USC Viterbi Information Sciences Institute

WRT-1012: Global Positioning Systems - Mission Engineering and Integration of Emerging Technologies

Sponsor: USSF Space and Missile Center

Dr. Michael Orosz (mdorosz@isi.edu)

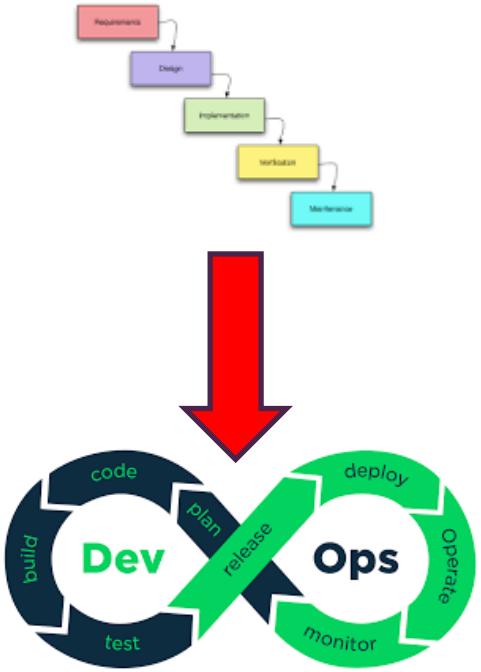
12th Annual SERC Sponsor Research Review
HOSTED VIRTUALLY ON: November 18, 2020
www.sercuarc.org

Outline

- Project Overview
- Test Environment
- Results
- Discussion
- Next steps

Project Overview:

- **Target:** 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

- **Partners**:
  - SERC
  - USC Information Sciences Institute (USC/ISI)
  - Georgia Tech Research Institute (GTRI)
- **Funding agency**: USSF Space and Missile Center's Production Corps (SMC/PC)
- **Period of Performance**: 26 June 2019 – 25 December 2020

USC University of Southern California
Information Sciences Institute

Georgia Tech Research Institute

SPACE & MISSILE SYSTEMS CENTER

UNITED STATES SPACE FORCE
DEPARTMENT OF THE AIR FORCE
MMXIX

The Test Environment:

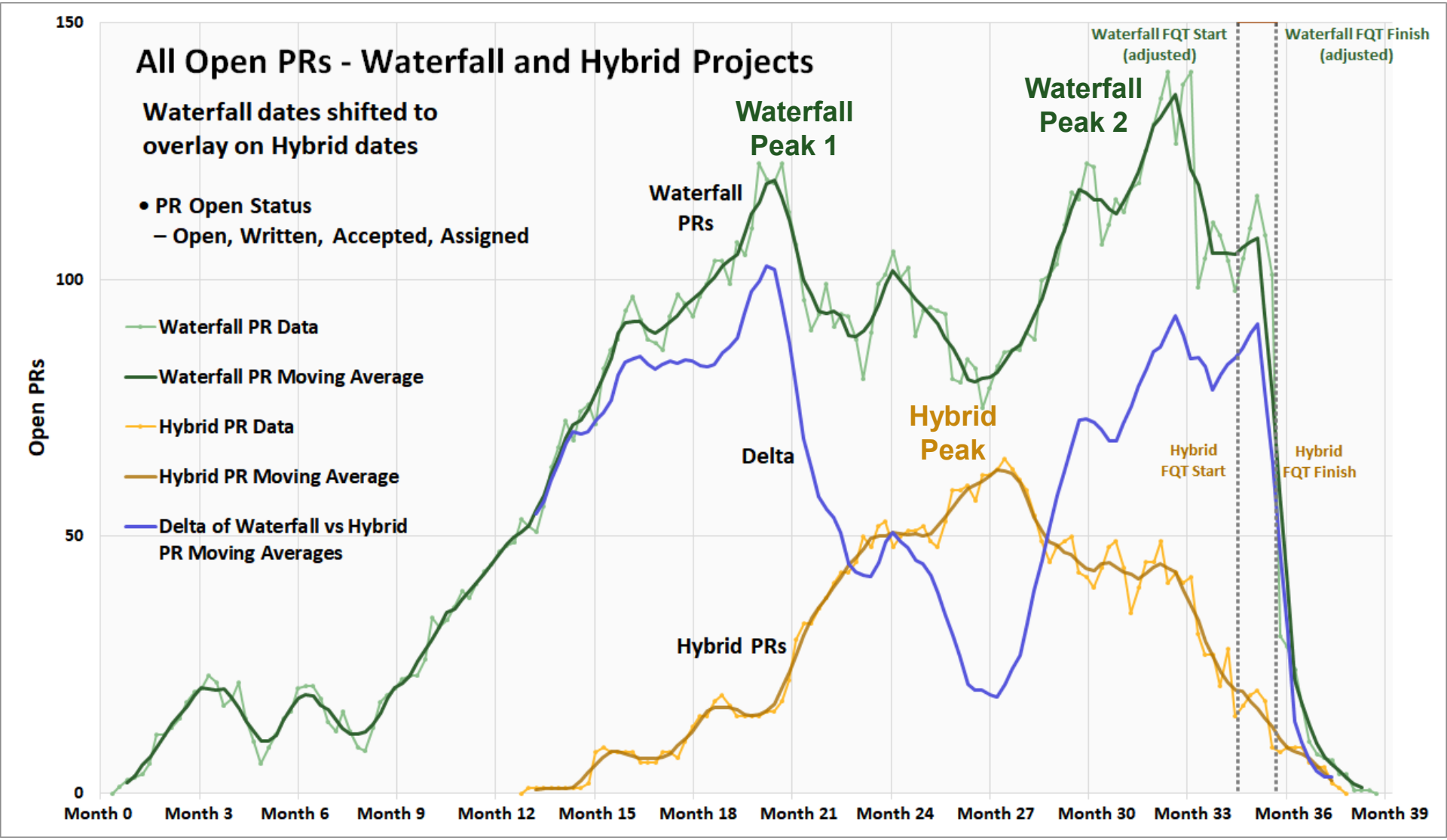
Two separate software development projects were developed as new components of an existing operational system.

- **Project A:** Traditional waterfall method used
  - Duration: 39 months (includes schedule extension)
  - Software lines of code (SLOC): 178K
- **Project B**: Hybrid composed of both waterfall and agile components
  - Duration: 25 months
  - Software lines of code (SLOC): 113K
- Software complexity of the two projects were gauged to be the same

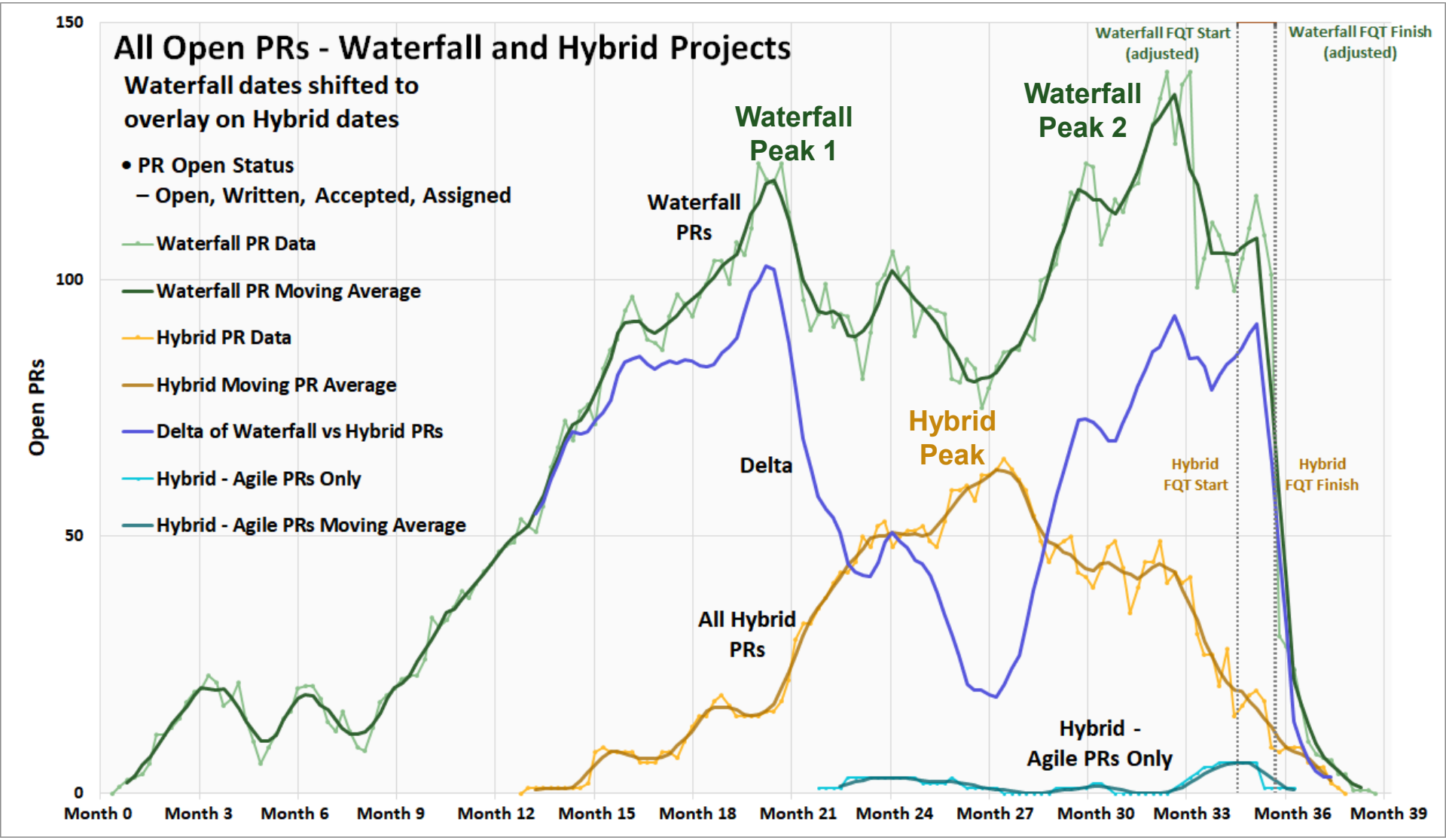
Analysis Approach:

- Problem reports (PRs) were collected upon the completion of software development.
- The number of PRs in an Open (unresolved) state were captured at the end of each week of software development for each project.
- The PR counts were plotted on a line graph to compare the development history of the two software development projects.
- The timeline of the waterfall project (Project A) was shifted so that the Formal Qualification Testing (FQT) period overlaid that of the Hybrid project (Project B).
- The PR counts of the waterfall project were proportionally reduced using the relative SLOC counts of the waterfall and hybrid projects.
- Adjusting the PR counts provides a more equivalent basis for comparing the productivity of the two development projects.

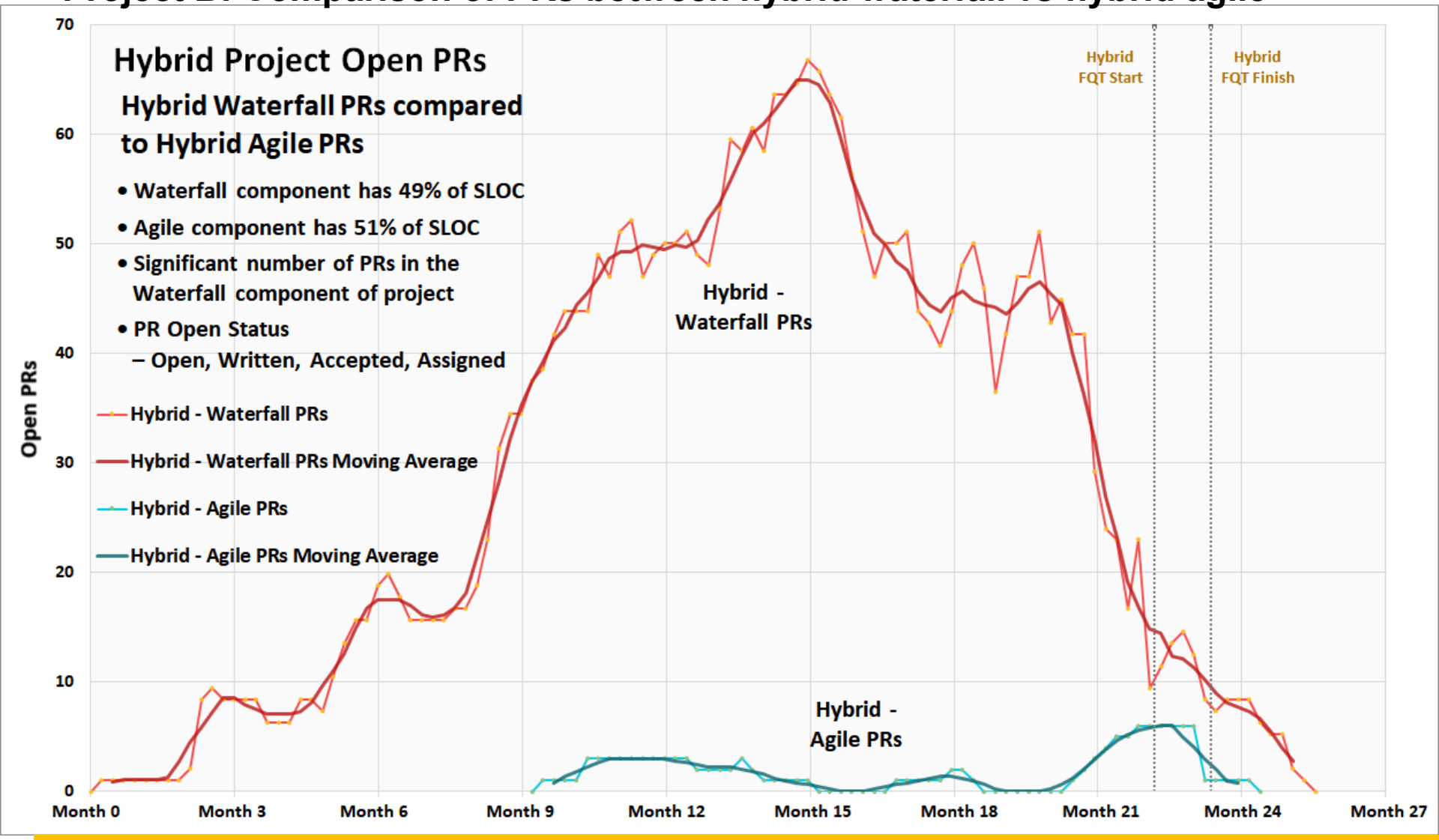
PR Comparison of Project A (Waterfall) and Project B (Hybrid)



Agile PRs of Hybrid Project Added to the Graph



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



Although the Agile component started 10 months later, the team completed the effort in less time with less PRs

Discussion:

- Overall PRs are lower for Project B (hybrid) than Project A (waterfall)...why?
- **Observation 1**: 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

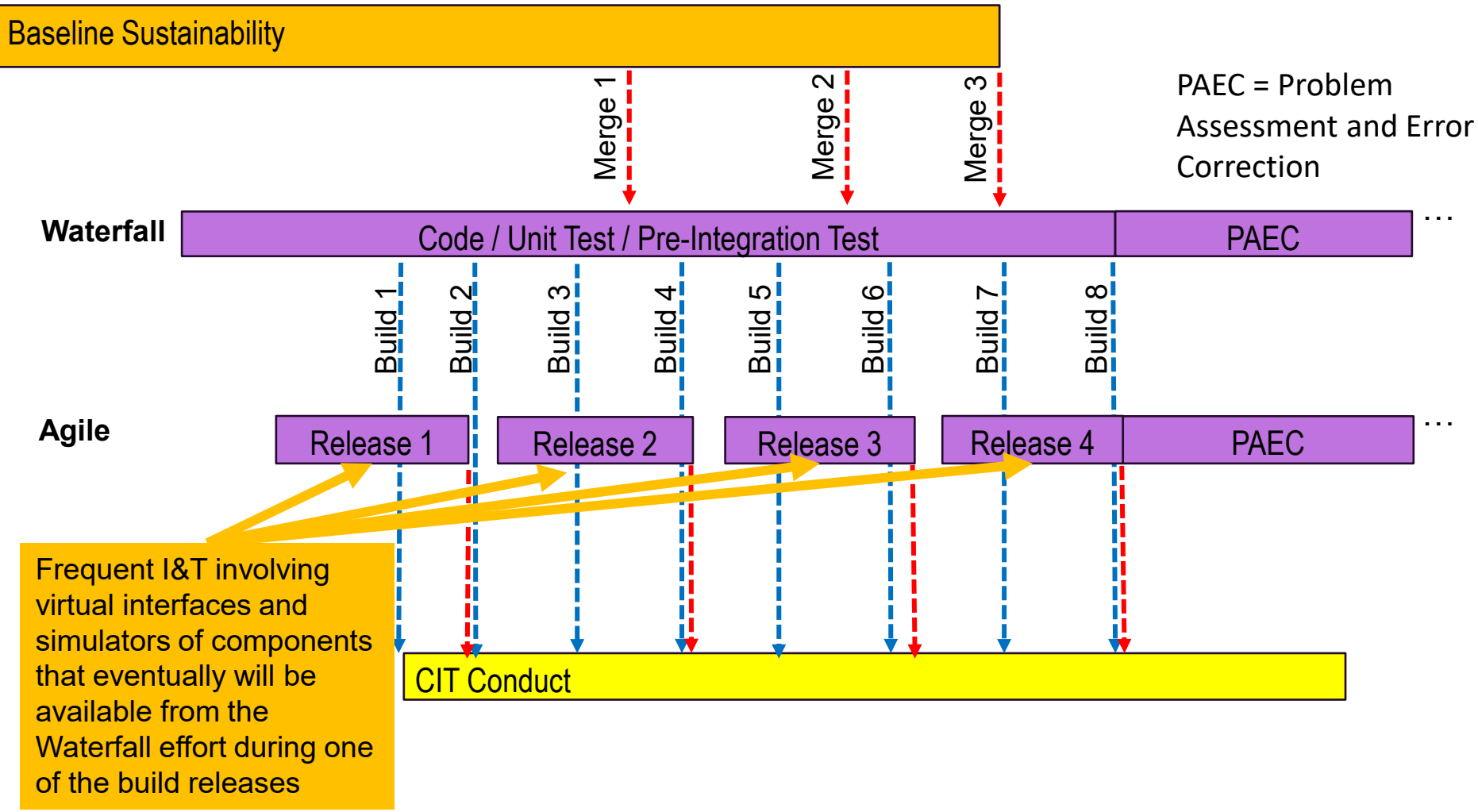
FQT: Formal Qualification Testing

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

Project B Schedule:



Discussion (cont.):

- **Observation 4**: In Project B, the technical complexity of the waterfall components compared to the agile components were roughly the same
- **Observation 5**: In Project B, 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.

Conclusion:

- Introduction of Agile/DevSecOps reduced PRs and helped keep the schedule and cost from growing.
- Some notes:
  - At this point, we only have data from two projects (i.e., Projects A and B). Data from other projects need to be collected
  - There was limited end-user engagement to influence future agile “builds”
  - The reduction of PRs in Project B (as compared to Project A) can be attributed to both the introduction of agile and to the introduction of the frequent “merge” events between the agile and waterfall efforts

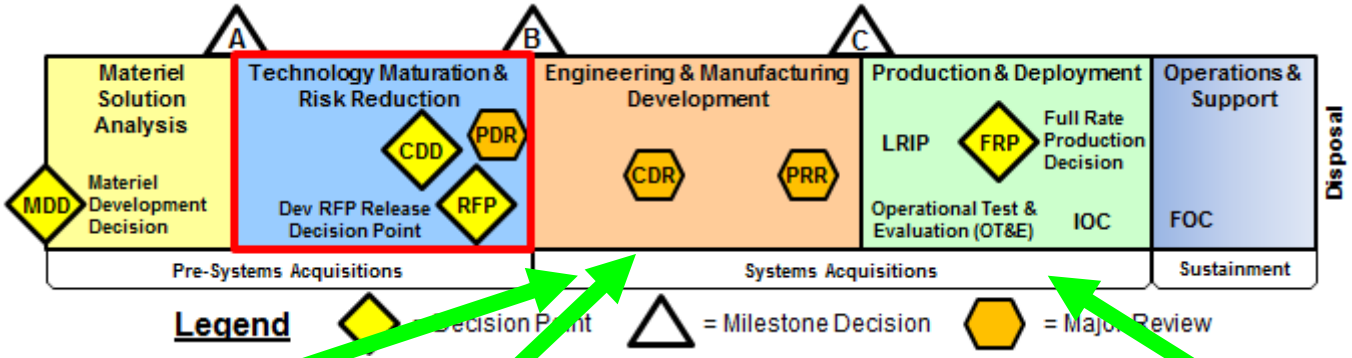
Next Steps: Project C:

- Project C is focused on implementing as much Agile/DevSecOps as possible in the overall project
- Target: A new project to enhance an existing platform that was developed using waterfall. Code complexity is very similar to projects A and B.
- Like the hybrid project, the new project exists within an acquisition management system that still relies on waterfall metrics (lines of code written/tested, number of PRs reported and worked off, etc.).

Project C: Initial areas of focus:

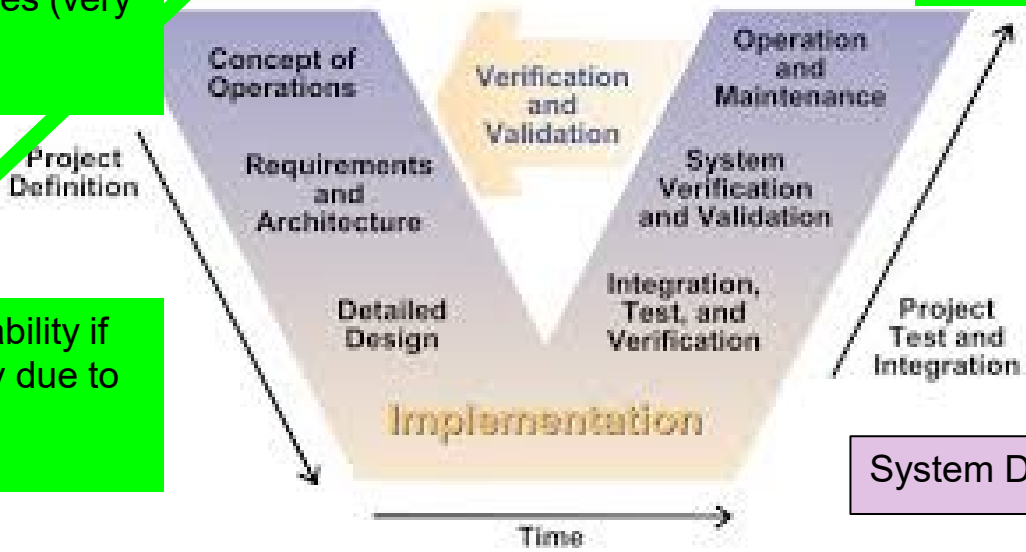
- **Translation**: Develop "Rosetta Stone" to map agile concepts to waterfall.
  - Contractor has selected SAFe framework for agile development
  - Each Program Increment (PI) can be viewed as a miniature "Development System V" found in the waterfall environment
  - Performance is measured by capability produced, not SLOC
  - Project backlog, velocity, capability vs. SLOC
- **Training**: Build training and "ramp-up" time upfront in the project...before Milestone B. Goal is to have a workforce ready for Agile/DevSecOps after Milestone B.
- **Customer engagement**: Government team becomes part of the development team. Participate in both PI and scrum planning, reviews and demonstrations

Refresher: Traditional Waterfall



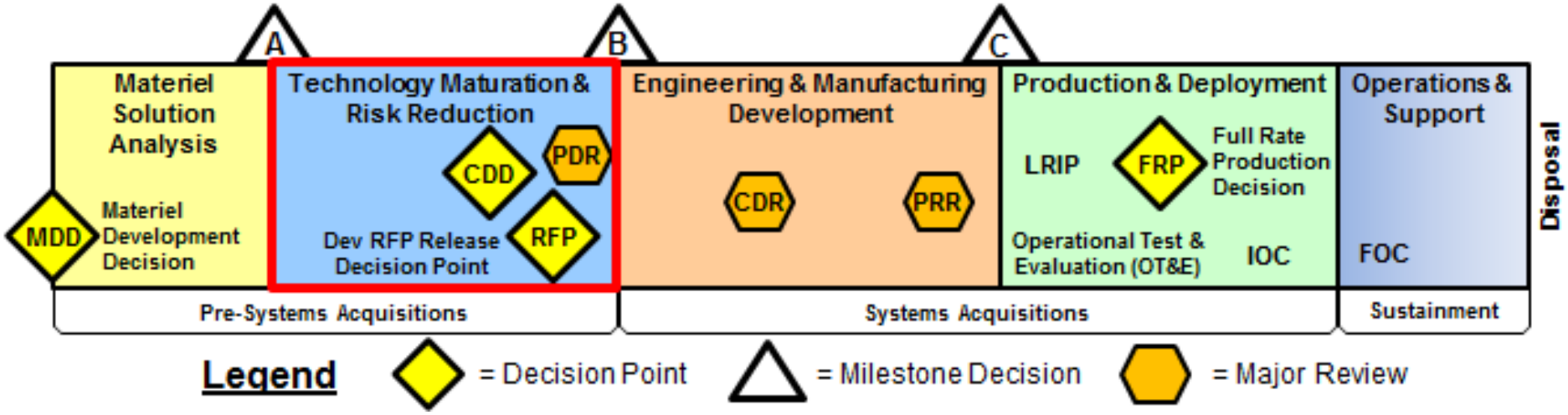
Very little opportunity to change requirements and priorities (very committed)

Capability is available here

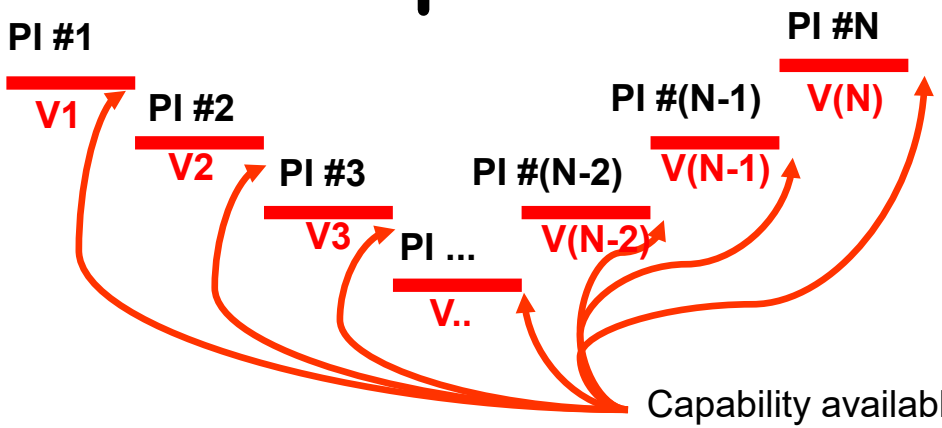


Very little realized capability if development ends early due to lack of funds

System Development “V”



- Divides the effort into multiple ***mini system development "Vs"***
- **These mini Vs** are call **Program Increments (PI)** and include development, integration and testing
- Capability is produced at the end of each PI. In many cases, this capability could be released and available to the end-user



Questions?