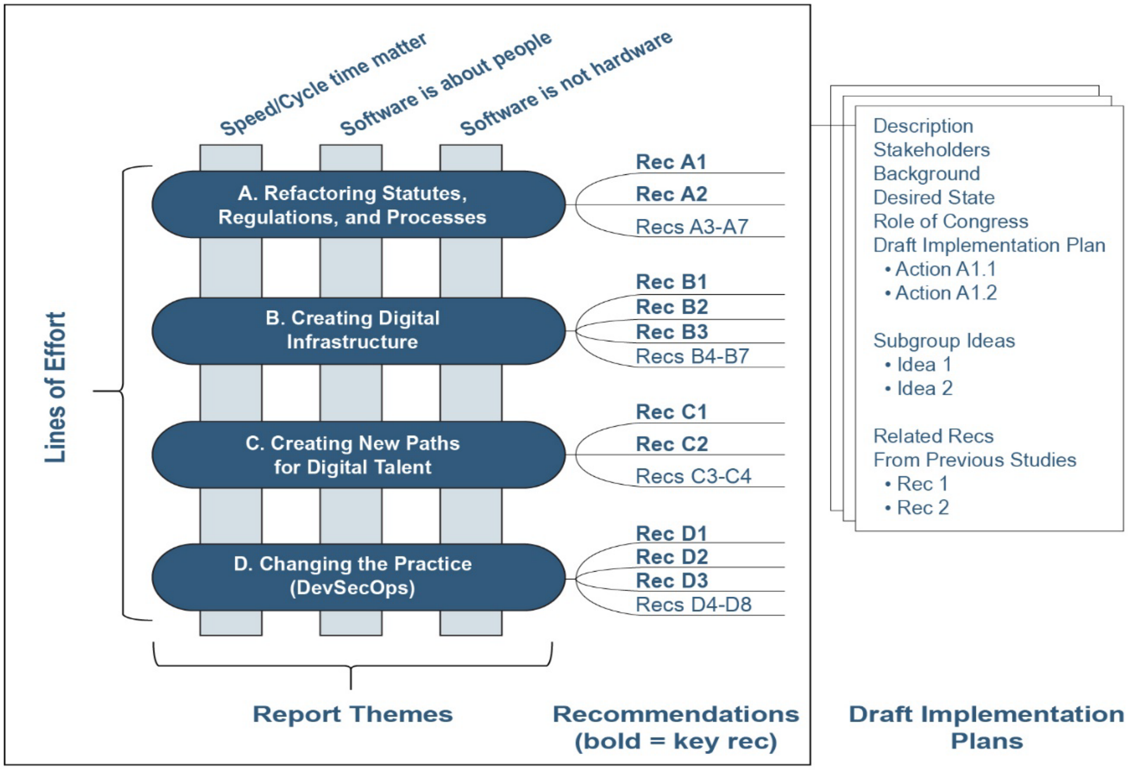

Figure 2. SWAP Lines of Effort. Source: SWAP (n.d.).

a. IMPROVING TRANSACTION SERVICES

For decades, the Department of Defense has sought greater interoperability and data sharing between military branches and defense agencies through leveraging shared services, a central infrastructure network, and cloud-based services. In 2011, the DOD implemented the Global Information Grid (GIG) infrastructure, which is a "globally interconnected end-to-end set of information capabilities for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policymakers, and support personnel" (DISA, n.d.-b). The GIG incorporates standard interfaces, protocols, and real-time information data formats that enable interoperability between different systems and applications to enable agile information capabilities (DISA, n.d.-b). All DOD acquisition programs will connect to the GIG as an effort to increase AI and machine learning (ML) technologies, which are critical technologies that can enhance the DOD decision-making processes, optimize resource allocation, and improve operational efficiencies (DARPA, n.d.).

In 2019, the DOD awarded the Joint Enterprise Defense Infrastructure (JEDI) which transitioned to the Joint War Cloud Capability (JWCC) to be the government's consolidated IT effort (Lohrmann, 2021). Figure 3 shows the JEDI cloud strategy and optimization. This cloud-based effort will span all three security levels across the DOD and provide the warfighter real-time data and AI capabilities in garrison and the battlefield (DOD, n.d.). During a media discussion at the Pentagon, John Sherman stated "the JWCC will serve that purpose and be a bridge to our longer-term approach, allowing us to leverage cloud technology from headquarters to the tactical edge, which will bolster our knowledge even further as we move to a full and open competition" (DOD, n.d.).

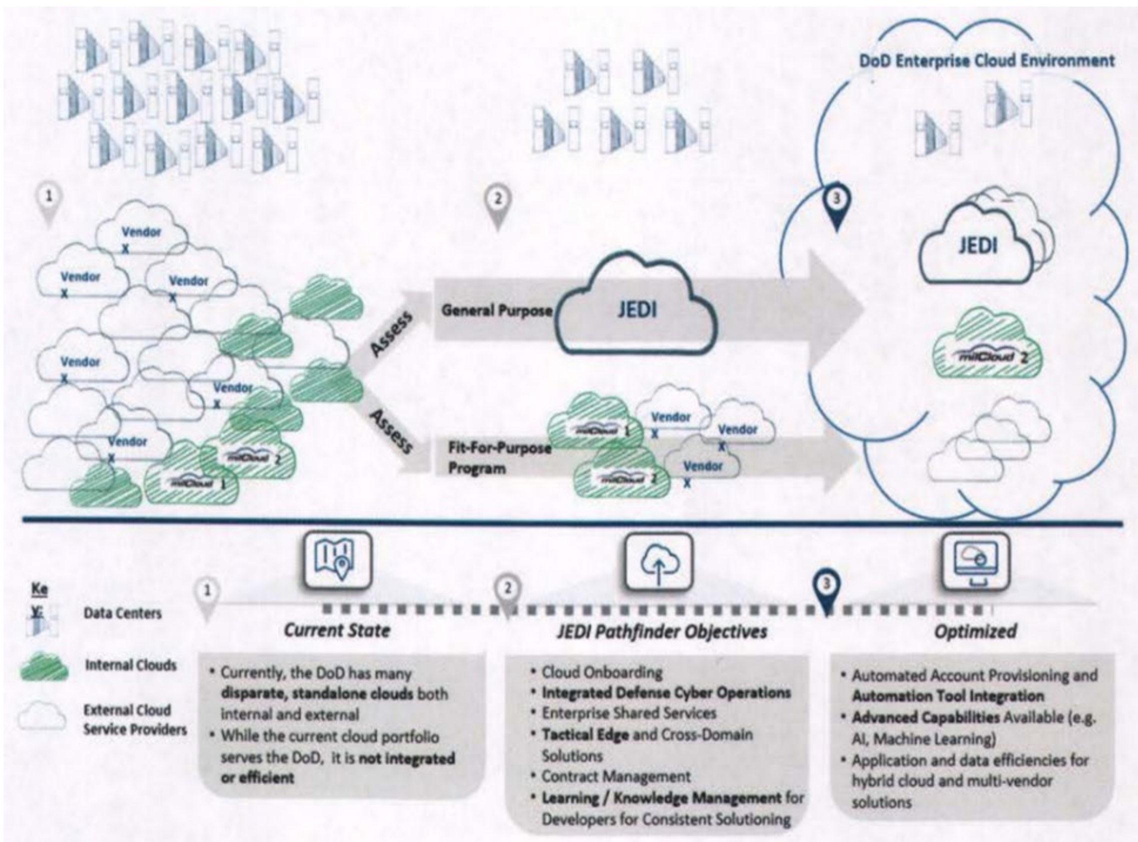

Figure 3. DOD Cloud Strategy. Source: DOD (2018a).

However, the DOD's reliance on software also poses significant challenges, particularly in the area of cybersecurity. The DOD's vast and complex software infrastructure makes it a prime target for cyber-attacks, and vulnerabilities in software can be exploited to compromise the security and integrity of the DOD's operations (Moser, 2021). To address this challenge, the DOD is investing in cybersecurity measures, such as the Cybersecurity Maturity Model Certification (CMMC) program, to ensure that its software applications meet minimum cybersecurity standards and posture through the supply chain (Department of Defense, n.d.). The CMMC program aims to ensure that all companies that do business with the DOD meet a minimum level of cybersecurity maturity, thereby reducing the risk of cyber-attacks on the DOD's supply chain.

C. DEVELOPING NEW SOFTWARE

Software development methodology is a structured approach for developing, maintaining, and delivering software systems. They provide a framework for organizing tasks and activities in software development, with the aim of improving the quality of the final product, reducing development time, and minimizing costs. According to Singh and Chhabra (2015), software development methodologies are divided into two main types: traditional and agile. Traditional methodologies, like the waterfall model, follow a sequential approach to software development. agile methodologies, such as Scrums, emphasize collaboration, flexibility, and iterative development (Ambler, 2009). The choice of method depends on factors, like project size, complexity, team size, and client requirements.

1. Waterfall

The waterfall methodology is a conventional project management approach that follows a linear and sequential process with specific deliverables for each project phase. Royce (1970) suggests that this methodology is suitable for projects with well-defined requirements, low risk, and stable technology base. The methodology includes "distinct phases, such as requirements gathering, design, implementation, testing, and maintenance, and each phase must be completed before the next one can begin" as seen in Figure 4 (Royce, 1970). Any changes in requirements or design must be carefully managed to avoid project timeline disruptions. Although the waterfall methodology has been widely used in software development, it has been criticized for its inflexibility and lack of responsiveness to changing requirements (Beck, 2000). However, proponents of this methodology argue that its structured approach can be effective in managing projects with well-defined requirements and limited risk (Royce, 1998).

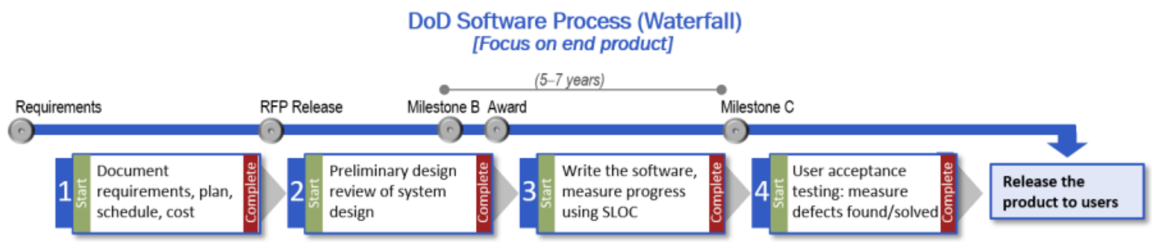

Figure 4. Waterfall Process. Source: DSB (2018).

The advantages of the Waterfall model include its structured and organized approach to software development, with clear phases and deliverables, ensuring timely and within budget project completion (Royce, 1970). It also has clearly defined milestones and deliverables, keeping stakeholders informed of project progress (Boehm, 1988). Moreover, this methodology is suitable for stable and well-defined projects where requirements are established at the project outset and are unlikely to change significantly (Royce, 1998).

However, the Waterfall model also has some disadvantages. It is inflexible to changes in requirements or design, resulting in delays or increased costs if changes are necessary (Beck, 2000). The model does not emphasize customer involvement during the development process, leading to a product that may not meet the customer's needs (Royce, 1998). Finally, the Waterfall model can have a high risk of failure, especially for complex projects where it may be difficult to anticipate all the requirements and risks at the project outset (Boehm, 1988).

2. Agile

Agile software development is an approach that emphasizes iterative and incremental development, frequent delivery of working software, and collaboration among cross-functional teams. The agile approach can be traced back to the early 1990s, when a group of software developers began experimenting with a new approach to software development that emphasized close collaboration between developers and customers and the use of small, self-organizing teams (Beck et al., 2001). In 2001, the collaboration produced The Agile Manifesto, which is a statement of the core values and principles of agile software development. The manifesto is defined by four core values and 12 principles that redefine software development to replace the traditional waterfall methodology (Beck et al., 2001). The four core values are (Beck et al., 2001):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan Becket (2001).

Agile development methodology is an iterative and "incremental approach to software development that emphasizes flexibility," collaboration, and rapid delivery of working software (Beck et al., 2001). According to the Agile Alliance, the Agile Manifesto values "individuals and interactions, working software, customer collaboration, and responding to change" (Agile Alliance, n.d.). Agile development typically involves working in small, self-organizing teams that work closely with stakeholders to deliver working software in short iterations, usually two to four weeks in length (Cockburn, 2001). Each iteration includes planning, development, testing, and review, with feedback from stakeholders used to inform the next iteration. The iteration process is better dined in the 12 key principles of agile development (Beck et al., 2001):

1. [The] highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly Becket (2001).

Agile software development methodologies have gained popularity in recent years due to their ability to improve software development processes and reduce costs. This has led the DOD to adopt agile methodologies for its software development projects. One major benefit of using agile software development methodologies in the DOD is the ability to deliver software projects in shorter time frames. This is important because many of the DOD's software projects have strict timelines and deadlines. According to Boehm and Turner (2004), agile methodologies allow for "incremental and iterative delivery of working software, allowing stakeholders to provide feedback and adjust requirements throughout the development process" (p. 37). This feedback loop allows for continuous improvement and ensures that end products meet the needs of the stakeholders as shown in Figure 5.

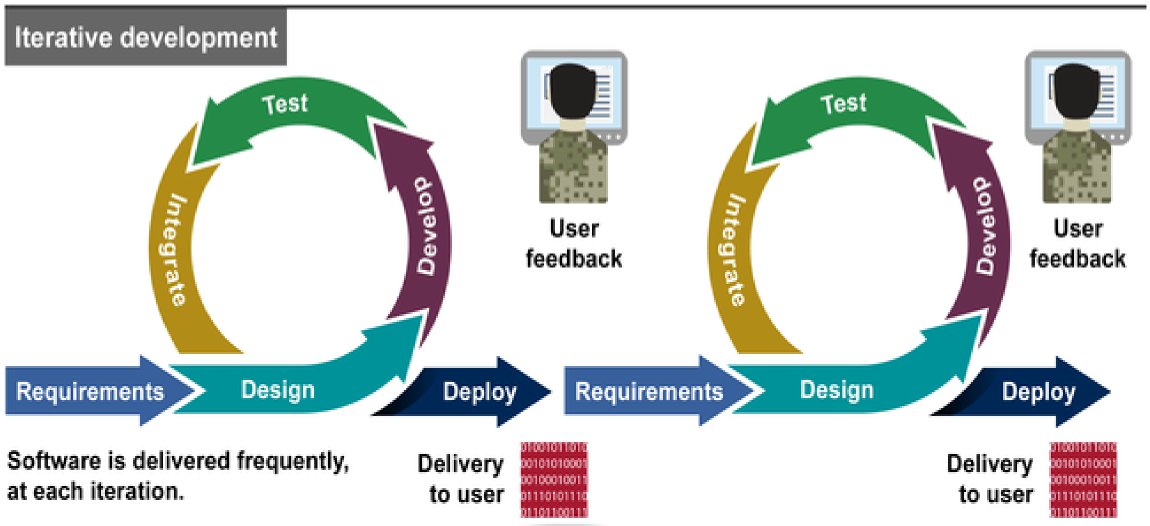

Figure 5. DOD Iterative Deployment Strategy. Source: GAO (2021b).

Use of agile software development methodologies in the DOD has increased collaboration between the development team and the stakeholders. Most of the DOD software projects involve complex requirements and multiple stakeholders across several geographic locations. According to Kim and Lee (2018), agile methodologies "emphasize close collaboration between the development team and stakeholders, enabling the team to respond to changing requirements and priorities in real-time" (p. 232). This collaboration ensures that the software meets the needs of the stakeholders and reduces the risk of project failure.

Despite the benefits of using agile software development methodologies, cultural resistance to change is a main challenge associated with its implementation. According to Swider and Zimmermann (2016), "the culture of the DOD is typically risk-averse, hierarchical, and resistant to change, which can make it difficult to adopt agile methodologies" (p. 57). Kassab (2018), furthered the argument noting "agile methodologies require a specific skill set and mindset, which may not be present in the current workforce" (p. 107). To be successful, the DOD must realize that not every new software application will use the agile method and having specialized software teams that are trained in traditional and agile frameworks will help guide the agency to formulating best decisions.

3. Scrum

Scrum is lightweight agile framework that helps people and teams through adaptive solutions of complex problems (Schwaber & Sutherland, 2017). The name scrum is a term used in the global sport of Rugby, where team members bind to each other and work together to achieve goals and support each other. Scrum is made up of five life "values of commitment, focus, openness, respect, and courage" (Schwaber & Sutherland, 2017). These values encourage and direct the team's actions and behavior.

Scrum theory is the basis for empiricism, which enables better decision-making by gaining knowledge through experience and observation (Schwaber & Sutherland, 2017). Transparency, inspection, and adaptation are the three pillars that sit upon this foundation and reduce risk when they combine. Transparency allows visibility to make the best decisions and enables inspection (Schwaber & Sutherland, 2017). Inspection is performed diligently and frequently to create a cadence where detection of variances and problems are quickly identified (Schwaber & Sutherland, 2017). Inspection leads to adaptation where the Scrum Team can adapt and adjust when information is learned during inspection (Schwaber & Sutherland, 2017).

a. The Scrum Team

Scrum Teams deliver products iteratively and incrementally to ensure a useful version of the working product is available. The Scrum Team consists of the "Product Owner, the Development Team, and the Scrum Master. The Product Owner is responsible for maximizing the value of the product" and managing the Product Backlog (Schwaber & Sutherland, 2017). A Development Team is a small four-to-eight-person team that is responsible for delivering a potentially releasable increment of a "done" product (Schwaber & Sutherland, 2017). The Scrum Master is a servant-leader that ensures everyone understands the product domain, goals, and scope.

b. Scrum Artifacts

Scrum artifacts shown in Figure 6, are important information the scrum team uses to define the product and what work is needed to create a "done" product. There are three constants a scrum team will reflect on over time:

- Product Backlog: An emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team. Product Backlog items that can be done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.
- Sprint Backlog: Is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how). The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.
- Increment: (or Sprint Goal) is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable. Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value (Schwaber & Sutherland, 2017).

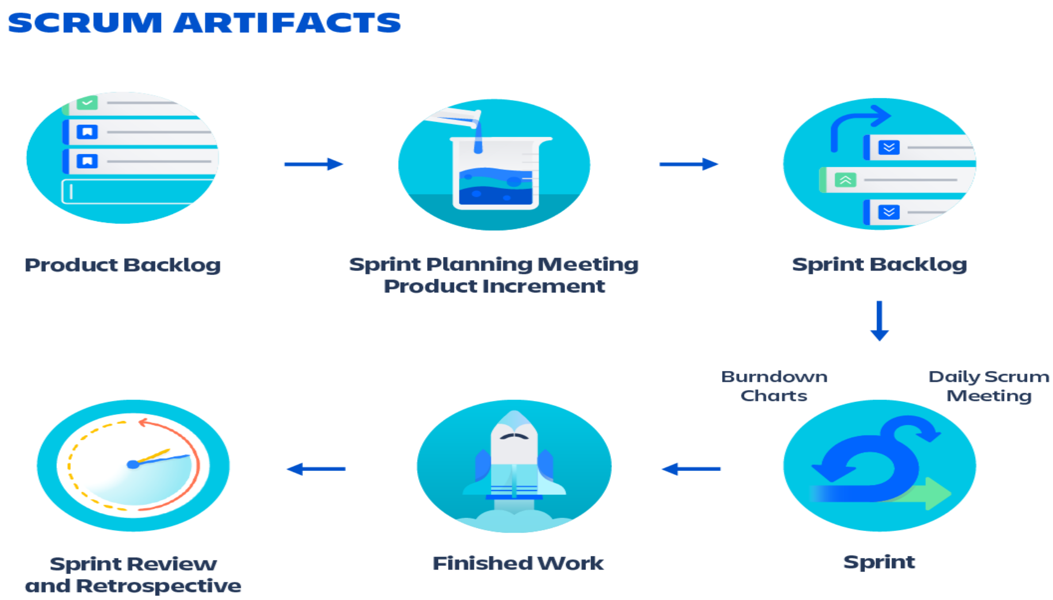

Figure 6. The Main Artifacts of Agile Scrum. Source: Atlassian (n.d.).

4. DevSecOps

Over the past few years, DevSecOps has become the most used agile framework for DOD software development. "DevSecOps is a core tenant of software modernization, technology transformation, and advancing the DOD's software development ecosystem to be more resilient" (DOD, 2021a). DevSecOps requires a significant cultural change to implement within an organization like the DOD. DevSecOps is not a different method like waterfall or scrum. The DevSecOps approach creates cross-functional teams that combine traditionally unrelated work scopes of development (Dev), cybersecurity (Sec), and operations (Ops) throughout the software life cycle. "As a unified team they follow agile principles and embrace a culture that recognizes resilient software is only possible" where quality, stability, and security intersect (DOD, 2021a). Figure 7 depicts the resilient software capabilities.

Figure 7. Pillars to Achieve Resilient Software Capabilities. Source: DOD (2021a).

DevSecOps attempts to remove shortfalls in incorporating security as an afterthought in software development as seen in Figure 8. The DOD's Chief Information Officer defines DevSecOps as (DOD, 2021a):

> An organization's cultural and technical practices, aligning them in such a way to enable the organization to reduce the gaps between a software developer team, a security team, and an operations team. Adoption improves processes through daily collaboration, agile workflows, and a continuous series of feedback loops DOD (2021a)

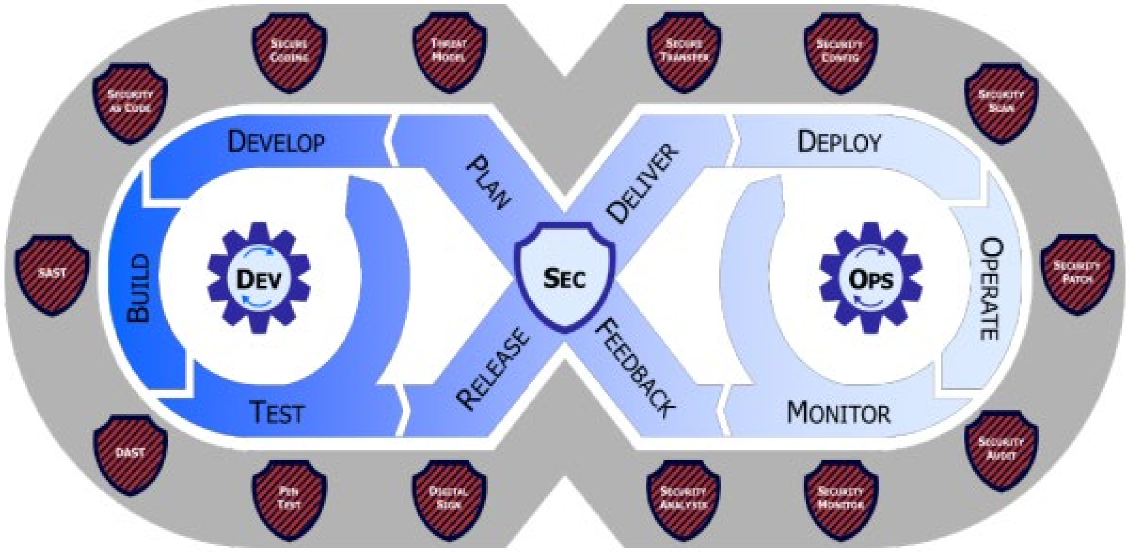

Figure 8. DevSecOps Life Cycle Phases and Philosophies.
Source: DOD (2021a).

Many of the DOD's current projects and missions lack agile standards and practices. This deficiency is prominent in cybersecurity frameworks as the focus is primarily on post-production deployments (DOD, 2021a). Currently, release cycles are "perceived as an uphill battle between development teams that attest to functionality, operational test and evaluation teams trying to confirm specific functionality, operations teams struggling to install and operate the product, and security teams bolting on protection mechanisms as an afterthought" (DOD, 2021a).

a. *DevSecOps Process*

The DevSecOps process requires efficient planning, design, and release of software through automated delivery paths. Implementation is a systematic approach consisting of ten distinct steps as shown in Figure 9. The DevSecOps Manifesto describes ten steps (Veritis, 2019):

1. Planning: Planning is the first approach to any task at hand and the core focus of DevSecOps—security—begins from here. In the planning stage, DevSecOps professionals must go beyond creating feature-based descriptions. The focus should also be on security and performance, acceptance test criteria, application interface and functionality and threat-defense models.

2. Developing: Developers should approach DevSecOps with a "how to do it" approach, rather than a "what to do" approach. It is important for developers to bring together available resources for guidance, have reliable practices and a code review system in place for themselves and for others in the team to follow.
3. Building: Automated build tools can uplift the whole DevSecOps implementation process tremendously. These tools ensure test-driven development, standards for release artifact generation and utilize tools to ensure the design aspect is in alignment with the team's coding and security standards through statistic code analysis.
4. Testing: Automated testing in DevSecOps should utilize strong testing practices including front-end, back-end, API, database and passive security testing.
5. Securing: Traditional testing methods always remain in place in DevSecOps exercise. However, somewhere down the line, there is a tendency to identify issues toward the end of the development process. Through advanced practices such as security scanning, we tend to become more aware of the issues and can determine if the threat is a serious one or not.
6. Deploying: Automated provisioning and deployment can fast-track the development process while making it a more consistent one. Infrastructure-as-code tools can perform the aforementioned audit properties and configurations and ensure secure configurations across the IT infrastructure.
7. Operating: Regular monitoring and upgrades are the Operations team's important tasks. DevSecOps teams ensure to deploy infrastructure-as-a-code tools to update and secure the entire organization's infrastructure in a quick and efficient manner with no scope for human error. Operations personnel have to be especially watchful of zero-day vulnerabilities.
8. Monitoring: Constantly keeping a watch for irregularities in security can save an organization from a breach. Hence, it is essential to implement a strong continuous monitoring program with real-time to keep a track of system performance and identify any exploits in their early stages.
9. Scaling: Gone are the days when organizations spent precious hours and money on the maintenance of large data centers. With the introduction of virtualization solutions and the cloud, organizations can scale their IT infrastructure or replace it in the event of a threat, which would be impossible to do with a traditional data center.
10. Adapting: Continuous improvement is key to any organization's growth. An organization will only be able to achieve the desired growth of it evolves in its practices including DevSecOps practices—security, functionality, and performance. Therefore, an organization should adapt to continuous improvement and external changing trends. (Veritis 2019)

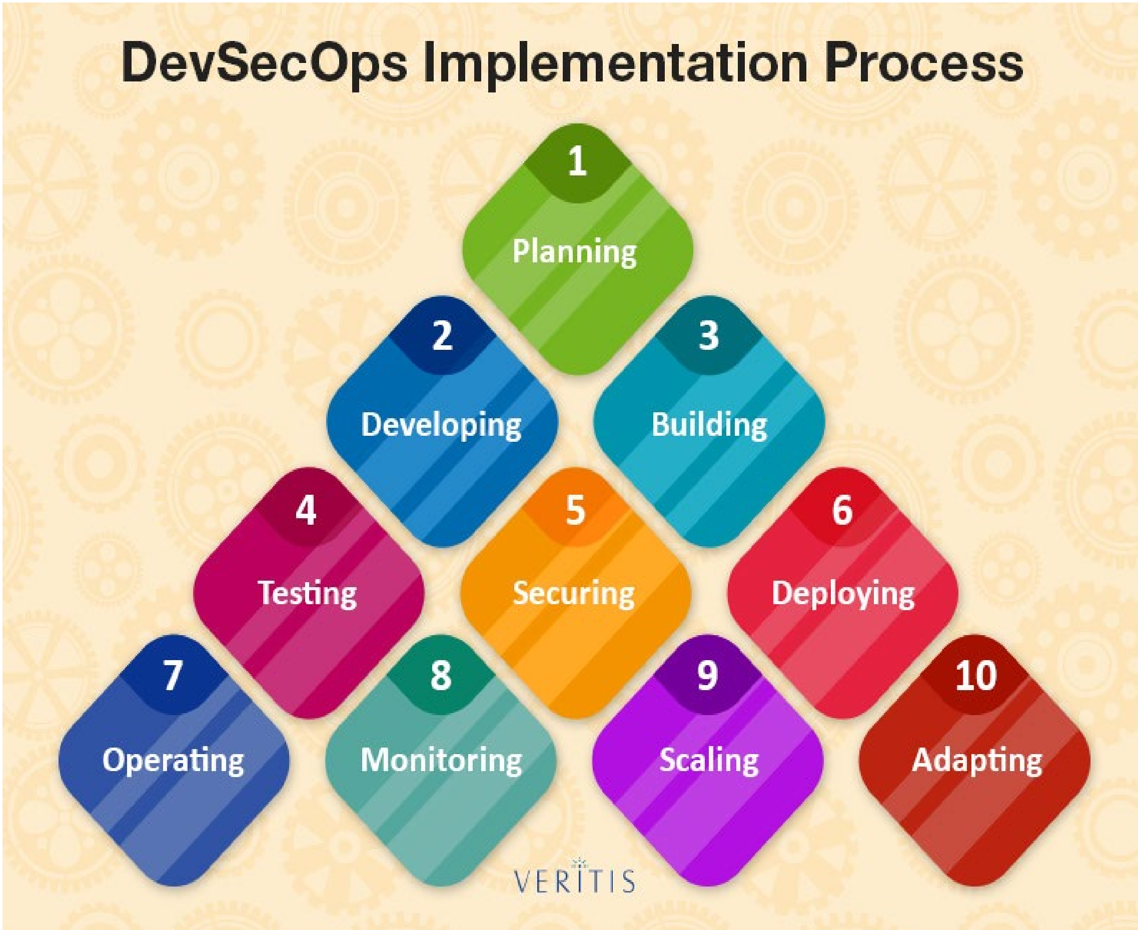

Figure 9. DevSecOps Ten Step Process. Source: Veritis (2019).

D. PERFORMANCE METRICS

Building the right team and culture is important for software development and implementation. But there needs to be a way to assess product quality and track team performance. Metrics provide key insights of agile team productivity and product capabilities as they move through the different stages of software development. Tracking and sharing agile metrics provides quantitative and qualitative feedback to agile teams, reducing confusion and frustration (Atlassian, 2019). The DOD tends to evaluate performance and quality through earned value management, where cost is the primary driver of a successful program, product, or service.

Agile methodologies use different metrics to evaluate incremental development through feedback, estimates, and charts or diagrams. In 2020, the DOD published its agile metrics guide to address the challenges and complexities of agile management. Agile metrics are (DOD, 2019a):

- "Story Points measure the complexity of a story" and are the building blocks for the team to estimate the complexities (or size) and work it can accomplish. Sizing is performed through sizing models like Fibonacci numbering or t-shirt sizzing.
- "Velocity measures the amount of work the agile team completes" during an increment. Velocity sets a benchmark for estimating and planning future increments and gives the team an average velocity over time.
- "Velocity Variance is the standard deviation of average velocity" to help with data point predictability and define acceptable variance thresholds.
- "Velocity Predictability is the measurement difference between planned and completed velocity" and is an indicator of process stability by identifying bottlenecks and work that stops first-in-first-out (FIFO) flows.
- "Story Completion Rate is the count of stories completed" divided by the count of stories planned and is a good way to communicate progress to the Product Owner and users.
- "Sprint Burndown Chart is used to provide a visual estimate of the pace and work accomplished daily." The agile team can see a line of the remaining hours of work and a line of the estimated completion date.
- "Release Burnup is a chart that measures the amount of work completed" for a given release based on totals of work planned. The agile team can see a visual of whether they are on track to com complete items needed for release.
- Cumulative Flow Diagram (CFD) provides a visualization showing the count of items at each step. The CFD shows Work in Progress (WIP), what is "To Do," and what is "Done" DOD (2019)

DevSecOps metrics measure what is in the organizations pipeline as well as the organization's ability to deliver, integrate, monitor, and restore products (DOD, 2019a). Organizations that actively measure efficiencies have greater insight into inefficiencies, pivot to improve the pipeline flow, and provide faster and consistent deliveries. DevSecOps metrics are (DOD, 2019a):

- Mean Time to Restore (MTTR) is the system response to an event downtime or defect that requires successive remediation.
- Deployment Frequency is a delivery cadence developed by the agile team for incremental iterative sprints and release cycles.
- Lead Time is a flow metric that represents the estimated time needed to deliver requirement solutions.
- Change Fail Rate is the percentage of releases to the operational environment (escapes) that requires successive remediation DOD (2019a).

Agile performance metrics help managers and decision makers improve the delivery of software to the field. These metrics provide stability and quality throughout the development and deployment delivery process.

E. SUMMARY

While the state of software and applications development in the DOD is challenging, there is optimism that future development will fully leverage agile methodologies and maximize legacy waterfall applications through modernization initiatives. Use of DevSecOps methodology and scrum framework enables faster software development the ensure the United States maintains its military superiority. This chapter concluded with performance metrics to identify best practices of software development and provide managers and product owners tools to manage agile teams and ensure delivery of "done" products.

III. PIEE PROGRAM ANALYSIS

The Procurement Integrated Enterprise Environment (PIEE), previously named Wide Area Workflow (WAWF), is a web-based Government Off The Shelf (GOTS) platform developed by the United States Department of Defense (DOD), providing a one-stop-shop in the Procure-to-Pay (P2P) process. Its focus on developing DOD enterprise capabilities, leveraging emerging technology, ease of access and use, maintaining and validating data, and interfacing with a variety of systems, increases efficiency and effectiveness throughout DOD procurement, payment, and related activities (Jacobs, 2023). The system streamlines and integrates procurement processes across the entire DOD enterprise, including the Army, Navy, Air Force, and other defense agencies (DLA, 2023c).

According to Defense Acquisition University (DAU) (2021), PIEE was developed as part of the Defense Procurement and Acquisition Policy (DPAP), now Defense Pricing and Contracting (DPC), initiative to modernize and simplify the procurement process. The DOD developed it as part of a broader initiative to create a paperless contracting workplace (DLA, n.d.-d). Originally merely an application launched in 1999, WAWF evolved into a portal or application suite in 2014. This enabled access to a wide range of procurement-related activities such as solicitation, award, contract management, and invoicing. Eventually its name changed to PIEE in 2018 when WAWF returned to the suite as an application (DLA, 2023d).

DAU (2021) further explains that PIEE is designed to be user-friendly and intuitive, with a focus on standardizing processes and reducing the time and effort required for procurement activities. The system also provides real-time visibility and tracking of procurement actions, which allows for greater transparency and accountability throughout the procurement process. The current state of the platform, as well as the overall vision for its future, works to attain the main goal of the Office of the Undersecretary of Defense (OUSD) for Acquisition and Sustainment (A&S), which is to "Enable [i]nnovative acquisition approaches that deliver warfighting capability at the speed of relevance" (Jacobs, 2023).

A. OWNERSHIP AND GOVERNANCE

The Defense Logistics Agency (DLA) owns and operates the PIEE suite platform. However, the governance of the PIEE suite, controlling its mission and overall direction, makes it a DOD enterprise platform. The governance structure of PIEE, represented in Figure 10, includes three main components: the PIEE Program Office, the PIEE Governance Board, and the PIEE User Group. The PIEE Program Office is responsible for the overall management and operation of the system. The PIEE Governance Board, the Electronic Business Configuration Control Board (EBCCB), provides oversight and strategic guidance to the Program Office. The PIEE User Group, made up of representatives from the military services (Air Force, Army, and Navy representing both the Navy and Marine Corps), defense agencies, and defense contractors, represents end-users of the system and provides feedback to the Program Office on system functionality and usability (DOD, 2019b).

The EBCCB is responsible for setting the strategic direction for the system, overseeing the development and implementation of system enhancements, and ensuring the system is aligned with the goals and objectives of the DOD. The Board is comprised of senior-level officials from various DOD organizations and agencies, including representatives from Defense Pricing and Contracting (DPC) in the Office of the Assistant Secretary of Defense for Acquisition (OASD (A)), the Office of the Assistant Secretary of Defense for Sustainment (OASD (S)), the Office of the Under Secretary of Defense (Comptroller) Business Integration Office (OUSD (C) BIO), and procurement leadership within the Fourth Estate, Army, Navy, Air Force, Defense Logistics Agency (DLA) (J6), and Defense Contract Management Agency (DCMA) (DOD, 2019b).

The PIEE User Group, on the other hand, is responsible for providing feedback to the PIEE Program Office on the functionality and usability of the system. The group represents end-users from across the DOD and provides valuable insights into system usage and how to improve the system. The User Group also helps to ensure the system meets the needs of its users.

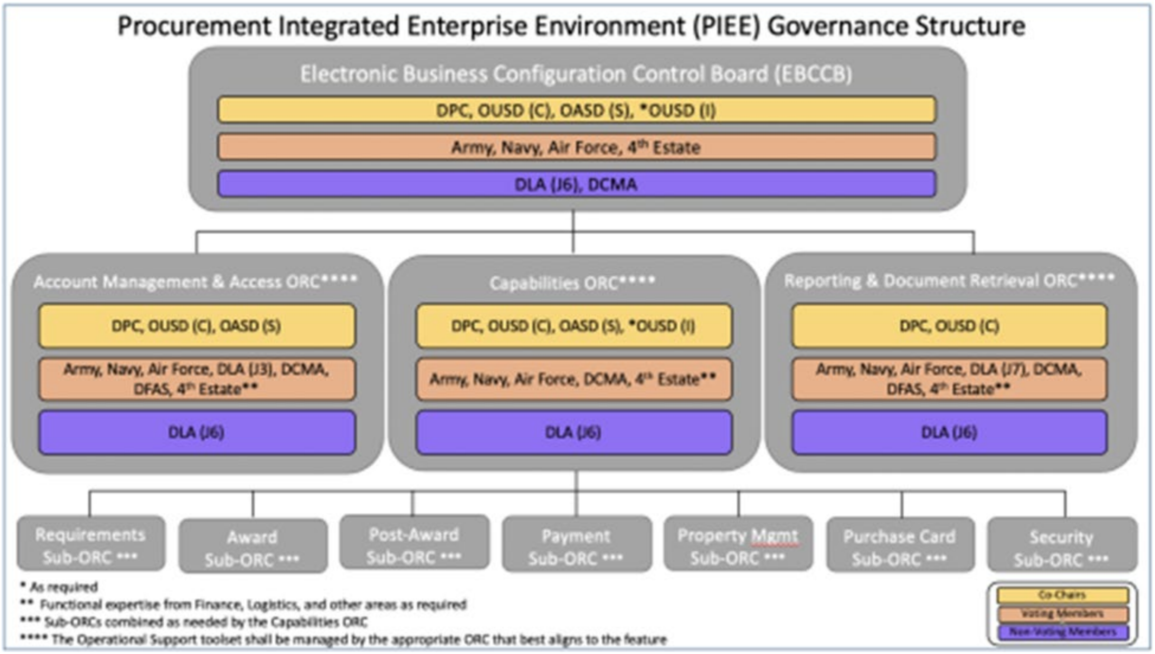

Figure 10. PIEE Governance Structure. Source: DOD (2019b).

The User Group provides its feedback through various Operational Requirements Committees (ORCs) and sub-ORCs, managing different aspects of the PIEE suite, from capabilities to account management and access control, as well as reporting and document retrieval (DOD, 2019b).

Overall, the governance structure of PIEE is designed to ensure that the system is effectively and efficiently managed and that it meets the needs of its users. By providing oversight, strategic guidance, and user feedback, the EBCCB and User Group help to ensure the system continues to be a valuable tool for the DOD and its associated agencies.

B. ARCHITECTURE AND APPLICATIONS

Due to its use of cloud hosting technology, PIEE facilitates expedient development and release of applications and enhancements to those applications (Propert, 2019). PIEE also uses a modular architecture, which allows for the addition and removal of components, as needed, in a plug-and-play manner (Propert, 2019). As shown in Figure 11, there are four layers in PIEE's architecture, including a data layer, the application and capabilities layer, an account management and access layer, and an operations support layer (Jacobs, 2023).

The Operations Support layer provides the program office and leads in the services and defense agencies with oversight and administrative functionality, such as access to the process model library, the ability to govern Department of Defense Activity Address Code (DODAAC) purpose code flags, the capacity to assign administration and payment office cognizance over Commercial and Government Entity (CAGE) codes, and the power to appoint individuals to official roles in the Government Purchase Card (GPC) program.

The Account Management and Access layer is the user's first point of contact with the platform, where user registration, access control, account management, and roles-based access is handled. This is one of the major advantages of the PIEE suite, as the platform provides a consistent, streamlined experience to users attempting to access numerous applications. Other procurement applications hosted outside of the PIEE suite commonly require cumbersome access management processes.

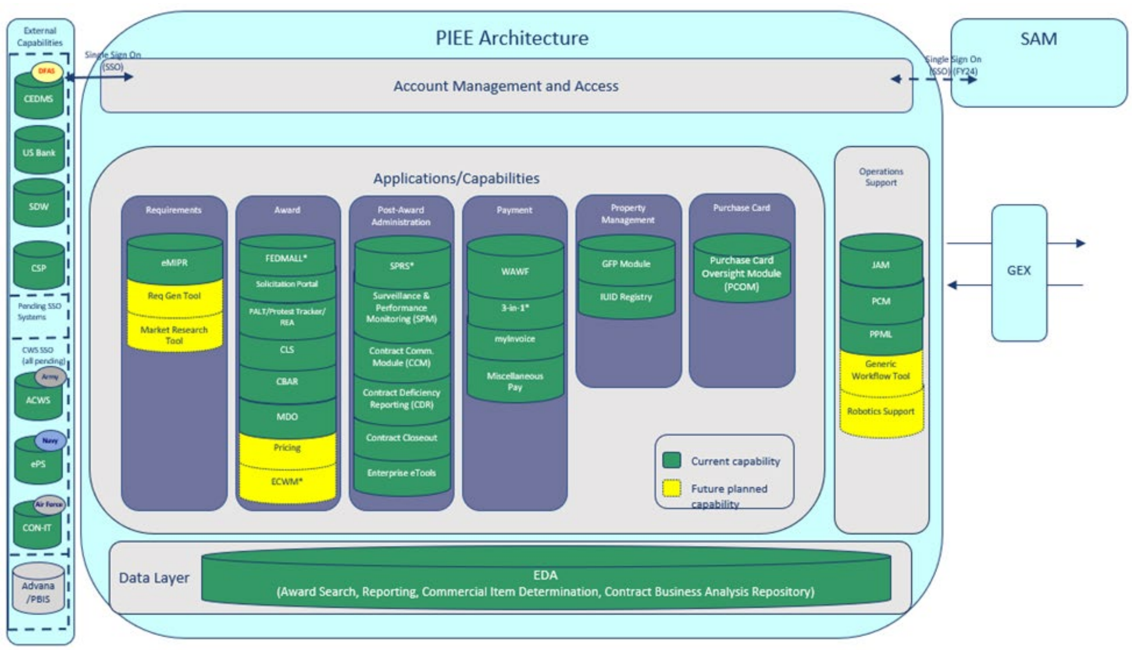

Figure 11. PIEE Suite Architecture. Source: Jacobs (2023).

Many of the applications accessed through PIEE are external to the platform, but leverage PIEE's Single Sign-On (SSO) functionality to grant and control access to these external applications in a manner consistent with every internally hosted application in PIEE, eliminating redundant processes while simplifying those that remain.

The Data layer is self-explanatory, as it provides users access to data, documents, and reporting throughout the PIEE suite. The data comes from the applications themselves, of course, but from other sources as well. For instance, contract writing systems send their data to the PIEE environment via the Global Exchange and this data is used to inform various systems within the PIEE suite, including Wide Area Workflow (WAWF) and Modifications and Delivery Orders, enabling pre-population of contractual data to facilitate payment and modification writing. Additionally, the jointly owned and operated Mechanization of Contract Administration Services (MOCAS) system provides data to the PIEE via and interface with Shared Data Warehouse (SDW) and this data is also used to inform and pre-populate applications, supplementing the data from the contract writing systems. The data from PIEE also transmits to downstream systems, informing them of contract closeouts, modifications and orders initiated in PIEE, and approved shipments and invoices, among other things. More recently, data from some PIEE applications started to be pushed to Advana, the DOD's acquisition data repository for data analytics. As the data becomes better and more readily available, this aspect of the PIEE suite will become the most important, as it is leveraged to inform, pre-populate, and automate processes and decisions.

Finally, the Applications and Capabilities layer, the most visible and familiar layer to most users, contains the 35 modules available for acquisition activities displayed in Figure 12. The applications are organized into eight categories, including Requirements, Award, Post-Award Administration, Payment, Property Management, Operational Support, Purchase Card, and Other. As mentioned previously, some of the listed applications are external to the environment, such as Contracting-Information Technology (CON-IT), Enterprise Contract Writing Module (ECWM), Army Contract Writing Systems (ACWS), Shared Data Warehouse (SDW), Clause Logic Service (CLS), and

Advana, but access to these applications is controlled within the PIEE suite to provide a single point of access, registration, and user roles management.

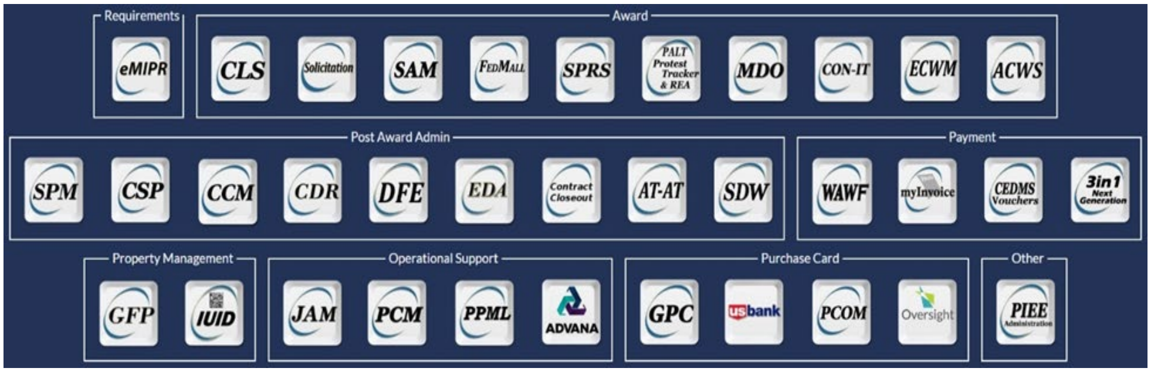

Figure 12. Current and Underdevelopment PIEE Applications. Source: DLA (2023e).

C. PIEE SUSTAINMENT AND DEVELOPMENT FUNDING

1. Sustainment Funding

There are a couple of mechanisms for funding the PIEE suite's sustainment and development costs. Since it is part of everyday business operations and use of some PIEE applications, like WAWF and EDA, are mandatory and used across the DOD every year, PIEE qualifies for funding via the defense working capital fund (DWCF). This also helps to ensure it remains funded across fiscal years, as funds do not expire and simply carry over into the next year. In fiscal year (FY) 2023, the total budget for PIEE covered under DWCF is nearly $37 million, listed under Wide Area Workflow (WAWF) in the DWCF, based on its previous name (DOD, 2022). The military services and three defense agencies, including the Defense Logistics Agency (DLA), Defense Contract Management Agency (DCMA), and Defense Finance and Accounting Service (DFAS), share the cost based on usage determined by the number of users from each group (Defense Working Capital, 2022). This helps to reduce the sustainment burden on any one entity, leverages DOD purchasing power, and provides another perk for organizations considering development within the PIEE suite, as sustainment costs are already factored into their budgets, for the most part.

2. Development/Enhancement Funding

Additionally, those entities using the PIEE suite or planning to use it fund development efforts by transferring funds to the DLA PIEE PMO in accordance with the Economy Act of 1932 via direct acquisition using Military Interdepartmental Purchase Requests (MIPRs). DLA's contracting office then puts these funds on contract for development within PIEE, following scheduling based on prioritization and ORC voting, as mentioned above.

D. SOLICITATION/CONTRACT REVIEW

The DLA contracting services office (DODAAC: SP4701) issued the PIEE Common Operation Environment (COE) solicitation (SP470120R0013) in May of 2020. This solicitation was for support and development of the PIEE suite. According to the solicitation, over half a million government and contractor personnel use the PIEE suite. The solicitation and eventual Indefinite Delivery Indefinite Quantity (IDIQ) contract vehicle consist of a single base year, plus two option years, enabling the PIEE program office to exercise two option years to extend the contract based on need and awardee performance. The solicitation promotes small business use through a partial small business set-aside, requiring a small business plan for teaming between a large and small business. The large business handles development efforts and supports the small business in sustainment and other efforts. The solicitation is for IT and Telecom services (Product Service Code: DA01) and resulted in an anticipated maximum award value of less than $29 million, assuming a three-year period of performance.

1. Task Areas

There are four task areas included in the solicitation and eventual Indefinite Delivery Indefinite Quantity (IDIQ) contract vehicle (SP470121D8002), including the optional Transition-Out period/plan, which is far less substantial than the other areas and covers developing and implementing a transition plan between the awardee and a separate awardee in the follow-on contract, if necessary. The first task area consists of providing program management support to the PIEE Program Management Office (PMO) and is covered under a firm fixed price structure. Similarly, the second task area is a firm fixed price pricing arrangement; this task area covers sustainment services related to the PIEE suite and its applications, including help desk support, testing and evaluation, fixing errors, addressing trouble tickets, maintaining systems/applications, and the like.

Finally, the third task area, software application development, is the most complex of the group and involves a cost-plus incentive fee (CPIF) pricing arrangement. System enhancements and development entails coding and software development; developing interfaces between applications within the PIEE suite and DOD partner business systems, including contract writing systems, administration systems, payment systems, contractor business systems, and other authoritative and/or enterprise-level systems (e.g., System for Award Management (SAM), DAASINQ/DODAAD, DLA CAGE, Clause Logic Service (CLS)); testing and testing support activities; and improving current PIEE registration/single sign-on (SSO), reporting, and applications.

2. Best Value/Tradeoff Approach

The issuing office used a competitive best value/tradeoff approach between non-price and price factors in this solicitation, where non-price factors pooled together carry far greater weight than price factors when evaluating proposals. The non-price factors include Past Performance, Management Approach, Technical Approach, Small Business Participation, and Key Personnel. The most important of these five factors are the technical approach, key personnel, and management approach, weighted equally, followed by small business participation and, lastly and of least significance, past performance. The price factor comes into play when proposals are closely related, as the government is looking for the best value for its money.

3. Observations and Recommendations

This solicitation is complex, especially given the typical solicitation and award issued by this procurement office. The award is a mixed-type IDIQ with one of the more complex pricing arrangements, using CPIF line items. This increases the upfront and post-award administrative burden on the contracting office. This contracting office does not issue many of this type of contract either, amounting to 0.16 percent of its contracting actions over the last ten years (Schmidt, 2023a). Given the infrequency of this pricing arrangement, the contracting office lacks experience in developing and administering contracts of this type. Even including cost-plus fixed fee (CPFF) pricing arrangements in the mix, less than one percent of contract actions from this office include either CPIF or CPFF (Schmidt, 2023a).

Additionally, the solicitation was approximately 240 pages in length, including eight attachments and other information beyond the attachments, as well as requirements for the awardee to develop its own performance work statements for the issued orders, rough order of magnitude estimates for engineering change proposals submitted for proposed development/enhancements, and teaming with a small business falling under at least one other category (e.g., Woman-Owned Small Business (WOSB), Veteran-Owned Small Business (VOSB), Service-Disabled Veteran-Owned Small Business (SDVOSB)). Six amendments to the solicitation followed, including extending the submission deadline and responding to over 160 questions about the lengthy solicitation, resulting in a 33-page amendment to answer all of the questions. The issuing office planned to award the contract early in fiscal year 2021, but failed to award until five months into the fiscal year.

Due to the complexity of the contract, requiring a lot from potential offerors, this may result in reducing the amount of competition. The number and type of questions supports this conclusion, as potential offerors found it difficult to meet all the requirements. Furthermore, there is potential for splitting the three main task areas into separate contracts, reserving 1–2 of them for small businesses and allowing large businesses to compete for the complex software development portion of the contract on a separate award. Alternatively, the onus of meeting small business concerns could fall on the prime contractor, requiring large business offerors to develop small business plans. The large business awardee could then manage the small business rather than teaming with a small business as part of a separate award process.

Moreover, although DLA awarded the IDIQ late and the contract is relatively complex, it is only issued for one year with two one-year option periods. This results in a maximum of three years of performance, requiring another procurement effort at that point, assuming there are no performance issues before then. Since sustainment of the PIEE suite and development of enhancements and additional applications is an ongoing requirement long into the future, it seems wise for the award to contain four one-year options to avoid having to complete the acquisition processes again in three years, assuming cost and performance do not create issues beforehand. With required tools like WAWF and EDA, the PIEE suite must be maintained long term and a lengthier contract provides stability for DLA and the contractor receiving the award.

a. DCMA Support

To offset the complexity of the overall requirement, DLA could enlist the Defense Contract Management Agency (DCMA) to administer the contract and its orders to ensure the administration capacity exists to handle all of the requirements surrounding a multi-year, mixed-type IDIQ with cost reimbursement line items, varying periods of performance, requirements for traceability, cost/pricing data, etc. For instance, FAR 42.302(a)(31) enables the program office to obtain assistance from DCMA to "Perform production support, surveillance, and status reporting, including timely reporting of potential and actual slippages in contract delivery schedules." This is currently relevant in the program, as two major applications, Modifications and Delivery Orders (MDO) and Audit Tracker and Action Tool (AT-AT), ended up behind schedule and the program office lacked awareness until late in development. Additionally, a handful of other development efforts have underrun, including Delivery Schedule Manager (DSM) and enhancements to Contract Closeout (CCO), and the program office was unaware until the end of the efforts due to limited forecasting. Similarly, FAR 42.302(a)(67) is another pertinent service DCMA provides by "Support [ing] the program, product, and project offices regarding program reviews, program status, program performance and actual or anticipated program problems." This support could alleviate some of the other problems the program faced, including timely review of submitted invoices and vouchers, comparisons between cost, schedule, and performance burndowns, risk analyses and mitigation strategies, and improved project estimates.

b. Contract Type Selection

Perhaps it is prudent to consider another contract type altogether, since the contracting office lacks experience issuing and administering CPIF pricing arrangements.

Although the contracting office also lacks experience with CPFF arrangements, they are simpler that CPIF. Plus, with the help of DCMA and DCAA, this could offset the complexity that remains. This also avoids issues with the schedule structure, since the current schedule design does not allow for overruns, meaning the back half or overrun portion of the CPIF's range of incentive effectiveness (RIE) is meaningless, since the schedule never allows the contractor to overrun and, thus, the contractor is never at risk of accepting a fee (profit) less than the target fee. This entails that the contractor is not incentivized to control costs due to a risk of reduced fee (profit). Instead, if the contractor fails to deliver within the allotted schedule, but incurs costs up to the target cost, the contractor receives the target fee (profit) without delivering. Additionally, if the office administering the contract does not perform its duties adequately, as has been the case in some instances as demonstrated in our analysis, the contractor could fail to incur costs up to the target cost, fail to deliver, and then be rewarded with an incentive fee above the target fee, thereby incentivizing the contractor to provide less effort toward delivering a final product once it is clear delivering on time is impossible. Essentially, given the design of the schedule, the overall complexity of the pricing arrangement and contract, and the lack of adequate resources to administer the contract, this arrangement incentivizes exactly the opposite of what it intends.

In contrast, a CPFF pricing arrangement is a higher risk pricing arrangement to the government and lower risk to the contractor than CPIF, as CPIF incentivizes the contractor to control costs to maximize the incentive fee available (Cuskey, 2016). However, with this program, CPIF becomes riskier to the government and less risky to the contractor, as the incentive fee is at least as high as the fixed fee with no risk of being smaller, despite failing to deliver. The contractor can obtain a higher fee with CPIF than CPFF under the exact same conditions as well. To reduce the complexity of the pricing arrangement, burden on the issuing and administration offices, and risk to the government, a CPFF pricing arrangement makes sense.

Along the same lines, cost reimbursement contracts, in general, increase the risk to the government and reduce it for the contractor, because the contractor must only promise to provide its "best efforts" and the government fully reimburses the contractor's incurred costs, plus typically provides some sort of fee (profit). The government frequently uses the pricing arrangements for software development efforts, because the contractor and government are unsure whether the contractor can deliver advanced software and/or at what cost, so the contractor is likely to avoid seeking firm-fixed price software development contracts or is likely to inflate its cost estimates to offset the risk involved. This adds risk to the government, as the field of offerors may shrink, those willing to provide proposals may be less experienced due to the risk involved, the contractor may fail to deliver, and/or this may substantially increase the offerors' estimates.

Due to the added risk to the government, administering such contracts requires substantial administrative effort, as well as additional effort from the contractors, as the government requires incurred costs submissions, must review costs, and the two parties negotiate rate agreements, which take time. The latter frequently results in the loss of development funds, as settling rate agreements takes so long the funds are no longer available for use on other projects. However, given the type of software development included in the PIEE suite, the length of time contractors have successfully developed this sort of software, and that much of the development is merely replacing and enhancing existing software, a firm-fixed price (FFP) contract type may be reasonable.

The government has substantial cost history to rely upon to predict development costs, including over 20 years developing and enhancing PIEE suite applications. Throughout this period, performance problems have not materialized and the same developer continues to perform the same sorts of development and sustainment services, with few significant changes beyond improved technology and a shift from waterfall to agile development processes. The development environment remains stable as well. Unlike the constantly shifting eBusiness landscapes within individual uniformed services and defense agencies, focused on maintaining and enhancing legacy systems and/or attempting to dramatically transform acquisition systems, the PIEE platform and tools, being enterprise in nature, remain supported and gradually developed, enhanced, and refreshed, with the various players' interests in mind.

Furthermore, although one major developer continues to provide most of the development and enhancement services within the PIEE suite, development on the platform remains competitive, as the low-code techniques on a GOTS platform enable numerous contractors to compete for PIEE development contracts. Finally, most of the development efforts of late begin with well-developed requirements packages based on replacing current functionality with renewed technology on an enterprise platform based on existing technology within the PIEE suite. Thus, the effort is predictable upfront.

Overall, this is an optimal situation for an FFP contract type, as it is predictable, stable, relatively simple, and competitive. Plus, per FAR 16.103(c), "contracting officers should avoid protracted use of a cost-reimbursement or time-and-materials contract after experience provides a basis for firmer pricing." This point was recently reiterated by Tenaglia (2023) in an Office of the Under Secretary of Defense for Acquisition and Sustainment memorandum providing guidance to implement DOD Inspector General recommendations based on DODIG-2022-137, *Audit of the Military Services' Award of Cost-Reimbursement Contracts*.

Of course, this is easier said than done. Traditional thinking surrounding FFP contracts is that they lack flexibility and the government is either stuck with exactly what it contracted for or must modify the contract repeatedly to incorporate desired changes, frequently at great cost to the government. Neither option is desirable, which is why more flexible cost-reimbursement contracts find favor. Additionally, developing software, and even more so when leveraging agile software development techniques, requires regular changes to requirements as new information presents itself, new technologies become available, and/or when developed functionality does not quite function as anticipated.

However, Freihofer, Dotson, and Maus (2021) explain how software development programs can leverage indefinite delivery contracts and FFP orders to incentivize contractor performance while remaining agile in development. Instead of incentivizing cost controls via cost-reimbursement contracts, which can negatively impact software development efforts where requirements change, delivered functionality is important, and stable and experienced teams are crucial, they argue for FFP orders designed around incentivizing development team stability (Freihofer et al., 2021). This is done by contracting for a specific development team at a specific price for a specific period on one task order and then increasing or decreasing the price of that same team on the next task order relative to whether the team remained stable throughout the development period or not, thereby incentivizing stasis over cost (Freihofer et al., 2021). Overall, Freihofer et al. (2021) assert this relies on trust between the government and contractor, of course. This should not be an issue with the current developer and program office, assuming the relationship remains into the future.

The solicitation itself and the structure of the IDIQ are good. The tradeoff approach and weighting of individual factors is sensible for this sort of complex and important effort, since the contracting office needs to ensure DOD enterprise acquisition systems remain reliable. This tradeoff approach using competitive processes also seems to have avoided exorbitant costs, as software development costs to the federal government have been increasing steadily, but PIEE suite development remains relatively inexpensive compared to alternatives covered later in this paper. However, more thought into the overall complexity of the requirement is necessary to ensure competition, a successful program, and avoidance of unnecessary and redundant work going forward, which is why delegating administration responsibilities to DCMA for this effort and separating the contract into its severable parts rather than grouping them together may be wise to reduce the workload on the program and contracting offices. Finally, as thoroughly explained above, the program office has opportunities to leverage other contract types, including CPFF, but especially FFP, to better achieve its development goals, reduce pre-award and post-award contracting burdens, and incentivize the correct development factors.

E. PROGRAM REVIEW

1. Estimating

Regardless of contract type, estimating plays an important role in every acquisition. For PIEE development efforts, entities planning to develop and enhance applications frequently create cost estimates for these efforts to support their budget requests years in advance as part of their Future Years Defense Program (FYDP) submitted annually along with the rest of the DOD submission as part of the President's budget. This serves as a baseline for programs, so the accuracy of such cost estimates is important. Every dollar matters given today's budget constraints, so programs are held accountable for their budget requests, whether too large or small.

However, this paper focuses more on the more detailed Independent Government Cost Estimates (IGCEs) related to PIEE development, as they are more readily available, detailed, and comparable to contractor Rough Order of Magnitudes (ROMs) and reported incurred costs. Although this paper reviews and compares estimates with one another and incurred costs, it avoids going into too much detail. The quantity of data available is substantial, given that many Engineering Change Proposals (ECPs) or software development projects are underway at once and completed in as little as every two weeks, a single Sprint or Iteration in the agile development schedule, though the average is closer to three months. The format of the data poses its own challenges, as the developer submits its ROMs in Microsoft Word and its incurred cost submissions in Portable Document Format (PDF). Given the quantity of projects and the two data formats, the work involved in compiling the data simply to begin analysis is prohibitive for this research effort. Finally, ROMs, IGCEs, and incurred cost submissions all contain Controlled Unclassified Information (CUI) and the authors desire to keep this paper available to a broad audience.

Requirements owners submit their ECPs to the PIEE PMO containing high-level features desired for development and some submit well-developed requirements packages, including DCMA and the Office of Naval Research (ONR), since the latter two are in the middle of developing replacement applications for their legacy systems and have experience with the processes and desired functionality. The PMO reviews the submissions, begins discussions with the requirements owner regarding scheduling, funding availability, product owners, and seeks clarifications related to anything unclear in the submission. The PMO also drafts an initial IGCE and provides the ECP submission to the contracted developer to provide a ROM. Once the developer completes the ROM, it is reviewed by the PMO, then provided to the requirements owner. At that point, the requirements owner reviews the ROM and begins negotiating with the developer if the estimate is higher than anticipated. After the requirements owner and developer agree on an estimate, the PMO provides the requirements owner with an IGCE and request for funding via MIPR by a specific date to allow the contracting office time to issue an order.

Although the PMO develops a draft IGCE prior to negotiations, it is not used during negotiations and it is unclear how much it is adjusted following negotiations, as the IGCE is not provided to the requirements owner until after negotiations end. Review of eight IGCEs DCMA had on hand as the requirements owner, all list estimates for the same 15 labor categories and the portions of the total estimate attributed to each labor category remain relatively consistent. Similarly, the four ROMs listing labor categories as part of the estimates DCMA possessed for comparison with the IGCEs consistently listed the same 17–18 labor categories. However, the IGCEs did not assign any hours or costs to nine of the labor categories included in the ROMs. Surprisingly, the IGCEs left off program manager, scrum master, web designers, web developers, and test engineer, despite PIEE software development efforts using all of these as part of agile development and testing of web applications.

Similarly, the IGCEs assigned costs to 7–8 labor categories the ROMs assigned no costs to. Review of the incurred cost submissions produces similar discrepancies where the developer included incurred costs for the labor categories listed in its ROMs. Furthermore, the incurred cost submissions included some labor categories included in the IGCEs, but not the ROMs, as well as some labor categories excluded from both the ROMs and IGCEs. Finally, despite these discrepancies, the estimated costs in both the IGCEs and ROMs remain remarkably similar. In fact, of six IGCEs and ROMs analyzed together, the IGCEs estimated total costs on average 0.18 percent higher than the ROMs, with the greatest difference being 0.41 percent higher than the ROM.

DCMA negotiated lower estimates for three of these six efforts. For one of these efforts, ECP 1350, DCMA negotiated a final cost estimate nearly 90 percent below the original ROM, yet the IGCE ended up just 0.6 percent higher than the agreed upon estimate. Similarly, DCMA negotiated ECP 1339 down by nearly 70 percent; the IGCE ended up 0.14 percent higher than that negotiated cost. Finally, repeated negotiations for a third effort, ECP 1338, following clarifications, multiple demonstrations, and discarding of unclear and complex requirements, resulted in a more than 55 percent reduction than the original cost estimate. Again, the IGCE ended up 0.05 percent higher than the negotiated cost.

The PIEE PMO scheduled ECP 1350 for nine months of development, but DCMA raised concerns over the pace of development, velocity, early on and continuously. Although the product released functionality into the production environment, DCMA had to obtain additional funds to proceed with development currently underway. This resulted in a nearly 47 percent increase from the original effort, but remains less than 17 percent of the original ROM estimate provided by the developer. Another software development effort, ECP 1292, was not negotiated down significantly, yet ended up overrunning, repeatedly, and remains under development. Although still under development, it has already cost an additional 67 percent, is expected to cost double the ROM estimate, and release into production was scheduled for May 2022, but is delayed until November 2023.

This is not a consistent trend, however; most efforts finish on time and on or below budget. For example, ECP 1339 mentioned above as being negotiated nearly 70 percent lower than the original ROM estimate, ended up underrunning by a few percent. Similarly, ECP 1338, negotiated over 55 percent lower than the initial ROM estimate, resulted in an underrun of 37 percent; the final cost before calculating the incentive fee amount is less than 30 percent of the original ROM estimate. Similarly, ECP 1304, completed in early March 2023, resulted in an underrun of nearly seven percent, despite adding requirements during development to handle items not previously identified as necessary.

2. Observations/Recommendations

Our research identified several issues with government and contractor estimating. First, DLA did not provide the IGCEs to requirements owners prior to receiving ROM estimates from the developer. Second, because of this, requirements owners did not use these government estimates to negotiate with the developer. Third, this resulted in dramatic shifts away from the developer instances, in some cases, without apparent support for such swings. However, these dramatic shifts do not appear to have resulted in a predictable result, as the largest negotiated shift did not result in the largest overrun nor did the overrun come close to the original estimate provided by the developer. Most development efforts, whether negotiated dramatically lower than initial estimates or not, resulted in underruns and some with the largest negotiated shifts still underran substantially. Finally, the labor categories identified as relevant in the IGCE failed to include many labor categories corresponding to developer ROMs and actual incurred costs and included many labor categories that did not correspond with the ROMs and incurred costs of past development efforts.

Per *DOD Instruction 5000.73: Cost Analysis Guidance and Procedures*, "Independent and sound cost estimates are vital for effective acquisition decision making and oversight. Cost estimates also support efficient and effective resource allocation decisions throughout the planning, programming, budgeting, and execution process" (Department of Defense Instruction, 2020a, p. 4). There is a plethora of historical PIEE development cost data, including invoices submitted monthly for years breaking costs out by ECP, labor categories, hours, and more. At the very least, this data should be leveraged to perform an actual costs estimate. According to Defense Acquisition University's (n.d.-a) Acquipedia page, this technique "is the most supportable" estimate type, especially when programs are mature, which includes PIEE software development, with a 20-year history of development.

The PIEE PMO can also use the analogy and expert opinion techniques to supplement the actual costs method by pointing to cost drivers, drawing analogous comparisons to previous efforts, and identifying potential risks (DAU, n.d.-a). Although the expert opinion technique ought to be used as a last resort on its own, as the DOD Independent Government Cost Estimate (IGCE): Handbook for Services Acquisition (2018) points out, using it to supplement other estimating techniques can refine and enhance those estimates. Similarly, analogy estimates can refine actual costs estimates by pointing out similarities and differences between past software development efforts and proposed efforts, determining whether estimates should be increased or decreased accordingly (DOD IGCE, 2018). Optimally, once more data is collected and combined, the PMO would shift to more accurate estimating techniques, such as parametric and engineering (DOD IGCE, 2018).

When reviewing the developer's incurred cost submissions, the authors of this paper noted something peculiar. The developer consistently submitted 2-in-1 invoices, which are invoices and receiving reports for service contracts, but reserved for fixed-price line item types and exclude cost, labor hour, and time-and-materials line item types. Per the contract, the developer is required to submit cost vouchers, which is consistent with the cost-type line item pricing arrangements associated with the developer's work. Submitting the wrong document may seem trivial, but in this case it adds significant risk due to the nuances and workflows associated with the two documents.

Since 2-in-1 invoices are invoices and receiving reports, when the Contracting Officer's Representative (COR) accepts these documents in WAWF, the COR accepts the work as complete and final and triggers payment of the invoices (DLA, n.d.-e). Cost-type line items are a form of financing or lending to the developer, however, and the effort and payments are not final normally until rate agreements are settled and in place. Additionally, since the contract lacks the FAR Clause 52.232-25, Alternate I, late payments on cost reimbursement line items do not rate interest payments, but invoicing via the 2-in-1 could trigger such automatic interest payments if late payments occur. Finally, when submitting cost vouchers per the contract, the developer is required to input the Department of Defense Activity Address Code (DODAAC) associated with the Defense Contract Auditing Agency (DCAA) responsible for auditing interim cost voucher submissions (DLA, n.d.-c). Submitting the 2-in1 invoice does not require input of the DCAA DODAAC, so DCAA is not notified of these submissions and has no way of identifying them to sample, review, and perform cost analysis. As mentioned previously, this increases the risk to the government, as these incurred cost submissions lack necessary oversight. The COR and developer's billing office require additional training to ensure the proper document is submitted to WAWF.

3. Schedule

The PIEE PMO's development schedule occurs over three-month periods consisting of a kickoff/planning period, five two-week development sprints, and a three-week testing period prior to code lock and release of new development into a production environment (Defense Logistics Agency, 2023a). Additionally, there are five scrum teams working simultaneously on different efforts throughout each sprint. Some projects occupy an entire scrum team for all five sprints of a development period and may continue into follow-on periods of development (DLA, 2023a). Other projects are smaller and may only require one sprint of effort for a scrum team or less than an entire scrum team's effort, depending on the sort of work involved (DLA, 2023a). The PIEE PMO must coordinate months in advance with the various military services and defense agencies interested in development to get their proposed ECPs, return ROMs and IGCEs, schedule ECPs for development, obtain required funding, and get the efforts on contract.

Throughout the development period, at the end of each sprint, the developer reports on progress made during burndown meetings (Defense Logistics Agency, 2023b). Each burndown meeting covers all of the projects underway and provides a burndown chart for each project displaying the previous two weeks of time (DLA, 2023b). The chart compares a guideline or glidepath/glideslope showing the average rate of expected burndown of remaining effort with the actual burndown of effort quantified as story points (DLA, 2023b). It resembles a burndown chart an individual could develop while charting a timecard over two weeks. The individual may expect to work 80 hours over two weeks, including eight hours per day, Monday through Friday, but the actual hours worked may appear different, with more and less than eight hours worked some days and even some work occurring over the weekend. Plus, the individual could work beyond the anticipated 80 hours. These burndown charts provide a snapshot and do not provide where the projects have been, nor do they project where the projects are likely to be in the future. Additionally, these burndown meetings provide a count of remaining user stories based on those currently developed, but excluding those not yet planned. Finally, the meetings explain challenges faced, potential opportunities, and expectations for the near future related to release schedules and testing needs (DLA, 2023b).

4. Observations/Recommendations

The pace of development is rapid and developers complete projects and release functional applications to production environments for users every three to nine months. This pace is consistent with DOD requirements, but industry best practices using agile methods presses for delivery in as little as every two weeks (GAO, 2021b). At the same time, most PIEE development work requires far more than two weeks to plan, develop, test, and release into a production environment simply due to the size and complexity of the applications. Most PIEE development efforts are complete within 3–6 months and almost all release at least some functionality within nine months (PIEE Enhancement by Release, 2023). One application, AT-AT, is the exception, as it has been in development for 15 of the last 20 months and is not expected to deploy usable functionality for another six months. However, this application is unique in a variety of ways, not the least of which is that it combines five legacy applications into one, requiring numerous functional specialists with varying degrees of expertise to serve as product owners and work to design and explain the desired functionality and workflows.

Opportunities exist for adjusting the schedule, especially as it relates to testing. Traditionally, the ten-week development period is followed by a three-week User Acceptance Testing (UAT) period. However, with a few larger ECPs, the developer piloted earlier testing by releasing newly developed functionality into a test environment after each two-week sprint, enabling product owners and JITC testers to test the functionality, identify issues, and resolve them much earlier in the process. This prevents substantial amounts of rework late in development, which can often put pressure on the short turnaround time between UAT, code lock, and release into production environments. Expanding this process to all ECPs and making it a permanent part of the schedule could lead to more rapid development turnaround times.

The most significant issue with the schedule relates to the burndown charts. As mentioned previously, the burndown covers only the last two weeks of development, nothing before and nothing afterward. Thus, the PMO does not get a complete picture of development. If the project is ahead or behind schedules leading into the burndown meeting, the chart does not reflect this reality. Being on the guideline on a given burndown chart does not mean the project is on the guideline, even though this is the message conveyed by the chart. Additionally, there is no forecasting taking place, since the burndown chart lacks estimates of future development rates and remaining work.

The forecasting issue relates to the issues with estimating covered earlier in this paper as well. Since user stories are not developed from the entire list of requirements ahead of time and, hence, cannot be estimated for complexity and necessary development time, adequate estimates of cost and schedule are incomplete and inaccurate. Furthermore, forecasting becomes difficult, as the remaining user stories and story points necessary to complete them are both unknown until shortly before either the project is complete or the schedule runs out. This means the PMO is unaware of potential overruns until late in the development period and cannot act to mitigate the risks earlier in development. The DOD desires software development to occur at the "speed of relevance," but this also means program offices require timely information to assist in rapid decision making (DOD Restructures Acquisition, 2017). A few ECPs suffered from this lack of planning and forecasting. For two of them, ECPs 1291 and 1292, overruns were identified late. Due to the lengthy budget cycle, development of these two ECPs were delayed further when issues could have been identified and resolved sooner or at least additional funds could have been requested timely. Other instances revolve around underruns where development efforts would finish early and funds could be shifted earlier to other projects. In the future, burndown charts and meetings ought to focus on the full development schedule of particular projects, enabling better program management decision making.

Finally, the PMO's burndown analysis lacks any comparison with the costs incurred through development. Tracking and comparing work completed, work remaining, funds expended, and funds remaining provides a fuller picture of the health of a project. For example, when development is burning through funds as expected or faster than estimated while the decline in work remaining is slower than projected, there is a good change funds run out before the effort is complete, unless there is a course shift. Similarly, if the schedule is running out and the work will not be completed on time, but funding is not being exhausted, the program office could push for more development during the time remaining to come closer to completion of the project before time runs out. Without including these various data elements in the burndown analysis, it is impossible to get ahead of issues like these.

This is what happened with ECP 1350 development, as the work was incomplete at the end of the period of performance, but funds remained for development. Worse yet, as noted above, since funds remained and adequate oversight was lacking, the developer was awarded with additional incentive fee for cost savings despite failing to complete the project. The developer is then incentivized to avoid expending all of the funds even if it means failing to complete projects. ECP 1338 experienced a similar issue, but was substantially underrunning. Fortunately, an experienced program manager identified the underrun in time to reassign the funds and effort to another project despite lacking the resources to identify the underrun via adequate burndown analyses of the effort and funds.

F. SIGNIFICANT SUCCESSES

Up to this point, the analysis of the PIEE program focused primarily on shortcomings, mentioning only a few highlights. However, the original impetus for this research project revolves around the apparent successes. Given the concerns identified throughout this research, most would assume the program is plagued with overruns, costly development, poor-performing applications, dissatisfied users, and numerous workarounds. However, none of this appears to be the case. Instead, the PIEE program manages to navigate waters where others struggle.

1. Sustainment Cost Comparison

Per the DOD (2021b), PIEE sustainment costs amounted to $34.5 million in FY 2021. A similar application suite sustained by DCMA, including numerous post-award applications and the Integrated Workload Management System (IWMS), required $34 million in sustainment funds as well (DCMA, 2021). DCMA is currently working to replace this legacy tool suite with modernized applications, including developing many within the PIEE suite. Despite being similar suites containing acquisition applications and requiring similar funding amounts, DCMA funds the entire amount of its suite's sustainment costs, while the PIEE suite sustainment costs are shared across the DOD. In fact, DCMA's portion of PIEE sustainment funding was $4.343 million in FY 2021, $3.448 million in FY 2022, and $2.085 million in FY 2023 (DOD, 2021b; DOD, 2022). Thus, DCMA funds sustainment of both application suites, but saves a significant amount of funds by sharing the sustainment costs of the PIEE suite with other DOD entities. Excluding initial development costs, replacing many of DCMA's acquisition applications with newly developed PIEE suite applications is expected to save DCMA much of the $34 million in annual sustainment expenses in the future without dramatically increasing its portion of PIEE sustainment funding.

2. Development Cost Comparison

Similarly, the cost of development in the PIEE suite compares well with similar development efforts. For instance, DCMA is finishing development in PIEE of a replacement for its contract writing system, Modifications and Delivery Orders (MDO). Overall, this development effort is estimated at approximately $1.7 million (SP4701-21-D-8002 SP4701-21-F-0401 & SP4701-23-F-0027). The application has already released over 25,000 modifications in the last 11 months (Schmidt, 2023b). In contrast, DCMA spent nearly $45 million on IWMS through 2017 (Schooten, 2018) and now plans to replace it with a newly developed application in the PIEE suite (Mims and Schmidt, 2023). Unfortunately, the original development effort resulted in an Anti-Deficiency Act (ADA) violation, compounding DCMA's development efforts (GAO, 2021a).

Alternatively, the United States (U.S.) Navy awarded a contract for $222.9 million to develop an electronic Procurement System (ePS) (CGI, Inc., 2019). Although significantly larger than DCMA's MDO development effort, both systems are contract writing systems, they must interface with a similar variety of other acquisition systems, and the requirements packages resemble one another, though the Navy's requirements list was three times larger than DCMA's list (Schmidt, 2021). This effort, regrettably, ended in contract termination after spending $25 million in development costs (Miller, 2021). Development of the application was not proceeding according to plan and the Navy claims "100% of [limited deployment] requirements NOT met–295 open defects (user acceptance testing (UAT) and IT) as of 6/3 (over 100 critical/high)" (Miller, 2021). Users also expressed dismay as the system performed worse than the one it was designed to replace, even after substantial training and support from the developer (Miller, 2021). Finally, estimated development costs skyrocketed to 350 percent of the original high estimate (Miller, 2021).

The Army awarded a $133.9 million development contract to replace its legacy contract writing systems with the Army Contract Writing System (ACWS) (CGI, Inc., 2017). The ACWS requirements package, though developing a system similar to the Navy's ePS, is approximately half the size of the Navy's package and twice that of DCMA (Schmidt, 2021). Again, this system is larger and more complex than DCMA's MDO, but it suffered a similar fate to that of Navy ePS (Miller, 2021). After spending tens of millions of dollars and experiencing similar issues as the Navy (Miller, 2021), the Army does not appear to be continuing in its development of ACWS with the same developer (W52P1J-17-D-0031 W51P1J-18-F-0271).

3. Capability Comparison

Although merely comparing requirement count between efforts is a poor evaluation on its own, it remains an initial indicator to start from. As mentioned already, DCMA's MDO, the Navy's ePS, and the Army's ACWS are all fundamentally contract writing systems. It is true that MDO's requirements package is much smaller and the system less complex overall. However, all contract writing systems must fulfill many of the same requirements regardless of size. They all must interface with the Federal Procurement Data System-Next Generation (FPDS-NG) to generate contract action reports (CARs) through its user interface, providing procurement data to the public. Additionally, writing systems need to generate Procurement Data Standard (PDS) XML data, transmit it through proper channels to the Global Exchange (GEX), and pass validation checks to get it distributed to downstream business systems for ingestion.

In addition to transmitting data, writing systems populate system inputs onto Portable Document Format (PDF) forms and continuation sheets for human-readable copies of contractual documents, Comma-Separated Values (CSV) index files for proper indexing within EDA, attachments, and attachment CSV files (Defense Federal Acquisition Regulations Supplement (DFARS) Procedures, Guidance, and Information (PGI) 204.1). Writing systems also interface with the Clause Logic Service (CLS) to identify necessary contract clauses and to import related XML associated with selected clauses. Of course, contract writing systems all enable users to create line items, schedules, and accounting information. These systems frequently interface with a variety of other applications to obtain official information, including SAM, DODAAD, PCM, and EDA.

DCMA’s MDO does all of these things, of course, but also enables users to create ARZ (mass) modifications to contracts; this functionality is a unique feature of MDO where it can generate the same modification to up to 20,000 contracts and orders via a single contract action, saving substantial DOD resources by avoiding having to draft them individually. No other writing system performs this function and none of them contained such requirements.

Additionally, MDO can perform P2P Handshake 2, validating funds with other compliant systems. It can also perform pre-validation PDS checks, determining whether its PDS XML will pass validation prior to releasing contractual documents. Following the pre-validation checks, MDO completes pre-conformance checks within EDA to determine whether the contractual documents will conform to the rest of the contract before releasing it. MDO is Standard Line of Accounting (SLOA) compliant, meaning it generates SLOA rather than the numerous legacy lines of accounting (LOAs); other contract writing systems generate legacy LOAs that must be translated into SLOA at the GEX for downstream system ingestion. Despite being smaller and less complex than the contract writing systems the Navy and Army plan to develop, MDO includes most of the complex aspects of these contract writing systems, contains additional functionality, has generated over 25,000 contract modifications to date, and costs less than one percent of the Navy’s original estimate for its system and just over one percent of the Army’s original estimate for the same. Finally, MDO being housed on the PIEE platform enables all of DOD to use it without licensing fees and at no additional cost to non-DCMA users.

4. Successful Collaboration

Perhaps the most remarkable aspect of the PIEE program is the amount of collaboration taking place. Although DLA’s PIEE PMO owns and operates the platform, the PIEE suite is open to all federal government users. Military services and defense and federal agencies can use, develop, and enhance applications on the platform. Some applications, like EDA and WAWF, are mandatory, while others remain optional. Still, over 200,000 government and contractor personnel maintain active roles in PIEE and 35 applications have been developed by various entities, demonstrating its span and appeal

(DLA, 2023c). Furthermore, these entities frequently collaborate and share funds and expertise to ensure the needs of the DOD enterprise and the rest of the federal government are met while saving taxpayer funds.

For example, DCMA and the Office of Naval Research (ONR) have collaborated on a few applications in PIEE, including the Enterprise Award File (EAF), Delivery Schedule Manager (DSM) (to be renamed Enterprise Communication & Deliverable Management (ECDM) following more development), and the Contractor Business Analysis Repository (CBAR). This collaboration led to DCMA aiding ONR in releasing over 2,600 contract modifications using MDO, which enabled ONR to avoid approximately 215 labor hours drafting, signing, and distributing the modifications, saving at least $20,000 (Haley and Schmidt, 2022). Other collaborative efforts exist as well, including the Navy, Army, DPC, and DCMA working together to develop SSO for Product Data Reporting and Evaluation Program (PDREP), streamlining user registration, role requests, and access control to that system. Like the ONR and DCMA collaboration, other defense entities collaborate on a variety of projects within the PIEE suite and have worked together to release ARZ (mass) modifications for various purposes, saving funds across the DOD.

G. SUMMARY

The intent of this analysis is to examine the PIEE program from various angles, identifying successes and opportunities for improvement. Program managers can then use this analysis to improve the PIEE program and similar programs, mimicking successful elements and refining areas incrementally that require additional attention. This analysis provided some background information about the program, what the PIEE platform includes, how it operates, who manages it, and how it is funded. Then the authors considered the solicitation and contract, providing suggestions on how to reduce complexity and alleviate strains created by the complexity of the contract. Next, the program's cost estimating and schedule were evaluated and the authors identified concerns in these areas, as well as means for improving them. Finally, the authors highlighted various successes, including the program's ability to leverage significant collaboration between the services, defense agencies, and contractors to develop functional enterprise software at reasonable cost to the taxpayer. They also emphasized other program accomplishments, including spreading sustainment costs across multiple entities, driving costs down for individual agencies; development cost comparisons with other less successful software development efforts; and capability comparisons with other programs. Based on this analysis, the PIEE program is a success, but there remains room for improvement. Furthermore, this analysis identifies additional topics requiring further investigation to get a better understanding of the program and the personnel involved, which the authors expand upon in the conclusion of this paper.

IV. CONCLUSION

The objective of this CAP is to identify best practices when developing and integrating software within the PIEE suite and how that information can be leveraged in current and future DOD programs. The benefit of this exploration provides software acquisition program managers with key insights of lessons learned with the collaborative efforts used to advance and incorporate the PIEE platform with waterfall legacy software through agile DevSecOps development. This chapter contains the purpose and analysis of the CAP with conclusions and recommendations.

A. RESEARCH QUESTIONS SYNOPSIS

1. What is the past, present, and planned/future state of the PIEE?

PIEE began as a single application, WAWF, an effort launched in 1999 to implement an effort to reduce workplace paperwork by streamlining processes and eliminating redundancy. Slowly at first, this application replaced more documents and processes, interfacing with vendor and government business systems alike. Then it expanded into an application platform, adding EDA to the mix. More recently, DCMA decided to replace its aging legacy administration applications with newly developed and enhanced applications within the PIEE suite. DCMA developed and continues to develop over a dozen applications and plans to develop many more over at least the next five years. ONR recently joined the mix as well, collaborating with DCMA to produce and enhance a few post-award applications. Additionally, the PIEE suite increased its interfaces with a multitude of defense and federal business systems. The enterprise functionality of the suite expanded dramatically. Along with the applications and interfaces, PIEE day-to-day business revolves around the data ingested, produced, validated, and transmitted to other systems and users. Future plans include further expansion of data and application use; pushing data to Advana, the DOD's data repository; and providing a single location for accessing major next generation contract writing systems, including the Air Force's CON-IT, the Fourth Estate's ECWM, the Army's ACWS, and the Navy's ePS. Given the expansion of functionality, quantity of DOD entities and contractors involved, and regulations requiring its use, PIEE's expansion and use does not appear to be slowing down.

2. How is the PIEE suite and development managed and can DOD leverage lessons to inform future, similar enterprise software development efforts?

Although DLA owns and operates the PIEE suite, governance and sustainment and development funding are shared. DLA's PIEE PMO, coupled with the PIEE Governance Board and PIEE User Group, make up the three groups in the governance structure. Together, the groups manage and operate the PIEE suite, guide its mission and strategic direction, conduct oversight of the platform and its operations, and represent and deliver feedback from end users of the applications in the suite. Overall, the governance structure and groups facilitate movement toward a goal of a one-stop-shop for procurement activities related to the end-to-end P2P process. Additionally, sustainment of the platform is shared by the military services and some defense agencies, ensuring the platform remains supported in the future and is not allowed to fall apart. Toward this end, some of the tools are required throughout the DOD, ensuring buy-in and support. Finally, although the PIEE PMO manages the development contract, all DOD entities can fund and develop applications and enhancements to applications within the PIEE suite while maintaining the enterprise vision and sharing tools with all users.

The governance structure, funding arrangement, and development processes all foster a culture of collaboration. Although DOD entities are not forced to develop applications nor enhancements within PIEE, being required to use and fund particular tools within the platform encourages them to engage and consider developing additions to the platform. Additionally, providing these entities and their user base with a voice in the process, its direction, and its development activities also ensures these entities work together to develop user-friendly processes and applications that are functional for all parties involved. The user-friendliness results in additional parties desiring to join, expanding the platform further. Since DOD entities already fund the platform via the WCF and sustainment funding is shared, further development does not increase sustainment costs to a noticeable degree and replacing legacy functions and applications can save sustainment costs by discarding other platforms funded entirely by individual entities; this is the situation DCMA finds itself in as it transitions from DCMA applications and platforms to the shared PIEE platform, saving tens of millions of dollars in annual sustainment costs. As part of every acquisition effort, including software development, procurement offices are required to conduct market research to determine what is available in the marketplace, what already exists, and whether what exists can meet their needs. PIEE suite development efforts are the result of such market research and future software development efforts, such as the next generation of contract writing systems and entitlement systems should take note. Given that the federal government follows the same acquisition and financial regulations, it is likely the DOD and federal agencies can all leverage similar functionality, sharing valuable, scarce resources, information, and expertise to collaborate and develop shared platforms and applications.

3. How do PIEE acquisition processes/decisions impact PIEE program management processes?

Many choices are made for the program management office due to acquisition processes and decisions prior to starting the project. As shown in this research, for example, the solicitation and resulting contract set many of the program management processes and pitfalls in motion. Overall, the complexity of the contract created challenges for post-award activities. Inclusion of four task areas, including program support, platform sustainment, application development, and a transition period, as well as the complex cost-plus incentive fee contract type for application development, small business teaming, the layout of the schedule, and the shortened length of the contract all produced downstream problems for the program office to manage. The decision to retain contract administration also prevented the program office from obtaining valuable support and expertise from DCMA.

Based on this research and the challenges the program office faced, the program office can work with the procurement office to eliminate some of the issues and mitigate others. Given the complexity of the contract type, for instance, and the conflicts between the contract type, the schedule, contractor performance and what is incentivized, it is wise to consider alternative contract types, from cost plus fixed fee to firm fixed price arrangements. Additionally, assigning contract administration to DCMA provides another avenue to spread the complexity out across more personnel and leverage its administrative expertise and contract support. Finally, the tasks could be broken up into separate contracts, the contractor could be required to develop a small business plan versus being forced to team with a small business, and the length of the award should be extended to five years to avoid having to unnecessarily complete acquisition processes too frequently.

4. What lessons can we take away from PIEE program management of software development efforts?

One significant lesson identified in researching the PIEE program's software development is the importance of developing and using high quality independent government cost estimates. Due to the lack of such quality estimates, requirements owners were left to negotiate without one, resulting in inconsistent results. For example, many projects were negotiated substantially lower than the contractor's initial proposal while others were not. Results varied, as some projects negotiated down significantly overran estimates, while others underran. Similarly, some project proposals that were not negotiated lower also overran estimates while others underran and came in on budget. Additionally, estimates provided after negotiations ended up slightly above the negotiated positions and bore little resemblance to labor categories provided by the contractor and the actual costs incurred through past development efforts. There is a significant amount of incurred cost data, as well as development data from user stories to interfaces and features and the story points required to develop these particulars. This can all be leveraged to produce better estimates and improve planning, negotiating, and inevitably development outcomes.

The development schedule poses its own challenges for managing the program. Each development cycle is relatively brief at three months in duration, including a planning phase, development phase, and a testing phase prior to releasing products into a production environment or continuing with another development period. This, coupled with requirements for scheduling development efforts, completing acquisition packages, securing funds from interested parties, and getting those efforts onto task orders, results in many conflicting demands. At the same time, this sort of fast-paced schedule aligns well with DOD software development demands for quick turnaround and the PIEE program pushes out new applications and enhancements on a regular basis. Some strategies for maintaining the schedule while mitigating some of the problems including better tracking of development efforts, tracing user stories back to requirements, obtaining the full development schedule picture when reviewing burndown charts, as well as comparing cost burndowns to performance burndowns to schedule burndowns. This enables the program office to forecast better and plan adjustments as needed within the short schedule. Additionally, the program office can implement continuous development testing as it has on occasion with larger projects, but employ it across the board on all projects. Instead of three weeks of testing after development is complete and when problems identified may disrupt significant portions of tool development that could take a lot of time and effort to resolve, continuous development testing enables product owners to identify issues early and often, resulting in earlier and less impactful fixes before developing on top of the problematic design.

There are many other lessons for program managers to take away from the PIEE program as well, including many successes. The amount of collaboration within organizations and across agencies and military services is remarkable. Despite all of the challenges, including competing interests, communication breakdowns, bureaucratic norms, and the many issues identified through this research, the PIEE program continues to produce valuable enterprise capabilities that do not exist elsewhere. As discussed in response to the next research question, this is all done at a fraction of the cost elsewhere.

5. How do PIEE efforts compare to the effectiveness, efficiency, and/or feasibility of similar efforts?

The results of the PIEE program stack up well compared with other efforts. For example, although the sustainment costs of the PIEE suite resemble those of other platforms, including DCMA’s eTools and IWMS, the quantity of users from across the DOD, as well as regular usage, is greater in PIEE than these other platforms. Furthermore, the sustainment costs of the PIEE platform are spread across the DOD, so entities like DCMA fund only a fraction of the sustainment costs in PIEE rather than the full amount of their own systems. Similarly, development costs as described throughout this paper are lower than many other similar efforts, including failed efforts. DCMA developed its IWMS system initially for approximately $45 million through 2017 and now plans to replace it within PIEE at significantly lower expense. The Army and Navy canceled their procurement system efforts after cost estimates continued to climb from their originally high $133.9 million and $222.9 million, respectively. In contrast, though much smaller, the DCMA-developed MDO contract writing system in PIEE already issues contract modifications and is anticipated to cost less than $2 million. Overall, development within the PIEE suite proves to be more efficient and effective than comparable efforts across the Department.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

While researching and analyzing software development in the PIEE program, the authors identified additional areas of research directly related to this topic. Unfortunately, this paper does not fully cover the PIEE program, as it is limited by both breadth and depth. The authors lacked the capacity to dig deeper into some areas, including cost estimating, forecasting, and agile metrics. Additionally, the authors were unable to review the entire program, to include the organizational culture of the various entities involved. Finally, although DCMA developed a value assessment covering some of the developed applications, there is an overall lack of data capturing user satisfaction and comparing PIEE outcomes with those in other software development programs. The following provides a sampling of areas for future research building on and expanding from the research in this paper.

1. Cost Estimating

Although this paper covered a variety of aspects and issues with the cost estimating processes, inputs, outputs, and outcomes associated with the PIEE program, the authors performed a shallow analysis of the topic overall. A plethora of data exists tied to the PIEE program. For example, the PIEE program possesses dozens of requirements packages from various entities developing applications and enhancements to applications in PIEE, as well as requirements for interfacing with other systems and applications, data transmission, and form population. Additionally, these requirements have all been developed into user stories, evaluated and estimated, and developed into applications and enhancements. This has all been tracked and documented, establishing average development rates or velocity over time. Furthermore, costs incurred for every effort, ECP, and sprint have been captured. The latter can be tied to the velocity, establishing estimates of costs associated with each story point. Then the requirements and user stories can be evaluated to determine the average story point counts for various categories of requirements and user stories. Finally, all of this data can be analyzed to produce more accurate cost estimates via various estimating methods. These estimates then provide the PIEE PMO with better negotiating leverage and the ability to schedule development timelines more accurately, preventing schedule slips and budgeting issues. These high-quality estimates could also be used to inform other software development efforts across the DOD. There is a lot of useful data presently unused in the PIEE program that could improve it dramatically.

2. Value Assessment

Originally, two of the research questions for this paper revolved around determining the value of the software developed via the PIEE program and end user satisfaction. Per DODI 5000.87 (2020b), these types of value assessments are necessary at least annually following fielding of software to determine whether development was successful, efficiencies are attained, end users are satisfied, investments are worth it, and if additional enhancements are needed. In researching this subject, the authors discovered a lack of such assessments, though DCMA recently completed one for its latest development efforts (Garris, 2023). DCMA's value assessment focused on whether the development efforts met the requirements of the agency and desires of the product owners involved, but also noted the quantity of outstanding trouble tickets by application, the value of individual features delivered by application, and a variety of other measures. Two notable and related missing elements from DCMA's assessment are 1) end user satisfaction and 2) end user change requests (Garris, 2023). However, DCMA developed support pages for each application for DCMA end users and is working to develop user satisfaction surveys and the capability for those users to submit change requests, so future value assessments ought to capture these elements.

Additionally, there is no comparison between the PIEE applications and similar acquisition applications developed elsewhere, whether within the DOD or elsewhere in the federal government. Research conducted for this paper touched on a comparison between PIEE MDO and other efforts to develop contract writing systems, but this analysis focused mainly on cost comparisons and a basic analysis of requirements, features, and the delivery of writing systems. An in-depth analysis comparing contract writing systems provides great value to the DOD and federal government. This research could better inform the acquisition community as to whether it remains necessary to develop separate writing systems for each military service, defense agencies, and the rest of the federal government and assist in identifying the best value for taxpayers. The PIEE platform provides a lot of value to the DOD as an enterprise-wide suite of acquisition applications. Since the DOD is the largest federal department, acquires the most, and procures the most complex systems, it is possible and perhaps even likely that the PIEE platform could expand to serve the entire federal government. Additional research into the value of the platform, user satisfaction, and a comparison with alternatives will inform program managers and federal and DOD acquisition leaders on how best to spend finite funds on procurement system development and modernization.

3. Organizational Culture

At the heart of any organization is the organizational culture. In reviewing the PIEE program, the authors discovered a few substantial problems that could easily doom any effort, but especially a DOD-enterprise software development effort with multiple program management offices, contractors, and defense agencies involved. For example, the program lacks quality cost estimates crucial to ensure the government gets the best value possible for its scarce resources. There is also a lack of traceability, velocity measures, and burndown analyses comparing cost and performance across the scheduled development period, leading to an inability to forecast accurately and mitigate issues timely. Finally, while researching the program, it was apparent many of the entities involved lacked the necessary personnel to perform all required duties and responsibilities, as many freely remarked as much. In spite of these conditions, the PIEE program continues to regularly release new applications and enhancements with more planned for the future. This begs the question "Why under such circumstances does the program continue to produce results and appear successful?" At first glance, it appears the unique collection of government personnel within various agencies and military services, government support contractors, and contractor personnel work hard in collaboration with one another to overcome such challenges. Wise (2010) points to such "public service motives" driving public servants to perform and behave in a certain manner to benefit the public good. Analyzing what drives the personnel associated with the PIEE program to push through difficulties and move the program forward is worthwhile to inform program managers about how to identify similar individuals to assign them to these efforts, incentivize such behaviors, and retain them in these positions long term.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

AcqNotes. (n.d.). *Global Information Grid (GIG)*. https://acqnotes.com/acqnote/careerfields/global-information-grid-gig

Agile Alliance. (n.d.). *Agile Manifesto*. https://www.agilealliance.org/agile101/the-agile-manifesto/

Ambler, S. W. (2009). Agile and iterative development: A manager's guide. *Pearson Education*.

Atlassian. (n.d.). *Learn about the main artifacts of agile scrum including prod*. https://www.atlassian.com/agile/scrum/artifacts

Atlassian. (2019). *Five agile metrics you won't hate | Atlassian*. https://www.atlassian.com/agile/project-management/metrics

Barno, D., & Bensahel, N. (2016). The Future of The Army: Today, Tomorrow, And The Day After Tomorrow. *Atlantic Council*. http://www.jstor.org/stable/resrep03676

Beck, K. (2000). Extreme programming explained: embrace change. *Addison-Wesley Professional*.

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., and Sutherland, J. (2001). Manifesto for agile software development. *Agile Alliance*. https://agilemanifesto.org/

Boehm, B. W. (1988). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 14–24.

Boehm, B., & Turner, R. (2004). Balancing agility and discipline: A guide for the perplexed. *Addison-Wesley Professional*.

CGI, Inc. (2017, June 14). *CGI awarded $133.9M contract to develop the U.S. Army Contract Writing System*. https://www.cgi.com/en/2017-06-14-CGI-awarded-133-9M-contract-to-develop-the-U-S-Army-Contract-Writing-System-ACWS

CGI, Inc. (2019, August 16). *CGI wins $222.9 million contract to develop electronic Procurement System (ePS) for U.S. Navy and Marine Corps*. https://www.cgi.com/us/en-us/2019-08-16-navy-marine-corps-electronic-procurement

Cockburn, A. (2001). Agile software development: The cooperative game. *Addison-Wesley Professional*.

Cuskey, J. R. (2016, Summer). Understanding the Mechanics of CPIF Contracts. *APTAC Connection*.

Defense Acquisition University (DAU). (n.d.-a) *Cost Estimation Methods*. https://www.dau.edu/acquipedia/pages/articledetails.aspx#!539

Defense Acquisition University (DAU). (n.d.-b). *Defense Information System for Acquisition*. https://www.dau.edu/glossary/Pages/Defense-Information-System-for-Acquisition-(DISA).aspx

Defense Acquisition University (DAU). (2021). *Procurement Integrated Enterprise Environment (PIEE)*. https://www.dau.edu/tools/t/Procurement-Integrated-Enterprise-Environment-(PIEE)

Defense Advanced Research Projects Agency (DARPA). (n.d.). *Artificial Intelligence*. https://www.darpa.mil/program/artificial-intelligence

Defense Contract Management Agency. (2021, March 3). *Modernization and Analytics Initiative (MAI): Business Case/Capabilities Need*. (Version 13.5). Department of Defense.

Defense Information Systems Agency (DISA). (n.d.-a). *Defense Information System for* Acquisition. https://www.disa.mil/disa-globals/disa-daa-organization/disa-daa

Defense Information Systems Agency (DISA). (n.d.-b). *Global Information Grid (GIG)*. https://www.disa.mil/Cybersecurity/Network-Services/Global-Information-Grid

Defense Innovation Board (DIB). (2018). *Software Acquisition and Practices (SWAP) Study: Volume 1: Findings and Recommendations*. https://media.defense.gov/2018/Oct/05/2002048904/-1/-1/0/DIB_SWAP_STUDYVOLUME_1_FINDINGS_RECOMMENDATIONS.PDF

Defense Innovation Board (DIB). (2020). *Software Acquisition and Practices (SWAP) Study*. https://media.defense.gov/2020/Oct/01/2002513056/-1/-1/0/DIB_SW_ACQUISITION_PRACTICES_STUDY_2020.09.29.PDF

Defense Innovation Board (DIB). (n.d.). *Software Acquisition & Practices (SWAP) Study*. https://innovation.defense.gov/software/

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (2023a, April). *Operations Requirements Committee (ORC)*. [PowerPoint slides].

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (2023b, April 5). *PIEE 6.16 PI 16 ECP Burndown Reports: Iteration 4*. [PowerPoint slides].

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (2023c). *Procurement Integrated Enterprise Environment (PIEE)*. https://piee.eb.mil

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (2023d). *Procurement Integrated Enterprise Environment (PIEE) Enhancements by Release*. https://pieetraining.eb.mil/wbt//portal/documents/PIEEEnhancementsByRelease.pdf

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (n.d.-a). *Procurement Integrated Enterprise Environment Training*. https://pieetraining.eb.mil/xhtml/unauth/web/homepage/functionalInfo.xhtml#:~:text=Specific%20DOD%20components%20and%20other%20government%20offices%20usi

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (2023e). *Procurement Integrated Enterprise Environment (PIEE) – Web Based Training (WBT)*. https://wawf-gt.eb.mil/wbt

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (n.d.-b). *WAWF - Wide Area Workflow*. https://www.dla.mil/Working-With-DLA/Applications/Details/Article/2924981/wawf-wide-area-workflow/

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (n.d.-c). *Wide Area Workflow (WAWF): Cost Voucher*. https://pieetraining.eb.mil/wbt/wawf/documents/CV.pdf

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (n.d.-d). *Wide Area Workflow (WAWF) Functional Information*. https://piee.eb.mil/xhtml/unauth/web/homepage/functionalInfo.xhtml

Defense Logistics Agency (DLA) Procurement Integrated Enterprise Environment (PIEE) Program Management Office (PMO). (n.d.-e). *Wide Area Workflow (WAWF): Invoice as 2 in 1*. https://pieetraining.eb.mil/wbt/wawf/documents/2n1.pdf

Defense Science Board. (2000). *Report of the Defense Science Board Task Force on Defense Software. Washington, D.C.: Office of the Under Secretary of Defense for Acquisition and Technology*. https://dsb.cto.mil/reports/2000s/ADA385923.pdf.

Defense Science Board. (2018, February). *Design and Acquisition of Software for Defense Systems*. https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

Defense Standardization Program Office. (n.d.). *About the Defense Standardization Program*. https://www.dsp.dla.mil/About-DSP/

Department of Defense. (2019a, September 23). *Agile metrics guide*. https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20Metrics%20v1.1%2020191122.pdf

Department of Defense. (n.d.-a). *Cybersecurity Maturity Model Certification (CMMC)*. https://www.acq.osd.mil/cmmc/index.html

Department of Defense. (2021b). *Defense Working Capital Fund: Defense-Wide Fiscal Year (FY) FY 2022 Budget Estimates Operating and Capital Budgets*. https://comptroller.defense.gov/Portals/45/Documents/defbudget/fy2022/budget_justification/pdfs/06_Defense_Working_Capital_Fund/PB_2022_DWWCF_Operating_and_Capital_Budget_Estimates.pdf

Department of Defense. (2022). *Defense Working Capital Fund: Defense-Wide Fiscal Year (FY) FY 2023 Budget Estimates Operating and Capital Budgets*. https://comptroller.defense.gov/Portals/45/Documents/defbudget/fy2023/budget_justification/pdfs/06_Defense_Working_Capital_Fund/PB_2023_DWWCF_Operating_and_Capital_Budget_Estimates.pdf

Department of Defense. (2021c). *Department of Defense Software Modernization Strategy*. https://media.defense.gov/2022/Feb/03/2002932833/-1/-1/1/DEPARTMENT-OF-DEFENSE-SOFTWARE-MODERNIZATION-STRATEGY.PDF

Department of Defense. (n.d.-b). *DOD Aims for New Enterprise-Wide Cloud by 2022*. https://www.defense.gov/News/News-Stories/Article/Article/2684754/DOD-aims-for-new-enterprise-wide-cloud-by-2022/

Department of Defense. (2018a). *DOD Cloud Strategy*. https://media.defense.gov/2019/Feb/04/2002085866/-1/-1/1/DOD-CLOUD-STRATEGY.PDF

Department of Defense. (2021a). *DOD Enterprise DevSecOps Strategy Guide Department of Defense* https://dl.DOD.cyber.mil/wp-content/uploads/devsecops/pdf/DODEnterpriseDevSecOpsStrategyGuide.pdf

Department of Defense. (2017, August 2). *DOD Restructures Acquisition, Technology Office to Improve Military Lethality, Speed*. https://www.defense.gov/News/News-Stories/Article/Article/1265231/DOD-restructures-acquisition-technology-office-to-improve-military-lethality-sp/https%3A%2F%2Fwww.defense.gov%2FNews%2FNews-Stories%2FArticle%2FArticle%2F1265231%2FDOD-restructures-acquisition-technology-office-to-improve-military-lethality-sp%2F

Department of Defense. (n.d.-c) *DOD Software Modernization Strategy Approved*. https://www.defense.gov/News/Releases/Release/Article/2924187/DOD-software-modernization-strategy-approved/

Department of Defense (DOD). (2019b, July 1). *Governance Charter: Procurement Integrated Enterprise Environment (PIEE)*. https://www.acq.osd.mil/asda/dpc/ce/cap/docs/piee/PIEE_Governance_Charter.pdf

Department of Defense. (2018b). *Joint Technical Architecture*. https://DODcio.defense.gov/Library/DOD-JTA/

Department of Defense Instruction. (2020a, March 13). *Cost Analysis Guidance and Procedures* (DODI 5000.73). Department of Defense. https://www.dau.edu/policy/_layouts/15/WopiFrame.aspx?sourcedoc=%2Fpolicy%2FPolicyDocuments%2FDODI%205000%2E73%20Cost%20Analysis%20Guidance%20and%20Procedures%2Epdf&action=interactivepreview

Department of Defense Instruction. (2022, June 8). *Operation of the Adaptive Acquisition Framework* (DODI 5000.02). Department of Defense. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/DODi/500002p.pdf

Department of Defense Instruction. (2020b, October 2). *Operation of the Software Acquisition Pathway* (DODI 5000.87). Department of Defense. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/DODi/500087p.PDF

Fossbytes. (2017, January 30). *Meet MOCAS — World's Oldest Computer Program That's Still In Use*. https://fossbytes.com/mocas-worlds-oldest-computer-program/

Freihofer, J. T., Dotson, D., & Maus, R. (2021, July 1). *Creating Incentivized Agile Contracts*. Defense Acquisition University: Defense Acquisition Magazine. https://www.dau.edu/library/defense-atl/blog/creating-incentivized-agile-contracts

Garris, W. A. (2023, March 13). Modernization and Analytics Initiative (MAI): Software Acquisition Pathway (SWP) Value Assessment (VA). Department of Defense: Defense Contract Management Agency, Functional Information Resource Management (FIRM) Center.

Government Accountability Office. (2021a). *Antideficiency Act Reports – Fiscal Year 2020*. (GAO Publication No. ADA-20-01). Washington, D.C.: U.S. Government Printing Office.

Government Accountability Office. (2020). *Defense Acquisitions: DOD Has Opportunities to Improve Oversight and Modernize Software Systems*. https://www.gao.gov/products/gao-20-407

Government Accountability Office. (2021b). *DOD Software Acquisition: Status of and Challenges Related to Reform Efforts*. (GAO Publication No. 21–105298). Washington, D.C.: U.S. Government Printing Office.

Government Accountability Office (GAO). (1998, January 30). *Financial Management: Seven DOD Initiatives That Affect the Contract Payment Process*. https://www.govinfo.gov/content/pkg/GAOREPORTS-AIMD-98-40/html/GAOREPORTS-AIMD-98-40.htm

Haley, E. T. & Schmidt, B. R. (2022, September). *Office of Naval Research (ONR) Saves Significant Time, Effort, & Money by Utilizing DCMA's New PIEE Modifications and Delivery Orders (MDO) Module*. Defense Contract Management Agency, Functional Information and Resource Management Center. https://DOD365.sharepoint-mil.us/sites/DCMA-Teams-PH-AQ-AQII/Lists/WAR/AllItems.aspx

Jacobs, M. (2023). *Procurement Integrated Enterprise Environment (PIEE) Overview*. [PowerPoint slides]. Defense Pricing and Contracting, Contracting eBusiness, Office of the Undersecretary of Defense for Acquisition and Sustainment.

Joint Artificial Intelligence Center. (2020). *JAIC Software Acquisition Framework*. https://www.ai.mil/docs/default-source/default-document-library/jaic-software-acquisition-framework.pdf

Kassab, M., Harris, A., & Nguyen, D. (2018). Understanding the challenges of agile software development in the Department of Defense. *Defense Acquisition Research Journal*, 25(1), 101–118.

Kim, Y., & Lee, K. (2018). An empirical study on the adoption and impact of agile methodologies in the Department of Defense software development. *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 15(3), 231–243.

Lohrmann, D. (2021). *What the DOD's JEDI cloud contract means for cloud computing*. Government Technology. https://www.govtech.com/computing/what-the-DODs-jedi-cloud-contract-means-for-cloud-computing.html

Mims, E. A. & Schmidt, B. R. (2023). *New Contract Administration Modules in PIEE*. [PowerPoint slides]. Defense Contract Management Agency, Functional Information Resource Management Center.

Moser, D. (2021). Defense Department grapples with modernizing its software. *Federal News Network*. https://federalnewsnetwork.com/defense-main/2021/02/defense-department-grapples-with-modernizing-its-software/

Propert, B. (2019). *Procure-to-Pay Capability Summary #29*. Procure-to-Pay Process Advocates Working Group (P2PPAWG). https://DODprocurementtoolbox.com/cms/sites/default/files/resources/2019-04/Procure-To-Pay%20Capability%20Summary%2029%20-%20PIEE.pdf

Royce, W. (1970, August). Managing the development of large software systems: Concepts and techniques. *Proceedings of the IEEE WESCON*, 1–9.

Royce, W. (1998). Software project management: a unified framework. *Addison-Wesley Professional*.

Schmidt, B. R. (2021). *Analysis of Navy ePS, Army ACWS, and DCMA MDO Contract Writing System Requirements Packages*. [Data set].

Schmidt, B. R. (2023a). *Analysis of SP4701 Contract Actions from 1 January 2013 – 26 March 2023* [Data set]. System for Award Management (SAM). https://sam.gov/reports/awards/adhoc

Schmidt, B. R. (2023b). *Analysis of S5102A-Issued Modifications from 1 June 2022 – 30 April 2023* [Data set]. Electronic Data Access (EDA). https://app.eda.fedmall.mil/award

Schwaber, K., & Sutherland, J. (2017). *The Definitive Guide to Scrum: The Rules of the Game*. https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

Singh, A., & Chhabra, J. (2015). Comparative analysis of software development methodologies. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(9), 274–277.

Swider, B., & Zimmermann, T. (2016). Cultural barriers to agile adoption in the Department of Defense. *Journal of Defense Software Engineering*, 29(1), 56–60.

Tenaglia, J. M. Defense Pricing and Contracting. (2023). *Guidance on Awarding Cost-Reimbursement Contracts*. Office of the Under Secretary of Defense (A&S).

Vergun, D. (2021, January 27). *Software Modernization Means Moving Faster, Smarter*. U.S. Department of Defense. https://www.defense.gov/News/News-Stories/Article/Article/2484578/software-modernization-means-moving-faster-smarter/.

Veritis. (2019, February 11). *DevSecOps Implementation Process and Road Map – Security at Every Step*. DevOps. https://devops.com/devsecops-implementation-process-and-road-map-security-at-every-step/

Wise, L. R. (2010). The Public Service Culture. In R. J. Stillman II, *Public Administration: Concepts and Cases* (pp. 320–330). Wadsworth Cengage Learning.

Acquisition Research Program
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net