SYM-AM-22-059

Excerpt from the Proceedings of the Nineteenth Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 11–12, 2022

Published: May 2, 2022

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).

Introducing Agile/DevSecOps into the Space Acquisition Environment

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. 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. 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]

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, Duffy retired from the United States Air Force with multiple assignments related to National Security Space acquisition programs and command and control systems. Duffy holds a Masters of Aeronautical Science degree from Embry-Riddle Aeronautical University and a Bachelors of Aeronautical and Astronautical Engineering degree 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 19 years on a number of 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.

Lieutenant Colonel Spear—has served in the U.S. Air Force for 19 years as a Developmental Engineer and Program Manager. He is currently a Materiel Leader at Los Angeles Air Force Base and Program Manager for a GPS ground control system. He has served in test, engineering, and staff positions for space, air armament, and fighter aircraft programs. Prior to his current assignment, he was Deputy Chief of the F-35 Integration Division in the U.S. Air Forces in Europe Headquarters. He holds a BS in mechanical engineering from Florida State University and an MS in systems engineering from the Air Force Institute of Technology.

Abstract

The University of Southern California (USC) and its Information Sciences Institute (USC-ISI) is undertaking research into improving the space-based systems acquisition process through the adoption of agile and DevSecOps methodologies. The USC-ISI team is currently undertaking research and systems engineering analysis to explore the mission engineering methods, analysis, metrics and training needed to transition from a traditional DoDI 5000.02 waterfall development environment to an agile/DevSecOps space systems acquisition environment. Over the past several years, the project team has been embedded at the U.S. Space Force's Space Systems Command, Production Corps (SSC/PC), developing performance measuring tools, collecting performance metrics and providing subject matter expertise on three projects – a traditional waterfall project, a hybrid parallel waterfall and agile development project and an on-going long-term highly agile development effort that is subject to traditional waterfall acquisition reporting requirements. This paper summarizes initial research results and lessons learned along with a discussion on next steps.

Introduction

As mandated by Congress via the 2018 National Defense Authorization Act (NDAA) the Defense Acquisition System (DAS) is to transition many if not all (where feasible) of its programs to agile and DevSecOps processes. This mandate reflects the need for the

Department of Defense (DoD) to rapidly develop warfighter capability to better meet the global challenges currently confronting the United States in today's environment. Competitors, using both increased manpower and technology, are producing systems at a much faster pace than traditional DoDI 5000.02 (DoD, 2020) waterfall processes can sustain. Under funding from the Systems Engineering Research Center (SERC), the University of Southern California's Information Sciences Institute (USC-ISI) is currently undertaking research and systems engineering analysis to explore the mission engineering methods, analysis, metrics and training needed to transition from a traditional DoDI 5000.02 waterfall development environment to an agile/DevSecOps space systems acquisition environment. Over the past several years, the project team has been embedded at the U.S. Space Force's Space Systems Command, Production Corps (SSC/PC), developing performance measuring tools, collecting performance metrics and providing subject matter expertise on three projects – a traditional waterfall project (Project A – the baseline), a hybrid parallel waterfall and agile development project (Project B), and an on-going long-term agile development effort that is subject to traditional waterfall acquisition reporting requirements (Project C).

After adjusting for differences in periods of performance and software lines of code, the hybrid waterfall and agile project (Project B) produced approximately 85.4% fewer open problem reports (PRs) than the traditional waterfall project (Project A). Both projects exhibited the same level of software and systems complexity.

An analysis of the performance of the waterfall portion as compared to the agile portion of the hybrid project (Project B) revealed that the agile effort produced approximately 95.7% fewer open problem reports as compared to the waterfall portion of the effort. Both efforts exhibited similar code complexity and software lines of code; however, the agile effort took 10 months less time to complete, and its workforce was considerably less experienced than the waterfall team.

Currently, the project team is embedded in Project C and is collecting performance metrics, developing and deploying additional performance measurement tools, providing subject matter expertise and developing a workforce agile/DevSecOps training program. An initial analysis of the collected data, a summary of lessons learned, and the impact of the training program is underway and is summarized in this paper.

Funding Support

This material 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 U.S. Department of Defense nor ASD(R&E).

Multiple Projects

As described in later sections, this paper is a summary of a long-term on-going research effort that crosses three space-based DoD acquisition efforts. As noted in Row et al. (2020), there has been considerable effort and reporting on the successes and challenges of adopting agile and DevSecOps approaches into traditional DoDI 5000.02 waterfall projects. Although these efforts have produced valuable and productive results, they are often focused on only one project. Perhaps an interesting feature of the research reported here is that the USC-ISI research team has been immersed and observing (serially) three DoD space-based acquisition efforts, each focused on the same mission space, of similar complexity and size, and ranging from a fully waterfall-implemented effort to projects with ever increasing levels of agile/DevSecOps implementation. This research approach is allowing the USC-ISI team to observe the benefits and challenges of implementing agile/DevSecOps through an incremental long-term process.

Methods

Project Immersion

To uncover the steps and processes required to transition an acquisition program from a traditional waterfall environment to agile and DevSecOps, the research team fully immersed into each of the space-based projects (i.e., A, B and C) and became part of the multiple project teams (i.e., members of various integrated product teams [IPTs]). This immersion included participating in daily scrums, grooming/refinement activities and various ceremonies – including sprint and program increment (PI) reviews and demonstrations. In addition, the research team participated in various PI planning events and working group activities focused on DevSecOps processes and request for comment (RFC) explorations.

The Projects

To compare the benefits of establishing and operating an agile and DevSecOps environment, it was necessary to initially identify and study two different projects which exhibited similar levels of software complexity. The first project (Project A) which serves as the baseline for the research effort is comprised of approximately 170K software lines of code (SLOC), required 39 months to complete and was undertaken in a traditional DoDI 5000.02 waterfall environment. Project B consists of approximately 120K SLOC, required 25 months to complete and was undertaken as a hybrid project – approximately half of the project was undertaken using waterfall, and the remaining half was undertaken using agile and DevSecOps processes. Project C, which is on-going, consists predominately of an agile and DevSecOps process but exists within a traditional waterfall administration/reporting environment. All three programs were judged to exhibit the same level of software complexity. Note the waterfall methods in Projects A and B follow a DoDI 5000.02 pre–January 23, 2020, approach. The agile methods followed in Projects B and C are using a tailored “Major Capability Acquisition” approach that incorporates agile because both projects’ acquisition strategies were developed before the latest DoDI version was written.

Data Collection:

As noted in Orosz et al. (2021), for projects A and B, problem reports (PRs) were collected daily throughout the project. Cost data was unavailable at the time of this paper but will be included in future analysis. For the on-going Project C, story and feature completion status are tracked via extractions from the contractor’s Jira® (Atlassian) issue tracking system. Status tracking includes identifying which stories and features completed as scheduled (i.e., within the assigned sprint or program increment [PI]), did not complete and spilled over into the next sprint or PI, were pushed to the next PI or future PI, and which were added to the PI backlog during PI execution. Once software development begins, PRs will also be tracked via data extraction from the contractor’s implementation of IBM® Rational® Dynamic Object-Oriented Requirements System (DOORS®) or DOORS Next Generation (DNG) system (IBM). In addition, cost tracking will also be collected.

Results

Projects A (Waterfall) and B (Hybrid)

As reported in Orosz et al. (2021), to compare results between project A (baseline waterfall effort) and project B (hybrid waterfall/agile effort), the following steps were taken in processing and analyzing the collected performance data:

1. Problem reports (PRs) were collected during the integration and testing phase of each project. For the agile portion of Project B, this included collecting PRs produced during the integration and testing phase of each sprint.
2. The number of PRs in an Open (unresolved) state were captured at the end of each week of software development for each project.
3. The PR counts were plotted in online graphs (Figures 1 and 2) to compare the development history of the two software development projects.
4. The timeline of the waterfall project (Project A) was shifted so that the Formal Qualification Testing (FQT) period overlaid that of the Hybrid project (Project B).
5. The PR counts of the waterfall project (Project A) were proportionally reduced using the relative SLOC counts of the waterfall and hybrid projects.

The decision to proportionally reduce the PR count of Project A (item 5) due to SLOC count differences with Project B was motivated by the observation that the code complexity of both projects was judged to be equivalent; therefore, proportionally reducing the PR count based on the ratio of SLOC between the two projects would provide a more equivalent basis for comparing the productivity of the two development projects.

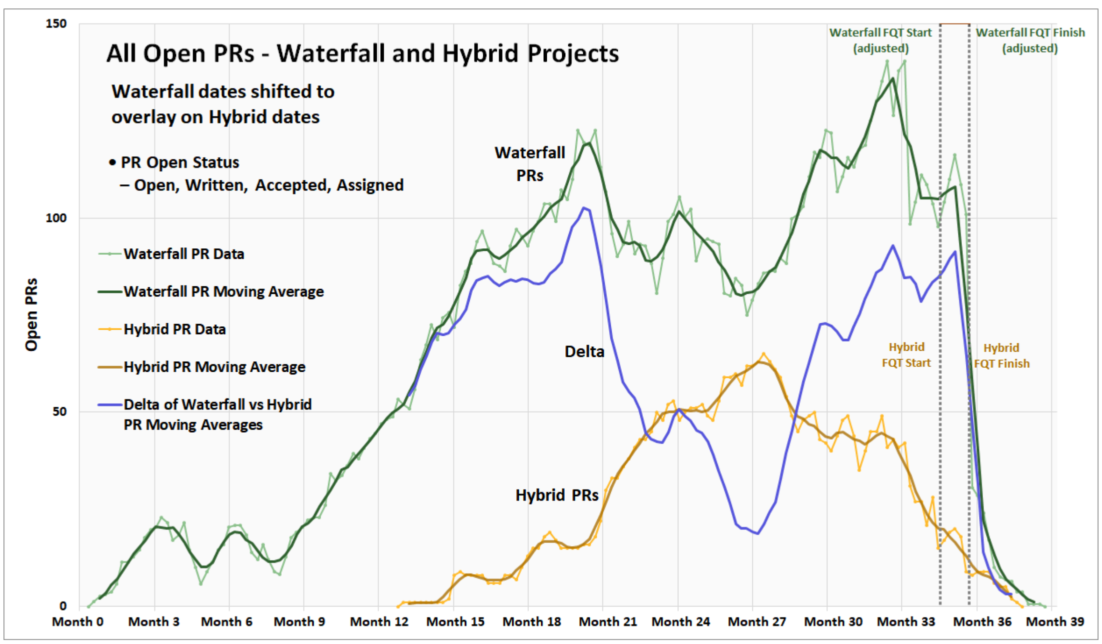

Figure 1. Comparison of PR Counts Between Project A (Waterfall) and Project B (Hybrid) (Orosz et al., 2021)

The project timeline for both projects were shifted so that the period of FQT for both projects coincide. The PR count for Project A is proportionally reduced to reflect the difference in SLOC between the two projects. Code complexity was judged to be the same for both projects. The green upper line is the PR trace for Project A. The yellow lower line is the PR track for Project B. The blue line is the difference in PR count between both projects.

In Figure 1, the two "PR peaks" for Project A reflect a situation where there were so many PRs identified during Component Integration and Testing (CIT) that the team was overwhelmed and had to stop CIT activities, address high priority PR events, and stretch the project timeline out to allow time to address the PRs before completing CIT (the second peak) and then going on to Formal Qualification Testing (FQT).

As also cited in Orosz et al. (2021), for Project B, the waterfall and agile teams worked in parallel, with periodic "merges" that underwent integration and testing. This helped reduce the "PR bow wave" because integration problems were discovered early during these "merge" events. In addition, the agile team undertook frequent integration and testing between the "merge" events (i.e., as part of each sprint). This allowed almost continuous integration and testing which resulted in problems being identified early (i.e., reduced PRs), before CIT was officially started with the merged waterfall and agile development components.

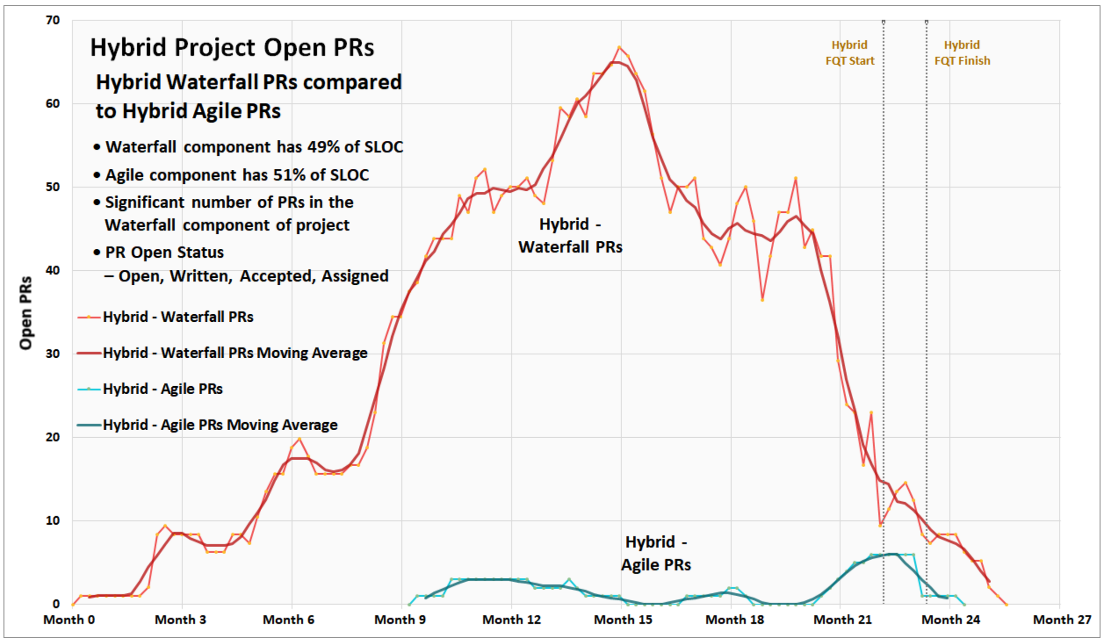

Figure 2. Comparison of PR Counts Between Waterfall and Agile Components of Project B (Hybrid)

(Orosz et al., 2021) Code complexity was judged to be the same for both efforts. Software lines of code (SLOC) were roughly the same for both projects. The red upper line is the PR trace for the waterfall effort. The dark blue lower line is the PR track for the agile effort. Note that the agile effort started 10 months after the program start and the project team was relatively inexperienced in working on agile projects or space programs in general.

As shown in Figure 2, the agile team in Project B produced considerably less PRs over the period of performance than the waterfall team. This was despite the agile team's late start (delayed 10 months) and the experience level of the agile development team was less than that of the waterfall team members (i.e., the agile effort included "ramp-up" training time for the team).

Study and Project C

Project C is focused on enhancing an existing software platform that was developed using waterfall. Code complexity is very similar to projects A and B. Like the hybrid project (Project B), Project C exists within an acquisition management system that relies on waterfall metrics (lines of code written/tested, CDRLs, EVM, number of PRs reported and worked off, etc.).

Due to the size of Project C, the contractor has implemented a modified Scaled Agile Framework® (SAFe) development environment. The project is comprised of a separate 15-month study phase consisting of six 10-week program increments (PIs) followed by an approximate 51-month-long execution phase, consisting of 19 program increments (PIs), with the first three PIs covering a 10-week performance period divided into five two-week sprints (Figure 3) and the remaining 16 PIs each consisting of a 13-week performance period divided into four three-week sprints and a one-week "PIT Stop" (Figure 4) reserved for demonstrations, innovation exploration, training and PI retrospective activities. The change from a 10-week PI to a 13-week PI resulted from the contractor's observation that the ceremony cadence in a two-week sprint was impacting feature team performance, so a three-week sprint was adopted.

In the study phase, the contractor A) undertook technical trade studies, B) established and operated a SAFe® agile development environment and C) established an initial DevSecOps pipeline infrastructure. As part of establishing the agile development environment, the contractor focused on system requirements decomposition into capabilities and features (with some stories) and initially populated the project backlog in preparation for the execution phase. During the study period, the government team also "ramped up" to become agile/DevSecOps "smart" – including how to manage an agile/DevSecOps project within a waterfall acquisition world (e.g., contracting, staffing, monitoring, managing, etc.). As part of the "ramping up," contractor-provided SAFe® training was made available to both contractor and government personnel. In addition, the government provided topic-specific training to its government personnel (to help ramp up the government's team understanding of agile/DevSecOps work processes).

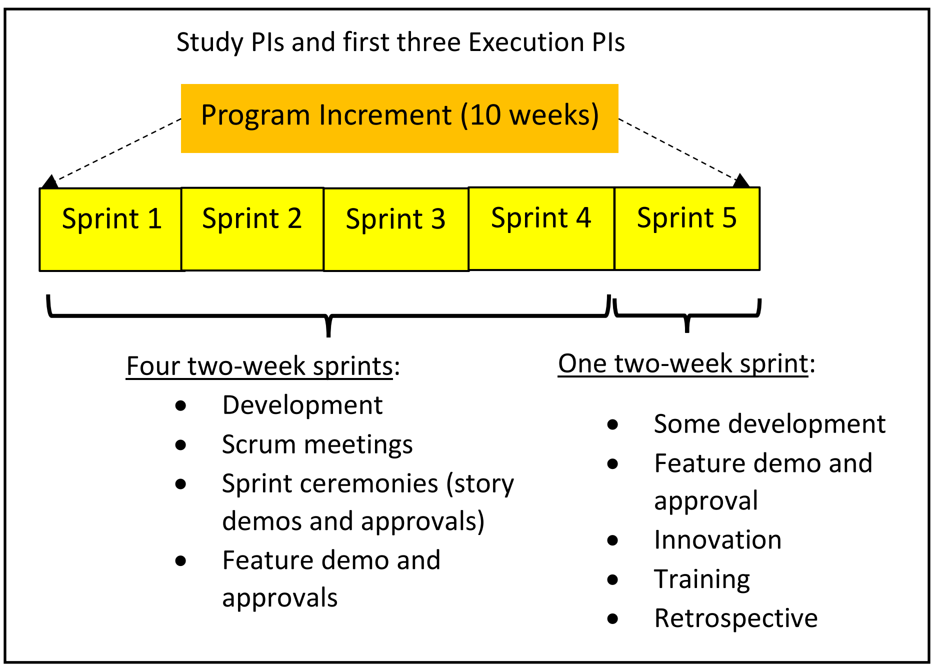

Figure 3. Project C relies on a modified SAFe development framework consisting of a 15-month study phase with six 10-week PIs and an approximate 51-month execution phase consisting of 19 program increments (PIs) with the first three PIs covering a 10-week performance period divided into five two-week sprints. From the study phase (initial 15-week effort), an initial project backlog was produced and divided into six team backlogs to coincide with the six development teams in the project.

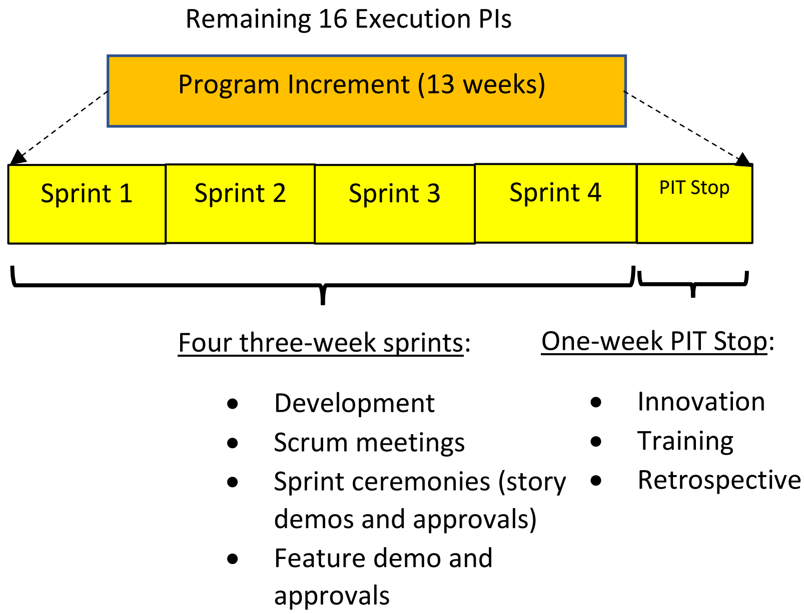

Figure 4. Project C relies on a modified SAFe development framework consisting of a 15-month study phase with six 10-week PIs and an approximate 51-month execution phase consisting of 19 program increments (PIs) with the first three PIs covering a 10-week performance period divided into five two-week sprints and the remaining 16 PIs each consisting of a 13-week performance period divided into four three-week sprints and a one-week “PIT Stop” reserved for demonstrations, innovation exploration, training and PI retrospective activities.

Tools Developed

During both the Project C study and initial eight months of the execution phase, the research team developed data extraction and analytic tools to help the government monitor and track program performance. These tools were developed to meet gaps in the availability of performance metric monitoring tools due to several reasons. First, the government team members could not access the contractor's implementation of Atlassian® Jira® and IBM Rational DNG platforms due to various cybersecurity and IT reasons. Second, many commercially available tools that provide similar capabilities are restricted from being used on the project due to import control protocols. In other cases, there are limits in license availability or the tools require a specific configuration in Jira or DNG that can’t be supported by the contractor executing Project C. In any case, these challenges necessitated the need to develop inhouse monitoring and analysis tools.

Many of the tools rely on extracting feature and story status information from daily Jira exports from the contractor. Microsoft® Excel® (with Visual Basic developed scripts) are used to process the extracted data and visually presented for government review. Table 1 summarizes the tools developed to date on Project C.

Table 1. List of Tools Developed to Date in Project C

Tool Name	Tool Description
Status board	Presents list of capabilities, features and stories by status (done, in-progress, blocked, backlog, etc.) and MVP/MMP, PI, and Sprint.
Issue viewer	Allows detailed inspection of any Jira issue (e.g., review acceptance criteria, definition of done, dependencies, PI assigned, priority assigned, etc.)
Feature Team Plan	Shows features and status assigned by Feature Team and the current PI assignment.
MVP vs PI Summary	Graphical table view of feature quantities as assigned by PI, MVP/MMP, and Release.
MVP-MMP Plan	Shows features and status assigned by MVP and MMP and the current PI assignment.
MVP-MMP Trace	Displays a hierarchy organized by MVP/MMPs with child capabilities, features, and stories; provides associated Jira information such as PI and status.
DNG Jira Summary	Presents program requirements and linked features with associated Jira information. Combines information from DNG requirements module export, DNG feature module export, and Jira export.

In Figure 5, the MVP-MMP Plan tool shows progress (features completed) towards meeting the first MVP (68% complete) deliverable (1-a). The tool also shows that there are two features assigned to the first MVP that are assigned to PIs that occur after the planned release of the first MVP.

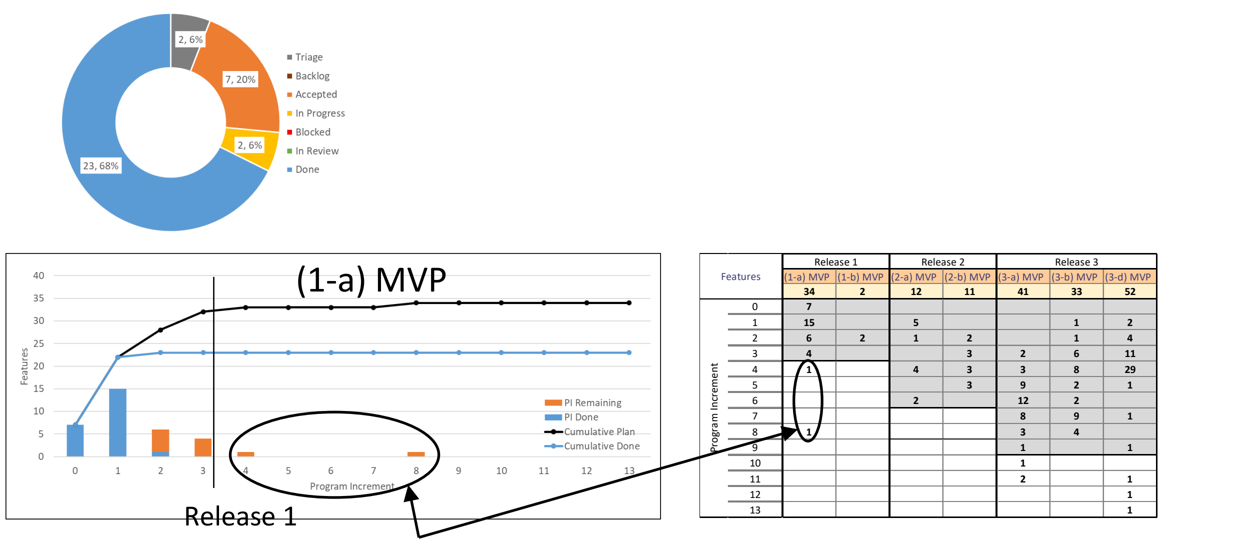

Figure 5. The MVP-MMP Plan Tool Showing Progress (Features Completed) on Meeting the First MVP (68% Complete). The tool also shows that there are two features assigned to the first MVP that are assigned to PIs that occur after the planned release date of the first MVP.

In Figure 6, the Feature Team Plan tool shows the current plan and progress (features completed, sized by story points) over time (PI) for a specific team. The tool shows that maximum PI capacity so far has been approximately 500 story points, PI 3 is in progress, and a surge of effort is planned for PI 4.

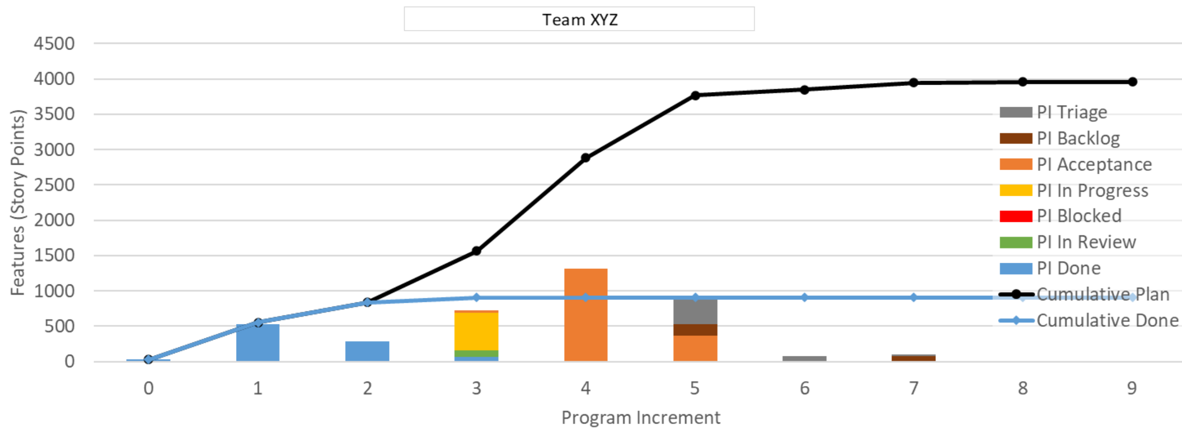

Figure 6. The Feature Team Plan Tool Shows the Current Plan and Progress (Features Completed, Sized by Story Points) Over Time (PI) for a Specific Team. The tool shows that maximum PI capacity so far has been approximately 500 story points, PI 3 is in progress, and a surge of effort is planned for PI 4.

Lessons Learned

During both the Project C study phase and during the initial eight months of the execution phase, the following lessons learned were observed.

Incongruity between agile and traditional programs with set budgets, requirements and timelines. The rigidity of the Capability Development Document (CDD) can cause considerable challenges during project RFP development, contract negotiations and program operation. For example, there is little “wiggle room” available to allow the contractor and government to *easily* shift priorities of requirements (via their decomposed features on the project backlog) and MVP/MMP release dates to account for temporary blockages (e.g., availability of an external dependency) or the introduction of a higher priority Request for Change (RFC) into the backlog. Undertaking such changes can be lengthy (in terms of time) as the proposed adjustments must be coordinated and approved up to the Joints Requirements Oversight Council or Service level. Additionally, with program funding matched to CDD threshold requirements level, the PM’s flexibility is removed. A possible solution is to allow PEO trade space among non-KPP requirements. Another approach (which is the subject of future research) is to identify “gray” areas within the definition and decomposition of CDD requirements that may be able to be traded for higher-priority items during the agile development process while still meeting the basic standards of the CDD.

Need to do engineering up front. Often, project teams will assume that with agile, there is less need for up-front engineering as the details of the design will unfold as the project moves along. Doing this will result in several challenges. First, dependencies between features (internal or external to a project team or CI) can often be overlooked if the proper engineering is not done up front. Second, to fully prioritize features on the backlog, there must be a roadmap and an initial design to help guide the decision making behind assigning priorities to the features on the backlog. Finally, it is necessary to fully decompose CDD/spec requirements (up front) to determine if all features have been identified for the project. This decomposition process can be quite complex as each requirement may require multiple features that cross multiple feature development teams. Further, the translation of feature verification up to FQT completion is also a major challenge. These features don’t have to have detailed designs behind them, but they do need sufficient detail so that when combined with other features, the requirement in which they are decomposed is fully covered. This upfront engineering and planning work is also critical for continually tracking the relationship between near-term, detailed plans and performance and higher-level plans for project completion.

Access to performance tools. As previously noted, government team members (at least initially) could not directly access the contractors' Jira and DNG platforms due to various cybersecurity and IT reasons. In addition, many commercially available performance tracking tools (e.g., Jira plug-ins) are not available to the contractor or government team members due to U.S. import restrictions. Many of these tools are developed and maintained by vendors outside of the United States. This is becoming a particularly difficult challenge as the performance metrics measuring tools marketplace continues to evolve with companies – many headquartered here in the United States – being acquired by foreign-owned companies. These challenges need to be addressed earlier in the program prior to the full ramp-up of the project.

PI length is too short. During the study phase, a 10-week PI divided into five two-week sprints was sufficient for technical innovation, project backlog population and in establishing an initial agile/DevSecOps acquisition framework. Once the execution phase started, however, the two-week sprint cadence, with its frequent planning, grooming/refinement and various ceremonies, didn't allow sufficient time for the feature teams to undertake the necessary work to complete stories within each sprint. The work on stories frequently spilled into and consumed most of sprint 5, leaving little time for feature demonstrations, innovation activities, training and retrospective activities. This realization led the contractor and government to extend the PI to 13 weeks divided into four three-week sprints followed by a one-week PIT Stop to allow for feature demonstrations, innovation and training activities. The key lesson here is that feature teams need the flexibility to adjust PI lengths and sprint cadence to meet the demands of the project. That said, it's important that all feature teams rely on the same PI length and sprint cadence to reduce synchronization and management challenges if teams are working different timelines.

Story Assignment Up Front. On multiple occasions, the project team observed that when stories are not initially (or tentatively) assigned to sprints in the upcoming PI, there is a high risk that the parent feature will not complete within its assigned PI. This is particularly a challenge with features that have many stories and story points. Much of the difficulty can be traced to not adequately matching anticipated team capacity to total story points assigned to the PI. In addition, keeping stories on the team backlog and not assigned to a sprint often resulted in the Product Owner (PO), scrum masters and team members overlooking these unassigned stories. When discovered later in the PI, it was often too late to adequately work the stories before the end of the PI. This often resulted in the feature team working stories through the last sprint/PIT Stop, allowing little or no time for adequate training, innovation and other non-development activities.

Too many story points in the PI. During PI planning, some feature teams allocated stories whose total story point count either met or exceeded the full capacity of the feature team. This resulted in no room for error in the event an assigned story took longer to complete than scheduled, a higher priority capability was added to the PI, there was reduced availability of manpower due to unexpected time off (e.g., sick leave), or some other unexpected event occurred. It is important that the PO leaves a buffer of capacity (say 20% of the total available capacity) to allow for unexpected events and activities.

Stay focused on MVP/MMP and the roadmap. Teams often drifted focus away from project priorities (meeting MVP/MMP and roadmap milestones) and instead to sprint planning activities and prioritizing completing as many features and stories as possible to meet velocity goals. This problem was often traced to PI planning where high-priority features are selected from the project backlog. The solution here is for the government to assign priorities to the project backlog based on MVP/MMP priorities and structure performance incentives based on progress towards the upcoming MVP/MMPs instead of on the number of features worked and completed. A roadmap should be developed and maintained in sufficient detail to continually identify the impacts on the total project caused by performance in near-term program activities.

Training. Members from both the contractor and the Government bring different experiences and expectations on agile and DevSecOps to the project team. In many cases, team members have little previous experience with either approach when on-boarding onto projects, are not familiar with scaling the approaches or bring prior experience from projects that have little resemblance to the current project (e.g., coming from industry). This challenge applies whether the team member is a software developer, a tester, or a project manager. In Project C, the government team is currently focused on establishing a training curriculum that starts with a focus on the foundational elements of agile and DevSecOps (e.g., what is agile, why use it, etc.), transitions to more advanced topics such as agile frameworks and contracting, and finishes out with a focus on the specific implementation of agile and DevSecOps to Project C. This training is targeting on-boarding of new team members and provides on-going training to address the evolving acquisition environment.

Need for an operations-like test environment as soon as possible. Since many projects are composed of both hardware and software components, with hardware lagging software in terms of availability, it is critical that a near operations test environment be available to the DevSecOps pipeline as soon as possible to facilitate continuous integration/continuous deployment (CI/CD) operations. This often means scheduling and prioritizing the development and acquisition of the near operations environment as early as possible while also putting in place backup plans in the event the planned near operations environment is not initially available when scheduled.

Next Steps

As of the writing of this paper, Project C is eight months into a 51-month effort. Software development is just underway, and the initial MVP deliverables are several months away. Going forward, the project team is focused on collecting and analyzing performance data such as 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 particularly 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 that is focused on delivering value rather than traditional waterfall metrics, such as software lines of code. For example, a key area of research is in 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

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

DoD. (2020). *Operation of the adaptive acquisition framework* (DOD Instruction 5000.02). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500002p.pdf

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

Roe, S., Novak, R., Staresina, P., & Raines, J. (2020). *Recommendations for injecting flexibility into acquisitions for software development and operations (DevSecOps) and agile software development* (MITRE Technology Report PRS-19-3551). https://apps.dtic.mil/sti/pdfs/AD1114449.pdf

Scaled Agile. (n.d.). https://www.scaledagileframework.com/

Orosz, M., Evans, J., Duffy, B., Charlton, C., & Mitchell, R. (2021). *WRT 1012: Global positioning systems - Mission engineering and integration of emerging technologies* (SERC Report SERC-2021-TR-001). https://sercuarc.org/publication/?id=231&pub-type=Technical-Report&publication=WRT%201012:%20Global%20Positioning%20Systems%20-%20Mission%20Engineering%20and%20Integration%20of%20Emerging%20Technologies

National Defense Authorization Act for Fiscal Year 2018, H.R. 2810. (n.d.). https://www.congress.gov/bill/115th-congress/house-bill/2810

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

www.acquisitionresearch.net