Continuous Iterative Development and Deployment Practices

Eileen Wrubel, Initiative Lead
Hasan Yasar, Technical Manager
Enduring Software Challenges

**Affordable**
Be Affordable such that the cost of acquisition and operations, despite increased capability, is reduced and predictable

**Trustworthy**
Be Trustworthy in construction, correct in implementation, and resilient in the face of operational uncertainties

**Capable**
Bring Capabilities that make new missions possible or improve the likelihood of success of existing ones

**Timely**
Be Timely so that the cadence of fielding is responsive to and anticipatory of the operational tempo of the warfighter

- Secure DevOps
- Agile/DevOps Automation
- Agile
- DevOps
Decrease Risk to Programs with Continuous Iterative Development and Deployment Practices

Source: Image adapted from Kessel Run, via Dr. Jeff Boleng
CMU SEI Objectives: DoD Outcomes

Automating the software development and acquisition lifecycle

DoD can produce assured software-enabled systems that are agile and responsive to mission

Continuous Iterative Development Practices

Modern Acquisition Lifecycle Practices
Automated Testing and Evaluation
Legacy Integration and Sustainment
Continuous Iterative Development and Deployment Practices

Why Is the DoD Software Landscape Different?
DoD Depends on Software but Does Not Control Development

Software and system complexity is increasing software cost and vulnerability, jeopardizing military capability

- DoD does not produce most of the software it uses, but it must maintain that software
- More and more capability results from software, and it will evolve for the lifetime of a system
- Latent cyber vulnerabilities, those exposed during operations, and those due to underlying dependencies are putting the DoD at risk
- Finding and fixing problems late causes rework and drives up costs
- Software cost overruns are overwhelming program delivery and sustainment

Modern software development and automated tools are critical
Defense Industrial Base and Commercial Companies Are the Main Developers of DoD Software

Commercial companies are widely adopting modern practices

- Use Agile, DevOps, cloud, automated test, continuous integration, and continuous release
- Goal: Ruthlessly automate all aspects of the development cycle

Defense industrial base is not incentivized to adopt modern practices

- Need new contracting and acquisition approaches

Challenges for DoD in adopting modern practices

- Risk acceptance
- Acquisition policy
- Testing
- Absence of DoD software ecosystems
- Slow adoption and enforcement of open standards
- Training
- Long delivery cycles
- Various software types on a single system
Modern Development Practices: Smaller Bites, Manageable Risk, Earlier in the Process

• Take advantage of tailoring allowed in acquisition policy
• Manage schedule in short, set increments
• Build capability in each sprint
• Automate test, integration, and assurance so they become ongoing engineering activities
• Build engaged, cross-functional teams that shift testing “to the left”
• Adopt approaches that fill technology gaps in the lifecycle
• Remember the importance of architecture

Agile, in its various forms, is an approach to software development in a DevOps environment, enabled by modern software factory tooling
Continuous Iterative Development and Deployment Practices

Agile and DevOps Approaches Support the DoD’s Drive to Provide Capability While Controlling Quality, Cost, and Schedule
Motivation for Agile Approaches

- Deliver performance at the speed of relevance

- Streamline rapid, iterative approaches from development to fielding

Source: National Defense Strategy Summary, Jan. 2018

“…advance the capabilities required to restore our overmatch, speed the rate in which we field these advanced capabilities, and improve the overall affordability of our fighting forces weapons systems....”
Hon. Ellen Lord
Under Secretary of Defense for Acquisition and Sustainment
DevOps Principles

DevOps is a set of principles and practices emphasizing collaboration and communication between software development teams and IT operations staff along with acquirers, suppliers, and other stakeholders in the lifecycle of a software system\(^1\)

Four Fundamental Principles

- **Collaboration** between all stakeholders
- **Infrastructure as code (IaC):** assets are versioned, scripted, and shared
- **Automation:** deployment, testing, provisioning, any manual or human-error-prone process
- **Monitoring:** any metric in development or operation that can inform priorities, direction, and policy

\(^1\) IEEE P2675 DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package, and Deployment
Benefits of DevOps

- Reduced errors during deployment
- Reduced time to deploy and resolve discovered errors
- Repeatable steps
- Continuous availability of pipeline and application
- Increased innovation time
- Responsiveness to business needs
- Traceability throughout the application lifecycle
- Increased stability and quality
- Continuous feedback
The DevOps Factory

- Feature to deployment
- Iterative and incremental development
- Automation in every phase of the SDLC
- Continuous feedback
- Metrics and measurement
- Complete engagement with all stakeholders
- Transparency and traceability across the lifecycle
Continuous Iterative Development and Deployment Practices

Real-World Results Using Agile and DevOps Approaches in Government Environments
Orion Independent Verification and Validation: The Challenge

Orion Multi-Purpose Crew Vehicle is NASA’s next human-rated spacecraft

IV&V supports rapid software composition by assessing untrusted components:

• Adds evidence-based assurance that minimizes the overall risk that Orion software will prevent the EM-1 flight from occurring safely and successfully

• Adds assurance that all safety-critical mission events happen as expected or with satisfactory responses to adverse conditions and appropriate protection against undesirable conditions

Orion IV&V previously analyzed entities in their entirety, with some entities not considered risky enough to analyze at all

Orion IV&V updated its flight software risk assessment and plan for what assurance would be added twice a year

• GNC: high
• Electrical: medium
• Video: low

Source: Briefing to SEI Agile Colloquium and TRISMAC by our collaborators, NASA Orion IV&V and Engility
Orion IV&V: Goals and Results

Switch to Capabilities/Follow the Risk

• Orion IV&V was uncomfortable with the residual risk that would have resulted from the previous approach

• IV&V decided to analyze the mission capabilities with highest risk regardless of their association to the entities

• Evaluating risk more dynamically and more frequently matches the changing risk landscape of the Orion Program

Orion IV&V changed delivery cadence from months to weeks

Following the risk focuses the Orion IV&V team’s effort on areas of highest concern

Stakeholders were happy with the changes: “IV&V's capability-based approach and ‘follow the risk’ strategy allows them to have relevant opinions on the most difficult issues the program is facing.”
The Agile Program Office: The Challenge

**Vision:** Demonstrate that Agile methodologies can be successfully implemented in a program-management environment

**Get real-life experience with Agile**
Government needs experience with
  • Agile processes
  • Agile tools

**Why?**
  • More practical knowledge of the art of the possible when collaborating with Agile contractors
  • Greater understanding for contractors when they have problems

**Take advantage of Lean principles**
Leverage small batches:
  • Increase flow
  • Improve productivity
  • Deliver quality

Visualize and manage work in progress (WIP):
  • WIP limits
  • WIP policies
  • See the work being done
# The Agile Program Office: Goals and Results

<table>
<thead>
<tr>
<th>Program Goals</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make timely and high-quality decisions</td>
<td>“We recently completed a $600K+ tech eval in only 17 days. This would normally take 90 days or more.”</td>
</tr>
<tr>
<td>Reliably repeat success and learn from failure</td>
<td>“We reconfigured the vehicle with added complexity in only 3 months. This would normally take a year to 18 months.” “We also demonstrated agility when we moved up the delivery date by 14 days to better accommodate another mission.”</td>
</tr>
<tr>
<td>Improve visibility of workflow; centralize data and info using cloud-based platform (Jira/Confluence)</td>
<td>“Our cloud-based tracking tool helps us to corral our work into manageable tasks. The filters help us manage the flow of work, which, in turn, accelerate the velocity…”</td>
</tr>
<tr>
<td>Understand the work better (flow, bottlenecks, anomalies)</td>
<td></td>
</tr>
<tr>
<td>Improve stakeholder management (transparency)</td>
<td>“With the customization in the tool, we’ve created a sort of bespoke platform that feeds us data how we want it, when we want it.”</td>
</tr>
</tbody>
</table>
JIDO SecDevOps: Pipeline → ATO in a Day!
Security Requires Automation with IaC, CI, and CD

- Security requirements and traceability
  - Risk Management Framework: (1) categorize, (2) select controls, (3) implement, (4) assess, (5) authorize, (6) monitor
- Code review and static analysis
- Automated security testing and verification
- Automated dependency vulnerability analysis
- Immutable system, infrastructure (re-)provisioning
SecDevOps

Security from inception to deployment and improvement with every delivery
Continuous Iterative Development and Deployment Practices

Ongoing Work: Applying Agile and DevOps Principles to Modernize Legacy Systems and Deliver Trusted Software Faster
Addressing Challenges in Research and Client Program Engagement

- More and more metrics in new contexts
- SecDevOps in safety-critical and hardware systems
- Organizational agility for programs
- Migration of legacy systems

SecDevOps in safety-critical and hardware systems

Migration of legacy systems
Research: Migration of Legacy Systems to Cloud Environment

Software engineering centers, software maintenance groups, and other sustainment organizations want to realize the benefits of IaC. They must first recover the technical baseline for the deployment, but:

- IaC doesn’t exist for legacy systems
- government has no data rights to ask for contractor deployment scripts
- often the only authoritative artifact is an instance of the running system

Can the deployment structure be automatically recovered from an instance of the running system?
Solution with IaC

Automatically recover a deployment model from a running system and generate IaC scripts from a model.

Model-based deployment enables automation:
- Port scripts to new tools or IaaS
- Analyze the model against design rules
- Transform the model (moving target defense)
Research: How to Build Trusted Systems

**Problem:** Modern software development challenges developers to build **trusted systems** that include increasing numbers of **untrusted components**

**Solution:** Component **scorecards** based on **project health** and aggregated **quality attribute indicators** will enable rapid delivery of software capability with greater developer confidence and 10% reduction in downstream rework

**Approach:** Apply existing automated analyses (e.g., code and repository analyses) mapping to common indicators from DoD projects, develop several candidate health and quality attribute indicators, and validate with open-source corpus and relevant stakeholders
## Score Card on Selected Components

### Project Health Indicators

<table>
<thead>
<tr>
<th>Properties</th>
<th>Indicator</th>
<th>Measurement approach or tool (example)</th>
<th>Ease of collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community health</td>
<td># developers</td>
<td>code-maat</td>
<td>Simple, with project history</td>
</tr>
<tr>
<td></td>
<td>Commit frequency</td>
<td>code-maat</td>
<td>Simple, with repo access</td>
</tr>
<tr>
<td></td>
<td>Bug frequency</td>
<td>code-maat</td>
<td>Simple, with repo access</td>
</tr>
<tr>
<td>Codebase</td>
<td>LOC</td>
<td>cloc</td>
<td>Simple, needs source</td>
</tr>
<tr>
<td></td>
<td>License</td>
<td>gh-license</td>
<td>Simple</td>
</tr>
<tr>
<td>Team</td>
<td>Countries of origin</td>
<td>LinkedIn/social metrics</td>
<td>Moderate</td>
</tr>
<tr>
<td></td>
<td>Previous experience</td>
<td>Social metrics</td>
<td>Hard</td>
</tr>
<tr>
<td>(Others)</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Quality Attribute Indicators

<table>
<thead>
<tr>
<th>Qualities</th>
<th>Indicator</th>
<th>Measurement approach or tool (example)</th>
<th>Ease of collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintainability</td>
<td># imported libraries</td>
<td>cvs_analy, sonatype</td>
<td>Simple</td>
</tr>
<tr>
<td>Level of coupling</td>
<td>DV8</td>
<td></td>
<td>Simple</td>
</tr>
<tr>
<td>Architecture flaws</td>
<td>DV8</td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>Performance</td>
<td>CPU load, peak</td>
<td>gprof</td>
<td>Moderate</td>
</tr>
<tr>
<td></td>
<td>Disk access</td>
<td>gprof</td>
<td>Simple</td>
</tr>
<tr>
<td></td>
<td>Memory use</td>
<td>valgrind</td>
<td>Moderate</td>
</tr>
<tr>
<td>Security</td>
<td># CVE violations</td>
<td>fortify, coverity, SCALe</td>
<td>Moderate</td>
</tr>
<tr>
<td>(Others)</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Continuous Iterative Development and Deployment Practices

Transition: Now and in the Future
SEI Agile and SecDevOps Resources

• Learn more:
  – webinars, courses, podcasts
  – *Agile in Government* booklet series
  – conference presentations: RSA, AppSec, Velocity, AgileDev, ARES, IEEE, O’Reilly
  – blogs at SEI Insights, dZone, and DevOps.com

• Join our community: 240+ member Agile Collaboration Group with 80+ organizations

• Attend our conferences and workshops:
  – Annual Agile Colloquium, Annual ADAPT Agile in Government Summit
  – AllDayDevOps, DevSecOpsDays
  – Secure DevOps, DevOps for Managers, DevOps in Practice

• Use our guidance:
  – partnered with DAU on Agile curriculum and course assets
  – GAO guidance for Agile programs uses SEI Agile Readiness & Fit Analysis
  – IEEE 2675 DevOps standard
Research Review 2018

SEI Client Program: Distributed Program Office Delivering (Non-Software) Services Now Tasked with New Function

How the SEI Helps This Program

Readiness & risk assessment: Considerations for Agile and DevOps adoption/transformation pilots

Training needs assessment: Review knowledge-specific needs for diverse program stakeholders

Expert coaching
- Basics through larger scale adoption (custom and COTS-based)
- agile methods for program teams, executive leaders
- Role/function-specific
- DevOps pipeline
- Identify best practices
- Process adoption support
- Change management
- Replication of success

Metrics development: Defining, implementing, and instrumenting for oversight

Program start-up workshops: Strategic goal-setting workshops

Devising workable hybrid options: Reconciling future approach with new and existing program policies and practices

Expertise in DevOps practices, deployment pipeline

Benefits

Access broad expertise in acquisition, Agile and traditional methods, and rich library of tools, methods, and courseware supporting Agile transformation in DoD

Apply lessons learned from other programs and research to avoid known barriers and pitfalls, take advantage of improved flow with DevOps/Agile

Apply unique and practical perspective, integrated and incremental development environment model

Enhance program-unique practices with practices that contribute to improved execution for other government agencies

Employ Trusted Broker
SEI Client Program: Large Program Adopting Agile to Better Manage Dynamic, Critical Changes to User Needs

How the SEI Helps This Program

<table>
<thead>
<tr>
<th>How the SEI Helps This Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Readiness &amp; risk assessment: Considerations for Agile and SAFe adoption/transformation</td>
<td></td>
</tr>
<tr>
<td>Training needs assessment: Review knowledge-specific needs for diverse program stakeholders</td>
<td></td>
</tr>
<tr>
<td>Expert coaching</td>
<td>• Basics through larger scale adoption (custom and COTS-based) • Agile methods for program teams, executive leaders • Role/function-specific • DevOps pipeline • Process adoption support • Change management • Replication of success</td>
</tr>
<tr>
<td>Metrics development: Defining, implementing, and instrumenting for oversight</td>
<td></td>
</tr>
<tr>
<td>Custom training</td>
<td>• Basics through role-specific support • DevOps process and practices • Process adoption • Workshop program-specific opportunities and barriers • Hands-on DevOps in practice workshop</td>
</tr>
<tr>
<td>Identifying workable hybrid options: Reconciling future approach with new and existing frameworks, program processes, and procedures</td>
<td></td>
</tr>
<tr>
<td>Expertise in DevOps practices, deployment pipeline</td>
<td></td>
</tr>
<tr>
<td>Benefits</td>
<td></td>
</tr>
<tr>
<td>Access broad expertise in acquisition, Agile and traditional methods, and rich library of tools, methods, and courseware supporting Agile/DevOps transformation in DoD</td>
<td></td>
</tr>
<tr>
<td>Apply lessons learned from other programs and research to avoid known barriers and pitfalls</td>
<td></td>
</tr>
<tr>
<td>Apply unique and practical perspective, innovative solutions</td>
<td></td>
</tr>
<tr>
<td>Enhance program-unique practices with known successful Agile-related practices used by other government programs and agencies</td>
<td></td>
</tr>
<tr>
<td>Employ Trusted Broker</td>
<td></td>
</tr>
</tbody>
</table>

Research Review 2018

© 2018 Carnegie Mellon University

[GOVERN] PROVE POTENTIAL IMPROVED PRACTICES

![Diagram of Traditional Acquisition Framework and Agile and DevOps Principles/Practices from Successful Commercial Use]
For More Information

Agile and DevOps: https://www.sei.cmu.edu/go/agile
https://www.sei.cmu.edu/go/devops

DevOps Blog: https://insights.sei.cmu.edu/devops

Webinars: https://www.sei.cmu.edu/publications/webinars/index.cfm

Podcasts: https://www.sei.cmu.edu/publications/podcasts/index.cfm

YouTube: https://www.youtube.com/user/TheSEICMU