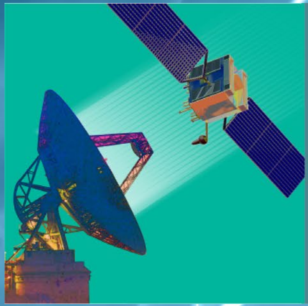
Working Group Session 11A Outbrief



Ground System Architectures Workshop

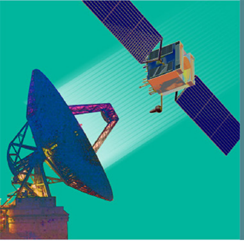
Opportunities in Data Exploitation

March 2–5, 2020 | Renaissance Los Angeles Airport Hotel

Agile Retrospective: Opportunities to Perform Agile Acquisition Differently

Leads:
Supannika Mobasser and
Jodene Sasine,
The Aerospace Corporation

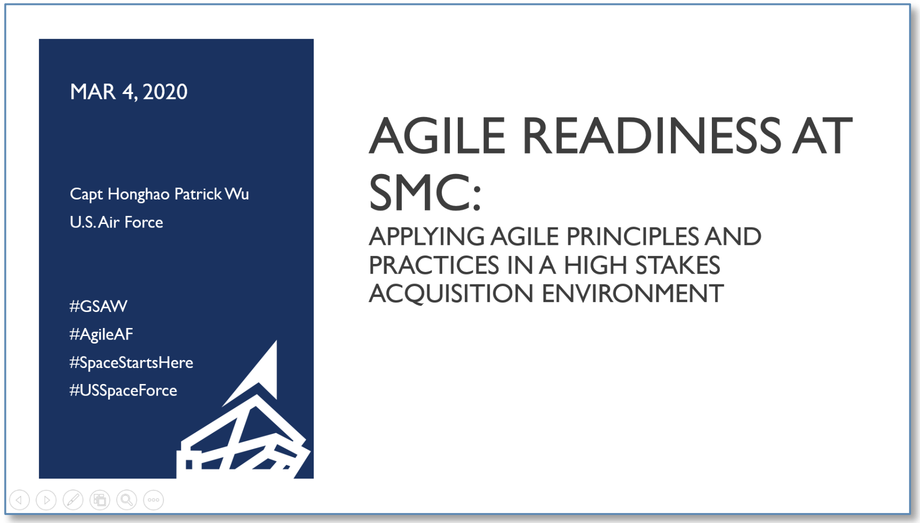
Approved for public release. OTR 2020-00511.



Schedule

Time	Presentation and Discussion
1:00 – 1:20pm	Session Overview
1:20 – 1:45 pm	*Agile Working Group 2019 Outbrief* Jodene Sasine, The Aerospace Corporation
1:45 – 2:10pm	*Agile Readiness at SMC* Capt Patrick Wu, SMC/ACX
2:10 – 3:00pm	General discussion • *Agile-compatible milestones and battle rhythm* • *Which CDRLs, when, and how?*
3:00 – 3:30pm	Break
3:30 – 3:50pm	*Using Organizational Baselining to Inform Adoption Planning of New Practices* Suzanne Miller, Software Engineering Institute
3:50 – 5:00pm	General discussion • *Continuous integration, verification, and testing* • *Just-in-time certification and accreditation* • *Smarter and faster data-driven metrics* • *Agile & MBE* • *Transparency and Openness*

Working Group Session 11A



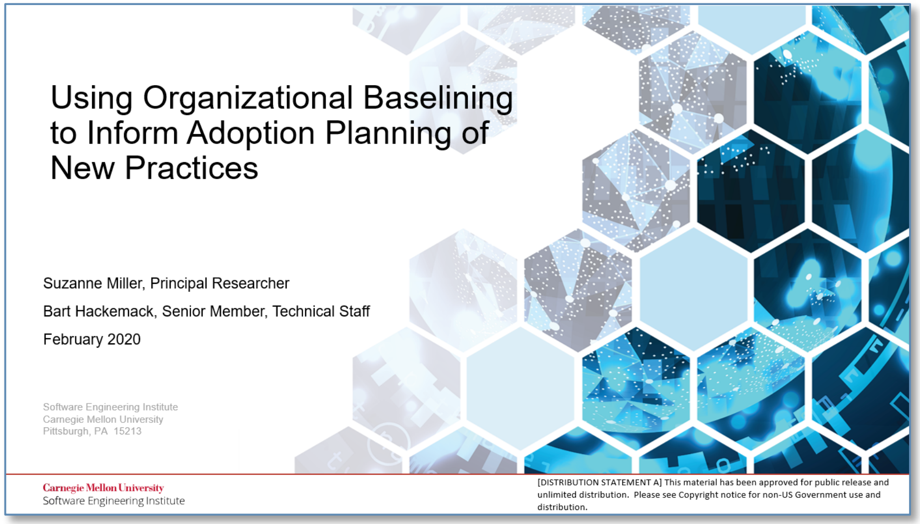
MAR 4, 2020

Capt Honghao Patrick Wu
U.S. Air Force

#GSAW
#AgileAF
#SpaceStartsHere
#USSpaceForce

AGILE READINESS AT SMC:

APPLYING AGILE PRINCIPLES AND PRACTICES IN A HIGH STAKES ACQUISITION ENVIRONMENT



Using Organizational Baselining to Inform Adoption Planning of New Practices

Suzanne Miller, Principal Researcher
Bart Hackemack, Senior Member, Technical Staff
February 2020

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Working Group Session 11A

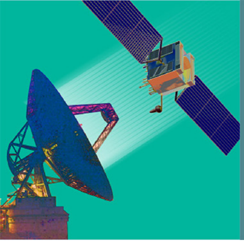
SMC
SPACE AND MISSILE SYSTEMS CENTER

THE STATE OF AGILE REPORT

Space and Missile Systems Center
United States Air Force
September 2019

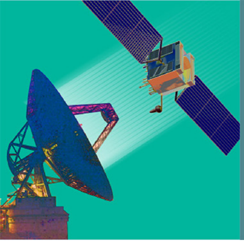
DISTRIBUTION STATEMENT A: This material has been approved for public release and unlimited distribution.

Download the SMC State of Agile Report at https://sites.google.com/view/atlasx/ARC



Pain Points about Agile Adoption

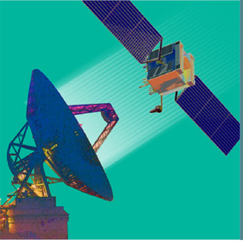
- Portfolio consistency on Agile adoption
  - *Agile adoption on System of Systems development*
  - *Coordinating with waterfall*
  - *Agile in sustainment and O&M*
  - *Agile in HW&SW development (complex cyber physical systems)*
- Agile architecture and design
- Budget, estimation, and tracking
- Agile mindset and culture
- Contract
  - *Limitation and inflexibility*
  - *Monolithic contract*



BLUF - Bottom Line Up Front

- Milestone reviews
  - *Keep SRR: Check the translation of requirements to Program Backlog*
  - *Keep other milestones incremental*
  - *Frequent but not too often, need to let the team works*
- CDRLs and document
  - *Develop a Product Roadmap pre-award to understand expectation and dependency*
  - *Use mission thread to tell story and priority*
  - *Keep ICD, test strategy, a document for platform infrastructure*
  - *Automate, auto-generate as much as possible*
  - *Don't request for CDRL upfront, high overhead to deliver a CDRL. Use live, online, incremental document*
- DevSecOps
  - *Keep Dev and Ops environments in sync*
  - *Integrate at the system-level as often as possible, cheaper rework, cheaper total cost of ownership*
  - *Ops might not be available for frequent deployment*
  - *Leadership may support frequent deployment, but AO is not available*

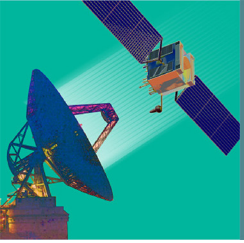
Working Group Session 11A



Agile-compatible milestones and battle rhythm

- Have you aligned traditional milestones with your program’s Agile planning and development battle rhythm? Did you tailor milestone expectations? How?
  - *SRR: Software Requirements Review*
    - *Kept this milestone; Translate Capability Development Document (CDD) into Program Backlog*
      - *Year one is decomposed to expected level of details (1-2 down from CDD)*
        - *Also, release 1 or 2 decomposed at this level, After year 1, keep at CDD level*
    - *Helps to start looking at Architecture, Leave room for learning by not over-specifying requirements*
  - *SDR, PMR, PDR, CDR, TRR, RRR*
  - *FDD: Full Deployment Decision*
    - *Kept this one in cyber-physical example*
  - *FD: Full Deployment*
- *other milestone activities are incorporated into other Agile events.*
  - *Apply incremental milestones as appropriate*

Working Group Session 11A

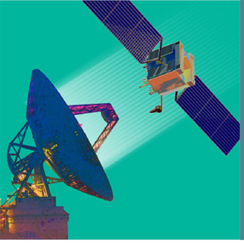


Ground System Architectures Workshop

Agile-compatible milestones and battle rhythm

- Product Roadmap – collaboration between Govt and Contractor
  - *Identify capability deliveries, then generate a product backlog per release/build*
  - *Capabilities are potentially shippable*
- Use Mission Threads and associate Epics to each Mission Thread
- Considerations for which milestones might be impacted by size of program
- Example program changed PMR structure to their 3-month increment cycle
- Adaptive Acquisition Pathways
  - *Lessons learned from 804, mid-tier programs, relaxed some requirements*
  - *Focused on meeting criteria as opposed to having events*
- Recognition that Hardware and Software can be treated differently

Working Group Session 11A

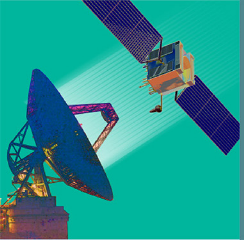


Agile-compatible milestones and battle rhythm

- What other Agile-compatible milestones or decision points are you using?
  - *Major and Minor Incremental Development Review (IDR) – every 4 weeks, demonstrate to govt*
    - Minor IDR - 1 day review, review software and some metrics
    - Major IDR – Revisit architecture

- Sustainment
  - *Develop a cadence for analysis of change requirements; does not require long-term analysis*

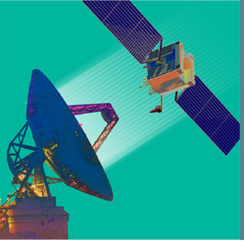
- Pathfinder Project
  - *Good candidate for Agile*
  - *Once project picked up, it imposed additional restrictions*
  - *How do you know if you are done? Change requests continuously*



Agile-compatible milestones and battle rhythm

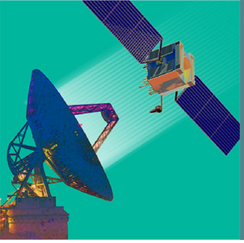
- Programs use different battle rhythms and terminology
  - *Program A: Build (9 months), Program Increment (3 months), Iteration (2 weeks)*
  - *Program B: Program Increment (3 months), Incremental Development Review (4 weeks)*
  - *Program C: Build Decision Review (8 months), System Demo (2 months), Sprint (2 weeks)*
  - *2 month increments, 1 month sprint*
    - Do this to align with EVM/Agile EVM
  - *A SME rep offers regular hours to be available for teams to ask questions*

- What battle rhythms that are not good?
  - *Examples*
    - 2 week sprint, 4 hour block meeting – not productive to Agile team



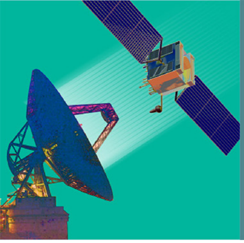
- How often do you release?
  - *Release to staging environment*
    - Every Sprint / Iteration (~1-4 weeks)
    - Every Release / Build (~3-6 months)
    - Every major milestone (~1 year)
    - Each Program Increment (~3 months)
    - One time Release at the end of development
    - Depends on component; typically 1/qtr; would like higher frequency
    - Continuous – CI/CD pipeline
    - Release on-demand
    - Window for deployment might be limited

Release defined as deployment to a non-development environment.



- How often do you release?
  - *Release to Ops Floor or Operation/Production environment*
    - Every Sprint / Iteration (~1-4 weeks)
    - Every Release / Build (~3-6 months)
    - Every major milestone (~1 year)
    - One time Release at the end of development
    - Sustainment: Quarterly to Ops floor
    - If deploy to multiple sites; each could pull from repo and upgrade at least 1 per year or more
  - ***Challenges:***
    - Ability for continuous ATO vs Ops floor is not ready to take the release

Release defined as deployment to a non-development environment.



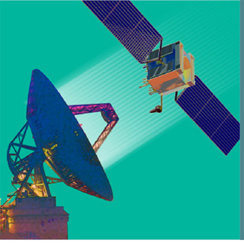
Ground System Architectures Workshop

Program Management Level CDRLs

- Which CDRLs have you excluded / tailored, or added for your Agile program?
  - *Program Management, Subcontractor Management Plan*
  - *System Engineering Management Plan, Software and System Measurements Report*
  - ***Integrated Master Schedule (lagging rather than leading)***
  - *Product Roadmap, Work Breakdown Structure*
  - *Software Development Plan*
  - *Quality Assurance Program Plan, Configuration Management Plan, Data Management Plan, Accreditation Plan, Financial Management*

  - ***Challenges: Budget, estimation, tracking***
    - **Agile EVM**
      - *Uses Agile-compatible WBS; sync EVM with sprint or release*
      - *3 month cycles are recommended for EVM*
    - How do you know if you are on track to complete on 3-5 year roadmap?
    - Prioritized backlog vs 5-year POM
    - Budget to sustain and evolve instead of POM
    - Value: Enterprise value vs just Operator Value
    - Minimum Viable Product (MVP)

Working Group Session 11A

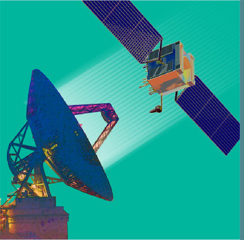


Requirement CDRLs

- Which CDRLs have you excluded / tailored, or added for your Agile program?
  - *System/Subsystem Specification, Software Requirements Specification*
  - *Technical Requirements Document*
  - *Requirements Traceability Matrix*
  - *Product Backlog*
  - ***Observations:***
    - Don’t let go of the ICDs
    - Need things that help tell the story
      - *Product backlog does not tell a story*
      - *Mission threads do help to tell the story*
      - *Help to provide priority*
      - *Option: Capability Needs Statement*

Architecture CDRLs

- Which CDRLs have you excluded / tailored, or added for your Agile program?
  - *Software Architecture Description, MBE models*
  - *Interface Control Document*
  - *System/Subsystem Design Description, Software Design Description*
  - ***Observations:***
    - Need something to define the platform architecture
    - Definition of the infrastructure
      - *Some programs fail due to not having a defined infrastructure*
      - *Dev and Ops environment aligned*
      - *Less emphasis on "to be"*
      - *More emphasis on "as built"*

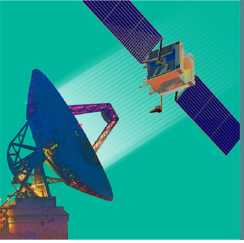


Testing CDRLs

- Which CDRLs have you excluded / tailored, or added for your Agile program?
  - *System Test Plan, System Integration and Test Plan*
  - *Software Test Plan, Software Test Description, Software Test Report*
  - ***Observations:***
    - TEMP – Test and Evaluation Management plan
      - *Define test strategies; incremental test detail in each appendix.*
    - Documentation of the test is via tool
    - Test plan for sustainment/sustained development
    - Identify the key test that you are going to run
    - Automated testing – autogenerate test documents

When and How?

- When are CDRLs delivered for your Agile program?
  - *Draft until final “As-built”*
  - *Align on battle rhythm (Release, Program Increment, …..)*
  - *Include in the “Definition of Done” for Build, Program Increment, Epic, Feature*
  - ***Observations:***
    - Depends of the purpose of the CDRL
    - Extremely high overhead cost to deliver a CDRL
      - *Contractor processes my inhibit quick delivery of CDRL*
    - Trust and good working relationship between Government and contractor
      - *Preferred option: Wiki-like, incremental, as-built documents*
    - Some documents need to be delivered in hardcopy such as technical manual



Ground System Architectures Workshop

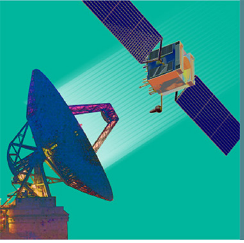
Continuous integration, verification, and testing

- How often do you integrate at the system-level on an Agile program?
  - *Every day (nightly build)*
  - *Every Sprint / Iteration (1-4 weeks)*
  - *Every Release / Build (1-4 months)*
  - *Every year (6 -12 months)*
  - *One time at the end of the development*
  - *Others? Depends*
  - ***Pros:***
  - ***Cons:***
  - ***Challenges:***
    - Determined by dependencies
    - Integration may cost less the more often you delivery and integrate
      - *Integrate early and often, lower rework cost*

DevOps : Synchronization between Dev and Ops

- Considering frequent software patching, updates, what level of consistency between Dev and Ops environments is necessary?
  - *Preferably 100% alignment between Dev and Ops*

- Techniques In ensuring the consistency
  - *Use the same baseline image*
  - *Use containerization*



Just-in-time certification and accreditation

- Do any of these suggestions from last year’s session work for an Agile program?
  - *Design for certification*
    - Microservice architecture
    - Use containers for accreditation scope
    - Artificial Intelligence (AI) driven 24 hour certification by DARPA
    - Automated compliance monitoring
  - *Stakeholders involvement*
    - Include accreditors, Authorizing Official (AO) as part of Agile team
      - *Accreditors need to know what they’re accrediting*

- Any new experiences regarding Continuous Authorization to Operate (ATO)?
  - *NASA has ATO out of the box, developer is using sandbox, but have to stay in the security boundary*
  - *Although leadership support, but AO is not available*

Working Group Session 11A