Where Our National Security Begins...
NRO
Industry Advisory Working Group

Working Session
25 June 2019
Welcome & Introductions

“Snapshots”

“S2P Corner” & “C2S Corner”

Ann Waynik floor time

Action Team discussions

Government Perspective

Open Dialog

No-Host Social
“Snapshots”

Events

- GEOINT 2019: 2-5 June
- Amazon Public Sector Summit: 11-12 June
- Azure + AI Conference: 11-13 June
- IAWG “Peer Review” session: 30 May

New Members

QUINTSPO briefing: 30 July, 1000-1100. Topics:

- Terms of Reference (25 min)
- Agile Acquisition / STC (25 min)
- Previous data call
  - Unintended consequences of AF/IF
  - Getting industry feedback on FGA architecture
“C2S Corner”

Latest & Greatest...

Topics & Issues Discussion
“S2P Corner”

Latest & Greatest...

Topics & Issues Discussion

Explore on CWAN/JWICS @ https://jportal.S2P.proj.nro.ic.gov
NRO IAWG
Action Teams & Topics

• Agile & DevSecOps 2019
  • Terms of Reference
  • Scaling DevSecOps Across Programs

• Agile Acquisition Strategies / Accelerating Speed to Capability

• Hybrid Cloud Business Model
1. Terms of Reference
2. Scaling Agile and DevSecOps across GED Program Offices
3. Contracting strategies, structures, and incentives
4. Training gaps, standardization, and program-specific implementations
5. ROI expectations and perceptions
6. E2E Mission Thread Closure in an Agile & DevSecOps world
Agile & DevSecOps 2019:

Terms of Reference

Matthew Reider (Team Lead)

Ken Laskey
Jay Eward
Rob Manogue

John Farrell
Scott Lawler
Agile & DevSecOps 2019: Terms of Reference Team

Challenge

Different government, industry, and vendors use different terms of reference to describe similar things, or use same terms of reference to describe different things.

Deliverable

A document containing Terms of Reference and the source information in order to align government and industry.

Approach

1. Identify commonly used agile and DevSecOps terms across GED and NRO
2. Identify terms that have most divergent or misunderstood definitions
3. Identify major commercial and agile government sources of definitions to create matrix
4. Highlight those definitions or parts of definitions that enhance common understanding
(U) GED's Requirements Flow Taxonomy

**SRs**

**Solution Level Epics**
Definition: Multi-increment, Multi-functional useful outcome that can be decomposed and allocated into multiple, single increment deliveries.
Example:
- OOS
- SRs: within Epic process data to generate products, accept support data for processing.

**Thread**

**Feature Level Epics**
Definition: Defines outcomes to a Project (e.g. OOS for M2C2) that delivers in a single increment (e.g. PI#1)
Example:
- GPF creates products IAW...

**Project**

**Program Roadmap**
Near term Project Level incremental features that support Solution-Level Roadmap

**Developer**

**Stories**
Definition: Decomposed functions into multiple iterations/sprints to accomplish increment outcome
- Typically will be over 2-4 week sprint/iteration timelines

**Software Development Sprints/Iterations**

Graphic is UNCLASSIFIED
### Agile Process Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint</td>
<td>A time-boxed period of work during which a specific amount of work is completed.</td>
</tr>
<tr>
<td>Release</td>
<td>The version of a product or service that is released.</td>
</tr>
<tr>
<td>User Story</td>
<td>A brief description of a user's needs or requirements.</td>
</tr>
<tr>
<td>Story Point</td>
<td>A specific aspect of a user story that can be implemented in a single sprint.</td>
</tr>
<tr>
<td>Story Point Normalization</td>
<td>The process of refining a story to ensure it is clear and concise.</td>
</tr>
<tr>
<td>ScrumMaster</td>
<td>The person responsible for facilitating the Scrum process.</td>
</tr>
<tr>
<td>Backlog Grooming/Refinement</td>
<td>The process of refining the backlog to ensure it is clear and prioritized.</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>An approach to software development that emphasizes testability.</td>
</tr>
<tr>
<td>Behavior Driven Development</td>
<td>An approach to software development that focuses on behavior over implementation.</td>
</tr>
</tbody>
</table>

### Organizational Terms

- Culture
- Mindset
- Policy
- Risk Averse

### Potential Definition Sources

- Scaled Agile Framework for enterprise (SAFe)
- Agile Alliance “Agile Glossary”
- SmartSheet “Ultimate Agile Dictionary”
- 18F Agile Lexicon

See Spreadsheet for Definitions
# Agile & DevSecOps 2019: Scaling DevSecOps across Programs

**Marc Kriz (Team Lead)**

- Shawn Lucas
- Themba Hinke
- John Jeremiah

- Pete Epstein
- Sam Stollar
- Kevin Chasse

- Eric Amberge
- Joel Doyle

---

*USGIF*
Agile DevSecOps document/outbrief divided into 4 sections:

1. Introduction
2. Aspects of DevSecOps and their definitions
3. Key Enablers of Agile DevSecOps
4. Alignment with GED practices
Why is GED transitioning to an Agile culture and moving towards DevSecOps?

Establishing Secure Application Development capabilities will:

- Deliver applications in a secure and timely manner increasing the ability for GED to provide a competitive advantage to downstream customers
- Bake-in and enforce cybersecurity functions and policy from inception through operations
- Enhance enterprise visibility of development activities and reduce accreditation timelines
- Ensure seamless application portability across enterprise and mission program environments
- Drive GED transformation to Agile and Lean Software Development and Delivery
Alignment with GED practices

• **Manage** - Mentality: leadership and a view, What metrics will demonstrate to managers that the “mechanics of management” are different?

• **Expect what you inspect.** Mechanics don’t matter if no one cares about them.

• **Additional Best Practice to consider: Create 3-6 metrics**
  - Document historical benchmarks of progress and leading indicators of transformation.
  - If workforce knows what is being tracked, they will change behavior accordingly.
GED scaling Agile and DevSecOps across Programs

The ability for teams of developers to work seamlessly across disparate network enclaves, with all artifacts, issues and code intact as if they were working on the
GED scaling Agile and DevSecOps across Programs

Enablers -

- Willingness to consider working differently - “digitally dissolving silo’s”
- Senior leadership AND middle management top cover for change
- Technology that enables all team members to be on the same page, with collaboration, visibility and metrics crossing all boundaries.
- Management from across the teams that agree on a common set of mission goals and outcomes.
GED scaling Agile and DevSecOps across Programs

Suggested Best Practice #1 -

- Run a Proof of Concept (POC), similar to an Open Source Software Project (see Commercial Open Source Software, or COSS as an example business model) whereby you have multiple teams, across enclaves, contributing code to a core application (or project) being managed by a core team that is the “author” or owner of the app or project.
GED scaling Agile and DevSecOps across Programs

Suggested Best Practice #2 -

- Pick a small, pilot project that incorporates multiple teams, across programs, or network enclaves (ex. Low to high), and agree on a common management strategy to be over the entire project, with authority that crosses teams.
Suggested Best Practice #3 -

- Pick a technology that enables cross domain DevSecOps and standardize on approach and agile DevSecOps methodology across the teams with the ability to easily export a project, with all of its artifacts and then import that project across domains or enclaves.
Suggested Best Practice #4 -

- Agree on the workflow and expected output, to include the handoffs between the teams. (see COSS business model for examples)

- COSS Business Model Definition Link:
GED scaling Agile and DevSecOps across Programs

Suggested Best Practice #5 -

- Measure each team on results, with incentives and recognition for those whose output (code contributions), velocity moves the mission, or core project, forward faster.

*Cycle time compression may be the most underestimated force in determining winners & losers in tech.* - Marc Andreessen
Action Team:

*Hybrid Cloud Adoption & Business Model*

Chris Arroyo (Team Lead)

Al Stewart   Robert Shelton   Keith Barber   Tim Stewart
Problem Statement
As the next generation of C2S, C2E will present the NRO with an opportunity to leverage a cloud agnostic environment with multiple options. Without a good understanding of how to adopt cloud agnostic model that works for NRO, the agency runs the risk of not realizing all the potential benefits. Failure to fully understand all the challenges involved and lacking a strategy for adoption will put the mission unnecessarily at risk. Additionally, lack of clarity and understanding from the government will inhibit industry's ability to meet the needs of the customer.
Hybrid Cloud Action Team

Objective

• Create a recommended framework to consider the impact of various cloud adoption scenarios on the NRO - including decision authority, business model impacts, and exit criteria.

Challenges

• What doesn't work today that could impact how the customer adopts a multi-cloud approach?

• What is your enterprise service approach? What is your data strategy? How are you servicing content to your end stream users?

Tasks

• How can the NRO utilize multiple clouds and what are our recommendations?

• Identify differences in business models

• Reconcile cloud contract language and terms and conditions across contracts

• Consideration of who decides which cloud(s) a program uses and in what proportion?
Proposed Framework

1. Identify Approach
   - Single Cloud?
   - Multiple Cloud?
     - Do you shop around? Are there performance metrics to ascertain the quality of one approach over another?
   - No Cloud?
     - Sunk costs of existing infrastructure...

2. Decision Authority
   - Programs? Directorate?
   - Criteria?

3. Business Model Impact
   - Recognition that different clouds may offer different models and costs, including:
     - Data migration
     - Integration and testing
     - Consultants

4. Exit Plan?
Action Team:
Accelerating Speed to Capability (STC)

Ben Chicoski (Team Lead)
Joe Chioda
Mike Moran
Scott Lawler
Ron Alford
Pete Epstein
Eric Viglione

Ben Chicoski
bchicoski@cloudbees.com
202.746.1124
**Speed to Capability**

**Purpose / Overview**

**Our Task**

- Identify process, development, and programmatic sources of delay to delivery of capabilities
- Make recommendations for concrete action. Operating principles:
  - Prioritize by need / impact
  - Be realistic – which needles can be moved?
  - Cite real-world examples from which to learn
- Iterate and refine based on Government-Industry feedback

**Desired Outcomes**

- Rapidly get new capabilities on contract, and then...
- Rapidly deliver approved, functional, relevant systems into users’ hands

**Questions to Address**

- How to enable / motivate STC
- How to pay for STC
- How to measure STC
**Speed to Capability**

**Purpose / Overview, cont.**

**Current Landscape**
- Lack of incentives for prime contractors to realize faster delivery cycles
- Change processes biased toward status quo
- Increasing complexity of acquisitions in areas like IT causing skills gap in acquisition workforce
- Pilots that don’t scale

**Summary Observations**
- This is a giant Lean Six Sigma problem: increase throughput, increase predictability, minimize variability
- Constraint: Requirements variability and evolution over shorter time cycles complicate the problem
- Shared services and segmented acquisitions reduce timelines at cost of increased integration complexity
- PREDICTABILITY is valued by government and industry alike
  - Acquisition timing
  - Procurement content
  - Clearances
  - OCI rules
  - Cost evaluation methods
STC members voted on which ideas are most worthy of presenting to GED leadership, based on these decision criteria:

A) Relevance to GED
B) Likelihood of having a significant positive impact
C) Whether GED is willing & able to take action on it
Now:
1) Review the down-selected list
2) Vote for five finalist ideas. Ask yourself:
   • Does it fit within the STC remit?
   • How well does it meet the decision criteria above?
   • Do we have sufficient supporting evidence (e.g., an exemplar program) to bolster our conclusions?

Next:
3) Assign a lead person to each of the finalists
   • Build out 2–3 slides per idea
4) Determine how best to portray the ideas in a briefing
Outcome-based contract that emphasizes speed and functionality without directing implementation (e.g., SOO vice SOW).

- Goal: Get approved, functional, relevant systems into users’ hands.
- Define the vision of why, who for, what; but don’t dictate the how
- Define value and then measure it (e.g., level of user satisfaction, net promoter score, ease of use, reliability, etc). Defining value as user satisfaction also enables adapting to changes to programs mid-flight because the work is pegged to the users’ real-time priorities.

Right-size the compliance burden to match the work being procured
Incentivize risk taking and fast failure on SMALL THINGS to LEARN FAST. Start small, be iterative, and build on success – or terminate quickly

Do away with CDRLs

Avoid complex RFPs with long planning phases with deliverables and milestones and fixed budgets. When we write RFPs, we assume we know what we need and know exactly what the solution is. This doesn’t allow for us to learn and adopt new ideas along the way.

- This is NOT scope creep but proper agile scope management

Start with Mission CONOPS and Capabilities as “requirements” then poll industry for solutions. Selected solutions immediately proceed to sole source awards to refine/deploy solution.

Involve end users up front to define the “what” not the “how”

Better communications pre-award: open, iterative, with the right people involved (for govt, that means end users; for industry, that means line-of-business management).

Concept of Minimal Viable Product as a strategy to increase speed of delivery to the warfighter based on the user’s highest needs

Contract structure (e.g., CLINs for innovation) that allows scope changes in flight
Government Perspective
Action Tracker (needs updating)

- Co-Chairs: Establish recurring engagement at GED “QUINT SPO” meeting
- Agile & DevOps Action Team Part 2: Form & Kickoff
  - Talent: Provide GED leadership with “DevOps Starter Kit” info for clearance sponsorship and IR&D instances on C2S/S2P
  - IAWG data call: examples of AF/IF criteria that created dis-incentives
  - FGA: Identify options for providing industry-wide feedback on FGA architecture in format more useful and open than RFIs
    - Get membership access to FGA 2025 architecture (government stated it was ‘released’)
    - Consider Action team and/or RID to engage
  - STC: Engage GED Contracts regarding “Contracting Performance” aspects and how industry would propose assessing them. What metrics?
  - Potential Action Teams:
    - FGA Integration Model & Cross-Segment/Cross-Program DevOps
    - STC or new team address Waterfall TTO to DevOps Continuous Delivery & Integration
Open Dialog

Additional Topics for Consideration

Actions & Next Steps

No-Host Social
NRO IAWG Contact Information

• Nick Buck: nick@buckgroup.net (703) 801-3405
• Ann Waynik: Ann.Waynik@tenica.biz (571)-376-5641
• Mike Moran: mmoran07@peraton.com (571) 524-1184

USGIF coordination:
• Shai Sobrino: shai.sobrino@usgif.org (571) 392-7205
BACKUP
## Speed to Capability
### Observations & Ideas

#### Requirements

**“Requirements Lock” Observation**
- Inhibits ability for programs to weave in new capabilities

**Contributing Factor**
- SOWs that “bake out” innovation or don’t articulate means to innovate (e.g., need to use study CLIN)

**Ideas**
- Dynamic Reqts Management
- Involve end users up front to define the “what” not the “how”
- Expand use of IDIQs

#### Acquisition/RFP

**“Solicitation Bloat” Observation**
- Drives away qualified performers, creates extra work – on both sides – without necessarily providing benefit

**Contributing Factor**
- Path of least resistance and lowest risk: “Include everything”

**Ideas**
- Right-size compliance documentation to effort size
- Involve security teams in the solicitation creation
- More two-way exchanges

#### Contracting

**No Contracting Metrics Observation**
- Lack of specific, shared objectives for improving contract(ing) performance. Lack of concrete actions means status quo wins

**Contributing Factor**
- Dearth of metrics for assessing quality of docs, RFPs, and contracting timelines

**Ideas**
- Measure against STC metrics based on industry standards and tailored to program profiles. [See template]

#### Development & Tech Transfer

**Waterfall Culture Observation**
- Inconsistent capability delivery chain and clunky transition to operations

**Contributing Factor**
- Inertia within current hybrid infrastructure and lack of end user involvement

**Idea**
- Create nexus where technologist, operator/analyst, MSI, and acquirer can ID and advance solutions immediately, then document “requirements”

---

**Goal:** increase throughput, decrease acquisition variances
## STC Metrics – Template

**Presumed:** STC is inversely proportional to program size

There might be other relevant categories besides size (e.g., environment – legacy, modern, hybrid).

### Time-Based Metrics

Plot each program’s Actual performance relative to its defined Threshold / Objective

### Speed to Capability

<table>
<thead>
<tr>
<th>Capability</th>
<th>Requirement-to-Award</th>
<th>Award-to-First-Capability</th>
<th>Government Performance</th>
<th>Industry Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Program</td>
<td>Actual Threshold Objective Benchmark</td>
<td>Actual Threshold Objective Benchmark</td>
<td>Speed to Award</td>
</tr>
<tr>
<td>$$$</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$$</td>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$$</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$$</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$$</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>e</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>h</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1st graphic represents Requirement-to-Award (aka government)
2nd represents Award-to-First-Capability (aka industry)

IAWG might offer an industry benchmark for the class of capability category

Track actual performance against predetermined threshold and objective specific to that program
While DevSecOps in principle is technology agnostic, a number of modern technologies have emerged that can be enablers to make agility and devsecops easier to adopt. To be clear, the principles of DevSecOps and agility are independent of specific technologies and any team can adopt the principles and goals of rapid delivery of secure and reliable software. There are several technologies we consider enablers:

- **Cloud** - Cloud technologies enable rapid deployment and scaling of infrastructure as needed.

- **Containerization** - Containers abstract the underlying hardware and operating system, making it easy to have resilient environments.

- **Container Management/Orchestration** - Container Management / Orchestration makes it easy to deploy, monitor, scale, and manage the infrastructure of multiple containers which support a given system.
Key Enabler of Agile DevSecOps: Automation

- **Build and Integration** - Streamlining the tasks of building, compiling and assembling source code and related libraries into a working application.

- **Testing** - Enable repeatable, accurate, and reliable testing of application functionality to ensure that application quality standards are maintained.

- **Security Scanning** - Evaluate source code, infrastructure, and running applications to identify potential security vulnerabilities.

- **Configuration** - Defining the application configuration as a consistent and repeatable set of steps, reducing the risk of manual errors which can introduce performance and/or security issues.

- **Deployment** - Enable dynamic and parallelized deployment of applications to support incremental roll outs, feature flags, and other techniques to manage the deployment process and mitigate risks.
DoD Enterprise DevSecOps Platform
(Software Factory)
DAU Presentation

DRAFT v1.3 – UNCLASSIFIED – PRE-DECISIONAL
From Waterfall to DevSecOps
Leadership can enable the adoption of DevSecOps by bringing its ENTIRE stack as a platform and by leveraging DevSecOps solutions.

It is imperative not to “select” a limited option of tools. The key of microservices and containers is to be able to use the best solution to achieve the outcome desired. We should not limit options to the extend possible.

Need to establish baseline requirements / thresholds for cybersecurity, test coverage, test results, documentation. This shouldn’t be reinvented per office but global to DoD as a standard practice to facilitate adoption.

Once those baseline requirements are set, OCIO can provide CI/CD Platform’ stack with embedded security as a side-car container and provide pre-ATOs for systems using the Platform stack and automatically integrating the OCIO security baseline requirements.

Bringing the entire stack as a Platform as a Service is key to avoid that each office reinvents the wheel and builds their own baseline requirements.

Understand that Failing is a GOOD thing! It is part of learning. It allows us to understand what works and what doesn’t - AS LONG AS we leverage rapid prototyping, which allows for QUICK failure and mostly painless. It also allows us to reprioritize rapidly and leverage learnings.
DevOps Challenges for Acquisition Office

- DevOps is a complete disruption of the traditional Acquisition model. We need to leverage Sec 873/874 of the 2018 NDAA (check out https://cdnapisec.kaltura.com/index.php/extwidget/preview/partner_id/2203981/uiconf_id/38241181/entry_id/1_gib6brbc/embed/dynamic).

- No more complex RFPs with long planning phases with deliverables and milestones and fixed budgets. Keep in mind that when we write RFPs, we assume we know what we need and know exactly what the solution is. This doesn’t allow for us to learn and adopt new ideas along the way. This is the main cause of current failures. The world/technology evolves, what we know evolves... We must be able to adapt continuously to guarantee success.

- This is NOT scope creep but proper agile scope management.

- RFPs should NOT define precise requirements with pre-defined technologies but focus on establishing mission outcomes and precise metrics to prove success of those end-goals.

- There is no “beginning” or “end” of a project, the project will continuously evolve based on mission needs. Yes, a project might be terminated but the idea is continuous evolution and development.

- The traditional accounting methods - where you have the R&D, development and deployment in production followed by maintenance/support phases - don’t apply anymore. We CONTINUOUSLY develop, deploy and learn. There are multiple releases per day.

- New procurement tools must enable continuous development and incentivize the use of containers, microservices and Agile methods.

- Most of the DevOps tools are opensource and the only costs are Cloud hosting/computing/storage. Some members are worried about costs of licenses and think about consolidation for cost saving - this is a non-issue. We pay for what we use as a IaaS/PaaS/SaaS model.
Martin Fowler describes the Strangler Application:

- One of the natural wonders of this area are the huge strangler vines. They seed in the upper branches of a fig tree and gradually work their way down the tree until they root in the soil. Over many years they grow into fantastic and beautiful shapes, meanwhile strangling and killing the tree that was their host.

To get there, the following steps were followed:

- First, add a proxy, which sits between the legacy application and the user. Initially, this proxy doesn’t do anything but pass all traffic, unmodified, to the application.
- Then, add new service (with its own database(s) and other supporting infrastructure) and link it to the proxy. Implement the first new page in this service. Then allow the proxy to serve traffic to that page (see below).
- Add more pages, more functionality and potentially more services. Open up the proxy to the new pages and services. Repeat until all required functionality is handled by the new stack.
- The monolith no longer serves traffic and can be switched off.

Seth Wambold (Action Team Lead)

Focus Topic 1: “Terminology Gaps/Terms of Reference”
Matt Reider (Topic Team Lead)
Jay Eward, Ken Laskey, Rob Manogue, John Farrell, Scott Lawler

Focus Topic 2: “Scaling Agile & DevSecOps across GED Programs”
Marc Kriz (Topic Team Lead)
Joel Doyle, Sam Stollar, Eric Amberge, Pete Epstein, Shawn Lucas, Themha Hinke

Work Schedule:
• 3/26-4/12: Focus teams kickoff and pull together first thoughts on their topic
• Week of 4/12: Full team discuss first thoughts from Focus teams
• 4/23: Full team brief IAWG working session on observations, ideas, reccos
• Repeat to closure, move to next Focus Topic