SERC RESEARCH REVIEW 2023 | NOVEMBER 15, 2023

Space Systems Command (SSC) Military Communications & Positioning, Navigation and Timing Directorate – Mission Engineering and Integration of Emerging Technologies

WRT-1069

PI: Michael Orosz (USC/ISI)

Collaborators: Capt David Pettis (USSF), Brian Duffy (USC/ISI), Craig Charlton (USC/ISI) and Fred Fritzen (Aerospace Corp)

Sponsor: US Space Force Space Systems Command Military Communications and Position, Navigation and Timing Directorate (SSC/CG)

USC Viterbi
School of Engineering
Information Sciences Institute

Agenda

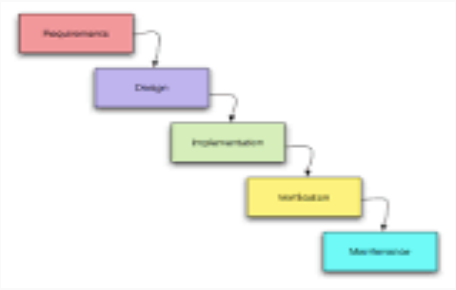
- Project Objectives and Timeline
- Brief Overview of Previous Results (WRT-1041)
- Current Project Status and Observations (WRT-1069)
- Next Steps (WRT-1069)

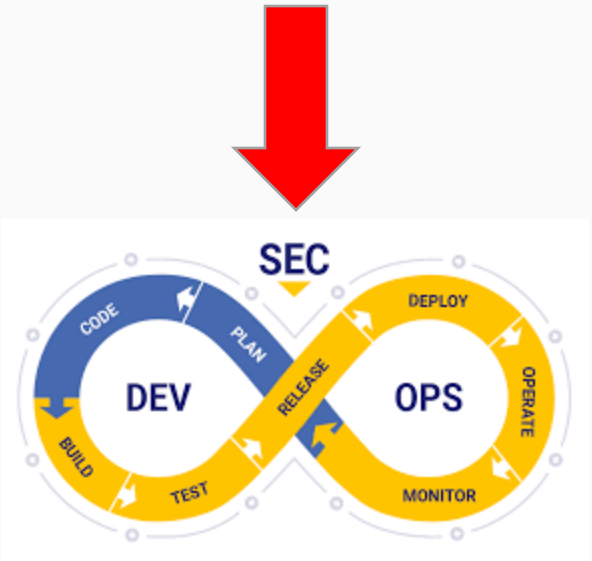
Project Overview

- **Funding agency**: U.S. Space Force and Space Systems Command Military Command and Positioning, Navigation and Timing Directorate (SSC/CG)
- **Team**:
  - —USC Information Sciences Institute (USC/ISI)
  - —SERC
- **Period of Performance**: August 2016 – January 2024 + Four One-Year Options

Objectives

- **Improve DoD competitiveness**: Specifically - improve existing DoD space-based *software* system acquisition processes
- **Goals**:
  - Determine the mission engineering methods, analysis, and metrics to transition from traditional DoD 5000 waterfall development environments 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 ...including workforce training

3. Incorporate processes and "lessons-learned" into a transition process to apply to other domains

Four DoD Acquisition Projects

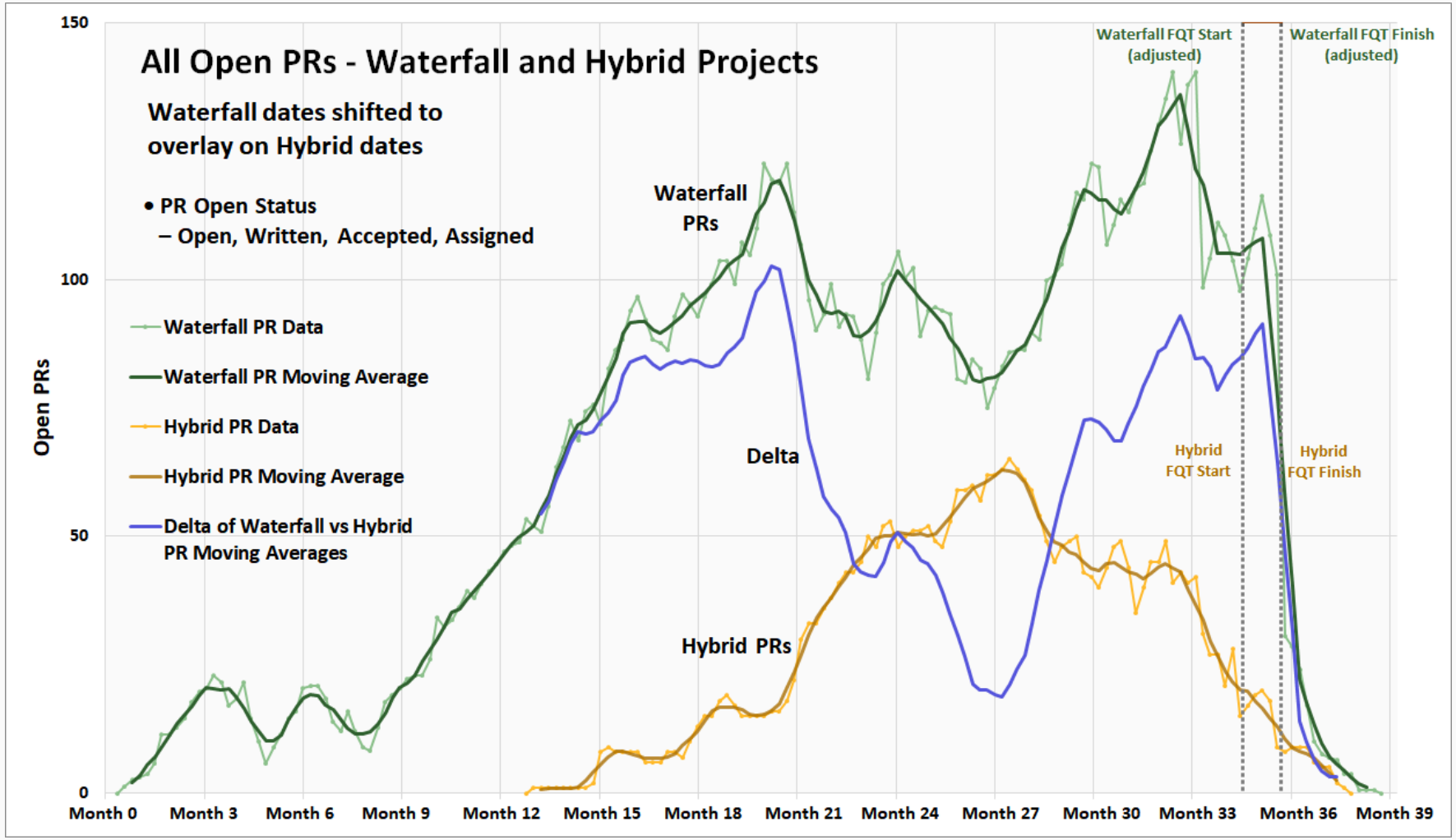
- **Project A:** Traditional waterfall method used (completed)
  - —Duration: 39 months (includes schedule extension)
  - —Software lines of code (SLOC): 178K

Baseline

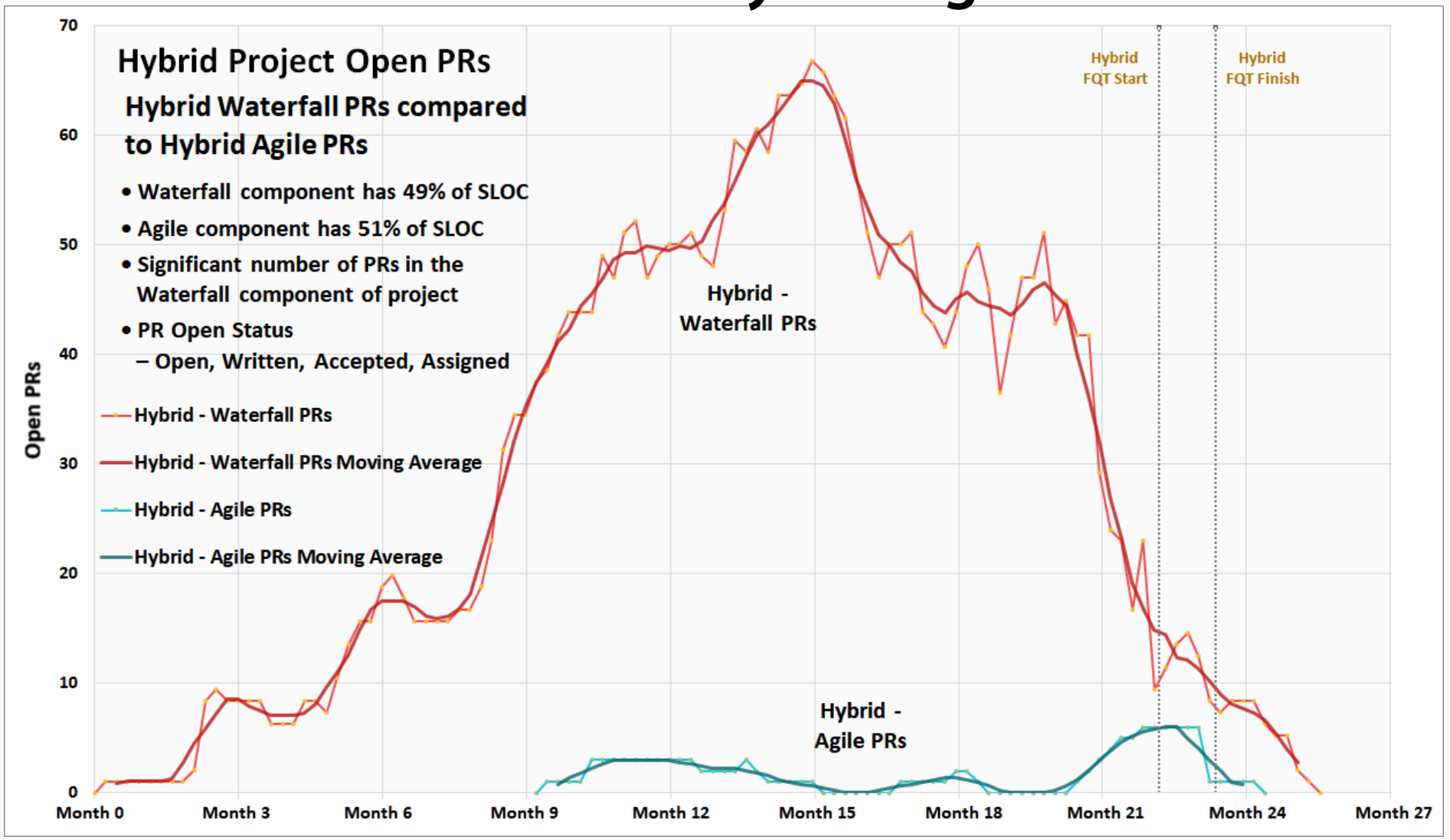
- **Project B**: Hybrid composed of both waterfall and agile/near continuous integration processes (completed)
  - —Duration: 25 months
  - —Software lines of code (SLOC): 113K
- **Project C**: Undertake technical explorations and stand up agile/DevSecOps environment in preparation for Project D (completed)
  - —Duration: 15 months
  - —Software lines of code (SLOC): None
- **Project D**: Agile/DevSecOps (**In Progress for 28 months**)
  - —Duration: Approximately 52 months
  - —Software lines of code (SLOC): TBD

Projects A & B – Example of the Benefits of Agile and Continuous Integration (WRT-1041)

Problem Report (PR) Comparison of Project A (Waterfall – *The Baseline*) and Project B (Hybrid)



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



Project C – A Study (No Software Development) WRT-1041

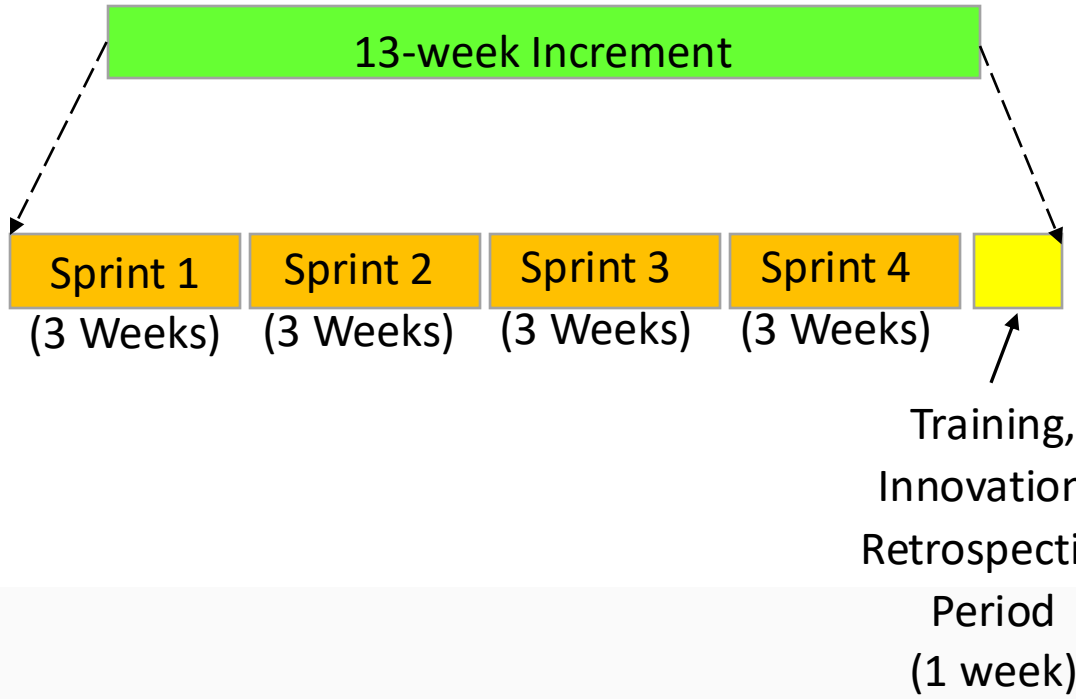
- **Study goal**: Undertake initial research into technical challenges, populate a project backlog, and stand up an Agile/DevSecOps software factory (SWF) environment in preparation for **Project D**.

  - **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 D (WRT-1041 & WRT-1069)

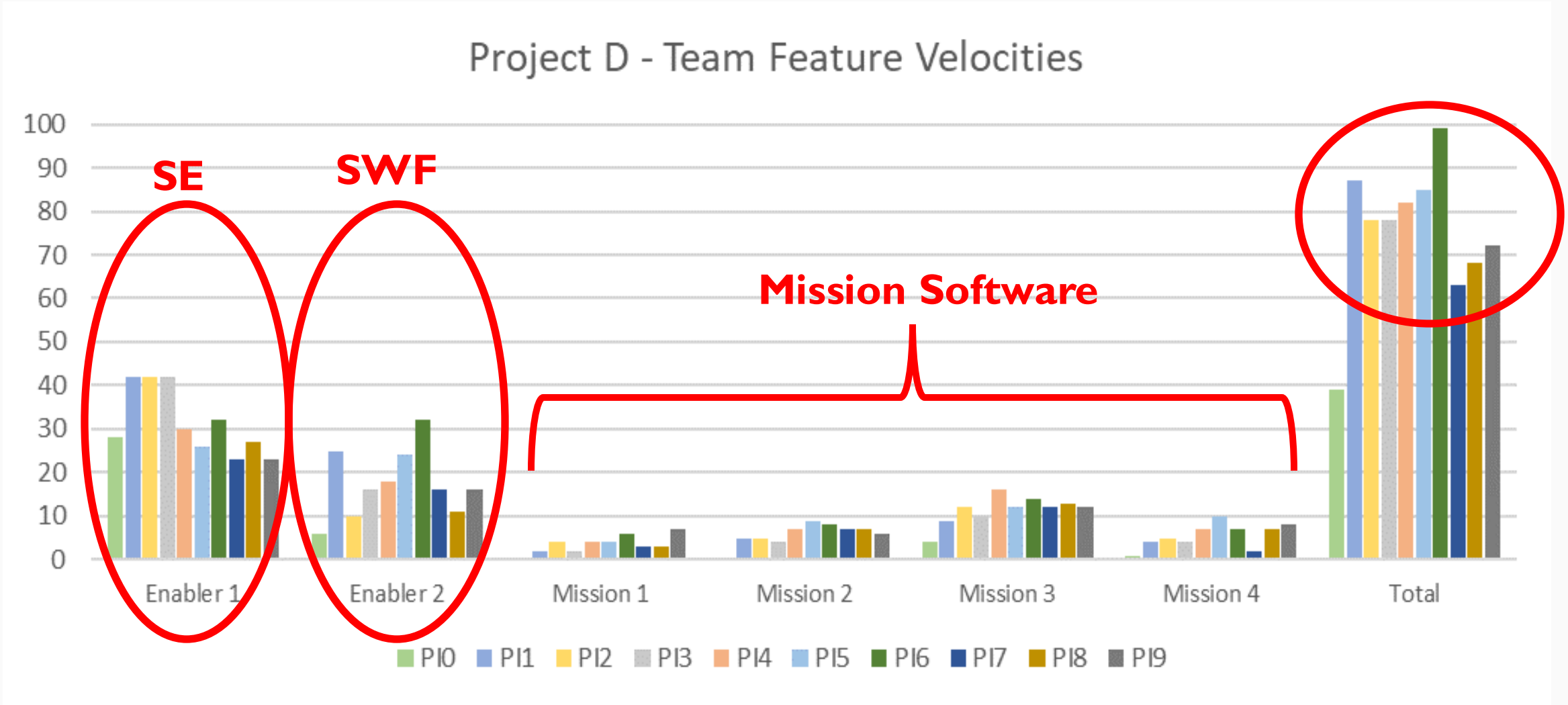
- Hybrid project
  - —Roughly 70% agile / 30% waterfall (mainly in the programmatic area)
  - —Duration: 52 months (currently in month 28)
  - —Software lines of code (SLOC): Not yet known
- Agile implementation
  - —Method: Modified SAFe® implementation
  - —Program Increment (PI): 13 weeks in duration with four 3-week sprints
  - —Last week of PI reserved for demonstrations, training, innovation and if necessary, "catching –up"
  - —Six scrum/sprint teams (4 are mission-focused teams, 2 are enabler teams)

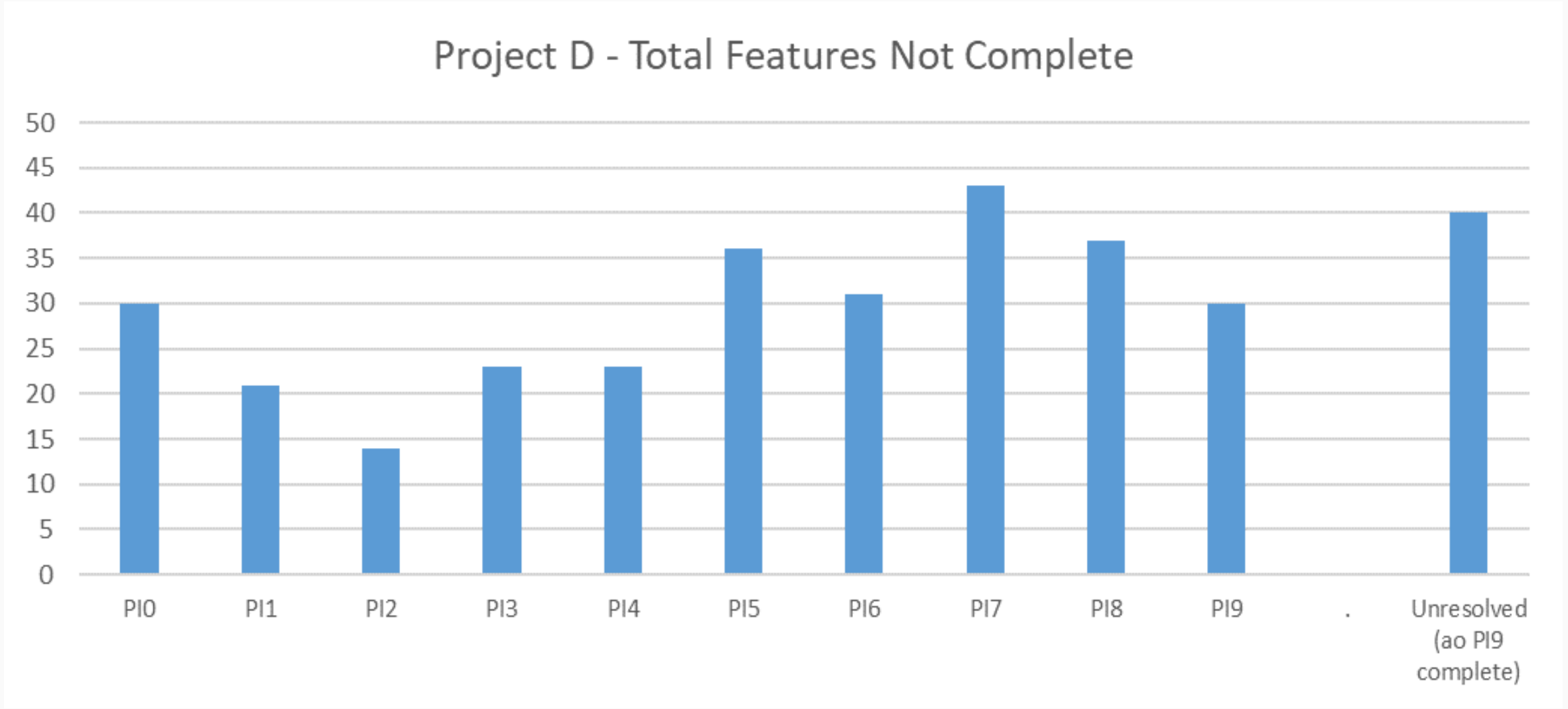


Starting with PI5, one of the enabler teams was split into three teams – producing 8 total teams.

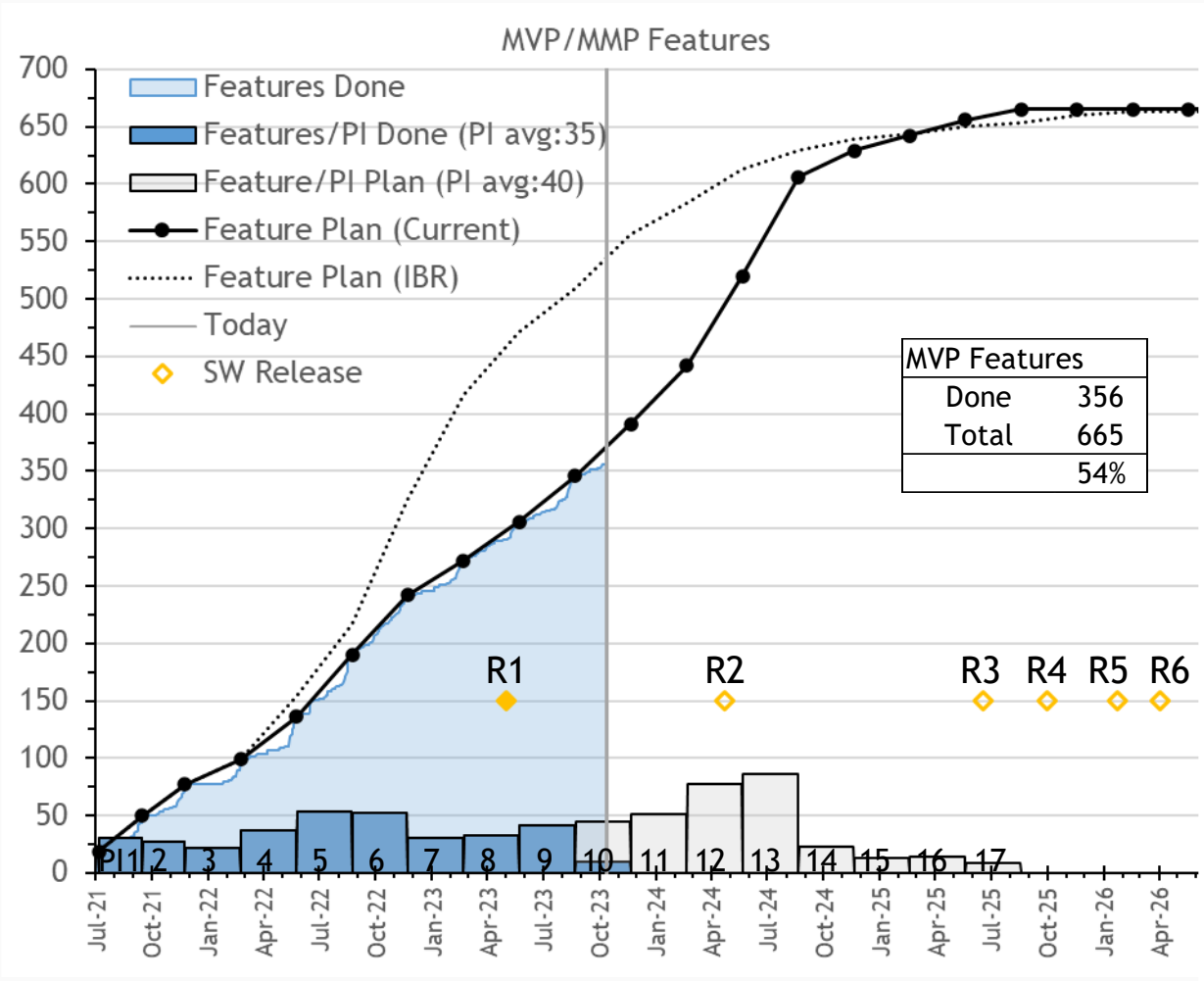
Results to Date

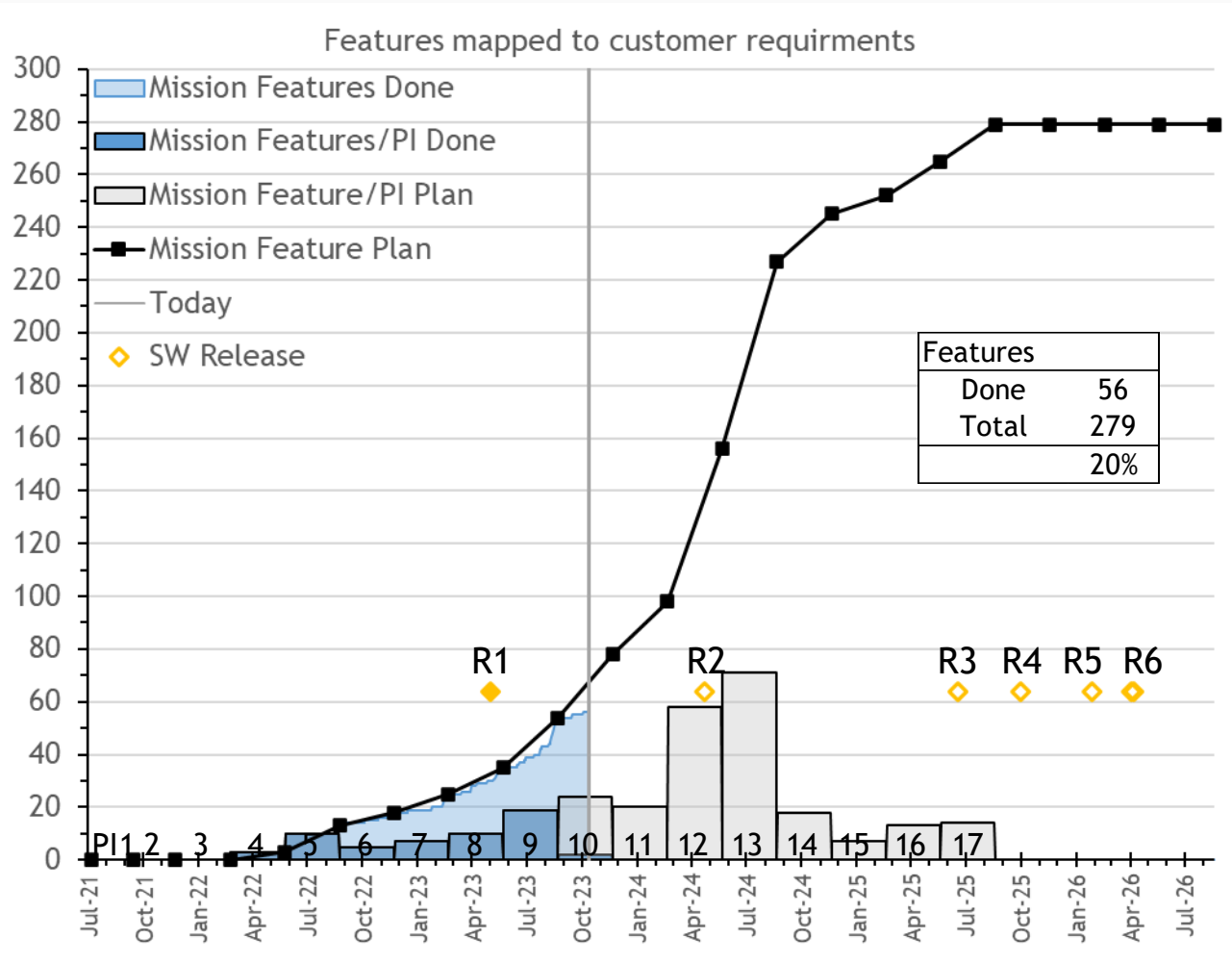
- Performance metrics being tracked:
  - Requirements → Features → Stories
  - Features completed and not completed within a PI (goal: complete features within a PI)
  - Stories completed and not completed within a sprint (goal: completed stories within a sprint)
- Key observation: stories and features spill over boundaries (spillage)
- Velocities stabilized, but are now starting to become volatile


Project D - Team Feature Velocities
100
90
80
70
60
50
40
30
20
10
0
SE
SWF
Mission Software
Enabler 1
Enabler 2
Mission 1
Mission 2
Mission 3
Mission 4
Total
PI0
PI1
PI2
PI3
PI4
PI5
PI6
PI7
PI8
PI9

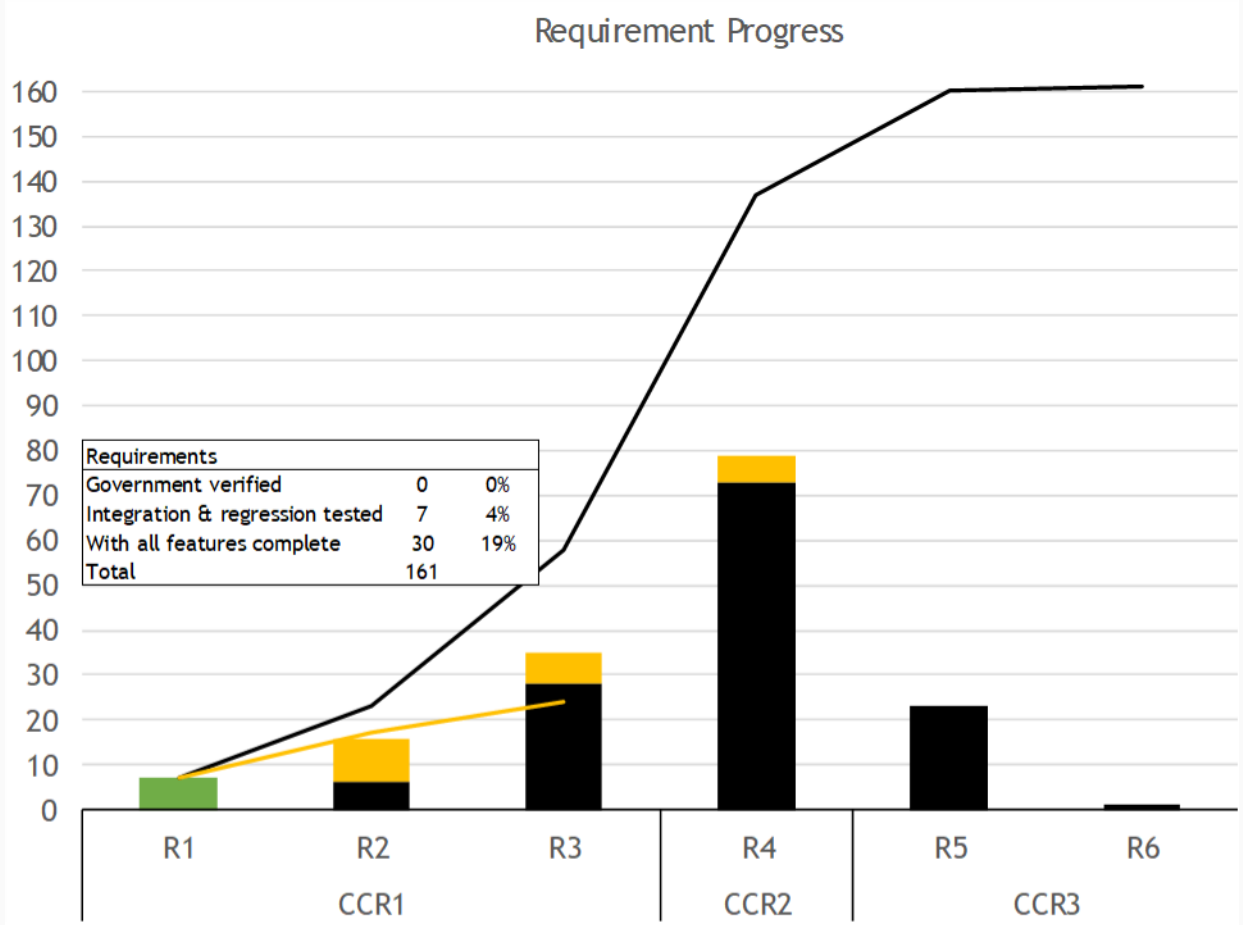

Project D - Total Features Not Complete
50
45
40
35
30
25
20
15
10
5
0
PI0
PI1
PI2
PI3
PI4
PI5
PI6
PI7
PI8
PI9
.
Unresolved (ao PI9 complete)

Tracking Features – All MVP/MMP vs. Mission MVP/MMP





Tracking Features – Features vs. Requirements



	R1	R2	R3	R4	R5	R6
Government verified	0	0	0	0	0	0
Rqmts int & reg tested	7	0	0	0	0	0
Rqmts w/features complete	0	10	7	6	0	0
Rqmts remaining	0	6	28	73	23	1

- Requirement completion plan and progress.
- For each planned release (i.e., R1, R2,…R6) a histogram of requirements remaining
  - all contributing features completed, integrated and tested, and verified by the customer
- **Solid black line**: Current cumulative requirements completion plan

Some Reasons for the Spillage

- Blockages
  - Lack of resources (software license issues, external dependencies, test facilities etc.)
- Competition for staff
  - In many cases, team members work multiple projects and can be "pulled" depending on project priorities
- Underestimating code complexity
  - Some of this can be attributed to "discovery"
  - Can also be attributed to a lack of understanding of the system requirements
- Changing priorities of features and stories due to various issues

Discussion Points

- Often in agile environments, there is an assumption that all developers and systems engineers are equivalent in skill and experience.
  - **Assumption:** simply replace a team member with another individual with the same experiences (i.e., a form of "plug and play").
  - **Reality**: This is far from true ***on specialized software projects***. New team members may need time to become familiar with the system
  - **Reality**: New team members – regardless of skill and experience – need support from a subject matter expert with knowledge of the project. This is difficult to support when the SME is pulled to another program.

Discussion Points

- Traditional project performance metrics often lag (by months, in many cases) the reality of the agile program.
  - **Earned Value Management (EVM)**. These metrics almost always lag actual values by months. This is due to the need for the prime contractor to collect and certify numbers and then the government requires time to review and analyze the information
  - **Integrated Master Schedule (IMS)**. The IMS almost always lags the current state of the project by months. This is primarily due to the length of time it requires to update the IMS to reflect the realities of the project.

Discussion Points

- Lack of full visibility into relationship between cost and the SoW.
  - **T-Shirt Sizing:** Commonly used in estimating (in hours) work (at least in the SAFe environment).
  - **Problem:** Difficult to determine which portion of the t-shirt is assigned to development and which is assigned to integration & testing and other tasking
  - **Problem:** T-shirt sizes come in discrete sizes (e.g., extra small, small, medium, large, extra large) with no support for partial sizes that may be more applicable to the work that needs to be completed.
  - **Problem:** Not clear of the labor breakdown within a t-shirt size (e.g., senior engineering hours vs. entry-level engineering hours, etc.)

Recommendations

1. Perform upfront engineering to help populate the project backlog, map features with compliance requirements, and to identify dependencies as early as possible in the program
2. The need to establish (early in the program) a near operational environment and high-fidelity simulators (for horizontal I&T)
3. Allocate stories to sprints at the beginning of a program increment
4. Plan margin into the sprints to handle unexpected events such as new technology insertion and/or unexpectedly complex stories
5. Focus on MVP/MMPs during Program Increment (PI) planning
6. Programmatic issues: get licensing, IP, accreditation, certification and other programmatic issues resolved early

Recommendations (Cont.)

7. Need for on-board and continuous training to ensure team members (both the contractor and acquisition team) are on the same page
8. Be prepared to customize performance tracking tools
   - Applies to all teams…government and development contractor
   - Issues:
     - Software incompatibilities
     - Foreign ownership of tools
     - Access challenges (e.g., VPN, security concerns, etc.)

Next Steps

- Work with government team to continue to address observations and apply lessons learned from the study (Project C) and initial phases of Project D.
- Continue collection of performance metrics with a focus on velocity and related metrics.
- Explore strategies to mitigating the challenges of using EVM, IMS t-shirt sizing and other cost and schedule performance tracking metrics within an Agile program.
- Continue developing/refining training materials and processes
- Join other projects to collect data and provide SME services

Thank you

Stay connected with SERC Online:

Email the presenter: Michael Orosz

mdorosz@isi.edu

Email the research team:

Michael Orosz

Michael.orosz.ctr@spaceforce.mil