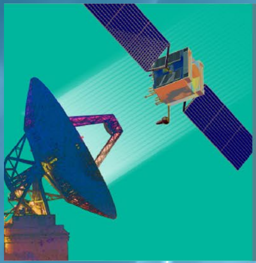
Working Group H Outbrief



Ground System Architectures Workshop

Driving Innovation for Enterprise Integration

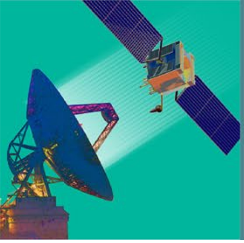
February 23–March 3, 2022 | Virtual Event

Agile Innovation for Enterprise Integration

Leads:
Jodene Sasine and
Jason McKenney,
The Aerospace Corporation

March 2, 2022

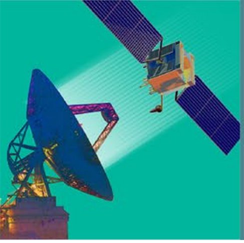
Approved for public release. OTR 2022-00620.



Session Goals

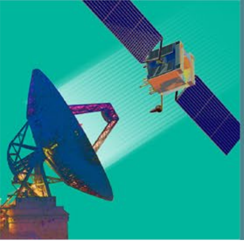
- This working group provides an opportunity for Agile practitioners to share their experiences and learn from others on several topics concerning integration

- Session Topics:
  - *Gather feedback and ideas from attendees on how to integrate Agile methods with Digital Engineering*
  - *Provide overview presentation on using Agile for hardware development*
  - *Provide overview presentation on Cybersecurity best practices for Agile teams*
  - *Describe techniques to set up Continuous Integration supported by Agile and DevSecOps development*
  - *Engage with the attendees to gather best practices based on their experiences with these topics*



Presenters / Panelists

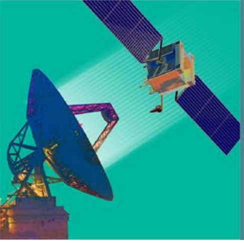
- Jodene Sasine, The Aerospace Corporation
- Donald Denham, The Aerospace Corporation
- Michael Roberts, The Aerospace Corporation
- Dr. Eric McCary, The Aerospace Corporation
- Jason McKenney, The Aerospace Corporation



Key Points

Expectations

- Examples of large, complex cyber-physical system adoption of lean/agile mindset
- General understanding of what the issues are and how to address
- How best to use Agile for a large integration project (customer/contractor resistant to creating architecture/design documentation - especially around interfaces and data exchange)
- How to understand dependencies given lack of documentation
- Sharing architecture/design models with stakeholders building the system who may not have access to Cameo, or know how to use such a tool

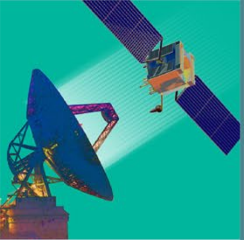


Key Points

Pain Points & Challenges

- Syncing Agile development cycles with non-Agile teams working to support the same product can be a challenge
- Security features will often be pushed to later in the development cycle
- Maturity level of Agile processes in support of Digital Engineering still needs work on most Government programs
- Handling different disciplines during the planning phase of a program can be a challenge
- Achieving Continuous Authority to Operate (ATO) is a good goal to have but it contains a lot of challenges
- Adapting Agile software best practices to Agile hardware development

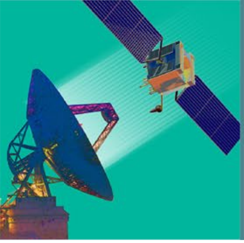
Working Group H



Key Points

Lessons Learned

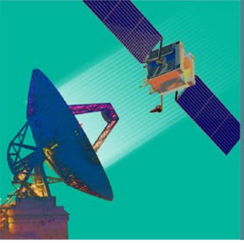
- Agile software development is gaining momentum in the Government sector and bringing new challenges along with it
- Hardware development can be supported by Agile processes
- Continuous Integration goals are best achieved with Agile development processes
- Continuous Integration pipelines can add more value to Ground System programs than simply being the “next new thing”
- Top-down modeling approach is most commonly used
  - *Models allow for quicker constraint identification rather than relying on Specification documents*



Key Points

Digital Engineering (DE) and Agile

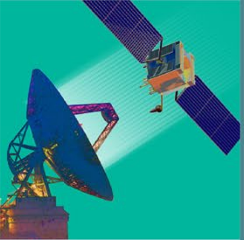
- Lessons Learned
  - *Role composition of the DE team (i.e., SE, Modeler, etc…)?*
    - Use Agile in their Digital Engineering Environment team
      - *Not using SAFe team organization constructs*
      - *Strong leadership with good communication mechanisms*
    - Dedicated Agile team working on model-based system/software engineering (MBSE), which enables prioritization across multiple project efforts within the program
    - Digital engineering component as part of each Agile development team
    - Program initially had SE embedded within project teams, however, with limited resources, it created conflicting prioritization.
      - *Moved to a single team with resources more aligned; team reviews all requests across program as inputs to their planning and leverages leadership prioritization to determine what fits based on their SE capacity*



Key Points

Digital Engineering (DE) and Agile

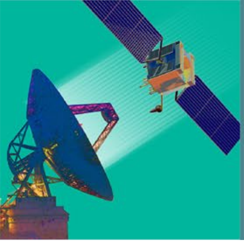
- Lessons Learned (cont.)
  - *How many DE teams are there?*
    - One DE team actually made up of multiple teams internal to the DE project
    - Multiple SEs within each Agile development team
  - *Are DE Teams Sprinting? Kanbaning?*
    - Program mostly using Scrum
    - Program previously using SAFe/Sprints
      - *Currently not using program-wide SAFe; still utilizing quarterly planning cycles*
  - *How do DE Teams get ahead of development teams?*
    - Vision for systems engineers to start the models and have accessible to development teams; development teams expected to update models as they implement, within the guardrails that are provided
    - Communicate knowledge points (key IMS elements, etc…) that are kept in Confluence as a living document



Key Points

Digital Engineering (DE) and Agile

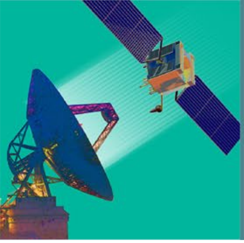
- Lessons Learned (cont.)
  - *What are your experiences on integrating Agile with DE? What was painful? What made you happy? What surprised you?*
    - Challenge: How to properly convey the appropriate level of maturity for each incremental delivery of engineering artifacts; how to agree with contractors on what is "just enough“
    - Painful: program that overkills MBSE use (e.g., draw activity/sequence diagrams of the DevSecOps pipeline)



Key Points

Continuous Integration Supported by Agile

- Lessons Learned
  - *Scrum supports CI/CD very well, but other Agile methods like Kanban can also be used*
  - *Code branching strategies need to be defined and understood by the development team and integration testing teams*
  - *Commits to the mainline should be done daily at minimum*
  - *Automated builds should be done with each code commit and should be kept at under 15 minutes to complete*
  - *Automated builds should include automated test cases for both new features and select regression*
  - *Broken builds should be fixed immediately*
  - *Test environment should mimic actual environment as much as possible*
  - *Release and rollback strategy should be defined in advance*
  - *Create dashboards to report on important metrics*
    - Metrics should be targeted and provide value



Conclusions

- Use of MBSE still has a learning curve to be attained by many programs
- Agile processes for hardware development will be similar but not always the same as those for software development
- Security features should be integrated into products as the new code base is being developed
- Automated Build processes must have support processes defined in advance (for example, what does the development team do when a build breaks or fails?)