This article provides insights into the current state of developmental testing (DT) and requirements management in Department of Defense information systems employing Agile development. The authors describe the study methodology and provide an overview of Agile development and testing. Insights are described for requirements, detailed planning, test execution, and reporting. This work articulates best practices related to DT and requirements management strategies for programs employing modernized Software Development Life Cycle practices.

DOI: https://doi.org/10.22594/dau.19-819.26.02
Keywords: Development and Operations (DevOps), DoD, Software Development, Automated Testing, Software Development Life Cycle (SDLC)
Industry software development efforts have used Agile and development and operations (DevOps) methodologies over the last 5 to 15 years. The Department of Defense (DoD) has applied these methodologies to various information system acquisition programs, and current guidance provides a renewed interest in pursuing these methodologies. The National Defense Authorization Act for Fiscal Year 2018 (NDAA, 2017) directs acquisition Program Management Offices (PMO) to pursue Agile or iterative software development by establishing pilot programs to use “Agile or Iterative Development methods to tailor major software-intensive warfighting systems and defense business systems” (§ 873) and “Software Development Pilot Program Using Agile Best Practices” (§ 874). Similarly, the 2018 Air Force Guidance Memorandum for Rapid Acquisition Activities (AFGM 2018-63-146-01) states that, “Agile software development and [DevOps] is required for all new initiatives unless waived” (Department of the Air Force, 2018, p. 9).

Background

This study was motivated by the 2018 NDAA and increased emphasis on Agile development throughout the DoD; the lead author collected data from April to September 2018 during a rotational assignment within the Office of the Deputy Assistant Secretary of Defense for Developmental Test and Evaluation, DASD(DT&E). The purpose of the study was to collect insights and best practices from DoD program offices practicing Agile software development methods, some of which use the DevOps development and deployment strategy.

The five DoD software-intensive systems shown below were selected for this study, as they have been informally recognized as successful development programs for their use of Agile methodologies, integrated testing, and/or iterative development. The authors are not aware of any independent review that determines if the programs are achieving their objectives.
• Joint Space Operations Center (JSpOC) Mission System (JMS), Air Force
• Distributed Common Ground System–Navy (DCGS-N) Increment 2, Navy
• Global Combat Support System–Joint (GCSS-J), Defense Information Systems Agency (DISA)
• Reserve Component Automation System (RCAS), Army
• Catapult/ANTS (Attack the Network Tool Suite), Joint Improvised-Threat Defeat Organization (JIDO)

Table 1 provides detailed characteristics of the programs that we surveyed and analyzed for their Agile development and test approaches. The programs are arranged into two groups: System of Systems (SoS) and Web Applications (Web Apps). The Acquisition Category (ACAT) is either: (a) Automated Information Systems with the Defense Acquisition Executive (IAM), or (b) DoD Component head (IAC), or designated delegates as the decision authority.

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
<th>Group</th>
<th>Component</th>
<th>ACAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMS</td>
<td>Space command and control situational awareness</td>
<td>SoS</td>
<td>Air Force</td>
<td>IAM</td>
</tr>
<tr>
<td>DCGS-N Increment 2</td>
<td>Tactical gateway for Navy-unique sensor data across the intelligence community</td>
<td>SoS</td>
<td>Navy</td>
<td>IAC</td>
</tr>
<tr>
<td>GCSS-J</td>
<td>Joint military integrated logistics through applications and tools</td>
<td>Web Apps</td>
<td>DISA</td>
<td>IAC</td>
</tr>
<tr>
<td>RCAS</td>
<td>Integrated, web-based software solutions to manage mobilization, safety, personnel, and force authorization requirements</td>
<td>Web Apps</td>
<td>Army</td>
<td>IAM</td>
</tr>
<tr>
<td>Catapult/ANTS</td>
<td>Catapult integrates global intelligence data; ANTS addresses improvised explosive device threats</td>
<td>Web Apps</td>
<td>JIDO</td>
<td>N/A</td>
</tr>
</tbody>
</table>

The SoS and Web Apps groupings are general in nature. The web application suites (GCSS-J, RCAS, and Catapult/ANTS) consist of one infrastructure system feeding multiple web applications, generally with loose coupling.
Replicating the production environment in a development ecosystem is usually straightforward. On the other hand, SoS (DCGS-N, JMS) are typically more complex than web application suites because they likely depend on tightly coupled, disparate systems that are sensitive to changes. Different vendors may have developed each subsystem, which increases integration challenges due to proprietary information or custom interfaces.

**Agile Software Development**

Agile Software Development Life Cycle (SDLC) methods have emerged as industry best practices. The Agile methodology emerged in 2001 when 17 leading software developers created the Agile Manifesto to design and share better ways to develop software. Agile values can be distilled into four core elements:

- Focusing on small, frequent capability releases
- Valuing working software over comprehensive documentation
- Responding rapidly to changes in operations, technology, and budgets
- Actively involving users throughout development to ensure high operational value

Each of the many existing Agile methodologies (e.g., Scrum, Extreme Programming [XP], Kanban) has its own unique processes, terms, techniques, and timelines.
DevOps Software Development

DevOps SDLC methodologies build upon Agile methodologies by addressing information technology (IT) operations team deployment activities, resulting in faster deployment to the production environment—daily or weekly instead of monthly or semiannually. DevOps works best when the system is in sustainment, when enhancements are smaller, and when a mature user base is established. Increased emphasis is placed on shifting sequential processes into parallel processes and deploying capabilities to the user faster, which entails (Cagle, Kristan, & Rice, 2015):

- Strong use of automation (code, build, infrastructure/architecture, test)
- Continuous development, test, integration, and demonstration of the software throughout the life cycle
- A high level of collaboration and integration between development, quality assurance, and IT operations
- Static, dynamic, and fuzz testing techniques
- Open source and reusable components, services, and development tools

A variant of DevOps is Secure DevOps, which emphasizes security throughout the entire process. This variant is known as SecDevOps and DevSecOps, and incorporates automated security testing, monitoring, and securely handling deployment, typically through hashed Docker containers.

Method

A team of DASD(DT&E) staff specialists developed a 38-question survey for PMO personnel to complete between June and August 2018, which contained questions in these areas, with specific survey questions available upon request:

- Agile development, specifically method, management tools, and structure of teams
- Requirements tool usage
- Documents used during actual development
• Test planning, test events, and test implementer (developer, DT&E, operational test and evaluation [OT&E])

• Greatest successes and challenges to DT&E program, and future goals

• Lessons learned and best practices in the areas of:
  o Requirements
  o Detailed planning
  o Test events and execution

Additionally, the team interviewed the 45th and 47th Test Squadrons (TS) at Eglin Air Force Base (AFB), FL, to learn more about their Agile developmental testing (DT) efforts. The 45th TS is the principal Air Force organization for strategic air and space, weapon platform, and business systems software DT&E. The 47th TS is the principal Air Force organization that evaluates cyber resiliency of both information- and avionics-based systems. Both belong to the 96th Cyberspace Test Group (F. B. Chavers, personal communication, June 13, 2018).

Analysis

The following sections compare and contrast the detail between the software development methods, requirements, detailed planning, and test events of the five programs studied. Emphasis is placed on automated testing, sprint-level testing, and user acceptance testing.
SDLC Method

Table 2 identifies each program’s SDLC method, Agile type, tools, and method of defining requirements.

<table>
<thead>
<tr>
<th>Program</th>
<th>SDLC Method</th>
<th>Agile Type</th>
<th>Tools</th>
<th>Requirements Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMS</td>
<td>Scrum/Waterfall</td>
<td>Hybrid</td>
<td>Jira, self-developed</td>
<td>User Story</td>
</tr>
<tr>
<td>DCGS-N Increment 2</td>
<td>Scaled Agile Framework (SAFe)</td>
<td>Hybrid</td>
<td>Jira</td>
<td>User Story</td>
</tr>
<tr>
<td>GCSS-J</td>
<td>Scrum</td>
<td>Agile</td>
<td>Rally, Jira</td>
<td>User Story</td>
</tr>
<tr>
<td>RCAS</td>
<td>SAFe, Secure DevOps</td>
<td>Hybrid</td>
<td>VersionOne</td>
<td>User Story</td>
</tr>
<tr>
<td>Catapult/ANTS</td>
<td>Secure DevOps</td>
<td>Hybrid</td>
<td>Jira</td>
<td>User Story</td>
</tr>
</tbody>
</table>

The programs use industry standard Agile SDLC tools, as referenced in the 12th Annual State of Agile Report, which states that 58% of self-reported agencies use Jira, 20% use VersionOne, 9% use CA Agile Central (Rally), and 46% use Excel (respondents could select multiple items) (CollabNet VersionOne, 2018). Within the report, “Respondents were asked whether they would recommend the tool(s) they are using based on their experience,” (p. 15) with 81% recommending VersionOne, 77% recommending Jira, and 72% recommending CA Agile Central (Rally), the top three tools.

Within the GCSS-J program, software developers use Rally for SDLC, and the testers use Jira for user stories and wireframes (detailed diagrams) to help design test cases. Jira provides issue/bug tracking and workflow management.

JMS uses Jira and a self-developed tool for internal tracking of requirements, which are user stories, workflows, and requirements documents. The user stories are derived from workflows or checklists from current operations. Test cases are built over a compilation of user stories that sum to a specific functionality.
VersionOne, used by RCAS, has a built-in requirements traceability matrix from capability to feature, to story, to task, to test case, which is extracted and delivered as a part of each release. The user stories contain the acceptance criteria used to validate each story during sprint-level testing. Teams also use various levels of workflows and conceptual design specifications based on the need of the specific feature.

One observation by the PMO is the need to fully train personnel in tool use to make maximum use of the capabilities provided by the tools they employ. Without additional system configuration, many of these tools do not work in DoD environments because of security and/or architecture limitations. An example is a defect-tracking system that uses its own server and structured query language database, which requires more than a client side installation.

**Agile method types and scope.** Table 3 summarizes the method, timeframe, and team attributes of the programs studied, which generally depended on the complexity and size of the project and the maturity of the team.

<table>
<thead>
<tr>
<th>Program</th>
<th>Method</th>
<th>Iteration Duration</th>
<th># of Iterations to a Release</th>
<th>Sprint Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMS</td>
<td>Scrum/Waterfall</td>
<td>1 month, 2 month integration</td>
<td>6 to 20 depending on Service Pack</td>
<td>1 Integrator 4 to 6 Vendors</td>
</tr>
<tr>
<td>DCGS-N</td>
<td>SAFe</td>
<td>1 month</td>
<td>3 sprints per Capability, 4 per Fleet Capability Release</td>
<td>6 working in parallel, started with 2</td>
</tr>
<tr>
<td>Increment 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCSS-J</td>
<td>Scrum</td>
<td>1 month</td>
<td>4, release capabilities as completed</td>
<td>4 in parallel</td>
</tr>
<tr>
<td>RCAS</td>
<td>Hybrid with XP, Kanban and Secure DevOps. SAFe compliant</td>
<td>3 weeks</td>
<td>4 sprints, quarterly releases, 1 week release planning</td>
<td>18 teams across RCAS portfolio 8 to 14 people per team, 2 separate locations</td>
</tr>
<tr>
<td>Catapult/ANTS</td>
<td>Hybrid Agile, Secure DevOps</td>
<td>2 days to 2 weeks</td>
<td>1 month minor, 3 months large</td>
<td>N/A</td>
</tr>
</tbody>
</table>

The Secure DevOps PMOs (RCAS and Catapult/ANTS) stressed the importance of teamwork and culture to support their efforts, while other PMOs talked about the benefits of good communication and dedicated
support from different acquisition team members. RCAS and Catapult/ANTS PMOs both remarked on how much time it took to achieve their current DevOps cultures. Both programs started around 2012/2013 and extensively customized their SDLC ecosystems. Both PMOs have delivered ecosystems that allow continuous planning, construction, integration, and delivery.

"The purpose of the study was to collect insights and best practices from DoD program offices practicing Agile software development methods, some of which use the DevOps development and deployment strategy."

Catapult/ANTS PMO further noted that it has taken 6 years to transition from waterfall to a Secure DevOps environment. Major efforts included bringing in coaches to teach and streamline processes toward Agile (Scrum) development approaches from 2012 through 2014, bringing in automated operating system-level virtualization tools in 2015, and finally achieving a continuous authorization risk management framework pipeline after 2016.

Requirements Gathering

Users provide requirements to enable the PMO to build a capability to meet the requested need. For requirements gathering, user community functional representatives send requirements to the development team, with details that vary based on the program’s SDLC approach (Table 2). For GCSS-J, the PMO requirements teams meet with functional requirements groups, and then work with the developer to craft user stories for the program. Catapult/ANTS PMO receives new requirements in three ways: (a) any user can submit a help desk ticket to address a software defect or new requested functionality; (b) JIDO’s mission IT development engineers can collect requirements from customers in the field; and (c) forward-deployed personnel collect requirements while embedded at the intelligence analysis centers. For example, personnel may see the value of a new requirement in an analysis ad hoc report and turn it into an enterprise solution for all analysis centers. The PMO also receives new mission requirements from DoD, JIDO, or other mission-related initiatives; and then development engineers work closely with users, Combatant Command staff members, and embedded analysts to vet, identify, and rank new requirements. RCAS PMO has a similar process for gathering requirements from the user, meeting with users on-site for each sprint.
**Detailed Planning**

All programs use initial iteration-planning meetings, and variations in their processes are outlined in this section.

For JMS PMO, user stories are sent to the integrator to be approved and executed. User stories that impact multiple requirements can be shared across vendors. However, the cross-coordination of requirements between vendors is a challenge.

RCAS and Catapult/ANTS PMOs both employ daily scrum meetings to manage development through their requirements backlog. Each backlog item has its associated tests identified as “To be built” and “To be executed.” Stories cannot be closed until all tasks are built, all tests passed, and both the product owner and user community accept the solution.

GCSS-J PMO holds weekly and/or biweekly working-level integrated product team (WIPT) meetings with all stakeholders to develop and document test plans for every release. Test dates are included in the integrated master schedule.

DCGS-N PMO assigns testers to each sprint team to participate in scrums, which are daily stand-up meetings. Weekly internal test-planning meetings are also held. External test and evaluation (T&E) WIPT meetings are held with stakeholders.

**Test Events and Execution**

The traditional roles (developer, integrator, and government tester) and test events are more efficient when reorganized in the Agile environment; however, Agile and DevOps bring new challenges to test events. The developer integration tests are typically executed by the development team, and System Integration Tests (SIT) are executed by the integrator or
government testers. System Qualification Tests (SQT), or system acceptance
tests, are run by government testers. Depending on the circumstance, the
roles may be dual-hatted (i.e., developers may both code and test), or the
phases may be merged, as DevOps removes the separate integration step.

**Role of the government tester during development sprints.** Even
when testers are tasked with development duties, they need to ensure that
the code developed during the sprint meets the “definition of done” by
thorough automated or manual testing.

With DCGS-N PMO, government testers are assigned to each sprint team
and participate in scrums to provide their perspective, working side-by-
side with the developer. The 45th TS is reviewing ways to better support
development sprints, such as retraining their personnel to better support
automated testing or provide specific manual testing solutions.

Catapult/ANTS PMO has encouraged its testers to participate in other parts
of the development chain so they can learn test automation, engineering,
requirements gathering, and other disciplines.

**Role of the developer during government DT.** Some programs have
separate government DT in the form of SITs, SQTs, or User Acceptance
Tests (UAT), where development contractors provide subject matter
expert support during government DT. DCGS N contractors also provide
user training, technical support, maintenance, and simulation support.
JMS vendors are present during 45th TS government testing to speed up
defect turnaround, so that when a system defect is found, the vendor can
immediately understand what is happening and start to develop a solution.

**Test types.** The developer and government testers perform the
following levels of test: automated testing, sprint-level testing, and user
acceptance testing. This is not an all-inclusive list.

**Automated software testing.** Automated software testing is
conducted as required, generally at each code compilation. Programs use, or
intend to use, automated tools as much as possible because of the recognized
time and cost savings. However, the hardware and software ecosystem can
make the return on investment (ROI) difficult.

Programs with a DevOps pipeline use robust automated tools because they
are also testing for integration. RCAS and Catapult/ANTS PMOs both
note that no single tool works best, and that automation frameworks are
a collection of tools used to support automation across the Agile solution
delivery life cycle.
Catapult/ANTS PMO has a continuous integration and deployment pipeline that uses Jira, Jenkins, and Docker tools. Specifically, the bulleted list of testing/quality gates is performed in the following order. If these tests are not passed, then code deployment is halted and left in the queue for examination (L. T. Garciga & D. Lederman, personal communication, July 2, 2018):

- **Unit Tests (Apache Maven, JaCoCo, SonarQube)**
  - Code Coverage Percentage
- **Static Code Analysis (SonarQube, Sonatype)**
  - Code Quality
  - Refactor Opportunities
  - Possible Injection Errors
  - Vulnerability Scan
- **Exchange-to-Exchange Tests (Selenium, Protractor)**
  - Automated Integration Tests
  - Ensure Application is Operational
  - Stress Testing
- **Penetration Testing (Open Web Application Security Project Zed Attack Proxy)**

Catapult/ANTS PMO further pursues a continuous risk management approach by providing the information assurance team with direct access to live tools including SonarQube and Twistlock. These tools help the PMO track and verify tasks in its plans of action and milestones, and provide an automated lockdown that addresses 240 of 242 Security Technical Implementation Guide (STIG) mandated controls (L. T. Garciga & D. Lederman, personal communication, July 2, 2018). The remaining two STIG controls must be verified manually.

RCAS PMO uses a broad variety of tools in its Secure DevOps framework, supporting a variety of heterogeneous architectures. Key tools for the RCAS portfolio include CodedUI, Specflow, Cucumber, Chef, Artifactory, C#, VisualStudio, PowerShell, Selenium, NUnit, Oracle, and Moq.
DCGS-N uses JMeter for functional testing and Stallion for automated message and data file injection, and the other programs use similar tools and static code analysis.

**Sprint-level testing.** Certain types of testing are more appropriate for the end of the sprint. The Space and Naval Warfare Systems Command (SPAWAR) performed additional testing during integration for JMS and DCGS-N. DCGS-N schedules cumulative functional testing before each Capability Drop and before each Fleet Capability Release. SPAWAR executes an integration test for each Fleet Capability Release before the system goes through a developmental testing/operational testing (DT/OT) cycle.

GCSS-J PMO performs developer testing at the sprint level. When sprint drops are integrated, the government does release testing, UATs, and system acceptance tests, which mark the formal end of government acceptance testing for the release.

RCAS PMO performs all testing within each sprint so that each backlog item is a potentially shippable microrelease. This testing includes functional and nonfunctional testing; unit and integration testing; performance testing; vulnerability testing; acceptance testing; white-box and black-box testing; and automated unit, smoke, regression, and vulnerability testing.

"**When government testers collaborate with the development team, the sharing of test plans, test cases, test data, and defects enhances testing and future software development.**"

**User acceptance testing.** Government UAT has different meanings among the programs, but usually it involves the first use by actual users in a government-led event. Release teams generally consist of the PMO, developer, government T&E, and user community personnel. Oftentimes, members of these development teams are drawn from organizations outside the PMO. The user community is a key component of the release team for programs using Agile development methodologies. Government UAT is generally performed at the end of a release, but earlier government test involvement during daily, sprint, and integration testing can add value to the effort.
GCSS-J’s test team receives operational mission threads from each command and incorporates them into testing. A team comprising Joint Interoperability Test Command (JITC) OT&E personnel and DISA DT&E personnel conducts an integrated SQT at the end of a release. If successful, this approach avoids a dedicated release OT&E. The DISA test team considers this working relationship with JITC and the stakeholders to be its greatest success.

At Vandenberg AFB, CA, JMS PMO hosts a program instance that can gain feedback from operators without their having to leave home stations. Acquiring dedicated operators for a dedicated test event is difficult, but this approach allows early user feedback. Additionally, JMS PMO tasks the 45th TS to provide its government DT, and the PMO prefers a smaller team dedicated to its effort, rather than the larger team that is shared among other 45th TS projects. For its current service pack, JMS PMO is working to involve the 45th TS more during contractor and integrator testing. The PMO is also involved with the Air Force OT&E Center, which provided an OT&E perspective earlier in the project.

For RCAS PMO, the user community must accept new features and releases during sprints for them to be considered complete. Catapult/ANTS PMO employs users as a test group in UATs, and further monitors user feedback after deployment through a series of engagements to monitor risks associated with community adoptions and mission effectiveness.

**Metrics.** Programs monitor their testing and development activities using manual and automated tools to track defects and test completion metrics, depending on the level of complexity and maturity of their solutions.

RCAS PMO tracks the productivity, capacity, and efficiency of their Agile Solution Factory (ASF) using story point analysis, which is conducted after each RCAS quarterly release to measure performance metrics across multiple levels from scrum teams to the enterprise level. The performance analysis delivers transparency and predictability of the ASF to the PMO. Other performance analysis approaches used in parallel include sustainment analysis, Agile earned value management, story point defect analysis to measure release quality, delivery velocity, automation ROI analysis, and help desk ticket analysis. A combination of these approaches allows RCAS PMO to continuously optimize the performance and efficiency of their program.

JMS PMO tracks findings using the deficiency report processes, and JIDO uses tools such as JaCoCo to provide test case code coverage metrics. DevOps organizations monitor their processes with automated tools.
Best Practices

The observed best practices are specific to each program; however, they can be generalized to provide a guide to improve software acquisition for other programs.

Develop a Tailored Approach with Continuous Delivery

Each PMO selected an SDLC methodology that best met its needs; no clear linkage emerged between the program groupings and their SDLC method. Table 4 shows the program groups and SDLCs used for each program.

<table>
<thead>
<tr>
<th>Program</th>
<th>Group</th>
<th>SDLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMS</td>
<td>SoS</td>
<td>Scrum</td>
</tr>
<tr>
<td>DCGS-N</td>
<td>SoS</td>
<td>SAFe</td>
</tr>
<tr>
<td>GCSS-J</td>
<td>Web Applications</td>
<td>Scrum</td>
</tr>
<tr>
<td>RCAS</td>
<td>Web Applications</td>
<td>SAFe, Secure DevOps</td>
</tr>
<tr>
<td>Catapult/ANTS</td>
<td>Web Applications</td>
<td>Secure DevOps</td>
</tr>
</tbody>
</table>

Drawing from the discussion in the Background section, the web application suites (GCSS-J, RCAS, and Catapult/ANTS) are more suitable for DevOps methodologies, as they are loosely coupled, allowing the development and deployment of small pieces over time. SoS complexity (DCGS-N, JMS) does not preclude the use of DevOps, but makes implementation difficult—both PMOs noted difficulty integrating input from multiple vendors. Deployment timing is also a factor; DCGS-N may deploy to a ship only once a year, which limits the ROI from a continuous deployment approach.

PMOs should also consider tailored approaches based on their requirements and time frames. Catapult/ANTS PMO uses different Agile methodologies based on requirements characteristics. For well-defined requirements with minimal user input, the PMO uses Agile Kanban SDLC methodologies to deploy a solution in as little as 2 days. For vague requirements or those that involve substantial user interaction, a traditional Agile Scrum approach is used. Wagner and Ford (2018), in their journal article entitled, *DoD Applications of Agile Software Development Methods*, also reach a
comparable conclusion. “No single methodology is suited for all software projects,” and “Developers, managers, and stakeholders (especially operational stakeholders) should jointly choose an Agile method based on project type, project size, the amount of documentation required, project risk, and the experience level of the development team” (Wagner & Ford, 2018, p. 52).

Multiple PMOs noted the merits of continuous delivery models, and JMS and GCSS-J PMOs are working to shift their processes toward continuous delivery models. JMS PMO specifically sees an advantage toward smaller pieces of capability that are effective by the user community. PMOs such as RCAS and Catapult/ANTS have implemented continuous delivery over the last 5 years in their Secure DevOps implementation. The Defense Science Board makes a similar recommendation that PMOs pursue a “Continuous Iterative Development” approach, with an emphasis on identifying Minimum Viable Product approaches and delegating acquisition authority to the PMO (DoD, 2018).

**Emphasize Obtaining Quality Feedback on Requirements and Solutions**

Agile methodologies stress the day-to-day involvement of the product owner and/or user community. Some PMOs have embedded traditional requirements development organization representatives within their PMO to facilitate rapid exchange of ideas, clarification of requirements, and real-time feedback.

Several PMOs have active requirements development or gathering teams within the PMO. The PMOs use these teams to refine requirements provided by the functional sponsor and/or users. The team also collects new requirements once the program has transitioned into sustainment, although DevOps programs may also practice this approach during development. Best practices are to delegate requirements development and refinement down to the PMO level and maintain a constant connection between the developers and user community.

Catapult/ANTS PMO has dedicated development engineers embedded at user facilities with intelligence analysts at the point of need. For example, personnel may see the value of a new requirement in an analysis ad hoc report and turn it into an enterprise solution for all analysis centers. In another example from Catapult/ANTS, the PMO selected an open-source graph database supported by the community, but after implementation, the open-source community had moved in a different direction with a different database. The PMO readjusted and supported the new database quickly rather than implementing a stale code release.
Invite Developers and Users to Participate in Government Testing

The key for successful DT is to communicate defect information back to the developer as quickly as possible. Although this objective could also be accomplished through a shorter reporting process, having the developer present when the errors occur has been shown to be one of the best methods. Development team members provide a useful resource to the test team during and after government testing. Their knowledge of system design can be compared to user expectations during testing. Developer presence during government DT provides critical feedback to support efficient and effective defect resolution. Additionally, user involvement helps find different types of errors because the users may use the system differently than the developer and/or tester imagined. Government test organizations can provide test jurisdiction with clearly defined processes and procedures in place to keep government DT independent. When government testers collaborate with the development team, the sharing of test plans, test cases, test data, and defects enhances testing and future software development.

In some cases, users cannot participate in a single test event. JMS PMO provides access to a program instance at the users’ home stations since obtaining time from dedicated operators for the test events can be difficult. This approach has consistently provided valuable feedback.
The Return on Investment for Automated Testing Depends on the Program

Several PMOs recommended pursuing automated test tools; however, both RCAS and Catapult/ANTS PMOs note that no single tool works best for automation tools. Automated testing involves a significant investment in automated software testing tools and training the workforce to use them effectively. Over time, automated software testing can reduce the time spent on repetitive regression testing, which is essential to achieve the flow that makes Agile software development effective. Although automated software testing has great benefits, PMOs will likely still need some degree of manual testing.

Each PMO team selects the appropriate tools for its development pipelines and reevaluates as new technology becomes available. Again, program length and type may impact the ROI.

Specifically, the Catapult/ANTS team has internal processes for validating their automated testing, which has resulted in changes to the automation tools that the team uses. They further mentioned that they initially performed manual and automated testing in parallel to build confidence in the automated tools, and noted that the automated tests hastened the process. The program has reached the point where even most user acceptance testing is automated, except for complex tasks. Portions of their user acceptance testing are executed either by recording user system interactions and replaying them, or by using a program to click through tasks.

Government DT can provide a similar process to guide and oversee automated testing. Government testers can guide automated testing by selecting tools, processes such as design of experiments or combinatorial testing, gates for the testing pipeline, and specify the extent of testing when configuring each tool. Once the pipeline is set up, government testers can oversee the process as the product moves through deployment gates to determine whether any issues arise.
Transition to Agile is Not a Quick and Easy Process

The two DevOps programs, RCAS and Catapult/ANTS, both developed their processes over a 3- to 5-year period. When considering Catapult/ANTS, the JIDO Chief Technology Officer remarked on the need for retraining testers, and JIDO’s continued move to Secure DevOps will require “Another 30 percent change in the skillset of the workforce” (Miller, 2018, para. 20). The 45th TS is also investigating focusing their workforce by either retraining its personnel to better support automated testing, or providing specific manual testing solutions. The RCAS PMO noted that the transformation to Agile at an enterprise scale requires a holistic, structured approach across three perspectives: (a) organization and people, (b) process, governance, and method, and (c) tools. Cultural transformations are difficult and require leadership driving the transformation from both the government and the system integrator.

"Agile and DevOps software development takes time and effort to implement, but the ROI from implementing automated procedures, requirements gathering teams, and better collaboration can be of great benefit to most software development programs."

For the RCAS program, both the government and the system integrator supported the continual evolution and transformation of their ASF (O’Brien & Green-Mack, 2018). The system integrator had a structured approach and training to transform to Agile, implement an ASF, and scale the ASF over time across the three perspectives noted previously. The RCAS PMO worked with the system integrator to adapt and enhance these structured approaches to integrate into both the RCAS PMO and RCAS mission space. RCAS PMO noted the benefits of having the system integrator host RCAS development and test environments, which included maximizing ASF performance, minimizing costs, increasing product quality, and cybersecurity. The system integrator can quickly integrate new tools within its cloud-based development test environments to drive better performance.

Initial efforts to implement this recommendation have already started. The Defense Acquisition University has undertaken an effort to create an Agile program management course designed to introduce Agile development precepts to pilot programs. This effort is not individual training, but is intended to instill a cultural change across the program management team.
as a group. This course is being developed to fulfill requirements of the 2018 NDAA (§ 891), which will provide training on “agile or iterative development methods to provide training for personnel implementing and supporting the pilot programs required by Sections 873 and 874 of this Act” (NDAA for Fiscal Year 2018, 2017).

Conclusions

The DoD continues to pursue Agile and DevOps software development methodologies, based on strategic guidance, and the potential for cost and schedule efficiencies. Although not all program types may benefit from these development techniques, the paradigm is shifting, and program managers and testers need to be aware of the circumstances where Agile or DevOps methods could benefit their programs. This study surveyed five acquisition programs in the areas of Agile development, requirements gathering, test planning, and test execution. The analysis summarizes their Agile and DevOps SDLC methodologies and the software development strategies. Each program uses unique management software, such as Jira, to track and manage software development. Requirements gathering and test event detailed planning are explored, and the specific test events and executions are covered with an emphasis on what is different during Agile and DevOps software development. The final section distills the findings into lessons learned and best practices. Agile and DevOps software development takes time and effort to implement, but the ROI from implementing automated procedures, requirements-gathering teams, and better collaboration can be of great benefit to most software development programs.
References


Acknowledgments

The authors are grateful for significant contributions from DASD(DT&E) staff members Chris DeLuca, Leland Hewitt, and Mark Price. Dr. Ferhan Kilical and Susan Gotta provided substantive input to compilation of the manuscript topics and content.
Author Biographies

Mr. Jeremy D. Kramer

is an operations research analyst for Air Mobility Command Test and Evaluation Squadron, Joint Base McGuire-Dix-Lakehurst, New Jersey. Mr. Kramer holds a BS in Mathematics from San Diego State University and an MS in Operations Research from North Carolina State University.

(E-mail address: jeremy.kramer.1@us.af.mil)

Lt Col Torrey J. Wagner, USAF,

is currently a faculty member at the Air Force Institute of Technology (AFIT). Lt Col Wagner was commissioned in 2000 from the University of Minnesota, Minnesota Air Force ROTC, with a BS in Electrical Engineering. His previous assignments included Air Force Research Laboratory (AFRL) section chief; AFRL branch chief; AFRL deputy division chief, and executive officer to the AFRL Commander; director of operations for a 60-member engineering analysis squadron; and as the Commander of an 80-person applied engineering squadron. Lt Col Torrey holds graduate degrees from Loughborough University, UK, and AFIT.

(E-mail: torrey.wagner@us.af.mil)