Working Group Outbrief

Ground System Architectures Workshop

Session 11D
Achieving Resiliency with Agile Methods

Supannika Mobasser and Jodene Sasine,
The Aerospace Corporation
Ground System Architectures Workshop

Session 11D

Participants

- Lauren Ballard, Nesbitt Discovery Academy
- Lyle Barner, JPL
- Emily Brison, Nesbitt Discovery Academy
- Doug Buettner, Aerospace
- Jay Bugenhagen, NASA
- Brook Cavell, Aerospace
- Roger Claypoole, Aerospace
- Eric Cohen, Lockheed Martin
- Enrique Fraga, GMV
- Judy Kelley, ASRC Federal
- Peggy Lou, Aerospace
- Paul Mallon, Aerospace
- Ugur Melihslizue, TAI
- Sue Mobasser, Aerospace
- Phuong Phan, Navy
- Jodene Sasine, Aerospace
- Jim Schier, NASA
- Scott Smith, SAIC
- Bruce Steiner, Aerospace
- Michael Thimblin, Aerospace
- Rolando Ventura, Harris
- Russ Wolfer, USG
<table>
<thead>
<tr>
<th>Time</th>
<th>Presentation and Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00 – 1:30pm</td>
<td>Session Overview</td>
</tr>
<tr>
<td>1:30 – 2:00pm</td>
<td>“Agile ground segment software development: cheaper, faster and better”</td>
</tr>
<tr>
<td></td>
<td>Enrique Fraga Moreira, GMV Aerospace and Defence</td>
</tr>
<tr>
<td>2:00 – 2:30pm</td>
<td>“SCRUB for Peer Review of Static Code Analysis Results”</td>
</tr>
<tr>
<td></td>
<td>Lyle Barner, Jet Propulsion Laboratory</td>
</tr>
<tr>
<td>2:30 – 3:00pm</td>
<td>General discussion - I</td>
</tr>
<tr>
<td></td>
<td>• <strong>Agile Battle Rhythm</strong>: who, what, when, where, why, how many</td>
</tr>
<tr>
<td>3:00 – 3:30pm</td>
<td>Break</td>
</tr>
<tr>
<td>3:30 – 5:00pm</td>
<td>General discussion – II</td>
</tr>
<tr>
<td></td>
<td>• <strong>Agile Architecture</strong>: build “-ilities” and resiliency in</td>
</tr>
<tr>
<td></td>
<td>• <strong>Agile Enterprise</strong>: cultural and paradigm shift</td>
</tr>
<tr>
<td></td>
<td>• <strong>Agile Mission Assurance</strong>: trust but real-time verify</td>
</tr>
<tr>
<td></td>
<td>• <strong>Agile Supporting Infrastructure</strong>: required product and process resources</td>
</tr>
</tbody>
</table>
• **Who?**
  – **Default:** Scrum Product Owner, Scrum Master, Developers and Testers
  – Team composition? Any special team, such as system engineering team, integration team, program management, customer liaison, Integrated Product (Process) Team (IPT)?
    • Embed Architect, SMEs, Requirements Engineer, SE into Agile team
    • Developers knowledgeable on cybersecurity or have a dedicated Cyber engineer
    • Release Train Engineer – manage multiple teams, release tempo
    • Human Factors Engineer – overall program, involve user communities
  – Who is your Product Owner?
    • Government – requires training/education; challenging due to frequent Govt rotation (consider if 2-3 month overlap is possible for cross training)
    • Contractor – ensure regular communication between Customer and Product Owners
    • Contractor Product Owner needs Govt counterpart to synchronize
  – Required certifications for Product Owner? Scrum Master?
    • Must be experienced
• **What?**
  – **Default:** Sprint Planning, Daily Stand-up, Sprint Demo, Sprint Retro, Story Grooming?
    • Scrum of scrums
    • Pre-release / Post-release (build / increment / iteration) Reviews
  – How to collaborate across teams?
    • Utilize Release Train Engineers
  – Any additional / tailored activities for the new roles?
    • Govt counterpart to Contractor Product Owner
    • Govt engineer/developer embedded/deployed into Contractor Agile team that Govt pays for
• **When?**
  - Sprint length? Release length? Number of Sprint per Release?
    - Sprint length: 2-4 weeks
    - Release length:
      - Greenfield: quarterly (minor); 6 months (major)
      - Enhancements: monthly
      - Stakeholders constrained: 9 months
    - Number of sprints per release: depends on release length
  - Any empty/buffer Sprint? At least one per release
  - Milestone reviews?
    - Build / Increment / Iteration Review, TRR, RRD, RRT
    - Agile metrics at each sprint review (i.e., velocity)
      - Useful – burndown, burnup, velocity, features delivered, technical debt
  - Frequency of system-level demo? Monthly
  - Are you using Integrated Master Schedule (IMS)? Any alternative?
    - EVM at release/iteration level
    - Portfolio report (Jira)
    - SEER-SEM (agile)
    - Most productive is 80% assigned
    - PMI: 1 day is 6 hours
• **Where?**
  
  – **Default:** collocated team members
  – Challenges on distributed teams? Mitigations?
  – Do you have collocated users?
    
    • If not, how do you collaborate? How often?
      
      – Visit contractor site at appropriate times – open hot desk
      – Online access to Contractor dashboard
        
        » Be careful on micromanagement
      – Skype, VTC helps but things lost in translation (facial, body language)
      – Need periodic person-to-person contact
  
  – Development environments? Demo environments? Staging or Operational-like environments?
    
    • System demo done in test or ops-like environment; depends on program
    • Leave development environment for development
    • Utilize Docker
    • Watch out for ‘it works on my machine’
Ground System Architectures Workshop

Session 11D

Agile Battle Rhythm

• Why?
  – **Default:** Four Manifesto Values and Twelve Principles
  – What works, what does not work?
    • Responding to change – sometimes means no change, expectation management, design for potential changes
    • For fixed price, Govt collaboration needs to be well understood
      – Be transparent, timely
    • Welcome changing requirements – typically no but tweak is ok

<table>
<thead>
<tr>
<th>Individuals &amp; interactions</th>
<th>over</th>
<th>Processes &amp; tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working software</td>
<td>over</td>
<td>Comprehensive documentation</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>over</td>
<td>Contract negotiation</td>
</tr>
<tr>
<td>Responding to change</td>
<td>over</td>
<td>Following a plan</td>
</tr>
</tbody>
</table>

1. Satisfy the customer
2. Welcome changing requirements
3. Deliver working software frequently
4. Stakeholders work together daily throughout the project
5. Motivated individuals
6. Face-to-face conversation
7. Working software is the primary measure of progress.
8. Sustainable development
9. Continuous attention to technical excellence
10. Simplicity
11. Self-organizing teams
12. Continuous Improvement
• **How Many?**
  - **Default:** 4-9 people per team
    • No more than 10 members
  - Ratio between Product Owner and teams?
    • 1 Product Owner to 1-2 teams (max)
  - Ratio between Scrum Master and teams?
    • 1 Scrum Master to 1 team
• **Approaches:** Design-as-you-go, Emergent Design, Architecture Runway, Enterprise Architecture, Release Train

• What is your approach in developing architecture and design in Agile development?
  – Knowledge of Interfaces and Interoperability
  – Embedded Systems Engineering (including architect) team
  – Design in resilience; need team members educated on what resilience means
  – Architecture Runway – have SE/Design teams work ahead of development team to flush out design prior

• How do you address non-functional requirements?
  – Part of Definition of Done; every commit checks (mostly automated); peer review, performance testing

• How do manage dependency between components?
  – Roadmap needs to be clear with dependencies represented
• Buy-in at Middle Management
  – Use short term incentives; MVP for short term win
• Culture Shock
  – Leader is no longer the boss, acts as a facilitator
  – Transition: Processes, Metrics, Tools, Infrastructure, Role & Responsibilities
  – Hire an Agile Coach
  – Executive involved at the beginning
  – Government: increased workload tremendously
  – Contractor: matrix management of agile developers created risk across multiple projects
  – Agile is more costly in the beginning; cheaper in “total cost of ownership”
• System Acceptance
  – Minimum at Feature and System level
  – Full System Test at the end
• Better way to adopt Agile, Process Improvement
• Midstream Agile Adoption
• RFP, FAR, Acquisition Milestones
• Scaling
• Interface to different processes
• System enhancement vs Greenfield development
• EVM, Project Planning, Quality Management
• Requirements Management
- Infrastructure and Resources must be ready (e.g., DevOps)
- Continuous Planning, Continuous Code Integration & Test
- Agile experience required; more important than certification
- Empower the team; but balance with checks and controls
- Release Engineer: Align integration team and system engineering team
- Assign 10% margin for reengineering

**All parties need to be Agile**