



Lessons Learned in Implementing DoD Space-based Agile/DevSecOps Programs

Michael D. Orosz¹, Lt Col Jacob Hempen², Lt Col John M. Turner², Brian Duffy¹ and Craig Charlton¹

¹USC Information Sciences Institute

310-486-2150

mdorosz@isi.edu

²United States Space Force

310-653-2547

jacob.m.hempen@spaceforce.mil

27 Jan 2025

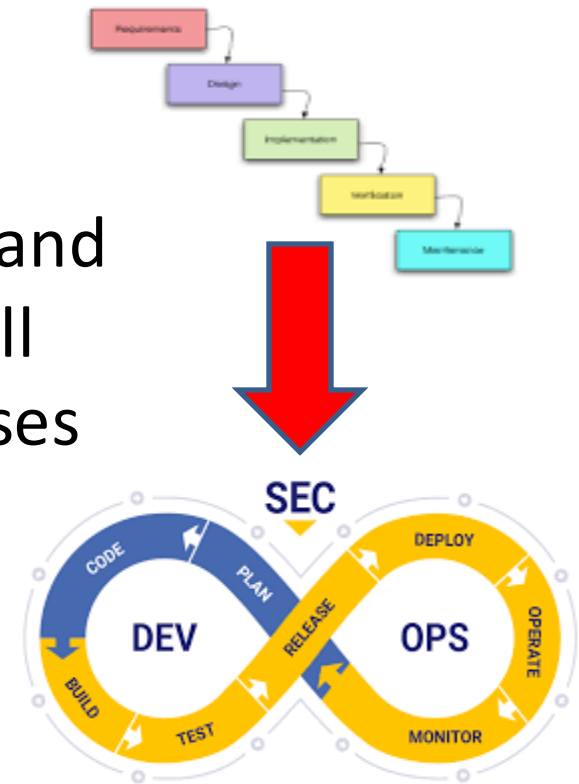
© 2025 by University of Southern California. Published
by The Aerospace Corporation with permission.



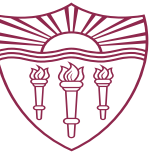


Overall Project Objectives

- **Improve DoD competitiveness:** Specifically - improve existing DoD space-based software acquisition pathway – *but generalize as needed for other domains*
- **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.



Four DoD Acquisition Projects



Baseline

- **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/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 42 months**)
 - Duration: Approximately 52 months
 - Software lines of code (SLOC): 100K-150K

Project D

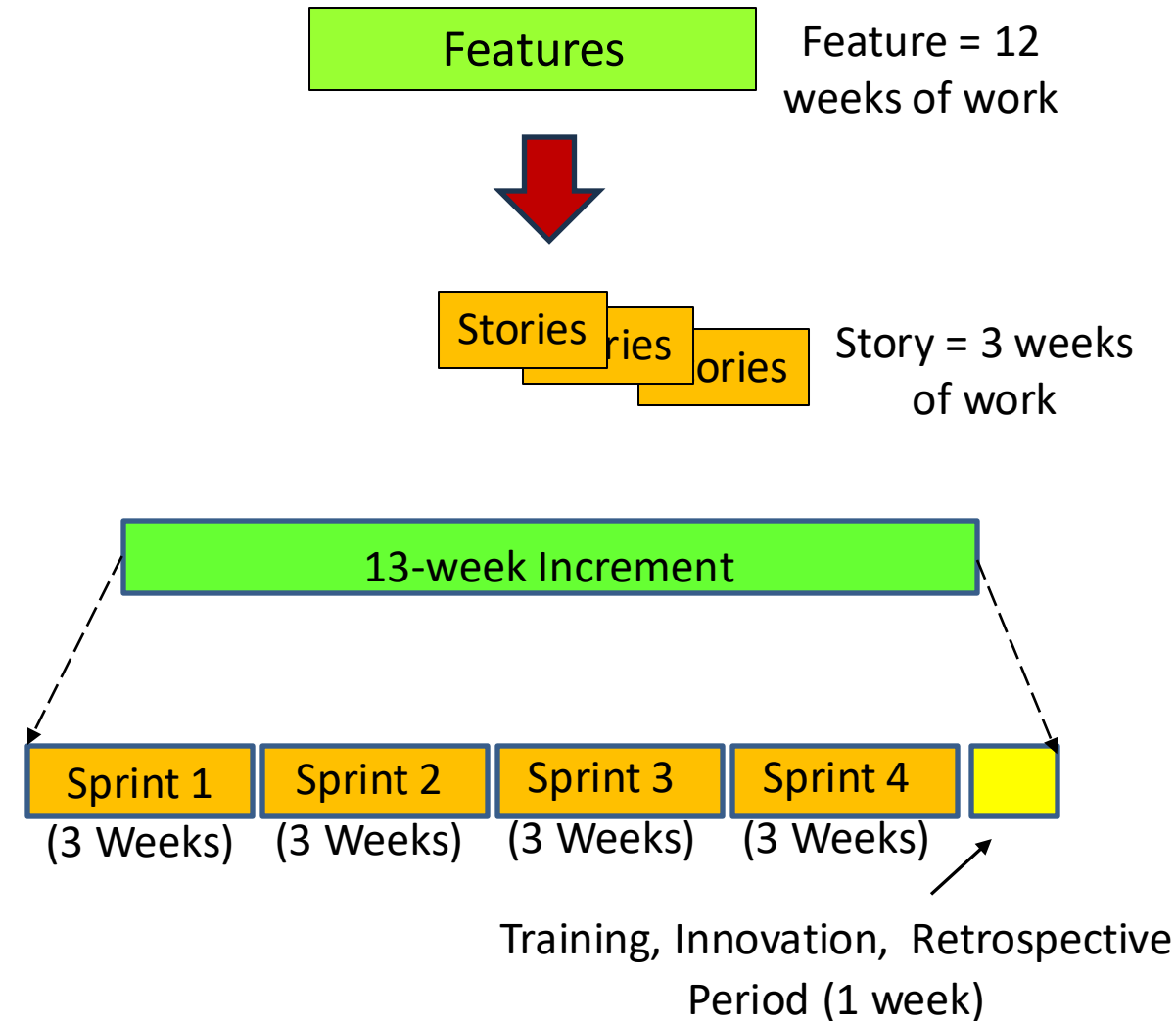


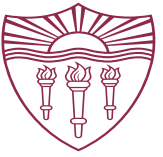
- Hybrid project

- Roughly 70% agile / 30% waterfall (mainly in the programmatic area)
- Duration: 52 months (currently in month 42)
- Software lines of code (SLOC): ~100K-150K

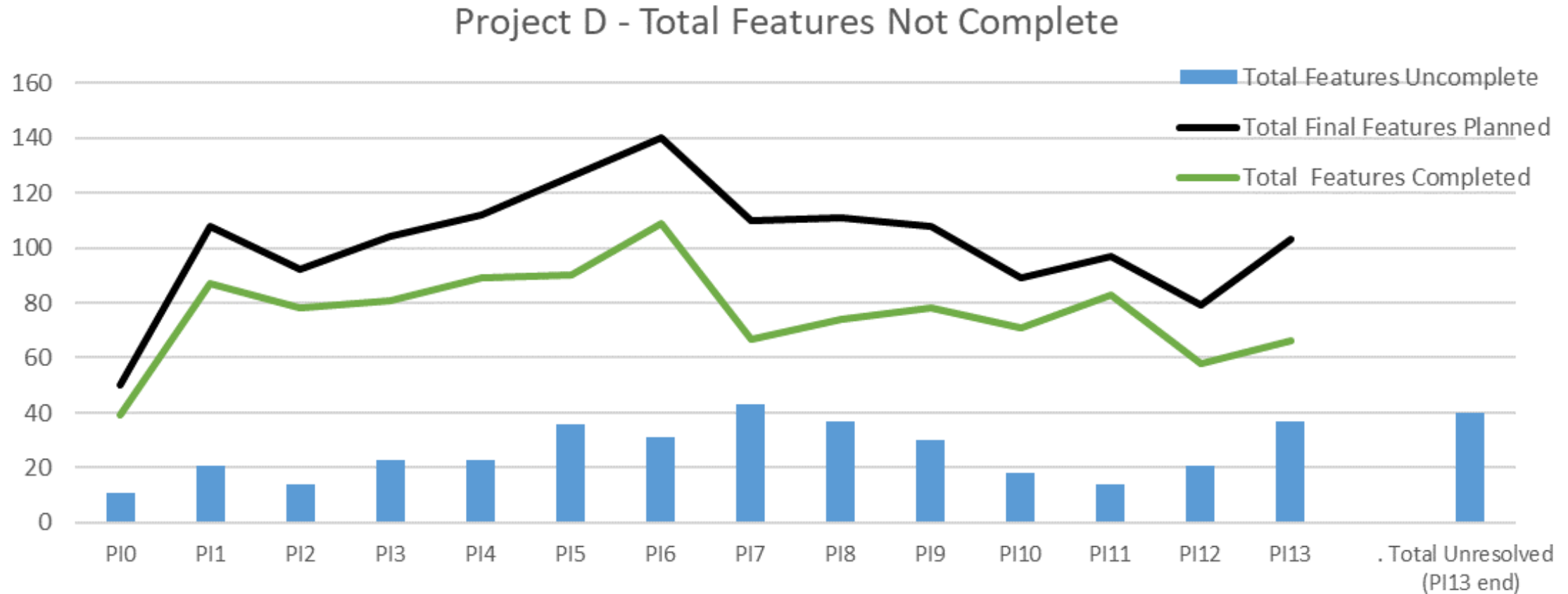
- Agile implementation

- 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)

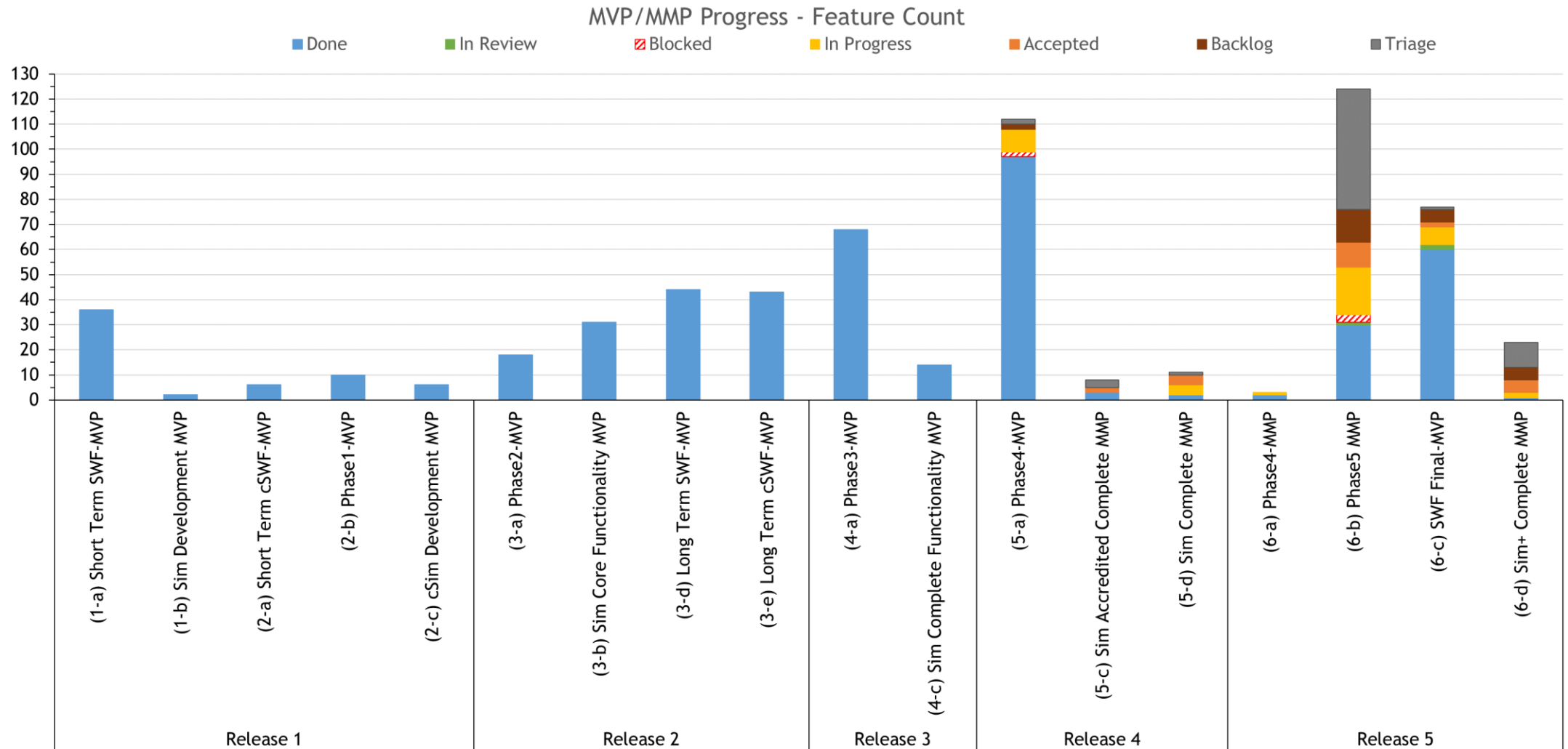
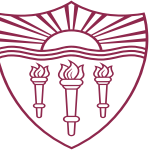




Features Slipping from PI to PI

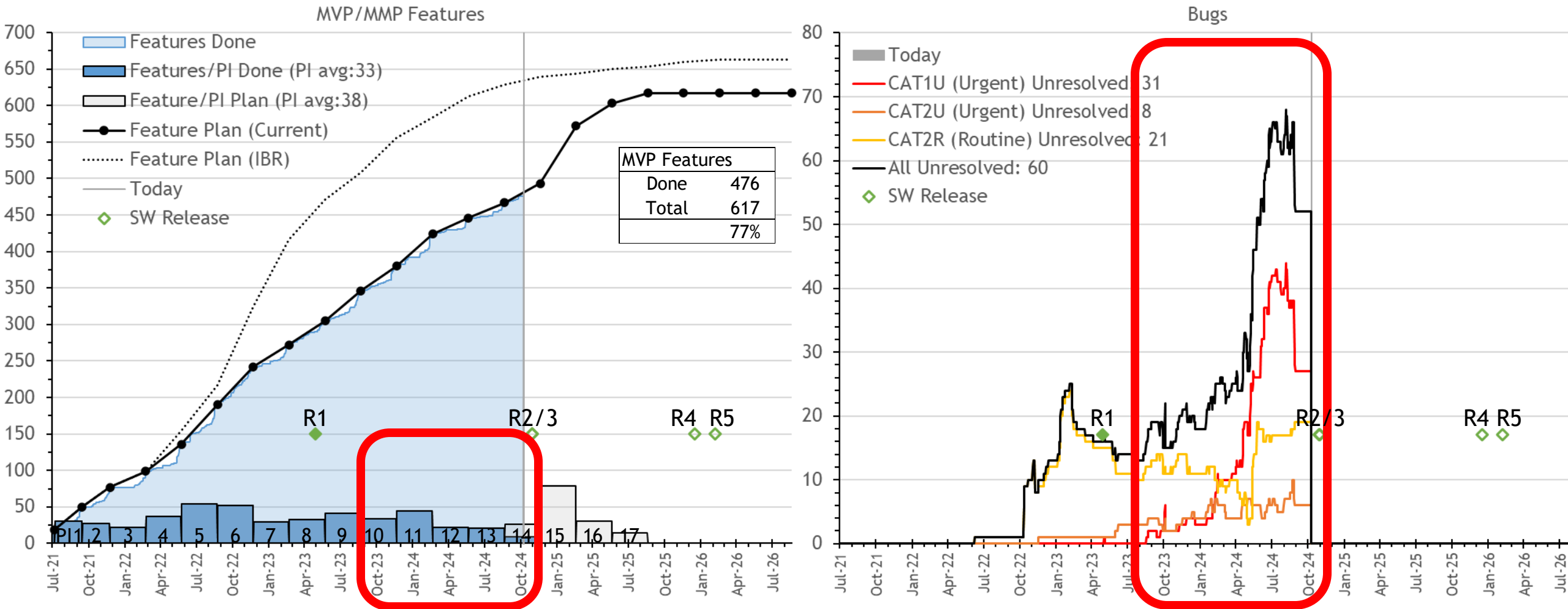


Are We Delivering Value Despite Features Slipping to Future Pls?





Bow Waves in Agile



"Bugs" discovered during testing

Information Science A bow wave and we're running out of runway!

USC Viterbi
School of Engineering

Some Reasons for the Building of a Bow Wave

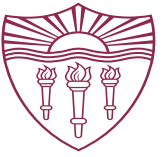


- Blockages
 - Lack of resources (software license issues, external dependencies, test facilities etc.)
- Competition for workforce
 - In many cases, team members work multiple projects and can be “pulled” depending on project priorities (new hires = expensive learning curve)
- Underestimating code complexity
 - Some of this can be attributed to “discovery”
 - Can also be attributed to a lack of understanding of the system requirements
- **Bugs & DRs due to delayed integration and testing of the system**
 - The high priority bugs & DRs consume workforce effort and crowd out mission features which are pushed to the “right.”



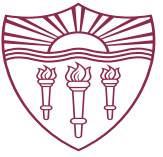
Recommendations/Observations

- 1. Upfront engineering:** Perform some upfront engineering to help populate the project backlog, map features with compliance requirements, ***identify test cases***, and to identify dependencies as early as possible in the program.
- 2. Not all staff skillsets are equal:** The amount and diversity of talent needs to be constant, so they can be moved/re-arranged to react to the changing needs (be agile) of the program.
- 3. Near operational test environments:** Establish (early in the program) a near operational environment and high-fidelity simulators (for horizontal I&T)
- 4. Sprint Margins:** Plan margin into the sprints to handle unexpected events such as new technology insertion and/or unexpectedly complex stories (**Agile 101**).
- 5. Licensing and other IP issues:** get licensing, IP, accreditation, certification and other programmatic issues resolved early.



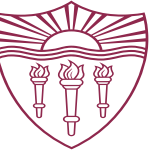
Recommendations/Observations (Cont.)

6. **Training:** Need for on-board and continuous training to ensure team members (both the contractor and acquisition team) are on the same page
7. **Costing Challenges:** For projects that are costed using “T-shirt sizes, costing needs to be segmented into development, integration and testing, and bug (problem) work-off. Also need sufficient number of T-shirt sizes to cover smallest to largest work packages
8. **Custom tools:** 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.)
 - EVM lags actuals – sometimes by months.



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



Questions?

Mike Orosz

mdorosz@isi.edu

michael.orosz.ctr@spaceforce.mil

310-486-2150