Working Group E Outbrief

25th Ground System Architectures Workshop

Adapting Critical Operations

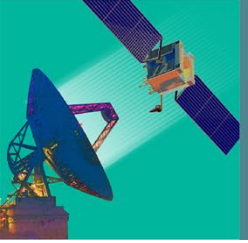
Starts March 1, 2021 | Special Online Series of Events

Agility with Continuous Adaptation

Leads:
Sue Mobasser and
Jodene Sasine,
The Aerospace Corporation

March 10, 2021

OTR202100612

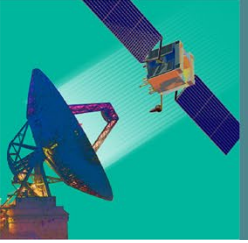


Session Goals

- Discuss adaptation of Agile concepts to the Ground System Acquisition lifecycle

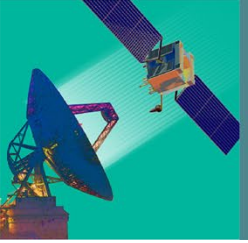
- Discussion Topics
  - *Agile Acquisition Policy and Guidance*
  - *Adapting to Face-to-Face Collaboration Remotely*
  - *Continuous Delivery through Smarter Software Factory*
    - DevSecOps, Tools and Infrastructure
    - Continuous Authority to Operate (cATO)
  - *Agile V&V, Quality Assurance*
  - *Incorporating Digital Engineering (DE) Concepts into Agile Programs*
    - Model-Based Software Engineering (MBSE)

- Share Agile adoption experiences and learn from others



Presenters/Panelists

- Dr. Supannika Mobasser, The Aerospace Corporation
- Jodene Sasine, The Aerospace Corporation

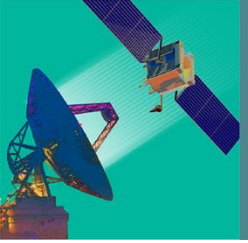


Key Points

Pain Points & Expectations

- Pain Points
  - *Still see resistance to Agile. Some people are still skeptical. We have to "educate" and make newcomers more comfortable with the switch to Agile.*
  - *Pain: Many people think Agile means "no process" so they can go rogue if you are not paying attention. Unfortunately, many WANT agile to be no process so they push for it.*
  - *How to trust and verify in an agile environment.*

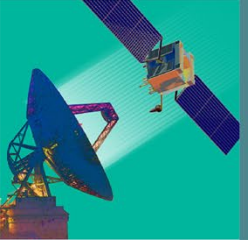
- Working Group Expectations
  - *Address Agile Application to SmallSat Proliferated space, quicker, faster, cheaper?*
  - *Agile interest - System design for an effective agile approach.*
  - *How to do agile in a less-agile environment.*



Key Points

Agile Acquisition Policy and Guidance

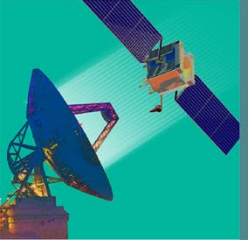
- Lessons Learned
  - *Don't give the contractor too much leeway to set when new features will be added, else it may be a LONG time*
  - *Using SOO (Statement of Objectives) or Statement of Capabilities instead of requirements*
  - *Long time contactors expect detailed requirements and are having trouble with just being provided capabilities to propose/design to.*
  - *Incorporating agile does not mean absence of Systems Engineering*
    - Using Architectural Runway or Product Roadmap to plan your work
  - *Be careful of templates. Sometimes those templates invite the old guard to continue doing things the old way.*
  - *Tailor CDRLs (what & when & how to deliver) to support Agile program*



Key Points

Agile Acquisition Policy and Guidance (cont.)

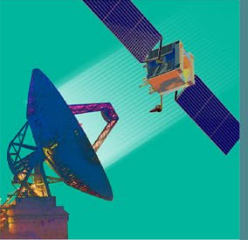
- Lessons Learned (cont.)
  - *Make sure you have full and complete transparency with contractor - this includes being invited to sprint planning mtgs, sprint and increment reviews, and ESPECIALLY access to burndown/burnup charts to track progress. Also - Make sure you ask for many different s/w metrics from their s/w quality staff to track # escapes, defect densities, KSLOC counts ,etc.*
  - *It's a huge organizational change for customers as well. If they aren't fully bought into agile, it's almost untenable.*
  - *Make sure your sprint planning meetings really ARE planning meetings. Some of ours were basically contractor brag sessions. (This is the same contractor that deferred new features...kicked that can WAY down the road.)*
  - *Re: metrics. They are frequently collected but not acted upon. How do you ensure they are incorporated into Agile?*



Key Points

Face-to-Face Collaboration Remotely Techniques

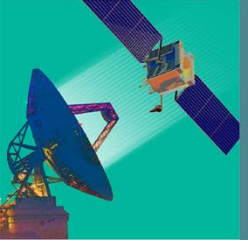
- Lessons Learned
  - *Use the right tools*
  - *Need team building activities*
  - *Use slack for cross-organization chat*
  - *Have attendees with camera's on so that they will be more attentive*



Key Points

Continuous Delivery through Smarter Software Factory

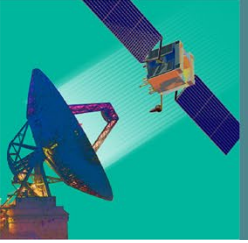
- DevSecOps Lessons Learned
  - *Need to educate people, so that we have common / correct understanding*
  - *Must think web- and microservices with lots of messaging, AND GIT repositories, AND containerization (with orchestration (think Kubernetes)) AND automation EVERYWHERE as much as possible. And also CI/CD pipelines.*
  - *Different variants of CloudOne and PlatformOne (or equivalent) can be adopted, which can and has led to competing subcultures within same large program.*
- *Continuous Authority to Operate (cATO) Lessons Learned*
  - *Collaborate with AO upfront, on features/capabilities release, so approval process can be faster*
  - *Extremely difficult to get to cATO; every generic change needs to be reviewed and approved by each mission before deployment; want to get to the point where the generic change is verified once and then deployed across all missions*
- Concerns with doing cATO?
  - *Getting an initial ATO is a major chore- getting continuous ATO will be nightmare unless ATO process is modified . It will require huge manpower also*



Key Points

Agile Verification and Validation (V&V)

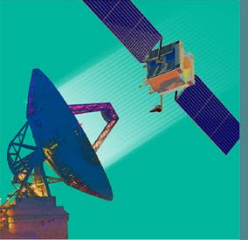
- Alternative V&V approaches for Agile
  - *Integrated V&V (not independent)*
    - V&V team is part of the Agile team
  - *Staggered V&V Sprints*
    - V&V team is running one Sprint behind to certify / evaluate the incrementally developed products
  - *Kanban*
    - No timebox, continuous flow of works; support varying sizes of tasks
  - *V&V-Driven-Development*
    - Develop metrics, threshold, or run test scripts to establish baseline, then incorporate continuous monitoring
  - *V&V Escrow*
    - Provide continuous oversight and performs periodic (quarterly) code inspection
  - *V&V in DevSecOps*
    - Automate the evaluation as much as possible to support continuous delivery



Key Points

Incorporating DE Concepts into Agile Programs

- Model-Based Software Engineering (MBSE) Discussion
  - *I've seen a lot of UML modeling integrated into agile programs, but modeling using MBSE and SysML seems to be more slowly understood and adopted, maybe because it more abstract. Fullblown Digital Engineering is even more "abstract" and slow to be understood and implemented.*
    - Just via the usage of activity diagrams, use case diagrams, state diagrams, etc.
  - *Someone presented a virtual store concept the other day. Same was presented for MBSE. I have never been able to browse/shop for an architecture product to use in my planning.*
  - *MBSE seems to provide strict/rigid modeling, which could be conflicting with agile that emphasizes flexibility. If too detailed, it became constraints. Need to stop at the level that does not change frequently.*
  - *Models are surely more "visual" and helpful to non-technical customers/execs, etc., but they are also more abstract (or esoteric?) since there are so many different types of models.*



Conclusions

- Discussion Topics
  - *Agile Acquisition Policy and Guidance*
  - *Adapting to Face-to-Face Collaboration Remotely*
  - *Continuous Delivery through Smarter Software Factory*
    - DevSecOps, Tools and Infrastructure
    - Continuous Authority to Operate (cATO)
  - *Agile V&V, Quality Assurance*
  - *Incorporating Digital Engineering (DE) Concepts into Agile Programs*
    - Model-Based Software Engineering (MBSE)