Working Group Outbrief

Ground System Architectures Workshop

Session 11F
Smarter Acquisition with Agile Approaches

Supannika Mobasser and Jodene Sasine
The Aerospace Corporation

Approved for public release. OTR 2019-00520.
Participants

- William Rossiter, NGA/GFCG
- Bart Hackenmack, SEI
- Enrique Praga, GMV
- Barry Boehm, USC
- Melissa Tucker, Noblis
- Britany Chamberlain, Aerospace
- Ernie Foster, Lockheed Martin
- David Wilson, Raytheon
- Gary Chinault, USAF
- Taiko Hine, Mitsubishi Electric
- Dwain Harris, Aerospace
- John Eichner, Aerospace
- Brian Bone, Kratos
- Neal Faradineh, Rocket Communication
- Jannell Villegas, Aerospace
- Barbara Mills, Sandia National Labs

- Jennifer DeNicholas, Radiant Solutions
- Sue Mobasser, Aerospace
- Jodene Sasine, Aerospace
- Scott Nigel, Aerospace
- Curt Holmer, Aerospace
- Marvin Dolin, Lockheed Martin
- Margaret Eckerman, Aerospace
- Tony Chiles, DOD Civilian
- Alan Annett, DOD Civilian
- Jeffrey Schloemer, Raytheon
- Emily Vieth, Raytheon
- Marta Verdigo, ISISpace
- Necdet Engm Oztuna, TAI
- LaDell Weinbach, Aerospace
- B. Hochstein, SMC/AD
<table>
<thead>
<tr>
<th>Time</th>
<th>Presentation and Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00 – 1:20pm</td>
<td>Session Overview</td>
</tr>
<tr>
<td>1:20 – 1:45 pm</td>
<td>Agile Working Group 2018 Outbrief</td>
</tr>
<tr>
<td></td>
<td>Jodene Sasine, The Aerospace Corporation</td>
</tr>
<tr>
<td>1:45 – 2:10pm</td>
<td>Scaled Agile in a traditional fixed contract world: A case from Satellite Monitoring and Control</td>
</tr>
<tr>
<td></td>
<td>Enrique Fraga Moreira, GMV Aerospace and Defence</td>
</tr>
<tr>
<td>2:10 – 2:35pm</td>
<td>Revisit on Agile Fit Check</td>
</tr>
<tr>
<td></td>
<td>Supannika Mobasser, The Aerospace Corporation</td>
</tr>
<tr>
<td>2:35 – 3:00pm</td>
<td>Agile Anti-Patterns</td>
</tr>
<tr>
<td></td>
<td>Supannika Mobasser, The Aerospace Corporation</td>
</tr>
<tr>
<td>3:00 – 3:30pm</td>
<td>Break</td>
</tr>
<tr>
<td>3:30 – 5:00pm</td>
<td>General discussion</td>
</tr>
<tr>
<td></td>
<td>• Smarter software factory and product delivery</td>
</tr>
<tr>
<td></td>
<td>• Smarter program oversight and incentive structure</td>
</tr>
<tr>
<td></td>
<td>• Smarter quality assurance, compliance, and accreditation</td>
</tr>
<tr>
<td></td>
<td>• Smarter practices and other domains</td>
</tr>
</tbody>
</table>
Do you agree with the following minimum essential elements of a software factory?
- Continuous integration, Continuous testing
- Tool chain with maximum automation
- Reusable code

How can we make it smarter?
- Templates: Pre-made application elements with placeholders for arguments.
- Recipe: Automate procedures in routine tasks
- Architecture guidance and patterns
- IV&V with machine learning?
- Data-driven
- Cloud-based?
- Continuous deployment
  - Should we / can we do that? Deploy to where?
    - For sustainment, DevOps delivery daily or quarterly
    - More frequent deliveries may need reduced oversight
• How can we make it smarter?
  – Better integration of system testing by External groups (LDTO, AFOTEC, ……) not at end
  – Continuous integration testing – reserve 6 weeks at end for independent system tests
  – Balance capability deliveries to Operations to reflect when needed
  – Use technical debt analyzer (avoid potential blow-ups of debt)
  – Require transparency of development pipeline for the Government
  – Create cohesive team (e.g., Civilian Govt and contractors on the team together)
    • Civ. Govt fosters/cultivates the team relationship and collaboration.
    • Civ. Govt PO makes final decision if team can’t get there.
  – Metrics wants
    • For Govt, provide real-time test results, development progress using Ktr tools
    • FFRDC suggests number of regressions, average number of bug (found outside of sprint), bug age, story point estimation
    • Re-brand a ‘bug’ found in a sprint to a “SAVE”
    • Understand type of bug (i.e., functional, screen color, etc…) and impact
Do you agree with the following minimum essential elements of Cybersecurity approach?

- **Automated Testing/Test Reporting**
  - Automated pipeline kicks off on code check-in; performs static code analysis
  - New automated testing written by independent developer (need to understand programming language of automated test tool); done within the sprint

- **Automated Security Scanning**
  - CI/CD integrated with source code scans (security and quality)
  - All deployment candidates scanned prior to deployment

- **Other techniques:**
  - Red team penetration testing
  - Embed in static analysis where critical; peer review based on static analysis

- **How can we make it smarter?**
  - Automated compliance monitoring
    - Embedded in continuous integration/build pipeline
    - FOSS testing / risk assessment
Do you agree with the following minimum essential elements of certification and accreditation process?

– Plan for early and upfront involvement
– Define as part of acceptance criteria and definition of done

How can we make it smarter?

– Composable certification [DARPA 2018]
  • Use the evaluated criteria of a subsystem as evidence in a system evaluation
– Automated evaluation [DARPA 2018]
  • Produce compelling, checkable assurance arguments backed by evidence
– Data-driven evidence
How can we make it smarter?

- Include accreditors, AO as part of Agile team
- Provide baseline of security controls to start from
- Accreditors need to know what they’re accrediting
- Certification and accreditation needs to cognizant of DevOps risks
- AI driven 24 hour certification by DARPA
- Use containers for accreditation scope
- Microservice architecture for accreditation to support continuous ATO
How can the government test be performed early and often?

How early?
- Pre-ATP, be part of the team to write RFP and SOW
- Very early; need a large paradigm shift for Govt to fully staff early
- Govt test team (i.e., external, AFOTEC, ops acceptance team (typically require 6-8 month lead time))

How often?
- Sprint-level, quarterly, annually, one-time
  - Deliver as often as possible based on operational availability and risk

How can we make it smarter?
- Govt tester sitting with developer
- When requirements are defined gain agreement of how it will be tested/verified/signed-off
- Include early testing in acquisition strategy (i.e., need agile testing strategy for verification/acceptance)
- AFSPC has stood up a test organization to install a better methodology for testing – contractor-level testing, embedded LDTO structure, ....
• “Be careful what you wish for”
• From Govt to contractor
  – What to incentivize?
    • Specific goal? Stretch goal? Innovation? Schedule? Quality?
  – What not to incentivize?
• From high level management to development team
  – What to incentivize?
    • Specific goal? Stretch goal? Innovation? Schedule? Quality?
  – What not to incentivize?
• From high level management to development team
  – What to incentivize?
    • Define mission value and quantify busy-ness
    • Incentivize based on number of fixes delivered in a determined amount of time; up award fee based on number of fixes
    • Incentivize during test phases based on requirements, capabilities
    • Ktr: if we deliver what we signed up for then incentivize (“doing what we said we’re going to do” (i.e., in an increment)
    • Incentivize developers for each bug they fix
    • Govt knows what they want then go fixed price, otherwise go cost plus or capacity (T&M)
  – What not to incentivize?
- Fight about how to do Agile
- Not sure how to do it
- How to fit in acquisition paradigm
- Shared understanding, same pace
- Government Agile rhythm, management
- Decision maker (with no authority)
- Buy-in, leadership onboard
- “Responding to change”
- Coordinating with waterfall
- Team coordination
Good Points

- Good and quick feedback
- Agile and Lean
- Quality, Speed