SYM-AM-26-078

Excerpt from the Proceedings

of the

Twenty-Third Annual Acquisition Research Symposium and Innovation Summit

Wednesday, May 6, 2026 Sessions

Volume I

"Accelerating Warfighting Capabilities"

Accelerating DoW Space-Based Acquisition Through the Implementation of Agile, DevSecOps and Digital Engineering Processes

Published: April 30, 2026

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

To request defense acquisition research, please contact:

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
E: arp@nps.edu
www.acquisitionresearch.net

Accelerating DoW Space-Based Acquisition Through the Implementation of Agile, DevSecOps and Digital Engineering Processes

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 John "Mack" Turner—joined the U.S. Space Force in 2022 after 15 years in the U.S. Army. He has held all Artillery positions at company level to include 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. [john.turner.66@spaceforce.mil]

Lt Col Trevor Johnson—is the Deputy Commander for the 31st Capability Development Squadron, Los Angeles AFB, California. He leads 35 military and civilian members in the development of the Next Generational Operational Control System. Lt Col Johnson earned his bachelor's degree in management from the U.S. Air Force Academy and was commissioned in 2006. He distinguished himself in a variety of leadership positions within the aircraft maintenance and acquisition career fields while at Grand Forks and Los Angeles AFB. Prior to assuming his current position, Lt Col Johnson led execution of the Future Operationally Resilient Ground Evolution (FORGE) program.

Brian Duffy—is a Senior Systems Engineer with the University of Southern California 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, Mr. Duffy retired from the U.S. Air Force with multiple assignments related to National Security Space acquisition programs and C2 systems. Mr. Duffy holds a Masters of Aeronautical Science from Embry-Riddle Aeronautical University and a Bachelors of Aeronautical and Astronautical Engineering from the University of Washington.

Craig Charlton—is a Senior Systems Engineer at USC's Information Sciences Institute (USC-ISI) and has provided acquisition support at Space Systems Command (SSC) at the Los Angeles Air Force Base during the past 20 years on several leading-edge satellite systems. Prior to his position at SSC, Mr. 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. Mr. Charlton received a BA in mathematics from California State University, Long Beach.

Abstract

Over the past nine years, using funding from the U.S. Air Force and the U.S. Space Force, System Engineering Research Center (SERC) researchers at the University of Southern California's Information Sciences Institute (USC/ISI) along with the active support and engagement of the USAF and USSF have undertaken a series of case studies focused on accelerating the Department of War's space systems acquisition process through the application of agile, DevSecOps, digital engineering and other advanced technologies such as AI.

The major focus of this research is discovering challenges and exploring solutions to managing process flow, resources and deliverables throughout the agile/DevSecOps system development process. Such challenges include managing dependencies on external and internal systems; supporting real-time continuous multi-system integration and testing; managing staff loading and

specialties over the course of the program; adapting to new and changing capabilities; arranging operator engagement throughout the development process; and the availability, collection and analysis of performance metrics for improved situational awareness and real-time decision-making.

Several observations and lessons learned will be covered in this paper. In addition, this paper discusses lessons learned to date in employing AI as a systems engineering aid throughout the development and deployment cycle.

Introduction

Secretary of War Pete Hegseth mandated that the U.S. Department of War (DoW; 2025) needs to field systems to the warfighter sooner, on budget and that meet warfighter needs. The purpose of the research undertaken by USC and the SERC UARC (Systems Engineering Research Center, n.d.) is to address this mandate by determining how agile (*Manifesto for Agile Software Development*, n.d.), DevSecOps (Microsoft Security, n.d.) and other modern acquisition methods and approaches—many already being successfully used in the private sector—can be successfully implemented in current and future DoW acquisition programs. Specifically, what tools and approaches can be incorporated into the acquisition process that allows programs to become agile and nimble to address dynamic conditions from both an operations perspective (i.e., what does the warfighter need now) and also from an acquisition perspective (e.g., how has technology changed, have new personnel joined the team, etc.)? In other words, by implementing processes which reduce program risk, the DoW will shift the discovery and resolution of program risk earlier in the program, leading to a more rapid delivery of warfighter capabilities.

The research discussed in this paper is currently focused on improving the acquisition of space-based software command and control (C2) systems, but much of what has been observed and learned can also be applied to hardware-only and hybrid hardware/software programs in other domains within the DoW, for example, agile processes applied to hardware-based programs that employ additive manufacturing (i.e., 3D-printing).

Over an approximate nine-year period, the USC-led project team has observed that agile, DevSecOps and related modern systems acquisition methods are able to improve DoW acquisition, though it is apparent that many challenges still exist and that adopting industry best practices requires considerable tailoring. For example, “code fast, fail fast and learn” is difficult to achieve in large mission-critical enterprise systems that cannot fail.

This paper extends lessons learned and observed from previously published work (Orosz, Hempen, et al., 2025; Orosz, Turner, et al., 2025).

Process Used

As noted in Orosz, Hempen, et al., (2025), the USC-led team is embedded into the space-based C2 acquisition environment. Project team members serve two roles: first, as systems engineering subject matter experts (SMEs) on the government team (e.g., members of one or more Integrated Project Teams [IPTs]), and secondly, as researchers. The advantage of embedding into the day-to-day acquisition process is that the team is able to observe which agile and other modern systems development processes and approaches either work or fail in improving and accelerating systems development and delivery. This embedding has helped establish strong working relationships with both the government and prime contractor teams (responsible for the actual development of the system) which has allowed experimentation with different approaches (e.g., making recommendations to both government and prime contractor) in the quest to improve and accelerate the acquisition process. Figure 1 summarizes the various projects the team has supported as part of this research effort. Currently, the project team is supporting Project D, and many of the research results presented in this paper are from this project. The team is not supporting Project E, but the project is listed here as it directly impacts Project D, as discussed later in the paper.

Project	Development Method	Duration	Lines of Code	Notes
Project A	Waterfall	39 Months	178K	Serves as baseline for research
Project B	50-50 Hybrid Agile/Waterfall	25 Months	113K	Agile effort greatly outperformed waterfall effort
Project C	Tech Study	15 Months	None	Prep for Project D. Agile training
Project D	70/30 Hybrid Agile/Waterfall	75 Months+	150K+	Currently underway
Project E	Waterfall	Many months	N/A	Impacts Project D

Figure 1. Table Summarizing the Space-Based Command and Control Projects That the USC-Led Research Team Was or Is Currently Supporting. The research team is not involved with Project E; however, Project E is linked to Project D, as discussed in the paper.

Observations

Project D relies on a modified implementation of the SAFe® (Lindsey, 2026) agile approach to development. As noted in Orosz, Turner, et al. (2025), requirements are decomposed into one or more work packages called features which are to be accomplished within a 12-week period of development work within a program increment (PI). Features are then further decomposed into one or more work packages called stories which will accomplish the work required to complete the overall feature (Figure 2). Each feature is assigned a priority on the project backlog which coincides with the PI the feature is to be worked. Each program increment is 13 weeks in length and consists of four 3-week sprints followed by a 1-week non-development effort used for demonstrations, training, PI planning, retrospection and other activities (Figure 3).

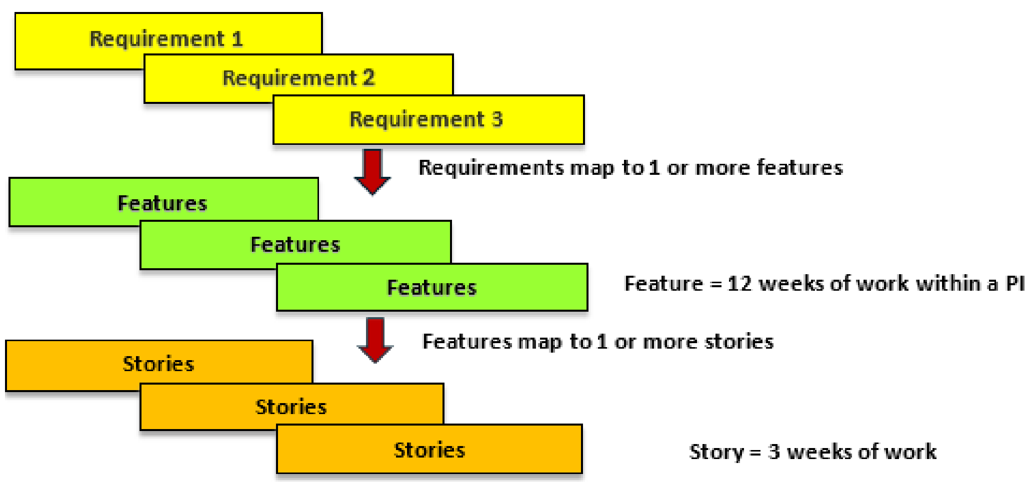

Figure 2. Project Consists of One or More System Requirements. Each requirement is decomposed into one or more features. A feature represents development work that can be completed within 12 weeks. Each feature is decomposed into one or more stories. Each story represents work that can be completed within weeks. Figure adapted from Orosz, Hempen, et al. (2025).

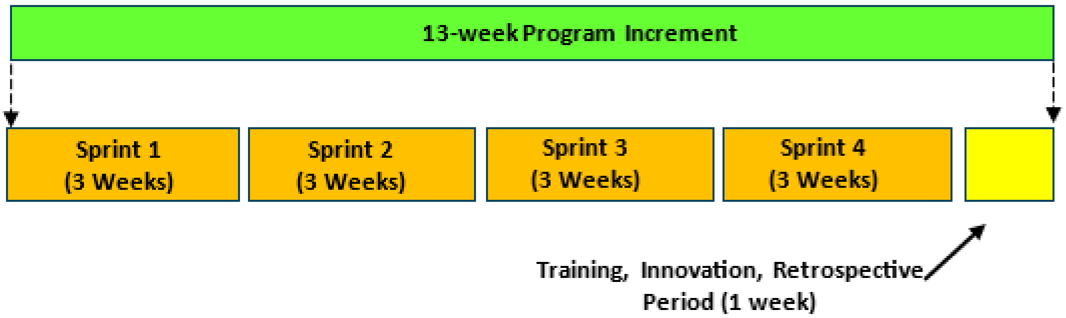

Figure 3. Features Are Assigned to a 13-Week PI. Within the PI are four 3-week sprints in which stories decomposed from a feature are worked. The final 1 week for each PI is used for non-development work such as training, innovation/research, retrospectives and other activities. Figure adapted from Orosz, Hempen, et al. (2025).

Figure 4 shows the number of features planned per PI (black trace), the number of features that were completed (green trace) and the number of uncompleted features that slipped from that PI to future PIs (blue bars). Slipped features indicate situations where the development team was not able to complete their assigned feature within the assigned PI.

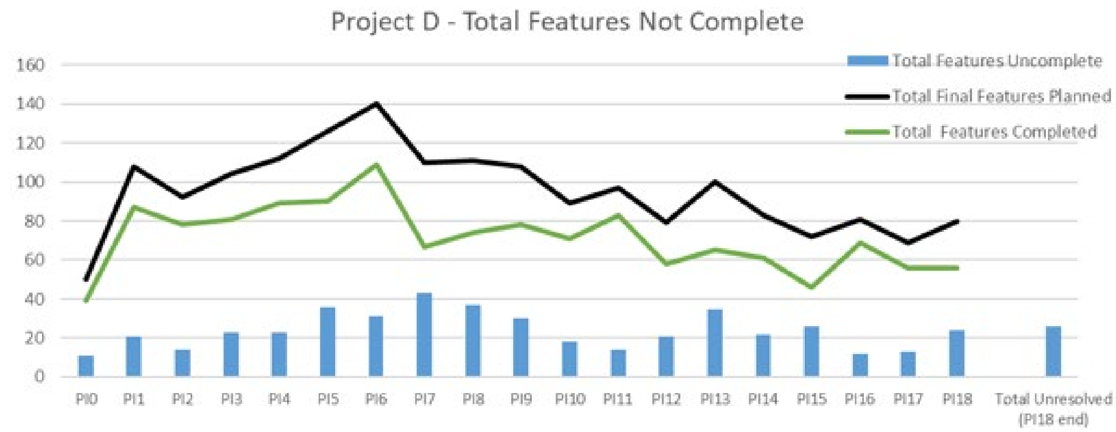

Figure 4. Features Assigned, Completed and Uncompleted per 13-Week PI. Blue bars represent features that did not complete within the 13-week PI and "spilled" over into future PIs.

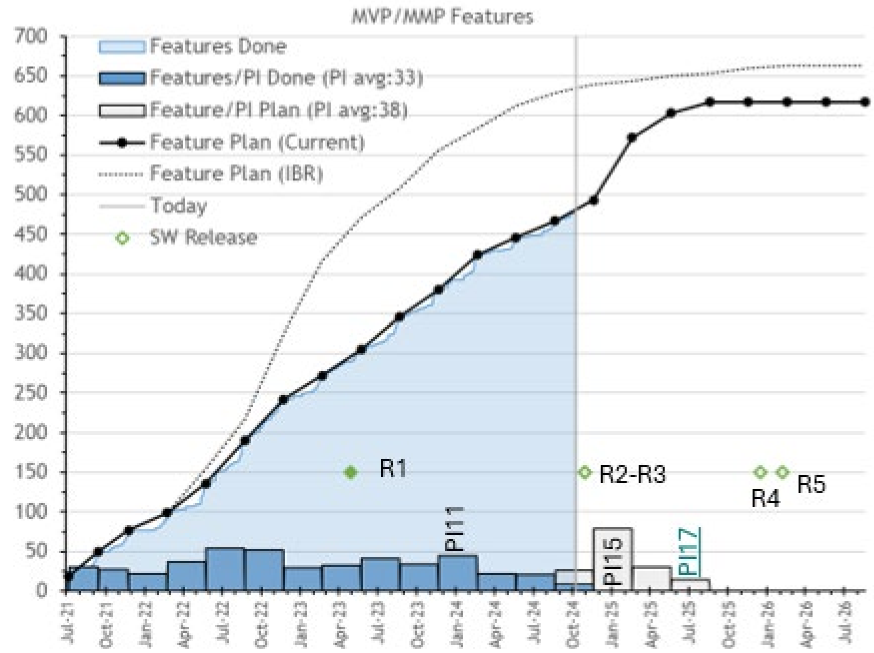

Figure 5. Plot of Features Completed per PI, as of October 2024. Note large number of features (a "bow wave" assigned to PI-15 [January 2025]). These features include uncompleted features that slipped into future PIs. All features are to be completed by the end of PI-17 (July 2025). R1-R5 are MVP/MMP releases milestones.

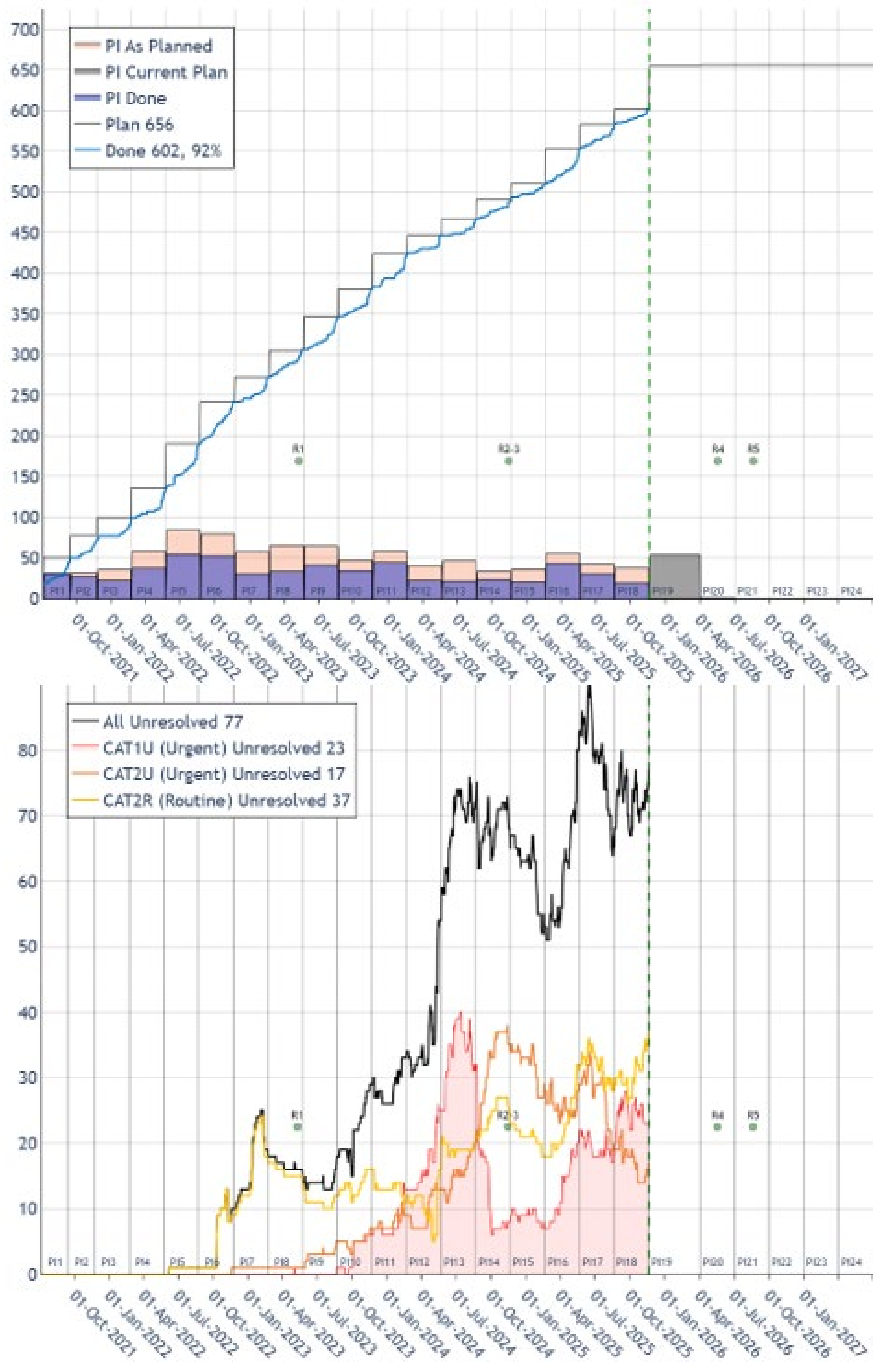

Figure 6. Upper Graph Represents Feature Work-Off Performance Approximately One Year Later Than That Covered in Figure 5. The development schedule has been extended to another three PIs (approximately 9 months) to PI-20. Milestone releases R4-R5 have shifted "right." Bottom Graph Shows Bugs That Have Been Detected via Integration and Testing Efforts. Large upticks in discovered bugs come at the expense of feature work-off as key workforce members are pulled from development to address critical bugs (CAT1Us), shown in red.

Figure 5 is a snapshot of program progress as of October 2024. The figure shows the number of features completed (dark blue bars) and the number of features planned for the current and future PIs (gray bars). As shown in the graph, the number of features assigned to be worked in the upcoming PI-15 (approximately 75), centered in January 2025, is considerably higher than feature velocity tracked from PI-0 through PI-14 (July 2021 to October 2024). This large number of features represents an evolving “bow wave” of work that has accumulated due to features slipping from PI to PI throughout the program.

At the top of Figure 6 is a similar plot as in Figure 5 but represents project performance approximately a year later, as of November 2025. The forecasted “bow wave” of features, centered around January 2025, has been “smoothed” out and shifted by extending the development schedule another three PIs (approximately 9 months) from PI-17 to PI-20. At the bottom of Figure 6 is the trace of “bugs” (known as CAT1s and CAT2s for “Category 1” [critical, with no work around] and “Category 2” [important, but not critical]) per PI. Notice that feature velocity is directly linked to the number of “bugs” detected and worked. The more bugs detected and assigned to be worked (i.e., CAT1U bugs), the lower the feature velocity.

The large number of bugs detected late in the development effort (starting roughly in PI-11) is attributed to the delayed start of integration and testing (to be discussed later). The CAT1U (critical with no work around) bugs have to be addressed, requiring key development team members to be pulled from feature development to work those bugs which is contributing to the low feature velocity that has been observed.

Discussion and Recommendations

What Is Working

Although the development schedule has slipped (at the time of this paper, approximately 9 months), there have been a number of “wins” observed (to date) in Project D. First, a major release was developed and underwent early testing by the end-user community via a near operational environment. The availability of this release (an MMP) was delayed by about 6 months due to challenges to be discussed below. Although the operating environment was simulated, this early testing and simulated use of the capability provided early feedback to the program which provided valuable input in the development of the next major MVP and MMPs.

A second win was the prime contractor’s management of the project backlog. When blocked features or stories were encountered, the development teams quickly pivoted and “pulled forward” and worked the next highest priority task from the project backlog. As blocked features and stories became unblocked, work once again resumed on those tasks based on their current priorities on the project backlog (which might have changed while blocked). Projects with a definitive end state (e.g., DD250, etc.), at some point, may be challenged to “pull forward” future high priority tasks as fewer tasks are available as the project backlog shrinks. Nonetheless, at the time of this writing, the development effort for Project D only slipped 9 months (i.e., 3 program increments) from the original 52-month (17 program increments) development schedule.

The third win was the unfettered access the government had to project technical data. The government had direct access to the project backlog, requirements and test databases throughout the program. In addition, the government teams participated in all scrums, ceremonies, demonstrations, and planning activities throughout the project. This unfettered access helped the government to have a real-time/near real-time picture of the current state of the project. Much of the data extracted from this access was formatted and analyzed by the government with tools developed using Python, MS Visual Basic for Applications, MS Excel and other applications. These tools allowed the government team to quickly analyze and provide real-time performance metrics to senior government leadership. In many cases, this information was shared with the prime contractor to help identify challenges in the acquisition process and establish priorities to address. Figures 4, 5, and 6 from above are examples of the metrics generated to provide real-time status of the project.

Perhaps the biggest "what worked" to date in Project D was the ease in which the prime contractor's scrum teams worked with the government's IPTs. Through frequent engagements via scrum meetings, ceremonies and other activities, team members from both sides developed a relationship that promoted good communications and established a common teaming environment while still respecting the business needs of the respective teams (e.g., the government is a representative of the warfighter/end-user, the contractor is the hired vendor, etc.).

Challenges

Late Integration and Testing: As noted in Orosz, Hempen, et al. (2025) agile-based projects need access to a near operational environment as soon as possible when implementing agile, DevSecOps and Continuous Integration/Continuous Deployment (CI/CD) approaches. The integration and testing in Project D were delayed due to the lack of availability of a near operational test environment. Of the testing that was undertaken early in the program, much of that testing consisted of isolated functional testing at the subsystem or Configuration Item (CI) level. Horizontal testing where one or more CIs are tested as an integrated unit (i.e., end-to-end testing) did not ramp up until around PI-11 (Figures 5 and 6 above). The result was a bow wave of "bug" (i.e., the CAT1Us and CAT2Us) discovery which led to a shifting bow wave of features to be completed (see Figures 5 and 6). In this case, the development team focused on the high-priority bugs rather than on the features and stories residing on the project backlog.

Workforce Management in a Dynamic Environment: A major challenge for the prime contractor is managing the workforce in an agile program. As program priorities change, often due to changing requirements and end-user needs, there is pressure (driven by these changing priorities) on the prime contractor to quickly pivot and focus on delivering newly identified high-priority capabilities. Often, however, the skill sets needed to meet these newly identified high priority capabilities may not be present in the current prime contractor's workforce. The prime contractor may often be in the difficult position of having to work with the workforce which they currently have on hand. Their skill sets may be more aligned with previous high-priority and scheduled work which is now further down in the project backlog (i.e., scheduled to be worked in the future) due to the recent focus on the newer high-priority work.

In such circumstances, development progress (e.g., velocity, etc.) can dramatically slow as the existing workforce has to "ramp-up" and become more familiar with the newly classified high-priority work to be completed. Further, under these conditions, there is a tendency of the prime to pull forward (i.e., raise higher in the project backlog) lower priority work to be worked with the current available workforce (with the more aligned skillsets) at the expense of working on higher priority work that needs to be delivered early. In other words, the prime contractor does not want a highly skilled workforce working tasks that actually require lesser skills. This is another example of "assuming the workforce has equal skill levels and experience and can be used on any effort" that has previously been identified (Orosz, Turner, et al., 2025).

Corporate Knowledge is Lost: A key factor contributing to schedule slippage is the constant churn of the government and prime contractor workforce. For many DoW programs, including agile programs, the program can span multiple years, even decades. In fact, if DD250s (AcqNotes, 2024) and other key milestones are viewed as MMPs, programs can be viewed as continuous evolving enterprises. Once a DD250 has been reached, the next phase (often an upgrade or sustainment) starts. As the program progresses, prime contractors change, government agencies restructure, developers come and go, and military and government civilians transition to other assignments. The challenge here is that knowledge of the system is often lost due to this churn in the workforce.

In this context, knowledge encapsulates the system being developed and sustained (i.e., system design, architecture, etc.), the acquisition process used to develop the system (i.e., waterfall, agile, hybrid, integration & testing approach, etc.) and the operational environment in which the system will operate (i.e., domain knowledge). Decisions behind design considerations, priorities, and other programmatic decisions are often lost when team members transition to other programs or opportunities. Often, priorities, stories, features, and tasks on program backlogs are set, and the rationale behind these decisions are lost as the program progresses. Capturing this corporate knowledge and making it available to new members of the workforce is a major challenge. Employing modern engineering processes can help (e.g., MBSE tools, etc.); however, often these tools and processes are not seamlessly interconnected. For example, linking system requirements to the model design to the developed software and system test results is still very difficult. In addition, capturing the decisions behind the design and approaches used is rarely captured, at least digitally, which can be problematic years into the future. Note that this "linking" challenge not only applies to the program office, but it also applies at the enterprise level.

Implementing DevSecOps: As noted in Orosz, Hempen, et al. (2025), in space-based enterprises, the implementation of DevSecOps and related Continuous Integration/Continuous Deployment (CI/CD) is challenging. This is also true in Project D. Often, the command and control (C2) segment of a space-based program is initiated and often completed prior to the availability of other systems and subsystems—including the space vehicle (SV) constellation. The lack of these external systems limits how much operational use (or testing) can be undertaken in the "Ops" portion of DevSecOps. This challenge also appears when external systems or their simulators are constantly changing or when commands or tasking (e.g., for commanding a space vehicle) frequently change. Enhanced enterprise engagement can help mitigate these challenges.

As noted elsewhere in the paper, project D suffered from the delayed availability of a near operational test environment that can be used for testing prior to deployment. In addition, the lack of available operators to test the incremental releases was also challenging to the project as these individuals are part of a small, highly trained cadre of operators that are in high demand. Scheduling time for these operators to test releases of the system was problematic and often involved "luck."

Loss of Architecture Modularity: Despite an initial focus and the best intentions of the prime contractor throughout the program to develop and maintain a modular architecture, there were many examples of tight coupling between multiple modules and Configuration Items (CIs) within Project D. This does not imply that the prime contractor never undertook refactoring during the development of the system. Refactoring was, in many situations, necessary as new capability was added to the system. However, such refactoring was chosen as the path of least resistance when adding new capability to the system. As the program progressed, however, multiple engineering change orders were introduced (via Requests for Change [RFCs]), high priority bugs (or discrepancy reports [DRs]) were worked, new team members transitioned onto the project, senior systems engineering leadership changed, and the constant pressure of maintaining high development velocity and production of releases (MVPs, MMPs, etc.) resulted in the tight coupling between various modules (within and between CIs). Refactoring to maintain modular architecture and code baseline was important, but a lower priority.

AI and the Acquisition Process: The government and USC/ISI team undertook an initial exploration on using available Artificial Intelligence (AI) tools to help with program management and metrics tracking. The motivation behind this exploration was to reduce the workforce effort required to collect, aggregate and report on program performance metrics. As with many other programs, without an AI or some other automated or semi-automated tool, collecting, aggregating, analyzing and then reporting on various aspects of a program is predominantly a manual process which is time-consuming and often error prone. In Project D, the government explored using NIPRGPT (U.S. Department of the Air Force, 2024)—a Large Language Model (LLM)—to collect, aggregate, analyze and generate program progress reports. The reports produced by the LLM required some final editing by a human subject matter expert (SME), but the savings in workforce hours (100s of hours) was deemed a success.

Recommendations and Observations

As previously noted, this paper summarizes recent research and findings to an ongoing series of case studies focused on accelerating an agile-based software program. Figure 7 summarizes recommendations and observations previously reported (Orosz, Turner, et al., 2025). The recommendations that follow are an extension to these earlier observations.

Recommendation/ Observations	Comments
Upfront engineering	Perform some upfront engineering to help populate the project backlog, map features with compliance requirements, identify test cases, and identify dependencies as early as possible in the program
Not all staff skillsets are equal	The amount and diversity of talent needs to be constant, so they can be moved/re-arranged to react to the changing needs (be agile) of the program
Near operational test environments	Establish (early in the program) a near operational environment and high-fidelity simulators (for horizontal I&T)
Sprint margins	Plan margin into the sprints to handle unexpected events such as new technology insertion and/or unexpectedly complex stories (Agile 101)
Licensing and other IP issues	Get licensing, IP, accreditation, certification and other programmatic issues resolved early
Training	Need for on-boarding and continuous training to ensure team members (both the contractor and acquisition team) are on the same page
Costing challenges	For projects that are costed using “T-shirt sizes,” costing needs to be segmented into development, integration and testing, and bug (problem) work-off. Also, need to have a sufficient number of T-shirt sizes to cover the smallest to largest work packages
Custom tools	Be prepared to customize performance tracking tools – Applies to all teams – Issues: • Software incompatibilities • Foreign ownership of tools • Access challenges (e.g., VPN, security concerns, etc.) • EVM lags actuals – sometimes by months.

Figure 7. Summary of Lessons Learned and Observations From Previous Research (Orosz, Turner, et al., 2025)

Capturing Corporate Knowledge: Recognizing that workforce churn will probably not abate even with increased incentivization as career growth and other opportunities will continue to drive employees to transition off a project, it is recommended that there should be a focus on 1) improving the capture of corporate knowledge that can be shared and 2) providing relevant on-boarding and continuous workforce training. The following are some recommendations for achieving this goal:

1. As features and stories are added, worked, and completed on the project backlog, design and coding decisions (from scrums, demonstrations, ceremonies, retrospectives and other meetings) should be captured and stored with the tracking ticket (e.g., a Jira [Atlassian, n.d.] Ticket) on the backlog. Such information could include a few bullets of text or, in environments that can support voice-to-text technologies (e.g., non-classified environments), a textual record of these meetings. For example, in programs that can support VTC meetings using tools such as MS Teams ("Microsoft Teams," 2026), the tool itself can capture voice traffic and automatically translate a textual record that can be attached to a tracking ticket. The advantage of having access to this information is that the thought process behind various decisions in adding, implementing, testing and then closing a feature or story is available to the future workforce.
2. Maintain and frequently update on-boarding training materials that describe the development approach used (e.g., SAFe, scrum, Kanban, etc.), design decisions behind the system, and other programmatic information that the workforce member needs to know. In addition, this on-boarding and continuous-training material should include domain-specific information to help the development and government team members to better understand the targeted end-user environment. For example, in a space-based command and control acquisition program, training (on-boarding and frequent refreshers) that highlights space-to-ground communications, cyber vulnerabilities, space vehicle constellation dynamics, orbital dynamics and other related information can help focus developers and the government on end-user needs and priorities.
3. Increase domain knowledge. A possible pathway is to rely on a qualification system similar to military aviation and other mission-critical job functions. For example, with aviation, pilots must not only achieve certain qualifications, but they also must maintain them via frequent "refresher" flights or training. The same could be used in the acquisition environment with domain-specific qualifications (e.g., space-based launch operations) and acquisition-specific qualifications (e.g., current agile best practices, etc.).

Frequent Engagement With the Enterprise: Team members (developers, integrators, testers, chief engineer, senior leadership and others) from both the prime contractor and the government need to be in frequent engagement with the whole enterprise (space segment, control segment, user segment, etc.). Whether this is in the form of working groups, tech exchange meetings (TEMs) or as part of other frequently scheduled meetings (e.g., project management reviews [PMRs]), they need to occur frequently to allow all stakeholders an opportunity to become aware of possible changes coming in the future, the status of the development/testing environments, and the overall status of the enterprise. In short, changes to the system design, requirements or concept of operation should not be a surprise to the development and testing environment.

Successful engagement with the enterprise is a particularly challenging problem with prime contractors due to a lack of mission expertise and historical links to the government. It is a surprisingly difficult task to engage with the enterprise. It is much easier to be successful at this task if the prime contractor has hired individuals who have prior relationships and are trusted as mission experts. Without this relationship, contractors are seen as "outsiders" who "don't understand the mission" and enterprise engagements will not be fruitful. Additionally, it is hard to do the upfront engineering of the mission threads without this expertise, and it is much harder to do integration without knowledge of where the "gotcha's" will come and without relationships with the other entities that need to be integrated with. The best software experts in the world will fail if they do not understand the mission.

Trained and Experienced Workforce. A trained and experienced project workforce is critical to accelerating an agile program. As noted in Orosz, Turner, et al. (2025), often both the government and the prime contractor view the workforce as a "plug and play" commodity where all team members have roughly the same experience and skill levels. This is simply not the case nor is the issue simply focused on the level of software development experience. Knowledge of the domain (space operations, ConOps, etc.), knowledge of development and testing environments, and awareness of best practices are also essential in the workforce.

As noted, the workforce is not a commodity. It is very difficult to hire a replacement that can be "plugged" into the development environment. Even experienced personnel will require time to "learn" the development environment, the ConOps of the targeted application, and workflow of the integration, testing and deployment. In project D, it was common for the prime contractor and the government to take up to 12 months to on-board new (and often) experienced personnel. It is recommended that both on-boarding and continuous training be an integrated part of the acquisition workflow. Although an expense, the investment can be spread over the life cycle of the program which often spans multiple years and, in some cases, decades (e.g., space-based systems).

Integration and Testing: As noted in Orosz, Turner, et al. (2025) and summarized in Figure 7, the sooner a near-operational environment becomes available, the sooner system-wide integration and testing can be undertaken. Equally important, however, is the need to define up front use cases or "mission threads" that define how the system works (or is envisioned to work) and is used. These cases included expected inputs, output, error and exception handling are directly linked to system requirements. These use cases inform the development of test scripts that are used during system-wide integration and testing. Additionally, including use case testing in a government statement of work could enable integrated "graduation" testing for releases, especially when a functional qualification test per release is not feasible.

Although the above recommendation is not new, it is often overlooked in agile-focused programs where the "code fast, fail fast and learn" approach is expected to drive the system design and test environment. This simply is not the case in complex OSW programs such as command and control of a space vehicle constellation where there are multiple venders and development approaches and timelines that span the enterprise.

For example, in many of these OSW agile-based software development programs, the development prime contractor will assume that the successful testing and verification of all features/stories (i.e., work products) that are decomposed from a single system requirement indicates that the system requirement has been met. Unfortunately, the sum of the parts does not necessarily equal the complete system. Only when all components are tested as an enterprise system, where all modules and CIs are connected and operational (even if simulated) and subjected to successful end-to-end testing, can it be declared that the system requirement(s) have been satisfied.

Increased Government Involvement: One of the successful aspects of all the projects that the research team has been involved with is the value of embedding knowledgeable government acquisition team members within the development environment. Government team members participate in scrums, ceremonies, planning, testing, and demonstration events. Such activities help build trust between the prime contractor and the government. For example, the government is aware and can appreciate challenges faced by the prime contractor (e.g., workforce changes, budget and scheduling challenges, code complexity issues, etc.). Similarly, the prime has an opportunity to appreciate the challenges faced by the government (especially the warfighter). Issues such as the importance of meeting system requirements, what those system requirements mean to the warfighter and the urgency of getting capability into the hands of the warfighter help with managing the project backlog, workforce and other resources.

In addition, this embedding of the government into the development workforce often allows the government real-time or near real-time access to prime contractor data that is used for tracking/capturing project performance metrics.

That said, government involvement requires a knowledgeable government workforce. As previously noted, a major challenge in agile-based project efforts is the need for an experienced workforce. Throwing a freshly minted lieutenant with no prior experience or knowledge domain into a scrum will almost always hinder system development.

Maintaining MOSA: As noted previously in the paper, the prime contractors behind each of the projects—including Project D—did focus on developing and maintaining a modular open system architecture (MOSA). The challenge is in maintaining modularity as the program progresses. Often, as pressure to meet upcoming milestones intensifies, the focus shifts to getting the capability completed, regardless of how it is done. This approach often results in unintentional interdependencies within and between modules. Although often discussed in the systems and software engineering community (Fowler, 2002; Schults, 2022), refactoring is often a second thought as such efforts add costs and time to a project. Although Fowler (2002) and Schults (2022) do not recommend adding refactoring stories or tasks specifically to the project backlog due to the challenges noted above, particularly when addressing systems architecture, it is recommended that refactoring stories/tasks be included in the project backlog. If not included in the project backlog, then at a minimum, the contracting language should require the prime contractor to prove that modularity is being maintained in the project. In other words, refactoring is just as important as the highest priority stories/tasks on the project backlog.

Conclusion

The purpose of the research undertaken by USC and the SERC UARC is to determine how agile, DevSecOps and other modern acquisition methods and approaches can be successfully implemented in current and future DoW acquisition programs to accelerate systems development and deployment. Specifically, the research is focused on determining what tools and approaches can be incorporated into the acquisition process that allows programs to become sufficiently agile and nimble to address dynamic conditions from both an operations perspective (i.e., what does the warfighter need now) and also from an acquisition perspective (e.g., what tech has changed, has new personnel joined the team, etc.).

A series of recommendations and observations have been produced from this and past studies. Chief among those recommendations is the need to do some up-front systems engineering to ensure that the necessary “scaffolding” is in place prior to the start of an agile program. In addition, it is important that a near-operational integration and testing environment be put in place as soon as possible to allow DevSecOps and CI/CD operations to be implemented. Another recommendation is that a trained workforce is absolutely necessary for the successful implementation of an agile-based program. This requires having the on-boarding and continuous training materials on-hand that cover programmatic details of the acquisition environment and the targeted domain.

As for next steps, the team will continue to explore introducing and implementing agile, DevSecOps and other modern systems development approaches into the DoW acquisition environment. Of particular interest is the challenge of transitioning from a completely virtual model of the proposed system to a final deliverable system by replacing the virtual components with the actual working components of the final system. An additional active area of research by the USC/ISI team is in employing AI in the capturing and dissemination of corporate knowledge. An open question is whether an LLM or other AI technique, when exposed to various system artifacts, can provide the necessary information when used to on-board and train new and existing workforce members.

References

AcqNotes. (2024, February 23). *Developing DD form 250*. https://acqnotes.com/acqnote/careerfields/department-of-defense-form-250

Atlassian. (n.d.). *Jira: Issue & project tracking software*. https://www.atlassian.com/software/jira

Fowler, M. (2002). Refactoring: Improving the design of existing code. In *Lecture notes in computer science* (p. 256). https://doi.org/10.1007/3-540-45672-4_31

Lindsey. (2026, January 21). *Framework - Scaled Agile framework*. https://framework.scaledagile.com/#big-picture

Manifesto for Agile software development. (n.d.). https://agilemanifesto.org/

Microsoft Security. (n.d.). *What is DevSecOps? Definition and best practices*. https://www.microsoft.com/en-us/security/business/security-101/what-is-devsecops#:~:text=DevSecOps%2C%20which%20stands%20for%20development,of%20a%20multicloud%20security%20strategy

Microsoft Teams. (2026, January 24). In *Wikipedia*. https://en.wikipedia.org/wiki/Microsoft_Teams

Orosz, M. D., Hempen, J., Turner, J. M., Duffy, B., & Charlton, C. (2025). *Lessons learned in implementing DOD space-based Agile/DevSecOps programs*. https://gsaw.org/wp-content/uploads/2025/10/2025-S12-2-Lessons-Learned-in-Implementing-DoD-Space-based-Agile-DevSecOps-Programs.pdf

Orosz, M., Turner, M. J., Pool, J., Duffy, B., & Charlton, C. (2025). *Lessons learned in implementing integration and testing in a DoD space-based* Agile/DevSecOps *environment*. https://www.dropbox.com/scl/fo/bwqt81qgp7gkznr0e38i6/AM1xi0g5QpOqPjP488FvoS0/Papers%20-%2034th%20Aerospace%20Testing%20Seminar?dl=0&preview=Michael+Orosz_Agile+Aerospace+Testing+Seminar_Paper_20250313+(1).pdf&rlkey=t80lb3igo9bgptii1wv7vu935&subfolder_nav_tracking=1

Schults, C. (2022, January 10). *The when, why, and how of refactoring in Agile*. https://linearb.io/blog/refactoring-in-agile

Systems Engineering Research Center. (n.d.). https://sercuarc.org/

U.S. Department of the Air Force. (2024, June 10). *Department of the Air Force launches NIPRGPT*. https://www.af.mil/News/Article-Display/Article/3800809/department-of-the-air-force-launches-niprgpt/

U.S. Department of War. (2025, November 10). *Secretary of War announces acquisition reform*. https://www.war.gov/News/Releases/Release/Article/4329487/secretary-of-war-announces-acquisition-reform/

Acquisition Research Program
Department of Acquisition, Finance, and Manpower
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net