



SYSTEMS
ENGINEERING
RESEARCH CENTER

USC
Viterbi
Information
Sciences Institute

WRT-1012/WRT-1041: SMC Production Corps – Mission Engineering and Integration of Emerging Technologies

Michael Orosz, University of Southern California Information Sciences Institute

Sponsor: US Space Force Space and Missile Systems Center-Production Corps (SMC/PC)

WASHINGTON DC VIRTUAL

NOV. 2-4 2021

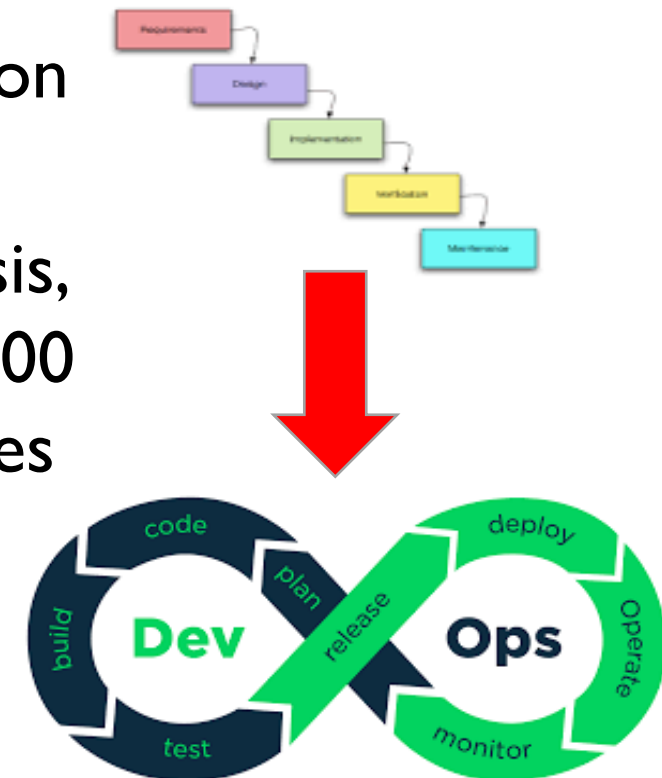
ANNUAL SPONSOR RESEARCH REVIEW

Agenda

- Project Objectives and Timeline
- Overview of Previous Results
- Current Status
- Next Steps

Objectives

- **Target:** DoD space-based system acquisition process
- **Goal:** Improve current space vehicle system acquisition processes
 - Determine the mission engineering methods, analysis, and metrics to transition from a traditional DoD 5000 waterfall development to Agile/DevSecOps processes
 - Includes integration of emerging technologies and related education for the future workforce



Process

1. Understand the current acquisition environment
 - Immerse into environment (become part of the team)
2. Develop approaches to transition acquisition elements from DoD 5000 to Agile/DevSecOps
3. Incorporate processes and “lessons-learned” into a transition process to apply to other domains

Project Overview

- **Team:**
 - USC Information Sciences Institute (USC/ISI)
 - SERC
- **Funding agency:** U.S. Space Force and Space Systems Command - Production Corps (SSC/PC)
- **Period of Performance:** 26 June 2019 – 29 April 2023 (via three different funded SERC projects)

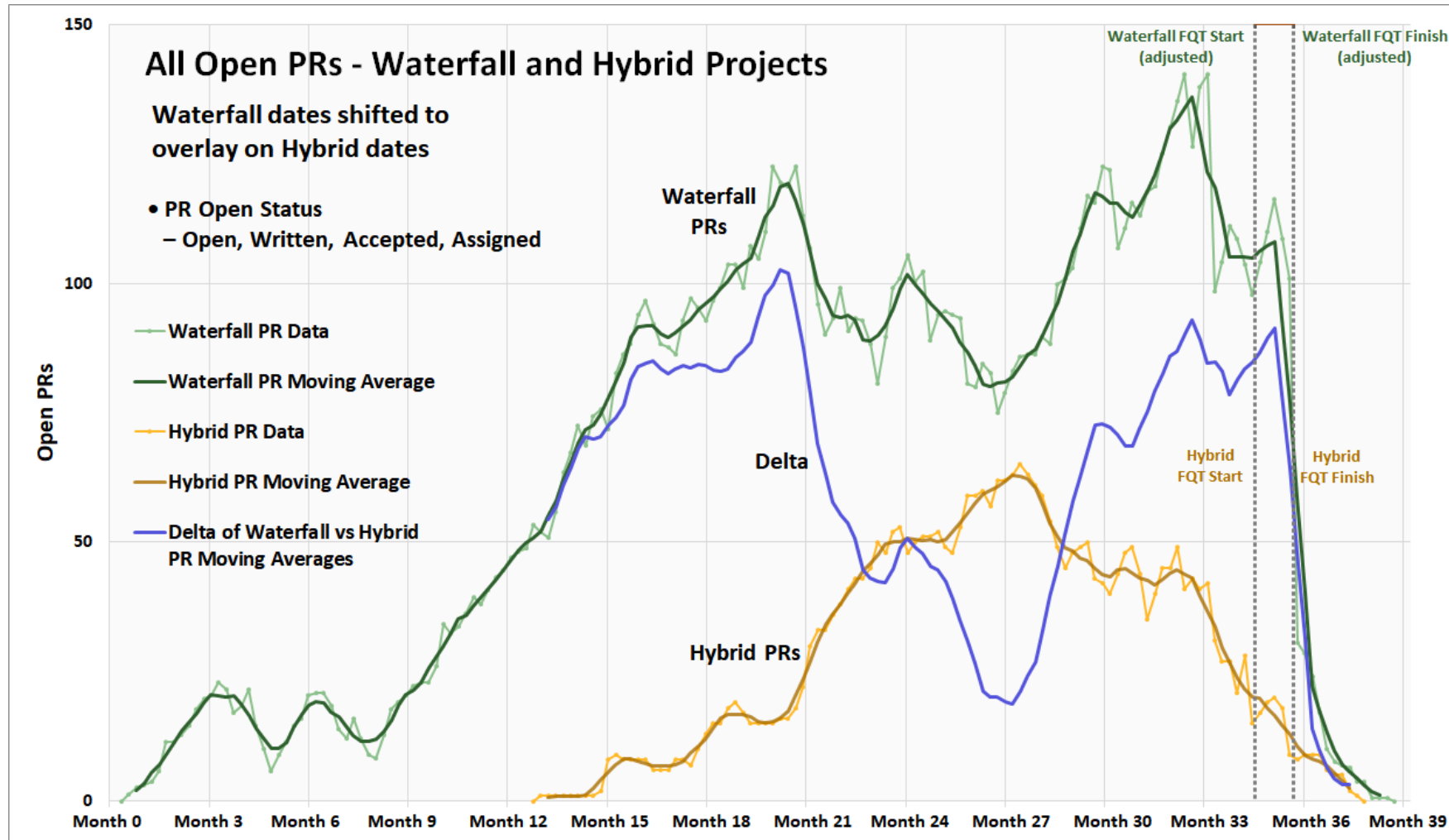


Four DoD Acquisition Projects

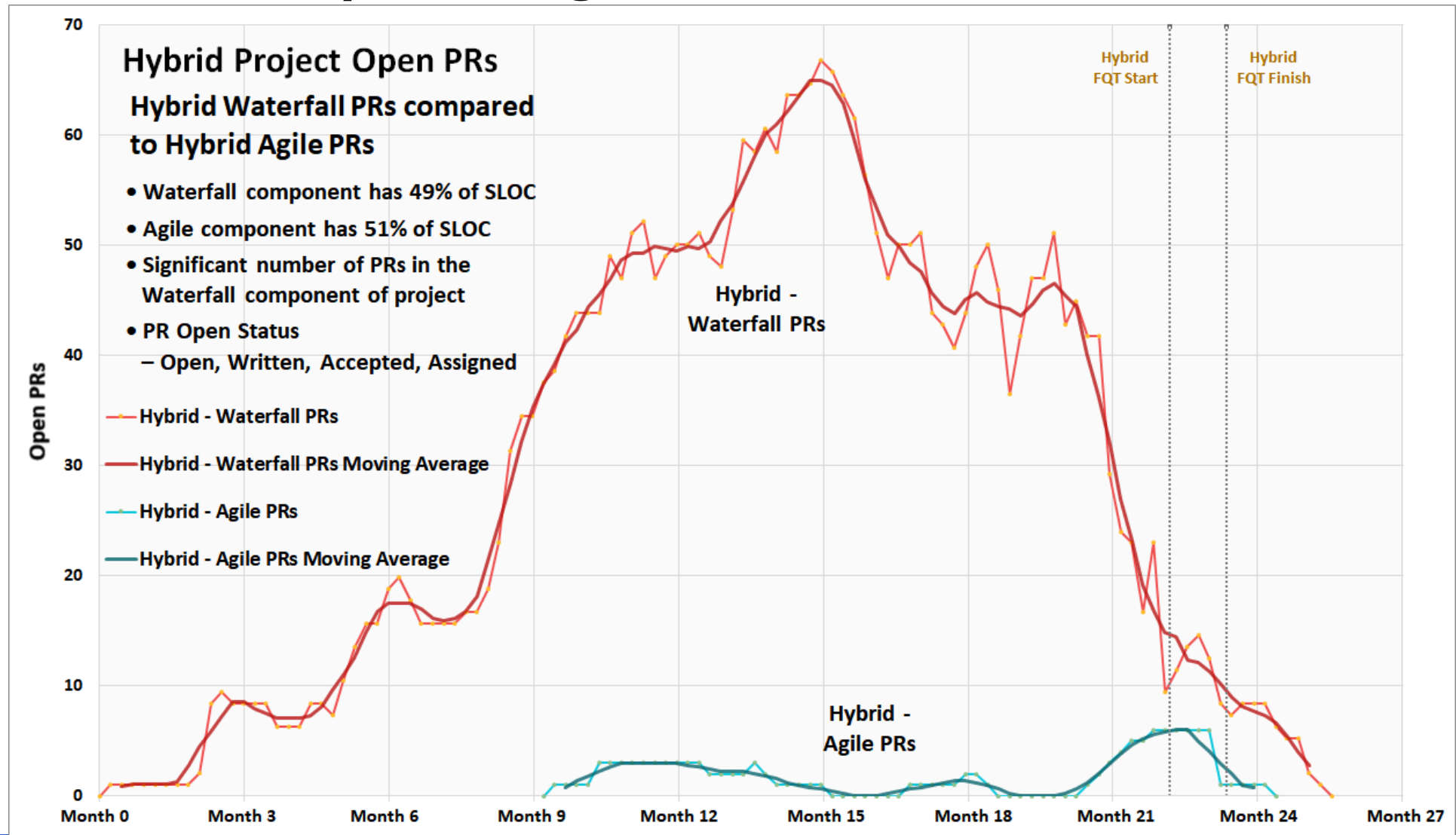
- **Project A**: Traditional waterfall method used (**completed**)
 - Duration: 39 months (includes schedule extension)
 - Software lines of code (SLOC): 178K
- **Project B**: Hybrid composed of both waterfall and agile components (**completed**)
 - Duration: 25 months
 - Software lines of code (SLOC): 113K
- **Project C**: Undertake technical explorations and stand up Agile/DevSecOps environment in preparation for Project D (**completed**)
 - Duration: 14 months
 - Software lines of code (SLOC): None
- **Project D**: Agile/DevSecOps
 - Duration: Approximately 48 months
 - Software lines of code (SLOC): TBD

Overview of Projects A and B

Problem Report (PR) Comparison of Project A (Waterfall) and Project B (Hybrid)

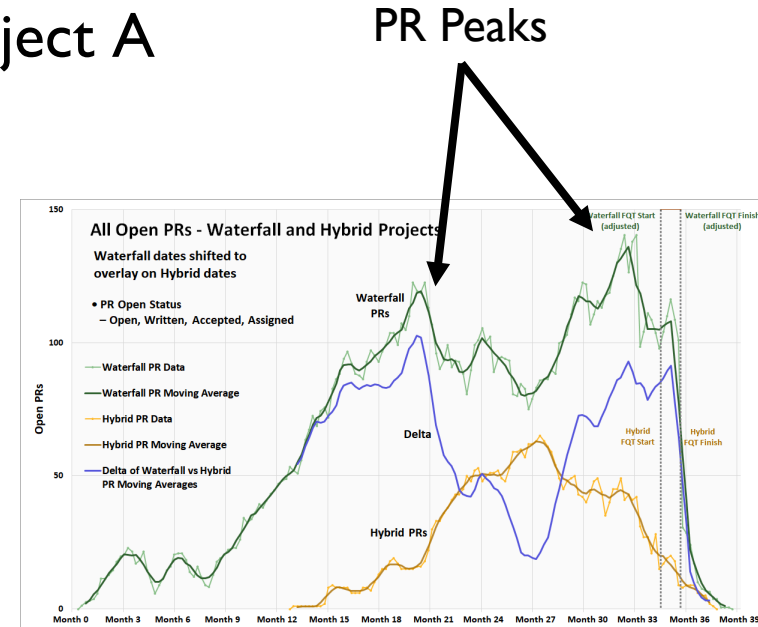


Project B: Comparison of PRs between hybrid waterfall vs hybrid agile



Discussion

- Overall PRs are lower for Project B (hybrid) than Project A (waterfall)
- **Observation I:** The two “PR peaks” for Project A reflect a situation where there were so many PRs identified during CIT that the team was overwhelmed and had to stretch the project out to allow time to address the PRs before completing CIT (the second peak) and then FQT.



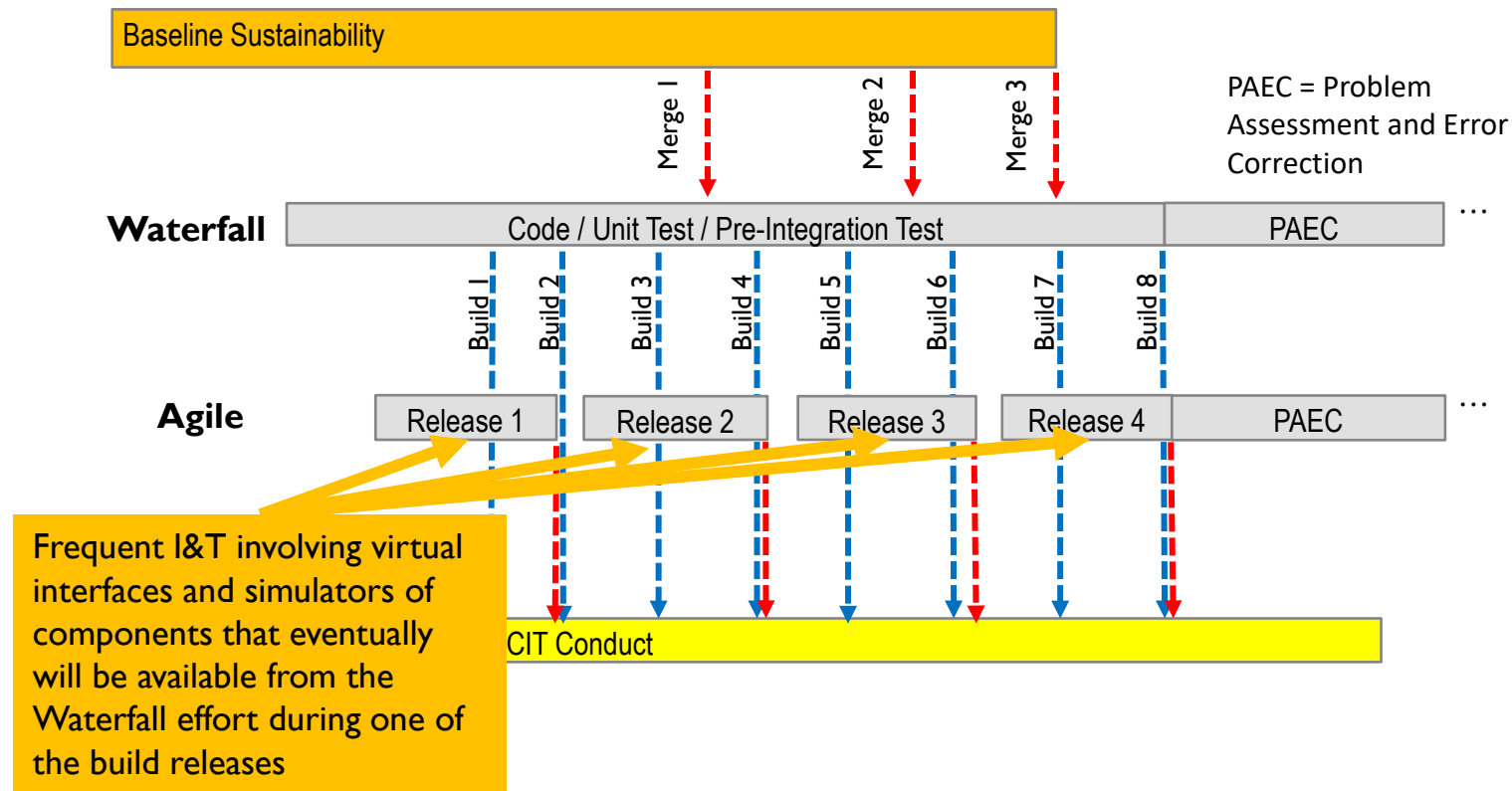
CIT: Component Integration and Testing
FQT: Formal Qualification Testing

Discussion (Cont.)

- **Observation 2:** For Project B, the waterfall and agile teams worked in parallel, with periodic “merges” that underwent integration and testing.
 - This helped reduce the “PR bow wave” because integration problems were discovered early during these “merge” events.
- **Observation 3:** For Project B, the agile team undertook frequent integration and testing between the “merge” events
 - This allowed almost continuous integration and testing which resulted in problems being identified early (i.e., reduced PRs), before CIT was officially started

Discussion (Cont.)

Project B Schedule:



Discussion (Cont.)

- **Observation 4:** In Project B (hybrid), the technical complexity of the waterfall components compared to the agile components were roughly the same
- **Observation 5:** In Project B (hybrid), the experience level of the agile development team was less than that of the waterfall team members (i.e., the agile effort included “ramp-up” time for the team)
- **Observation 6:** Despite having a less experienced team and a 10-month later start, the agile team produced fewer PRs during system development and testing than the waterfall team.

Summary of Projects A and B

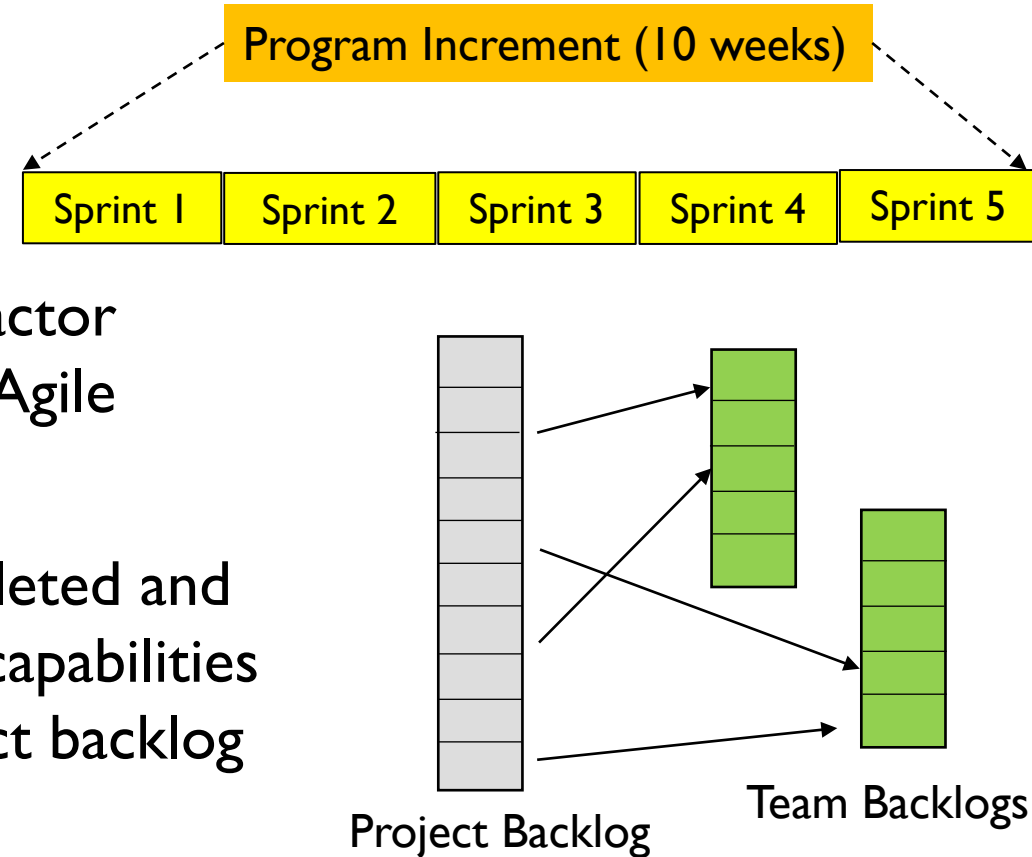
- Introduction of Agile/DevSecOps reduced PRs and helped keep the schedule and presumably costs from growing.
- Agile portion of Project B performed better than Waterfall portion of Project B despite starting later with a less experienced workforce
- Note:
 - Code complexity between Project A and B is the same
 - Code complexity and SLOC between the waterfall and agile portions of Project B were approximately the same
 - Cost data was not collected
 - The comparison of projects is only a single data point
 - There was limited engagement (“during development”) with the end-user community

Project C – A Study (No Software Development)

- **Study goal:** Undertake initial research into technical challenges, populate a project backlog and stand up an Agile/DevSecOps software factory (SWF) environment in preparation for **Project D**.
 - **Project D:** A new project to extend an existing waterfall-developed platform. Code complexity is very similar to projects A and B.
- Like Project B (hybrid), Project D exists within an acquisition management system that continues to rely on waterfall metrics (lines of code written/tested, number of PRs reported and worked off, EVM, IMS, etc.).

Project C Results

- An Agile environment was stood up and used throughout the study
- Due to the size of upcoming Project D, contractor implemented a nuanced version of the Scaled Agile Framework (SAFe®) environment.
- Modeling of technical issues successfully completed and system requirements were decomposed into capabilities and features (with some stories). Initial project backlog was populated
- **Training:** Contractor provided SAFe® training to contractor and government personnel. Government provided topic-specific training to government personnel (to help ramp up government team to Agile/DevSecOps work processes)



Project C: Conclusions

- **Training, Training and Training:** Foundational training (e.g., definition of Agile, SAFe®, Kanban, etc.) is not sufficient. *Tailored training focused on the nuances of the implementation of Agile/DevSecOps in the project is also necessary.*
- **Metrics:**
 - Very little software was developed (as planned)
 - The project focused on the creation of features (and some stories) and population of the project backlog.
 - Customized performance tracking tools needed to be developed due to the difficulty/cost of licensing equivalent software.
- **Government Involvement:** Increased considerably (e.g., scrums, ceremonies, refining, etc.). An open question is what is the proper balance (e.g., does government need to participate in every scrum?)

Next Steps

- Project D has started
- Work with government team to continue to address observations and apply lessons learned from the study (Project C).
- Continue collection of performance metrics with an eye towards velocity and related metrics.