SYM-AM-24-027

Excerpt from the Proceedings

of the

Twenty-First Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 8–9, 2024

Published: April 30, 2024

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

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

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

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

Managing Resources in a DoD Space-based Agile/DevSecOps Program

Michael Orosz—directs the Decision Systems Group at the University of Southern California's Information Sciences Institute and is a Research Associate Professor in USC's Sonny Astani Department of Civil and Environmental Engineering. Dr. Orosz has over 30 years' experience in government and commercial software development, systems engineering and acquisition, applied research and development, and project management, and has developed several successful products in both the government and commercial sectors. Dr. Orosz received his BS in engineering from the Colorado School of Mines, an MS in computer science from the University of Colorado, and a PhD in computer science from UCLA.. [mdorosz@isi.edu]

Lt Col Jacob M. Hempen—is Squadron Commander and Materiel Leader for the PNT Integrated Mission Delta, Software Engineering Squadron, Los Angeles AFB, CA, where he is responsible for three GPS Command and Control programs. Lt Col Hempen's career spans UAS flight test operations in Afghanistan/Iraq, cyber development, GPS Navigation Warfare programs, and High-Altitude SIGINT satellite development. Recently, Lt Col Hempen led an NSC-chartered missile and space launch technology committee under the State Department. In his last assignment, he was the GPS User Equipment Program Element Monitor for the Assistant Secretary of the Air Force for Space Acquisition and Integration. [jacob.m.hempen@spaceforce.mil]

Lt Col John "Mack" Turner—joined the Space Force in 2022 after 15 years in the U.S. Army. He has held all Artillery positions at company level including both Battery and Company Command. After supporting PEO Ammunition, he joined PEO C3T modernizing Mission Command software. In his last assignment, he oversaw Artillery software requirements for the Fires Center of Excellence. Since entering the Space Force, Mack has supported the follow-on effort for the Next Generation Operational Control System (OCX 3F). He holds a BS in Electronic and Information Technology Systems and an MA in Defense and Strategic Studies.

Brian Duffy—is a Senior Systems Engineer with the University of Southern California's Information Sciences Institute (USC/ISI). He conducts research and analysis to determine system engineering methods and metrics necessary to transition Major Defense Acquisition Programs from a traditional waterfall development to Agile/DevSecOps processes. Prior to USC/ISI, Brian retired from the United State Air Force with multiple assignments related to National Security Space acquisition programs, and command and control systems. Brian holds a Master of Aeronautical Science from Embry-Riddle Aeronautical University and a Bachelor of Aeronautical and Astronautical Engineering from the University of Washington.

Craig Charlton—is a Senior Systems Engineer at the University of Southern California's Information Sciences Institute (USC-ISI) and has provided acquisition support at Space Systems Command (SSC) at the Los Angeles AFB during the past 20 years on several leading-edge satellite systems. Prior to his position at SSC, Charlton acquired more than 25 years of experience as a software engineer and in managing software projects in the commercial world, primarily in the fields of engineering and of law enforcement. Charlton received a BA in mathematics from California State University, Long Beach.

Abstract

The University of Southern California's Information Sciences Institute (USC/ISI), along with funding and the active support and engagement of the USAF and USSF, has undertaken a series of case studies focused on developing lessons learned and identifying best practices when agile and DevSecOps methodologies are introduced into the space-based software-only acquisition environment.

A major focus of this research is discovering challenges and exploring solutions to managing resources throughout the agile/DevSecOps system development process. Such challenges include managing dependencies on external and internal systems; staff loading and specialties over the course of the program; the introduction of new capabilities; and the availability,

collection, and analysis of performance metrics for improved situational awareness. It is important to efficiently manage a program as it progresses toward the later stages of the development effort. The stories and features with low resource demand typically are already completed by the later stages, leaving stories and features with high resource demand to be addressed late in the program.

Introduction

Funded by the USAF and then by the USSF, the University of Southern California's Information Sciences Institute (USC/ISI) has led a series of case studies focused on developing lessons learned and identifying best practices when agile and DevSecOps methodologies are introduced into the space-based software-only acquisition environment. Although much has been written in applying agile and DevSecOps to DoD acquisition programs, much of this research has taken a broad view of project management (e.g., Proctor & Daniels, 2020). The research described in this paper is focused on the day-to-day (in some cases almost hour-by-hour) operations of several agile/DevSecOps-based projects with a major focus on discovering challenges and exploring solutions to managing resources throughout the agile/DevSecOps system development process. Such challenges include managing dependencies on external and internal systems; staff loading and specialties over the course of the program; the introduction of new capabilities; and the availability, collection, and analysis of performance metrics for improved situational awareness. It is important to efficiently manage a program as it progresses toward the later stages of the development effort. The stories and features with low resource demand typically are already completed by the later stages, leaving stories and features with high resource demand to be addressed late in the program.

The seven-year effort (to date) involved three major projects that span a fully waterfall effort (serving as the baseline) to a predominantly agile/DevSecOps hybrid project. The focus of this paper is primarily on the predominantly agile/DevSecOps hybrid project that is currently about mid-way through scheduled completion. Initial observations and lessons learned include:

- Perform necessary upfront systems-engineering to help populate the initial agile project backlog, map features with compliance requirements, identify staff loading and specialties, and to identify dependencies as early as possible in the program.
- Establish early in the program a near operational environment with high-fidelity simulators for system-wide integration and testing.
- Plan sprints with sufficient margin to manage unexpected events such as emerging technology insertion or unexpectedly complex stories.
- Ensure that licensing, intellectual property (IP), accreditation, certification and other programmatic issues are resolved early in the program.
- Plan for on-board and continuous training to ensure a productive workforce.
- Be prepared to create customized performance tracking tools.

Funding Support

The material covered in this paper is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ003419D0003. The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense or ASD(R&E).

Multiple Projects

As reported in Orosz et al. (2022), the study covers three software-focused acquisition programs summarized in Table 1.

Table 1 List of projects in the study

Project	Description	Waterfall/ Agile	Comments
A	Added new capability to an existing space-based command and control application.	100% waterfall	178K Software lines of code. Serves as the **baseline** for the multi-case study. Project completed.
B	Added new capability to Project A.	50% agile/50% waterfall	128K Software lines of code. Similar code complexity as Project A. Project completed.
C	Add new capability to a system that itself is in the final stages of development.	70% agile/30% waterfall	150K Software lines of code (estimated). Similar code complexity as Projects A and B. The 30% waterfall portion is primarily in support of EVM (DAU, n.d.), CDRLs (AcqNotes, 2024a) and IMS (AcqNotes, 2024b) activities that are normally associated with a DoDI 5000.02 waterfall effort.

Although this paper is a summary of lessons learned and best practices for all three projects, the findings reported predominantly reflect the efforts from Project C. The development effort in Project C relies on a nuanced implementation of SAFe® (Scaled Agile Framework, 2024) agile process along with multiple DevSecOps pipelines. As noted in Table 1, the 30% waterfall portion is primarily in support of activities involved with the Integrated Master Schedule (IMS), Earn Value Management (EVM), and the completion of multiple documents listed in the Contract Data Requirements List (CDRL; AcqNotes, 2024a).

Methods

Project Immersion

To fully understand the resource challenges in agile/DevSecOps-based projects, members of the USC/ISI team fully immersed in each of the projects. USC/ISI researchers are members of multiple integrated project teams (IPTs; AcqNotes, 2024c) and participate in all events and activities associated within a space-based agile effort. This includes participating in sprint and program increment ceremonies, reviews, planning sessions, scrums, Kanbans, demonstrations, working groups, technical evaluations, trade studies, management meetings such as PMRs, and other activities associated with an agile-based project.

In addition, immersion includes interacting with prime contractors and their subcontractors as well as other government agencies that compose the overall (system of systems) enterprise. As part of the immersion, USC/ISI researchers collect and analyze project performance data and provide systems engineering subject matter expertise. The USC/ISI team also collects lessons learned from both the government and the prime contractor.

Data Collection

Data collection involves both observations (from day-to-day immersion activities) and the collection of project performance data which includes the daily tracking of completed story points, features, and system requirements, as well as progress toward minimum viable product (MVP) and minimum marketable product (MMP) milestones. Much of the performance data was collected via customized tools that extracted sprint and program increment (PI) data from the prime contractor's Jira® (Atlassian, n.d.) issue tracking system and the DOORS Next Generation (DNG; IBM, n.d.) requirements tracking systems.

The collected data was analyzed and compared against MVP and MMP milestones based on an Integrated Master Schedule (IMS). Performance tracking also included tracking the number of stories and features that are not completed within the assigned timebox and "spill over" from one sprint or PI to the next (or future sprint or PI). Observations include tracking workforce movements (on and off the project) and the availability of external resources to the project. As Project C matures, the USC/ISI team will be collecting bug and discrepancy reports (DRs) that result from the integration and testing component of the DevSecOps pipeline.

Results and Lessons Learned

Upfront Engineering

As noted in Orosz et al., (2023) it is important to perform the necessary upfront systems-engineering to help populate the initial agile project backlog, map features with compliance requirements, and identify the initial "scaffolding" of the system design. From a resource management perspective, undertaking initial systems engineering is also important to help identify staff loading and specialties, and to identify dependencies (e.g., high-fidelity simulators, algorithm development, or interfaces), as well as the availability of alternatives if those resources are not available (i.e., build a simulator). This is also the time to establish the necessary policies and practices that promote one or more of the eight aspects of systems engineering agility (Dove et al., 2023) such as identifying the types of performance data to collect to improve program situational awareness. As noted in Dove et al. (2023), such measures can greatly mitigate systems rework and other challenges as the program evolves.

It is important to note that this upfront systems-engineering effort is not about undertaking a detailed design (common in waterfall efforts); rather the effort involves making high-level design trades and defining the system's architecture down to "black box"-like descriptions (preferably in a MBSE application such as Cameo (CameoMagic Solution, n.d.)). These descriptions should define interfaces (internal and external) and performance windows.

An output of the upfront systems engineering effort is the initial population of the project backlog (list of features and associated sizes) – along with priorities. Project backlog priorities should focus on developing useful functionally (a tenet of agile), instead of advancing multiple MVPs at the same time. During the upfront engineering process, it is important to recognize that for most acquisition projects, particularly software-based efforts, there will come a point in the effort when a decision could be made to eliminate or defer capability. These decisions must be made early, or quickly once encountered, to avoid wasting development resources. Although in an agile environment, eliminating capabilities normally involves removing or reducing the priorities of work on the project backlog, the challenge is for programs where a portion of the capability to be removed or deferred has already been developed (i.e., developed code). In many of these DoD programs, there is usually a security requirement (Nord et al., 2021; SD-Team, n.d.) that specifies that no code should be delivered in the operational system. In such cases, additional cost may be involved to rework the existing baseline to remove this dead-code. Although this situation is not uncommon in systems engineering and acquisition, failure to consider this situation when first developing the outlines of the system architecture and project backlog sequencing can lead to resource challenges later in the program. This challenge is particularly an issue for hybrid waterfall/agile programs where multiple MVPs are planned in parallel (e.g., to take advantage of available resources). If one or more MVPs are dropped from the program, the developed source code may have to be removed from the system.

Adequate upfront systems engineering is also important for projects where the workforce frequently changes, and the resulting "corporate knowledge" is lost. In this case, the underlying reasoning and rationale behind the creation of the initial feature definition and priority on the project backlog is lost too and often hampers future feature refinement activities. What typically happens on larger programs is the initial systems architecture, use cases, and project backlog are defined; however, backlog features lack detail (or high-level detail). These under-defined features are assigned to be worked in the future, long after the original team members involved in the original design have left the program. In such cases, considerable effort is required for the existing development team to "understand" the system or MVP – including the intent of the system requirements.

Not All Staff Skillsets Are Equal

In complex projects with evolving requirements (usually based on changing customer needs and priorities) or unavailable resources, the project backlog is usually subject to frequent changes. This usually involves a shuffling of features or stories within the project backlog with lower priority features/stories promoted in priority, completable with what is ready at the time, while higher priority features/stories are demoted in the project backlog and blocked until a specific resource is available. Often, certain requirements and their linked features (i.e., the work) require specific skill sets that may be unique to a handful of developers.

In many agile implementations, it is assumed by management that developers can easily jump from one feature to another. While a team of interchangeable skillset developers is a program manager's goal, in practice this is rarely the case, and can result in situations where a sprint team has staff, often experienced and expensive, that are not familiar with the current high-priority features at the top of the project backlog. Also, newer staff may not have the requisite experience with a highly technical area of the system. Adding new personnel often results in project delays as the prime contractor or government must find and hire a developer with the required skillset. In many cases, it can take up to a year – from the initial position announcement (the req) to the day the individual is hired, and this does not include any training that may need to be included before the new hire is productive.

To reduce the impact to the project development efforts from key personnel transitioning on and off a project, there are a few steps a program can take.

- Implement a continuous training program to help quickly ramp up new staff and help keep existing staff up to date on the project and on evolving technologies and system acquisition processes.
- Rely on MBSE and other digital engineering processes and applications to capture the decision-making behind the system design. It is not enough to capture the design of the system; it is also important to capture the decision-making behind the system design.

Near Operational Test Environments

Most DoD acquisition programs are quite complex and consist of a system of systems configuration involving both internal and external elements (i.e., systems and subsystems). To adequately test various elements requires access to all elements of the enterprise. This is particularly true in operations that involve DevSecOps pipelines that frequently (daily, weekly, etc.) undertake integration and testing runs to ensure content recently added to the system does not “break” existing development content. In addition, to sell off a system requirement (i.e., confirm that the system meets the specifications of the requirement), the system must be executed in the actual operating environment (which is almost impossible) or in a near operational environment that closely simulates the actual operating environment.

The challenge, as noted in Orosz et al. (2023) is that one or more of these internal and external elements may not be available (either in the form of the actual system or as a simulator). The element may currently be under development (often by a third-party vendor) or is in use by other programs or the operating environment. This can result in program schedule slippage or delay due to the lack of an adequate integration and test environment. These delays also often result in key personnel with high-demand skills being pulled to support other programs with no guarantee of returning to the program – necessitating the need to hire replacements, and thus adding an additional source of schedule delay.

This necessitates the need for high-fidelity simulators and near-operational environments much earlier in an agile program than in waterfall. Due to the fast-paced nature of agile/DevSecOps, a program cannot wait for these systems to materialize. Simulators and near-operational environments need to be developed as soon as possible, as enablers for system integration, testing, and ultimately rapid delivery of new functionality. It is recommended that this criterion be part of the contract language and that contract performance be linked to the completion of these systems. In addition, the project backlog and MVP/MMP sequence need to reflect the availability of the simulators/near-operational environment to help drive the development of these elements. Doing this will also allow the program to complete stories/features when the appropriate integration and test environment is available (a form of risk management). Ideally, these simulators should be part of the system model (i.e., MBSE).

Sprint Margins

As noted in Orosz et al. (2023), there is a tendency by developers to completely fill a sprint or program increment timebox with work (e.g., features/stories) leaving no margin for unexpected work, unexpected code complexity, or unplanned staff challenges. If margin is not present, there is an increased risk of stories/features slipping into the next development interval (i.e., sprint or program increment) causing the schedule to stretch/extend. A continuing cascade of slipped stories/features will cause a bow wave of work to emerge in the project backlog, resulting in some work being addressed later, possibly months or even years into the future. Such delays may require additional personnel to be hired, or worse, there is a risk that key personnel with the necessary skillsets will be “loaned out.” While the program waits for the delayed work to begin, there is a possibility of the loaned personnel never returning to the project. This can be a particular challenge where subcontractors must leave a project when a contract ends, and they are no longer available when the delayed work can finally be addressed.

Licensing and Other IP Considerations

Also as noted in Orosz et al. (2023), it is critically important to address licensing and Intellectual Property (IP) considerations up front prior to the start of system development. It is surprising how easy it is to underestimate the amount of time required to address IP and licensing issues. Although such issues are typically part of a vendor’s proposal, often there is insufficient understanding of the full scope of the project, which often leads to a bill of material (BOM) being inadequately defined. When new licensing and IP needs are identified later in the development process, the delays in negotiating terms can greatly impact project performance.

Although it is impossible to fully understand every need of an agile program upfront, considerable “pain” can be avoided in the future if some upfront engineering is undertaken (as already discussed).

Also be aware that some third-party vendor solutions may be A) incompatible with the current or evolving system and/or B) foreign developed and restricted from use on the project due to the classified nature of the environment. In addition, there are also security issues that need to be addressed when introducing new tools and IP into a system. For example, in classified programs, time needs to be allotted to the screening of applications for security and safety reasons. All these delays can result in the development schedule shifting to the right, putting stress on costs and on the available workforce.

Costing

There are many costing challenges in agile-based acquisition projects (e.g., EVM lagging the project backlog, etc.), but a particularly challenging area is in the use of T-shirt sizing, used in some agile projects for gauging capacity, but then extended to estimate labor hours and cost for a particular task. Costing using T-shirt sizes can be misleading as the method is typically based on a “one-size” fits all approach to estimating labor hours. This is particularly a challenge in programs where the T-shirt size approach includes development, integration & testing, and discrepancy report (DR) work off an all-in-one quote.

Often, a contractor will estimate the effort using a range of T-shirt sizes: extra-small, small, medium, large, extra-large, and so on. These sizes often cover a range of hours (e.g., extra-small may involve tasks that range from 1–300 hours to complete, a large T-shirt may range from 1,000–3,000 hours, etc.). These T-shirt sizes are usually based on the proposing contractor’s experience on similar projects. For a new project, these estimates are probably suitable as there are many unknowns and so relying on the contractor’s best judgment is reasonable. The challenge can emerge when new capability is inserted into the project backlog.

Presumably, as a system is developed, the contractor (and the government) will have a better understanding of what effort is required to complete tasks of various complexities. As such, the contractor will have a better understanding of what an extra-small T-shirt size of effort really involves. When new capability is added to a project (via a Request for Change (RFC)), the T-shirt sizes quoted rarely reflect experience gained on actual costs. This is particularly a challenge when T-shirt sizes include development, integration and testing, and DR work-off.

In addition, when RFCs are placed on contract, often the capability added is similar to capability that has yet to be developed on the original contract. In some cases, integration and testing could cover multiple capabilities – existing and new RFC capabilities thus reducing the workforce costs for undertaking individual I&T efforts for each individual capability.

If T-shirt sizing is used, it is recommended that an estimation system should be maintained by updating the capability T-shirt sizes when “as run” information becomes available.

Customized Performance Tracking Tools

As noted in Orosz et al. (2023), program management will need to be prepared to develop performance tracking tools. Although there are third-party performance tracking tools, many of these tools cannot be used due to licensing or import control issues (i.e., foreign owned applications). In other cases, existing performance tracking tools, particularly those that are waterfall oriented (such as EVM and IMS applications) present data that often lags the actual project performance by months. By the time a problem is recognized using these tools, it is often too late for corrective action, resulting in significant cost and schedule delays.

Developing or customizing performance tools typically involves leveraging a program's existing data sources. For example, in agile-based projects, Jira is often used to track day-to-day performance of the project backlog, DNG is used for requirements, and MS Project the IMS (tied into EVM). Readily available applications, like MS Excel, Visual Basic for Applications, and Python, can be used to create customized tools tailored to produce specific performance-tracking information, which can be created from the fusion of these data sources. In many cases, these tools can be developed and applied across multiple programs.

For Project C, several tools were developed and integrated to provide the following performance metrics.

- Intra-program historical information: features and PI assignments, story and sprint assignments, feature and MVP/MMP assignments, and status changes (Figures 1 through 4).
- Identification of features mapped to compliance requirements (Figure 5).
- If a ticket management system (e.g., Jira) and an IMS are used, then a tool is needed to synchronize data or determine if the systems are showing different entries.

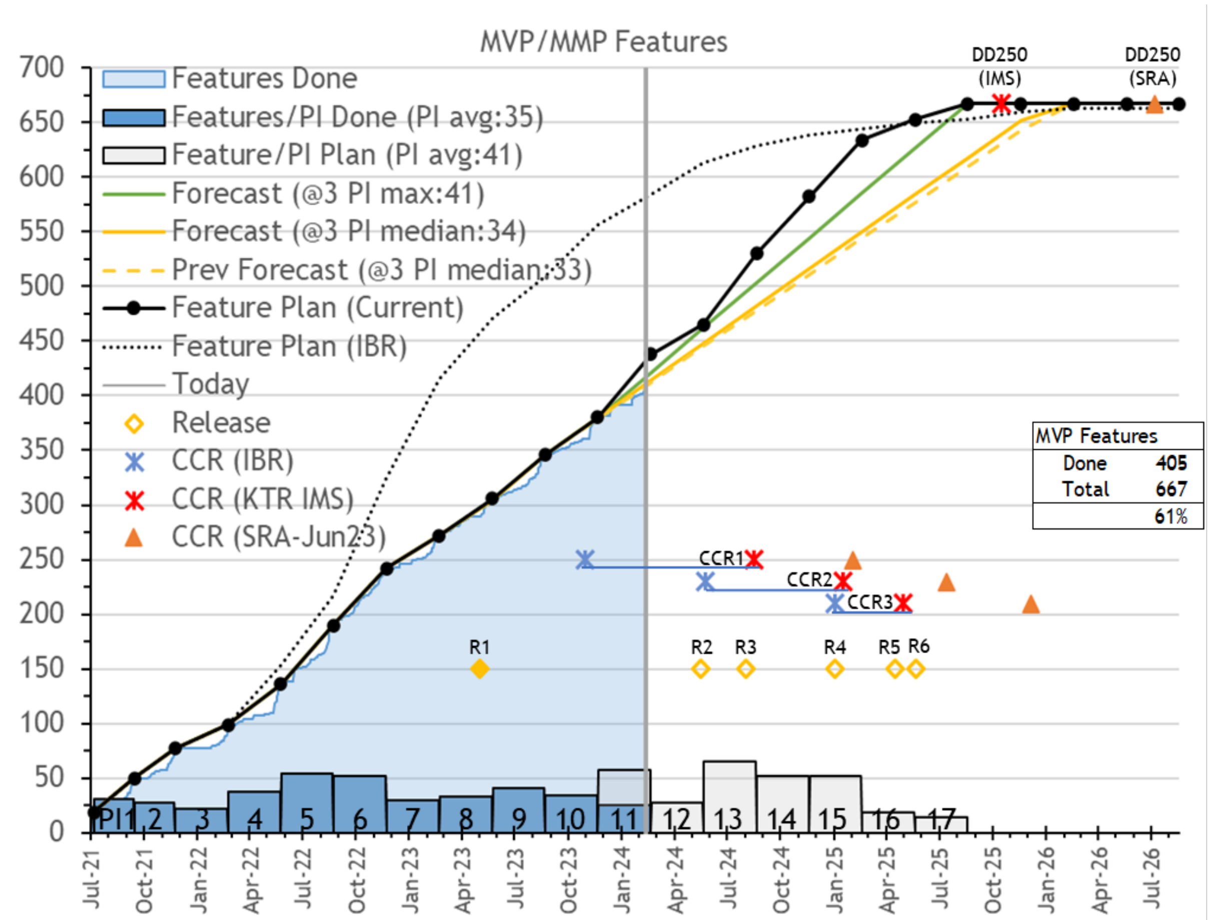


Figure 1 – MVP/MMP Plan and Progress Chart. (Orosz et al., 2023b)

The graphic in Figure 1 shows the progress the project is making toward completing features linked to assigned MVPs and MMPs (i.e., a "burn down" chart). The dotted black line is the original feature to Program Increment (PI) allocation plan created at the time of the program's Integrated Baseline Review (IBR). The solid black line is the current cumulative plan for features assigned to PIs and becomes the "as run" when PIs complete. The blue fill area shows the cumulative completion of features. The green and yellow lines show forecast completions at maximum and median rates, and as a comparison the previous PI's median forecast is also displayed. A breakout of the individual PIs (features planned and done) is along the bottom of the graphs. The milestones show Critical Capability Releases (collection of MVPs and MMPs releases) from the IBR, latest Schedule Risk Assessment (SRA), and contractor's Integrated Master Schedule; and incremental software releases (collection of MVPs/MMPs). What is not shown in Figure 1 is current and planned workforce (available but removed as sensitive information). Of note is that the contractor FTE (Full Time Equivalent) count remained relatively steady (+/- 5%) from PI 4 through PI 11.

Figure 1 shows that if the project continues at the current "burn up" rate, the development will not be complete until what looks like PI19. The original plan called for the bulk of the features to be completed by PI 13 with "clean-up" features to follow.

Feature Team	Key		Status	Δ Date	Points	%	MVP/ MM	PI History	MVP History	Linked Rqmts (DNG)	PI	Issue Type	Updated
TMC	DEVC2-8843	F	In Progress	28-Nov-23	34 | 38	89	(6-b)	1-06),PI10(2023-03-26),PI11(2023-07-11)	(4-a) 2022-08-09,(6-b) 2023-10-19	RQ4342	11	Feature	02-Feb-24 13:23
SIM	DEVC2-477	F	In Progress	07-Sep-23	12 | 18	67	(3-b)	1-27),PI10(2023-04-11),PI11(2023-11-22)	08-29,(4-c) 2022-02-05,(3-b) 2022-02-11	RQ3279, RQ4439	11	Feature	02-Feb-24 13:23
SIM	DEVC2-8386	F	In Progress	26-Oct-23	23 | 31	74	(4-c)	1-27),PI10(2023-04-11),PI11(2023-05-04)	Non-M 2022-07-06,(4-c) 2023-01-18	RQ4386	11	Feature	02-Feb-24 13:23
SIM	DEVC2-8385	F	Done	15-Dec-23	9 | 9	100	(4-c)	2-13),PI10(2023-04-11),PI11(2023-07-11)	Non-M 2022-07-06,(4-c) 2023-01-18		11	Feature	02-Feb-24 13:23
SIM	DEVC2-813	F	Done	01-Feb-24	33 | 33	100	(3-b)	2-13),PI10(2023-08-23),PI11(2023-11-22)	(3-b) 2021-08-29	RQ4380, RQ4381, RQ4384, RQ4385, RQ4	11	Feature	01-Feb-24 18:48
SIM	DEVC2-1675	F	Done	01-Feb-24	28 | 28	100	(3-b)	5-22),PI10(2023-08-23),PI11(2023-11-22)	(3-b) 2021-08-29	RQ2750, RQ4398	11	Feature	01-Feb-24 18:43
TMC	DEVC2-554	F	In Progress	10-Apr-23	44 | 47	94	(5-a)	1-12),PI10(2023-04-10),PI11(2023-07-10)	(5-a) 2021-08-29	RQ1008, RQ1235, RQ370, RQ4050, RQ40	11	Feature	01-Feb-24 16:03
SFG	DEVC2-4430	F	In Review	01-Feb-24	0 | 1		(6-c)	3-11),PI12(2023-01-04),PI11(2023-07-07)	(3-d) 2021-10-04,(6-c) 2022-01-10		11	Feature	01-Feb-24 11:45
SFG	DEVC2-4429	F	In Review	01-Feb-24	0 | 1		(6-c)	0-06),PI10(2023-01-04),PI11(2023-07-10)	(3-d) 2021-10-08,(6-c) 2022-01-10		11	Feature	01-Feb-24 11:44
TMB	DEVC2-13770	F	In Progress	28-Nov-23	25 | 41	61	(6-b)	PI11(2023-10-10)	(4-a) 2023-10-10,(6-b) 2023-10-12		11	Feature	01-Feb-24 11:04
TMB	DEVC2-2268	F	In Review	01-Feb-24	16 | 16	100	(5-a)	7-10),PI12(2023-09-14),PI11(2023-10-09)	(5-a) 2021-08-29		11	Feature	01-Feb-24 11:01
TMA	DEVC2-1551	F	In Progress	20-Nov-23	56 | 72	78	(4-a)	1-18),PI10(2023-02-06),PI11(2023-07-12)	(4-a) 2021-10-29		11	Feature	01-Feb-24 10:13
MET	DEVC2-13005	F	In Progress	28-Nov-23	25 | 37	68	Non-M	PI11(2023-09-19)	Non-M 2023-09-19		11	Feature	01-Feb-24 10:13
MET	DEVC2-13086	F	In Progress	27-Nov-23	52 | 76	68	Non-M	PI11(2023-09-21)	Non-M 2023-09-21		11	Feature	01-Feb-24 10:13

Figure 2 – Status Board of Features in Current PI

The graphic in Figure 2 shows a status board of daily collected Jira data for status, status change date, assigned cumulative story points, percent complete, and assigned MVP/MMP (orange columns) filtered for features in the current program increment (ending 20 Feb 24, column removed for brevity) and sorted by update date (green columns). Jira exports, collected daily, allow for off-line analysis such as tracking a feature's assigned PI or MVP and change date (left and center blue columns). Off-line analysis tools also enable data fusion such as the association of requirements information from DNG and features maintained in Jira (right blue column).

While the prime contractor's Jira platform natively contains this information, it is only available while logged into platform. Collection of data for off-line display and analysis enables the government program management team to have an independent historical record.

Feature Team	Key		Status	Δ Date	Points	%	MVP/ MM	Milestone ID	IMS Start	IMS Finish	IMS&PI Matcl	PI	Issue Type	Updated
TMC	DEVC2-8843	F	In Progress	28-Nov-23	34 | 38	89	(6-b)	DEVC2.4241	04-Dec-23	13-Feb-24		11	Feature	02-Feb-24 13:23
SIM	DEVC2-477	F	In Progress	07-Sep-23	12 | 18	67	(3-b)	DEVC2.1975	13-Sep-23	09-Feb-24		11	Feature	02-Feb-24 13:23
SIM	DEVC2-8386	F	In Progress	26-Oct-23	23 | 31	74	(4-c)	DEVC2.3686	25-Oct-23	22-Jan-24		11	Feature	02-Feb-24 13:23
SIM	DEVC2-8385	F	Done	15-Dec-23	9 | 9	100	(4-c)	DEVC2.3685	25-Oct-23	15-Dec-23		11	Feature	02-Feb-24 13:23
SIM	DEVC2-813	F	Done	01-Feb-24	33 | 33	100	(3-b)	DEVC2.2717	10-Jul-23	08-Feb-24		11	Feature	01-Feb-24 18:48
SIM	DEVC2-1675	F	Done	01-Feb-24	28 | 28	100	(3-b)	DEVC2.1969	23-Nov-22	08-Feb-24		11	Feature	01-Feb-24 18:43
TMC	DEVC2-554	F	In Progress	10-Apr-23	44 | 47	94	(5-a)	DEVC2.3791	10-Apr-23	13-Feb-24		11	Feature	01-Feb-24 16:03
SFG	DEVC2-4430	F	In Review	01-Feb-24	0 | 1		(6-c)	DEVC2.3343	03-Jan-24	13-Feb-24		11	Feature	01-Feb-24 11:45
SFG	DEVC2-4429	F	In Review	01-Feb-24	0 | 1		(6-c)	DEVC2.3442	22-Nov-23	13-Feb-24		11	Feature	01-Feb-24 11:44
TMB	DEVC2-13770	F	In Progress	28-Nov-23	25 | 41	61	(6-b)	DEVC2.4265	14-Nov-23	05-Feb-24		11	Feature	01-Feb-24 11:04
TMB	DEVC2-2268	F	In Review	01-Feb-24	16 | 16	100	(5-a)	DEVC2.2858	22-Nov-23	13-Feb-24		11	Feature	01-Feb-24 11:01
TMA	DEVC2-1551	F	In Progress	20-Nov-23	56 | 72	78	(4-a)	DEVC2.3745	27-Nov-23	12-Apr-24		11	Feature	01-Feb-24 10:13
MET	DEVC2-13005	F	In Progress	28-Nov-23	25 | 37	68	Non-M	DEVC2.4273	22-Nov-23	13-Feb-24		11	Feature	01-Feb-24 10:13
MET	DEVC2-13086	F	In Progress	27-Nov-23	52 | 76	68	Non-M	DEVC2.4287	22-Nov-23	13-Feb-24		11	Feature	01-Feb-24 10:13

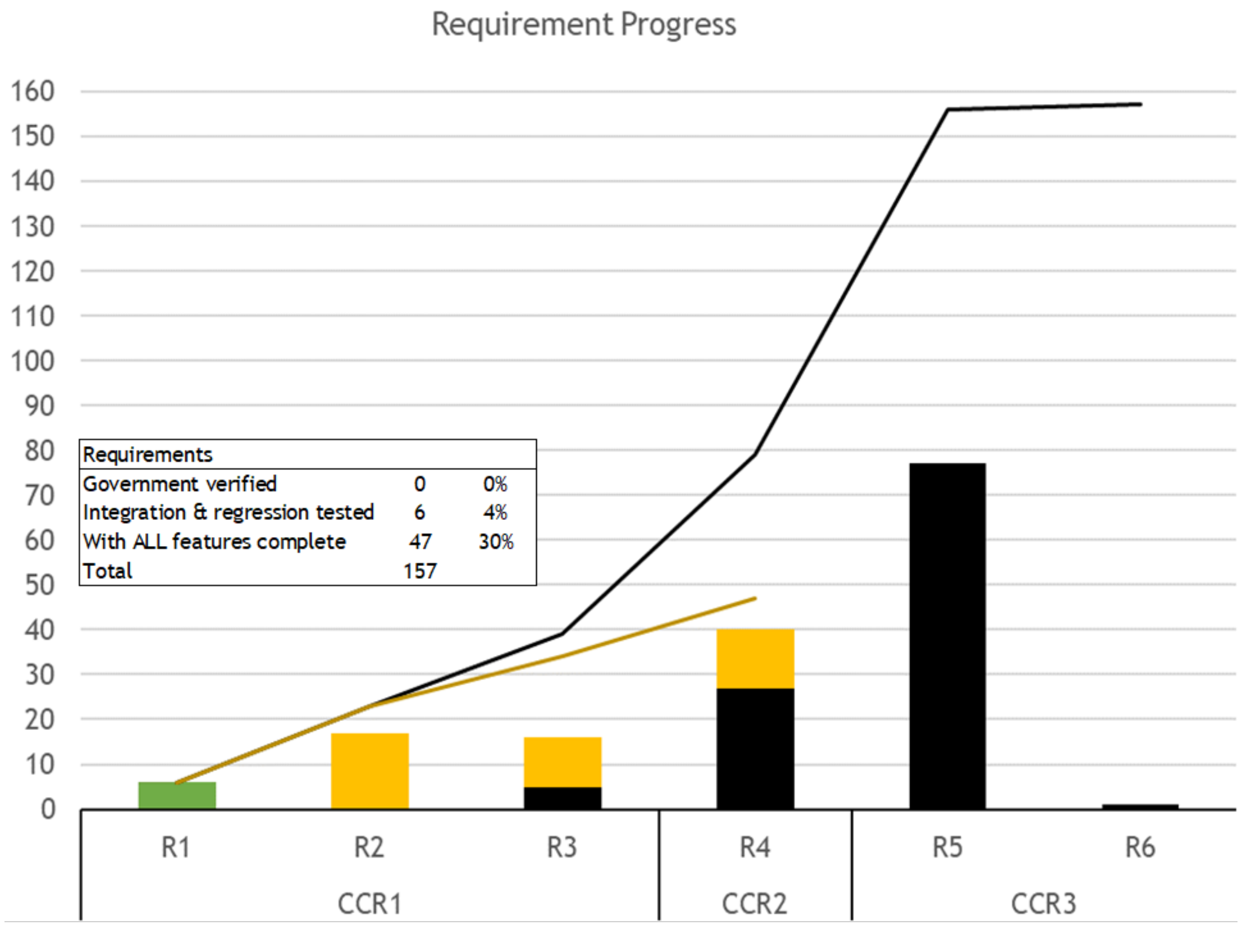
Figure 3 – Status Board of Features with IMS Flags

Figure 3 shows the same PI features as before, with monthly IMS data incorporated and the results of an automated "IMS & PI Match" analysis displayed – IMS Start and Finish dates are compared with PI start and finish dates to flag if the IMS activity occurs within or outside the planned PI timebox. In this figure many features began before the current PI (unfinished from a prior PI); however, DEVC2-1551 (third from last) is scheduled to finish after the current PI.

Feature Team	Key		Status	Δ Date	Points	%	MVP/ MM	Sprint History	PI	Issue Type	Updated
SIM	DEVC2-14735	S	In Review	05-Feb-24	3	100	(3-b)	PI11-SP4(2024-01-24)	11	Story	05-Feb-24 08:09
SIM	DEVC2-14728	S	In Review	04-Feb-24	1	100	(4-c)	PI11-SP4(2024-01-23)	11	Story	04-Feb-24 22:10
TRN	DEVC2-13645	S	Done	02-Feb-24	3	100	Non-M	PI11-SP4(2023-10-09)	11	Story	02-Feb-24 16:26
SIM	DEVC2-14724	S	Done	02-Feb-24	1	100	(4-c)	PI11-SP4(2024-01-23)	11	Story	02-Feb-24 14:31
TMB	DEVC2-10961	S	In Review	02-Feb-24	3	100	Non-M	5),PI12-SP2(2023-10-09),PI11-SP4(2024-01-09)	11	Story	02-Feb-24 12:01
TMB	DEVC2-10756	S	In Progress	02-Feb-24	4		Non-M	5),PI12-SP3(2023-10-09),PI11-SP4(2024-01-21)	11	Story	02-Feb-24 12:01
SIM	DEVC2-14729	S	In Progress	29-Jan-24	3		(4-c)	PI11-SP4(2024-01-23)	11	Story	02-Feb-24 11:17
SIM	DEVC2-14002	S	In Review	02-Feb-24	1	100	Non-M	PI11-SP4(2023-11-08)	11	Story	02-Feb-24 10:55
TMA	DEVC2-13472	S	In Progress	02-Feb-24	1		(4-a)	PI11-SP4(2023-10-10)	11	Story	02-Feb-24 10:46
TMA	DEVC2-13474	S	In Progress	02-Feb-24	1		(4-a)	PI11-SP4(2023-10-10)	11	Story	02-Feb-24 10:46
TMA	DEVC2-13473	S	In Progress	02-Feb-24	1		(4-a)	PI11-SP4(2023-10-10)	11	Story	02-Feb-24 10:46
TMA	DEVC2-13470	S	In Progress	02-Feb-24	1		(4-a)	PI11-SP4(2023-10-10)	11	Story	02-Feb-24 10:46
TMA	DEVC2-13468	S	In Progress	02-Feb-24	1		(4-a)	PI11-SP4(2023-10-10)	11	Story	02-Feb-24 10:46
TMB	DEVC2-11377	S	In Progress	02-Feb-24	1		Non-M	PI10-SP3(2023-07-07),PI11-SP4(2023-10-09)	11	Story	02-Feb-24 10:40

Figure 4 – Status Board of Stories in Current Sprint

Figure 4 shows a status board of collected Jira data (orange columns) filtered for stories in the current sprint and sorted by update date (green columns). Similar to tracking a feature's PI history, a story's assigned Sprint is displayed with change dates (blue columns). This view shows two stories that were pulled early from PI 12 into PI 11; the last story was previously assigned to PI 10 – Sprint 3, and later reassigned to PI 11 – Sprint 4.



	R1	R2	R3	R4	R5	R6
Government verified	0	0	0	0	0	0
Rqmts int & reg tested	6	0	0	0	0	0
Rqmts w/features complete	0	17	11	13	0	0
Rqmts remaining	0	0	5	27	77	1

Figure 5 – Requirements Completed Toward Critical Release Chart. (Orosz et al., 2023b)

The chart in Figure 5 shows the requirement completion plan and to-date progress. Each release (R1, R2, … R6) represent one or more MVPs/MMPs in the project. Critical Capability Releases (CCR1, CCR2 and CCR3) consist of multiple releases. For each planned release (e.g., R1, R2, etc.) a histogram shows how many requirements are assigned, requirements that have been verified by the government (blue – zero so far on Project C), requirements whose features have been completed and individually verified/tested (green), requirements with completed features but not yet individually verified by the government (gold) and requirements that have features that have yet to be completed (black).

Next Steps

As of the writing of this paper, Project C is 30 months into a 51-month effort. Software development has been underway for roughly 18 months, and the initial MMP deliverables are several months away. Going forward, the USC/ISI project team is focused on collecting and analyzing performance data such as bugs/DRs, cost data, DevSecOps pipeline performance and project velocities and other performance metrics. In addition, based on observations and lessons learned, the project team will continue to offer subject matter expertise to the government on monitoring and managing the agile/DevSecOps project.

Of particular interest to the team is how to better transition from an environment that relies on well-defined waterfall performance metrics to an evolving agile software development environment. The agile environment is focused on delivering value rather than the traditional waterfall metrics, such as software lines of code. For example, a key area of research is improving the synchronization of PI planning with the Integrated Master Schedule (IMS), which drives many EVM metrics. Finally, the development of more effective workforce training processes and materials will also be undertaken. Results will be published in a future paper.

References

AcqNotes. (2024a, February 4). *Contract Data Requirements List (CDRL) - ACQNotes*. https://acqnotes.com/acqnote/careerfields/contract-data-requirements-list-cdrl#:~:text=Contracts%20%26%20Legal-,Contract%20Data%20Requirements%20List%20(CDRL),data%20requirements%20and%20delivery%20information

AcqNotes. (2024b, February 4). *Integrated Master Schedule (IMS)*. https://acqnotes.com/acqnote/careerfields/integrated-master-schedule

AcqNotes. (2024c, February 6). *Integrated Product Team (IPT)*. https://acqnotes.com/acqnote/careerfields/integrated-product-teams

Atlassian. (n.d.). *Introduction to Jira family*. https://www.atlassian.com/software/jira/guides/getting-started/overview

CameoMagic Solutions. (n.d.). *Cameomagic Solutions*. https://www.cameomagic.com/

DAU. (n.d.). *Earned value management (EVM)*. https://www.dau.edu/acquipedia-article/earned-value-management-evm

Dove, R., Lunney, K., Orosz, M., & Yokell, M. (2023). Agile systems engineering – Eight core aspects. *INCOSE International Symposium*, *33*(1), 823–837. https://doi.org/10.1002/iis2.13055

IBM. (n.d.). *Overview of Rational DOORS Next Generation*. https://www.ibm.com/docs/en/elm/6.0.6?topic=capabilities-rational-doors-next-generation

Nord, R., Ozkaya, I., & Woody, C. (2021, July). *Examples of technical debt's cybersecurity impact*. https://apps.dtic.mil/sti/trecms/pdf/AD1144728.pdf

Orosz, M., Spear, G., Duffy, B., & Charlton, C. (2022). Introducing agile/DevSecOps into the space acquisition environment. *Naval Postgraduate School 19th Annual Acquisition Research Symposium Proceedings*, *1*, 405–416. https://dair.nps.edu/handle/123456789/4541

Orosz, M., Duffy, B., Charlton, C., Saunders, H., & Thomas, E. (2023). Unique challenges in mission engineering and technology integration. In *Systems Engineering for the Digital Age* (1st ed., pp. 665–681). https://doi.org/10.1002/9781394203314.ch31

Orosz, M., Col, L., Hempen, J., Turner, J., & Duffy, B. (2023). Tracking performance in a DoD Agile/DevSecOps environment. *NDIA 26th Annual Systems and Mission Engineering Conference.* https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2023/systems/Thurs_1556340_Orosz.pdf

Proctor, S., & Daniels, C. B. (2020). Implementing agile project management in the US Department of Defense. *Space Infrastructures: From Risk to Resilience Governance*, *57*, 337.

Scaled Agile Framework. (2024, February 8). *Safe 6.0 framework*. https://www.scaledagileframework.com/

SD-Team. (n.d.). *CWE-561 - Security Database*. https://www.security-database.com/cwe.php?name=CWE-561

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net