

TRACKING PERFORMANCE IN A DOD AGILE/DEVSECOPS ENVIRONMENT

Michael Orosz, Research Professor and Director,
Lt Col Jacob Hempen, USSF
Lt Col John Turner, USSF
Brian Duffy, Senior Systems Engineer,
Craig Charlton, Senior Systems Engineer

Contact:

mdorosz@isi.edu

Michael.Orosz.ctr@spaceforce.mil

University of Southern California Information Sciences Institute
Marina del Rey, CA
October 2023



Agenda

- Project Objectives and Timeline (Review)
- Performance Metrics
 - Observations
 - Solutions
- Programmatic observations
- Next steps

Project Overview



SYSTEMS
ENGINEERING
RESEARCH CENTER

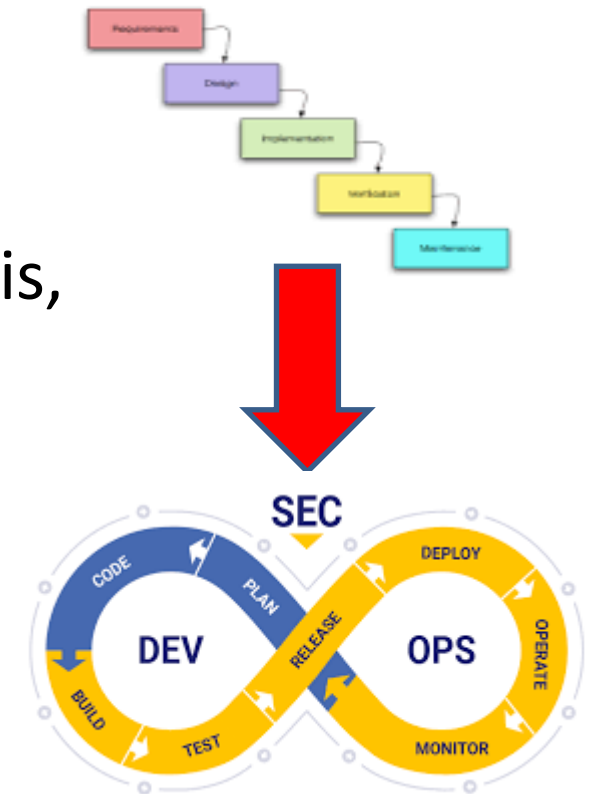
- **Team:**
 - USC Information Sciences Institute (USC/ISI)
 - USSF SSC
 - SERC
- **Funding agency:** U.S. Space Force and Space Systems Command Military Command and Positioning, Navigation and Timing Directorate (SSC/CG)
- **Period of Performance:** August 2016 – October 2023 (via multiple SERC projects)



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
 - Promote the 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 for applying 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
- **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: 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

Baseline

Performance Metrics

1. Determine metrics to collect for highly agile projects
 - Emphasis is hybrid waterfall/agile projects
2. Determine how to collect for highly agile projects
3. Develop analytical tools and processes

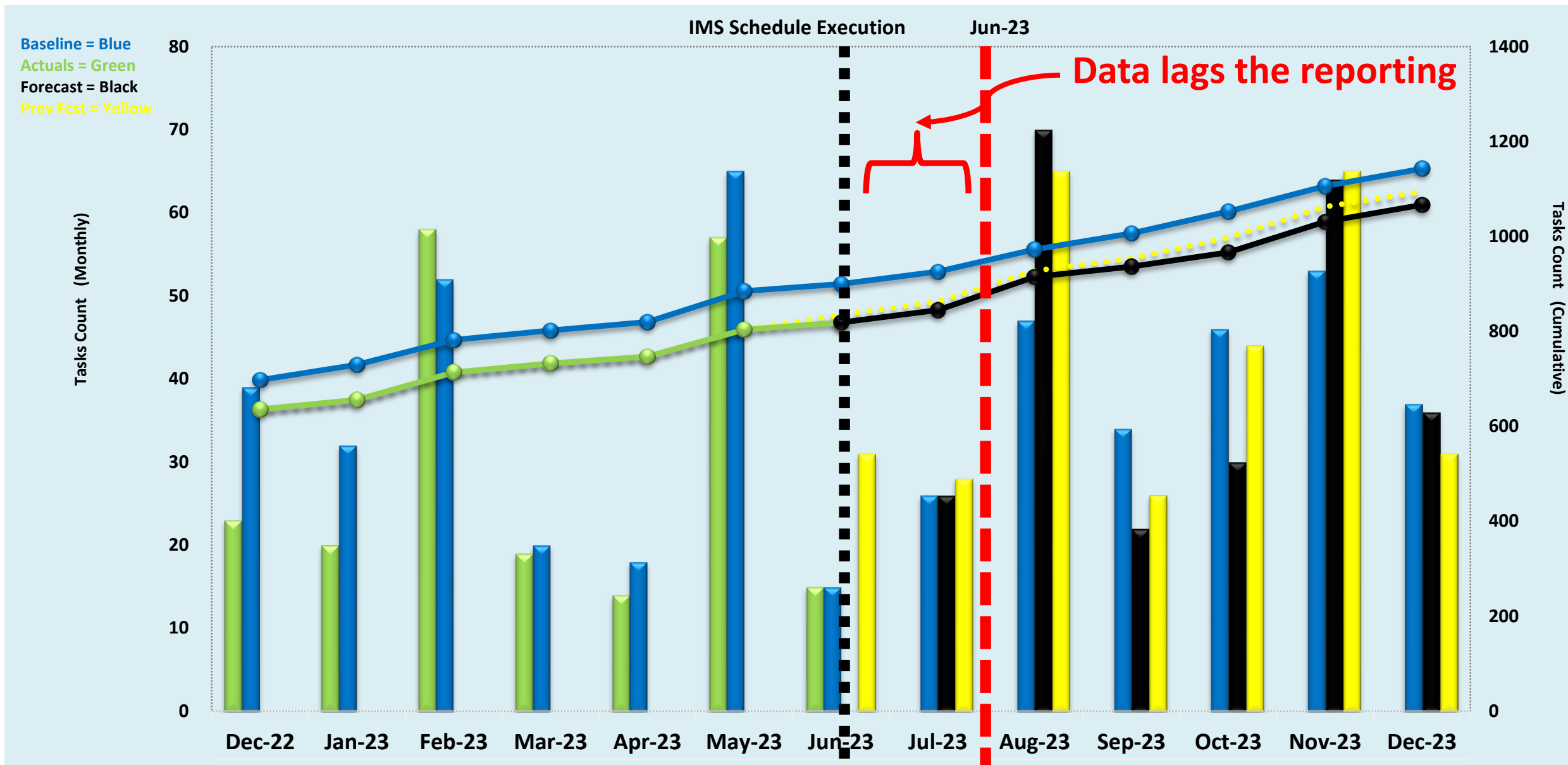
Performance Metrics - Observations

Challenges with Current Practices

Many agile programs – particularly hybrid waterfall/agile programs – rely on traditional performance metrics

1. Earned Value Management (EVM)

- Often lags current performance by several months.
- Reporting periods don't always align with agile increment periods (e.g., if EVM end date and program increment end date don't align, the metrics may show poor performance one month and aggressive performance the next – see next slide).



Challenges with Current Practices (Cont.)

1. Earned Value Management (EVM) - Continued
 - Often, EVM and other traditional approaches don't communicate changing priorities or opportunities that result from taking advantage of dynamic conditions (e.g., having expert manpower available so a future feature is moved to the "left" – higher in priority in the agile project backlog).
 - IMS is always catching up to the actual state of the project
 - Can lag by months depending on the program and how often the project backlog changes

Challenges with Current Practices (Cont.)

2. Existing performance tracking tools

- Due to foreign ownership, some tools are not available to the government or the development contractor
- Licensing challenges – including expenses
- Incompatibility of tools and the system development pipeline

Challenges with Current Practices (Cont.)

3. DoD needs to track complex compliance items in addition to common agile metrics
 - Example: Atlassian Marketplace has many tools to show common agile metrics (feature-to-story relationship roadmaps, sprint velocity, and completion forecasts, etc.)
 - DoD needs: DoD projects also need to track items such as requirements linked in many-to-many relationships with the completion of requirements.

Challenges with Current Practices (Cont.)

4. Access challenges

- Cleared defense contractors may need to host the feature tracking applications (e.g., Atlassian's Jira) on their internal network to integrate with other development systems.
- The contractor's hosting of agile program tracking applications (e.g., Jira) may limit access by customers.
- The challenge: Additional approvals and steps may be required to allow external user (i.e., the government) access and ensure cybersecurity which may limit user access or impact user experience.

Performance Metrics - Solutions

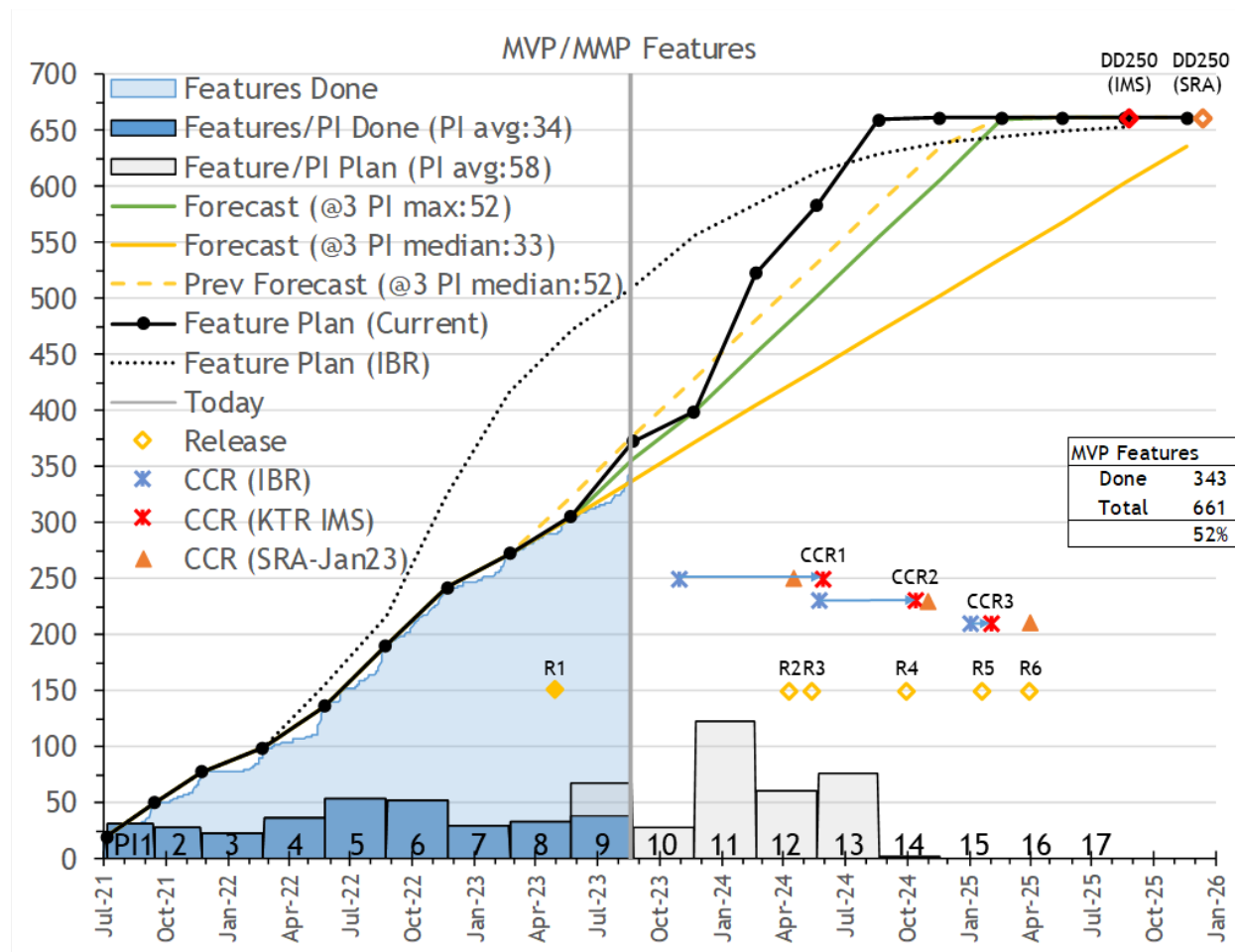
Solutions – General Observations

1. Develop solutions which combine new tools with existing solutions
 - Example: Combine Jira, DNG (Doors Next Generation), MS Excel with project-developed applications using Visual Basic, Python, etc.
2. Collect and store metrics on a daily basis (store in a database).
 - Many tools don't have built-in histories or, if available, are not easily exposable to the common user.
 - For example, Jira provides a snapshot of current project status, but historical data is not available for reviewing past performance.

Solutions – General Observations (Cont.)

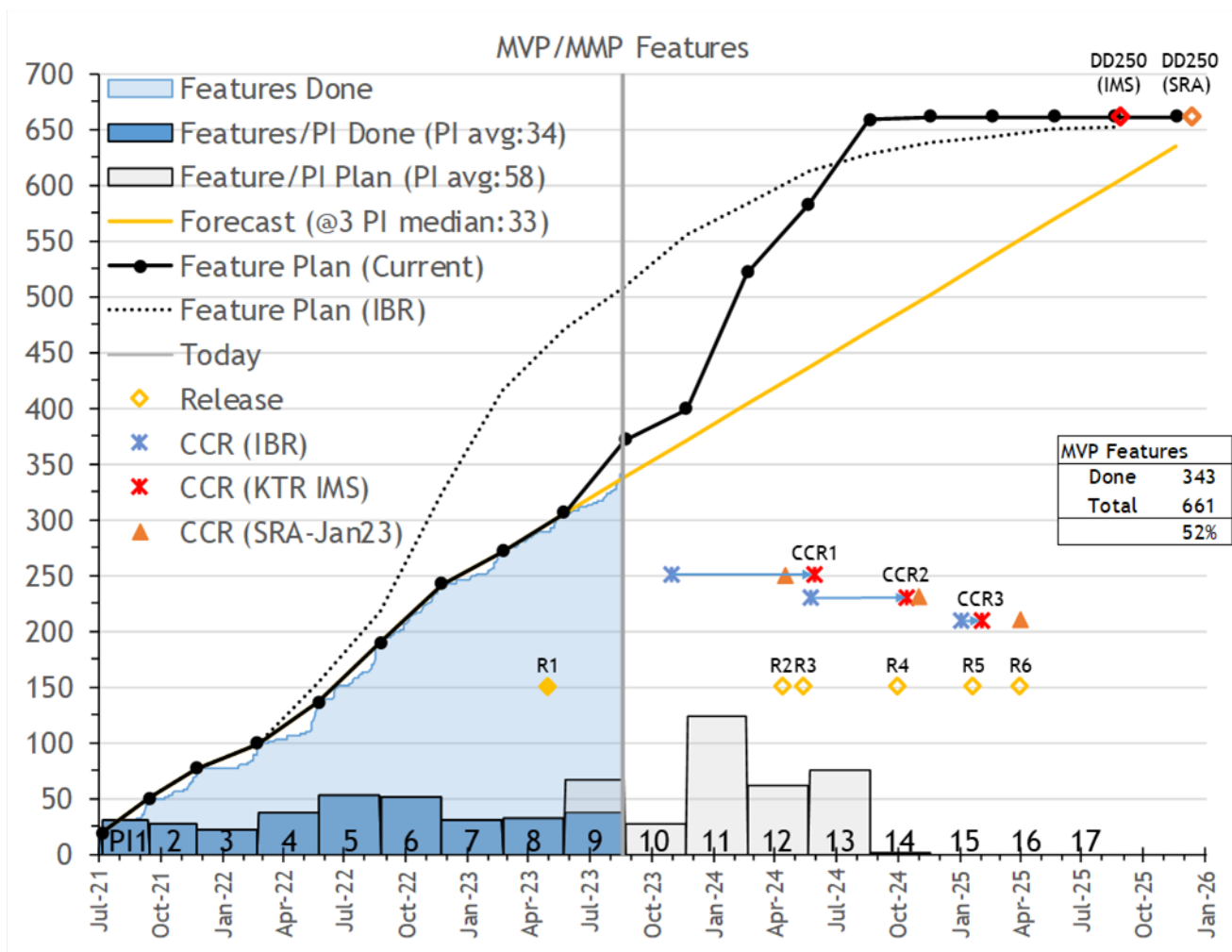
3. MS Excel and Visual Basic are invaluable for capturing program performance.
 - Often used for analyzing current and (collected) past performance data.
 - Important to have team members who are familiar with MS Excel and Visual Basic.
4. Python and a lightweight relational database are useful for exporting performance data from an application such as Jira

Metrics – MVP/MMP Plan and Progress



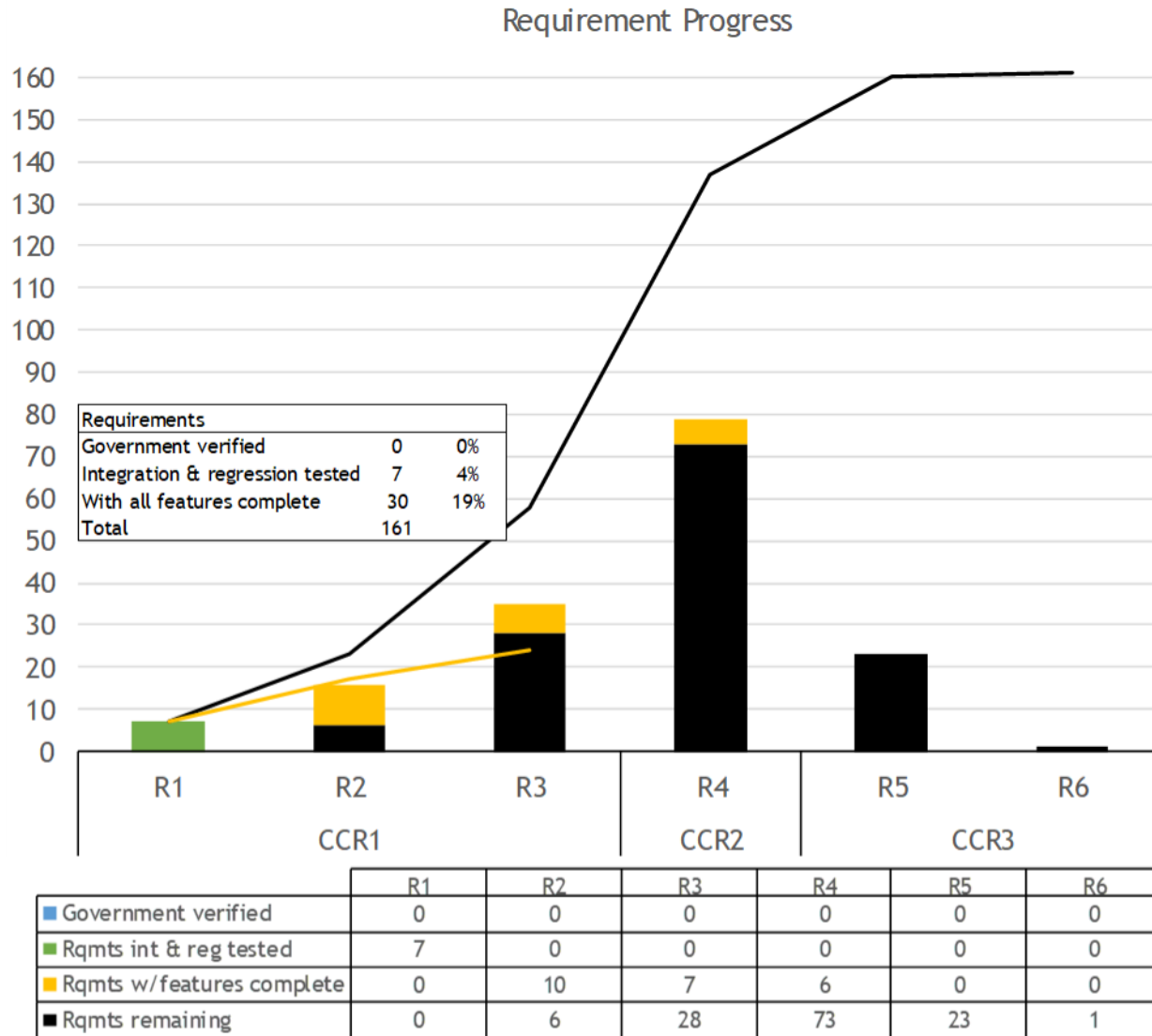
- **Dotted black line:** original feature to Program Increment (PI) allocation plan set at program’s IBR.
- **Solid black line:** current cumulative plan for features assigned to PIs and becomes the “as run” when PIs complete.
- **Light blue filled area:** shows the cumulative completion of features.
- **Green and yellow lines:** show forecast completions at maximum and median rates.
- **Not shown:** Current & planned FTE.

Metrics that worked – MVP/MMP Plan (De-cluttered)



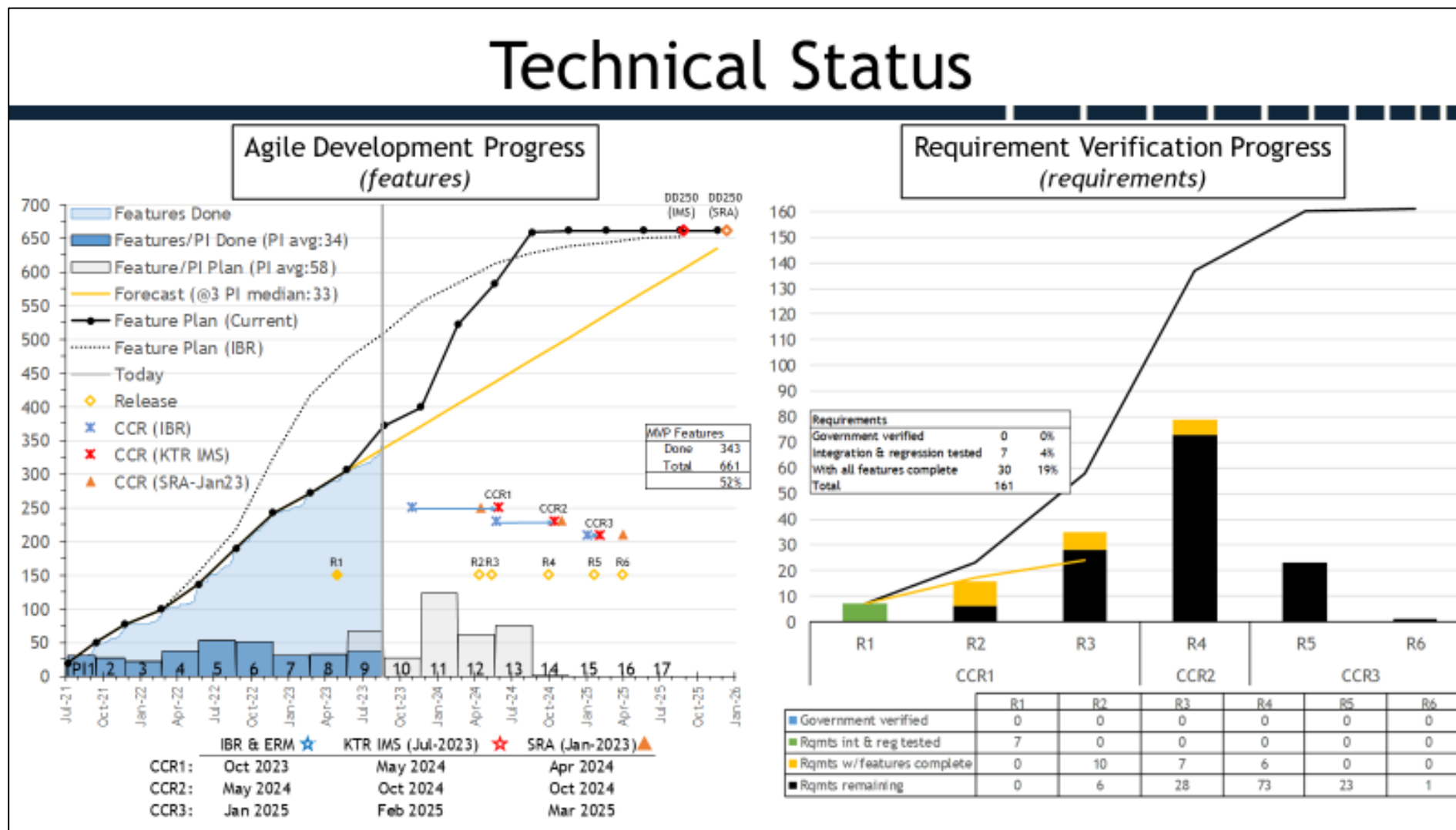
- Burn up of features with a direct association to MVPs and MMPs.
- Forecast lines are limited to only the current median rate (yellow line) as the most-likely case.

Metrics that Worked – Requirements Chart



- 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

Most Useful – Combine both metrics (presented to SAE)



Overall Recommendations

1. There are advantages in capturing software development data for local or off-line analysis.
 - a. Enables the generation of metrics in areas of interest that are not covered by off-the-shelf task management systems.
2. It is important to understand the project's software development configuration management system and methods for extracting development data in a form that is suitable for the generation of metrics.
3. It is important that the customer has independent and direct access to software development data for the purpose of data analysis and reporting.

Programmatic Observations

Programmatic Agile Relationships

1. Agile changes how work cost is calculated when incorporating and costing new efforts to be added to an agile program (e.g., RFC – Request For Change)
 - a. It adds a layer of obscurity to the customer who is trying to evaluate the cost of new work.
 - b. EVM (mentioned previously) lags and struggles in keeping up with agile. In addition, government costing methods also struggle.
 - c. Costing methods that rely on t-shirt sizes (for example) make it very difficult to deduce scope of work and to fully understand increases in cost due to external events such as inflation.

Programmatic Agile Relationships (cont.)

2. Agile can be flexible via certain program variables being fixed or stable, however, when instability occurs , agile flexibility can push into the realm of the complex (for both developer and customer)
 - a. For example, when experienced personnel are unexpectedly pulled away from the project to support another project (for an extended period)
 - b. This typically means junior or new team members are left on the program scrambling to ramp up and “learn the system.” This often leads to “analysis paralysis.”
 - c. Agile needs senior members to provide cadre leadership.

Programmatic Agile Relationships (cont.)

3. If the program has a major technical dependency from another company (peer, not sub) and that dependency becomes out of sync (by years) with the programs cadence, there is only so much agile can do to catch up.
 - a. This usually means that the program will need to be re-baselined which pushes the timeline to the right...and...
 - b. ...triggers costing which runs into the aforementioned cost issue.

Next Steps

- Continue exploring improved performance metrics, particularly on what is needed at the SAE level.
- Continue collecting lessons learned and development recommendations on how to 1) transition into an agile program and 2) how to manage an agile program in a DoD enterprise environment.
 - For example, can we find solutions to the three observations just discussed?

Questions?

Contact: Mike Orosz

mdorosz@isi.edu

michael.orosz.ctr@spaceforce.mil