Get Actionable Program Management Data from the DevSecOps Pipelines

October 2023

Bill Nichols
Julie Cohen

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Make Program Decisions!

Your SOCOM commander has learned that an adversary has unexpected capabilities.

You must **reprioritize** your **capabilities**.

Whiz-Bang Software for Sensor Fusion, which was added to the roadmap 18 months ago, is now **top** priority with a need of within the next 6 months.

Two capabilities that were to be the next priority, due in 3 and 9 months, are critical and must be completed as close to the original due dates as feasible.

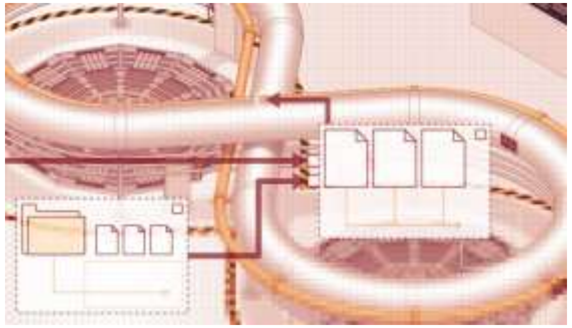
You need to know the following:

- How can the two next-priority capabilities and the new, highest priority capability be delivered without affecting the staffing?
- To deliver the Whiz-Bang capability within six months while maintaining the other two capabilities' schedules, **will more teams be required**? **How many**? **At what cost**?

What Do You Do?

How do you make informed, defensible decisions?

Agenda



Defensible decisions by automating data collection and analysis

Description of ACE/PoPs

Indicators

Observations and Lessons

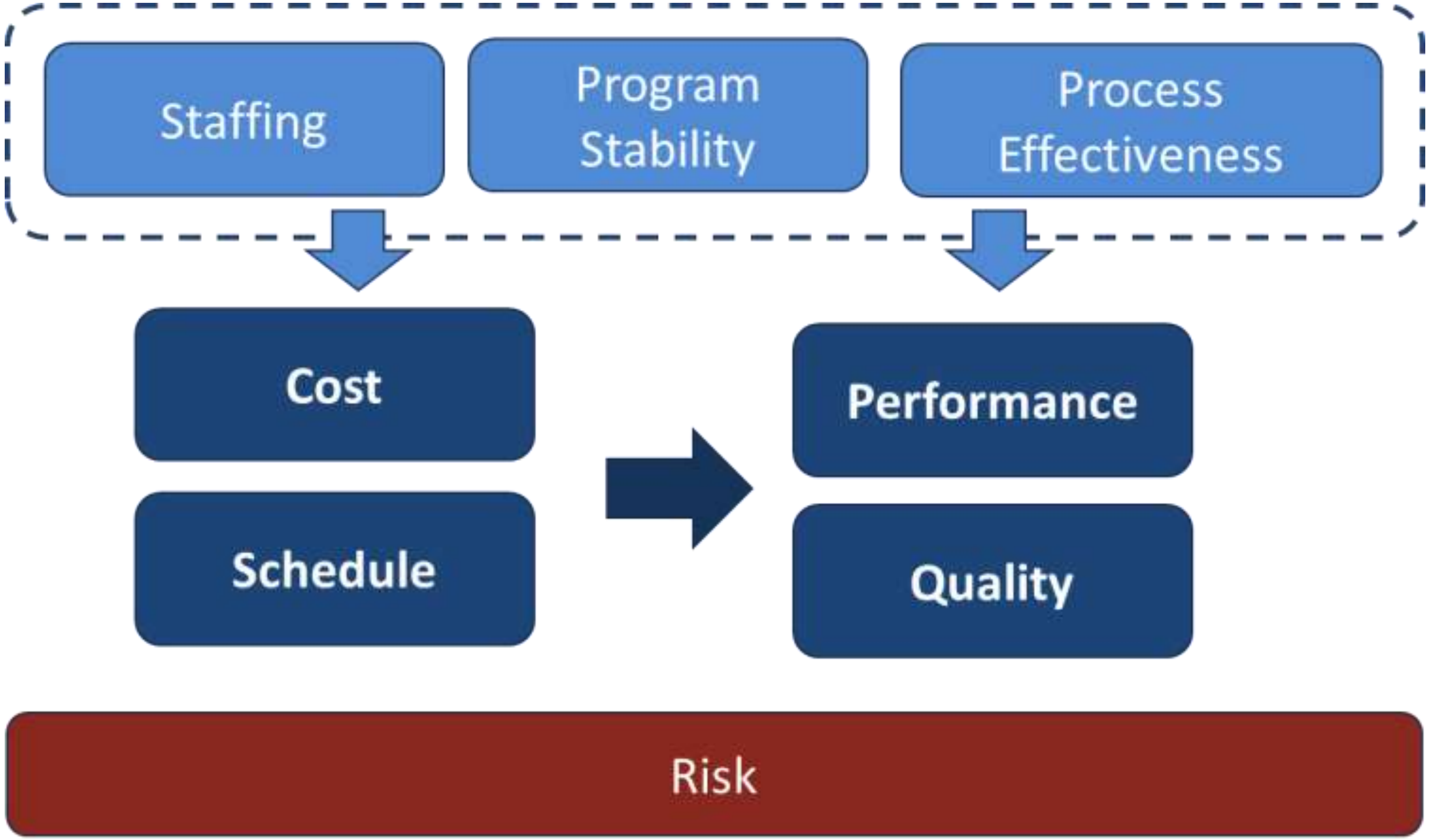
Call to Action

ACE/PoPs Overview

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Get the information you need

Information needs: What are your Targets? What can you manage?



Automated Continuous Estimation for **Pipelines of Pipelines**

Automation drives continuous integration and delivery of software, but outpaces program control

To solve this problem: Automate data collection Model DSO systems with **Monte Carlo**, and provide continuous reporting.

- Determine status
- Project future events
- Provide evidence for corrective actions

Goal: Programs using DSO(DevSecOps) have constant access to information needed to monitor and control schedule and cost commitments.

Status and projection models should be available in real time.

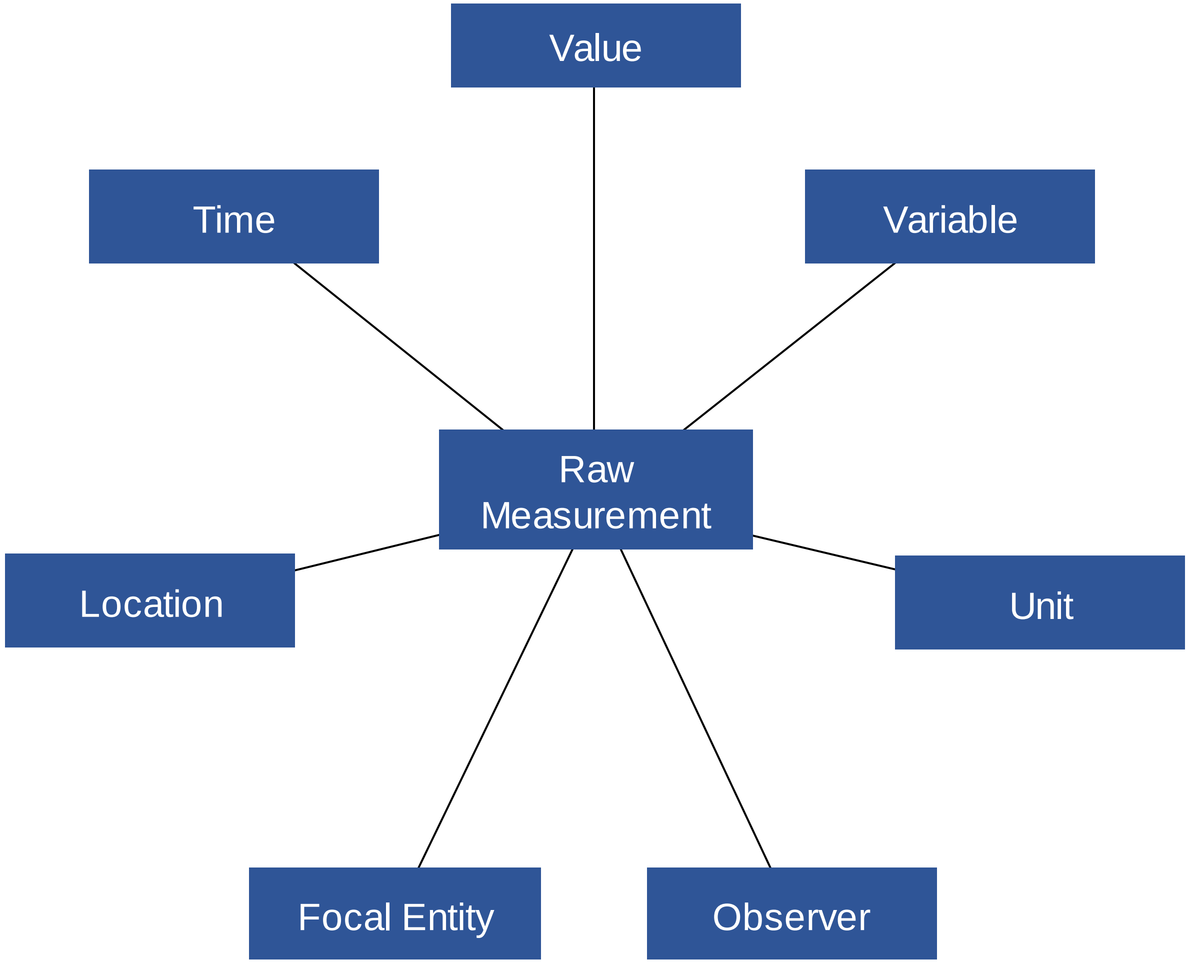
Model pipeline and pipeline-of-pipeline systems.

Automate data collection and Program Management Status Reporting for DevSecOps pipelines.

Directly collect data from DevSecOps pipeline tools

- Automate data collection, storage, and reporting
- Correlate data to project outcomes
- Present completion to-date and milestone predictions to Program Management in smart dashboards

Prototype DSO Measurement



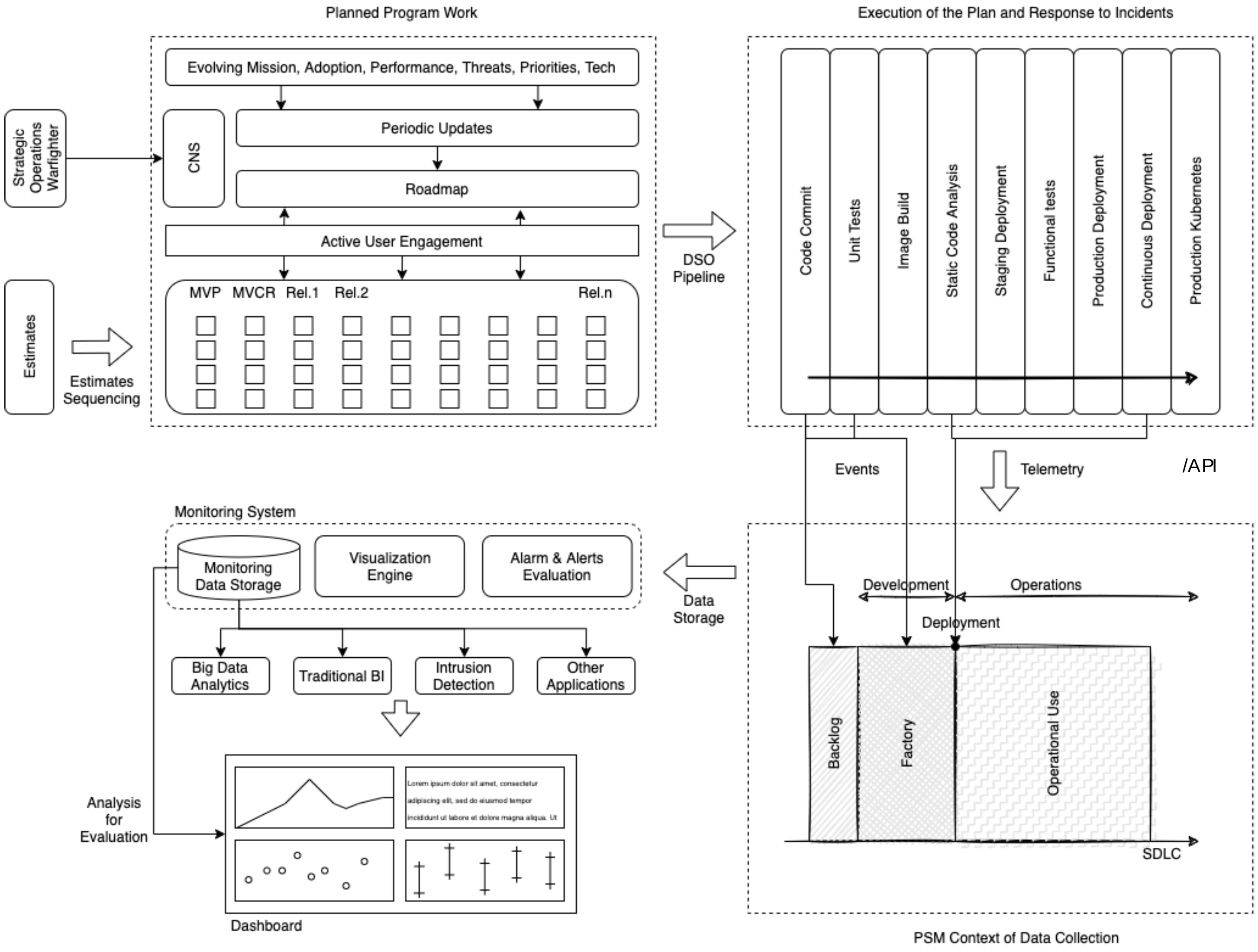
- Measurement: *"A set of observations that reduce uncertainty where the result is expressed as a quantity." - Douglas Hubbard*

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

ACE/PoPs Model: Data Collection Context

Managed with Jira, Gitlab, Rally

Factory Pipelines



Planned work includes the WBS, work packages, work sequencing, and estimates.

Work packages **execute** plan development stages. Tools trigger events (time stamps, package labels).

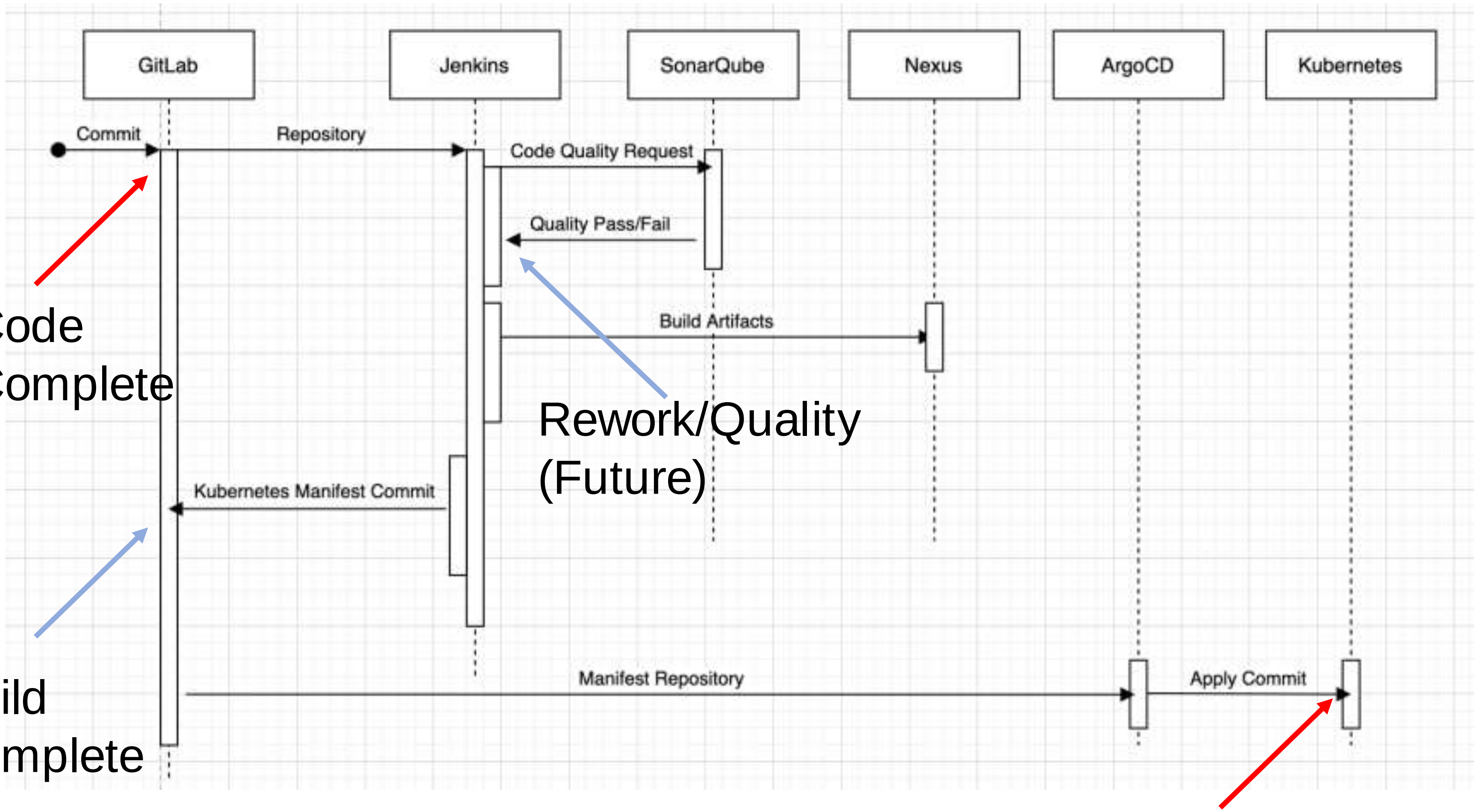
Data is collected and **transformed** for storage.

The **warehouse** loads the data and provides the interface for analysis and dashboards.

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Track events through the Pipelines

Extracting metrics https://youtu.be/u96OFTXgr0g



Code Complete

Rework/Quality (Future)

Build Complete

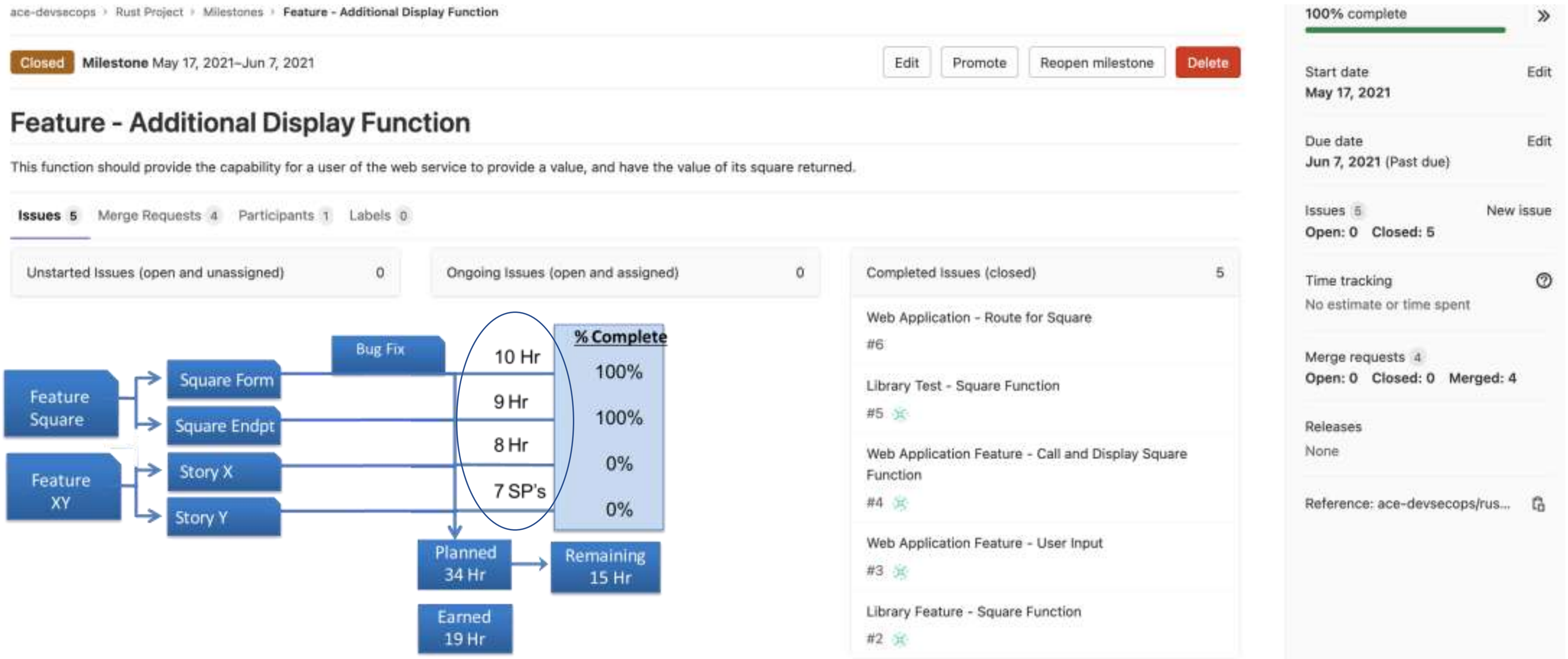
Deployed = Done

- Date is collected from key events.
- The data specification is on the following slide.
- Use Labels to connect WBS, RoadMap, and Backlog to work packages.
- Lead Times
- Estimated Dates
- Actual Times

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Track From the Roadmap Through the Pipeline

Work Completion https://youtu.be/X-R1mIZ3sPk



ace-devsecops › Rust Project › Milestones › **Feature - Additional Display Function**

Closed **Milestone** May 17, 2021–Jun 7, 2021

Edit | Promote | Reopen milestone | Delete

Feature - Additional Display Function

This function should provide the capability for a user of the web service to provide a value, and have the value of its square returned.

Issues 5 | Merge Requests 4 | Participants 1 | Labels 0

Unstarted Issues (open and unassigned)	0

Ongoing Issues (open and assigned)	0

Completed Issues (closed)	5
Web Application - Route for Square #6	
Library Test - Square Function #5	
Web Application Feature - Call and Display Square Function #4	
Web Application Feature - User Input #3	
Library Feature - Square Function #2	

100% complete

Start date — Edit
May 17, 2021

Due date — Edit
Jun 7, 2021 (Past due)

Issues 5 — New issue
Open: 0 **Closed: 5**

Time tracking
No estimate or time spent

Merge requests 4
Open: 0 **Closed: 0** **Merged: 4**

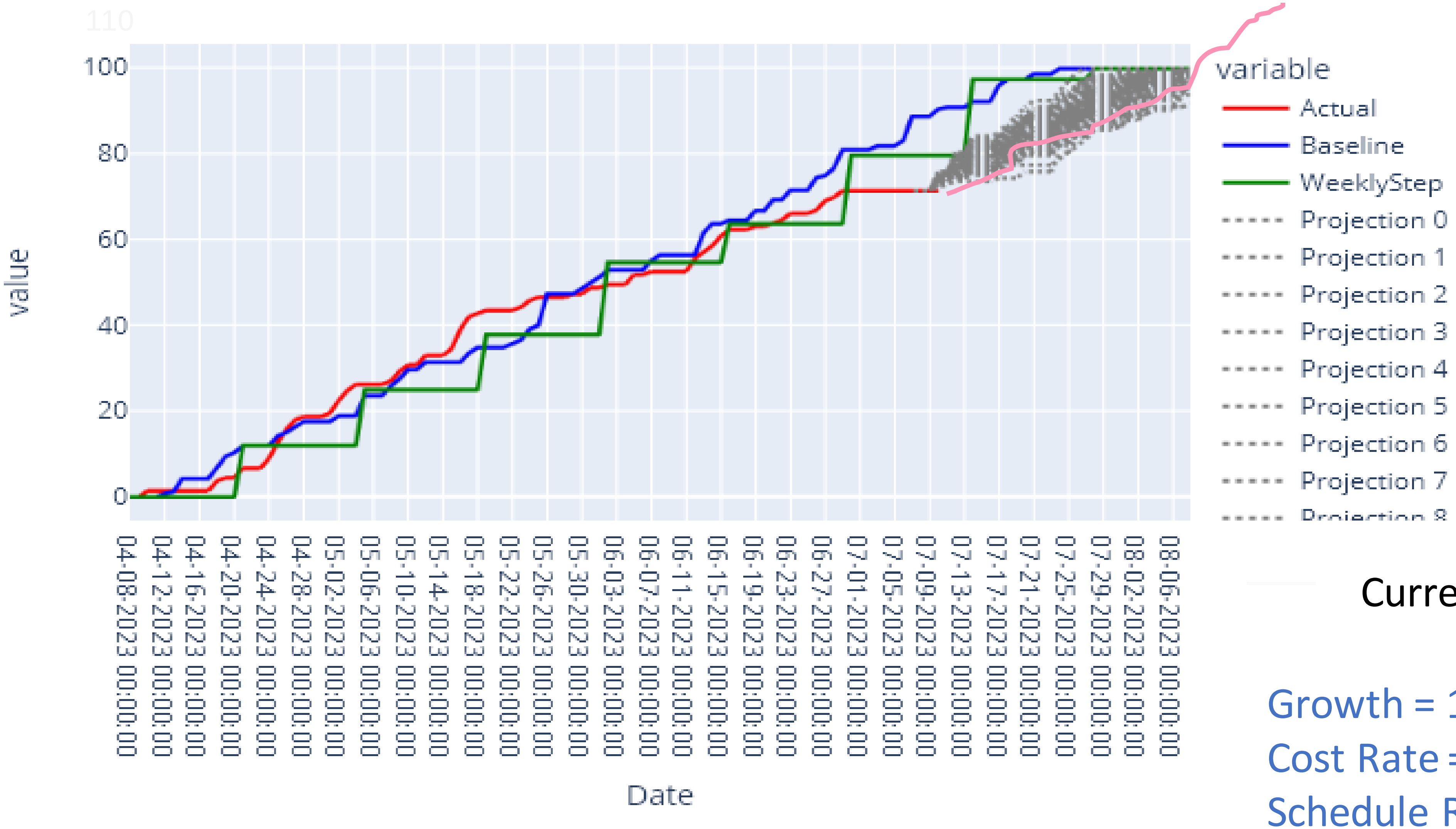
Releases
None

Reference: ace-devsecops/rus...

Indicators

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Track Status and Project outcomes of an Increment



Current Plan

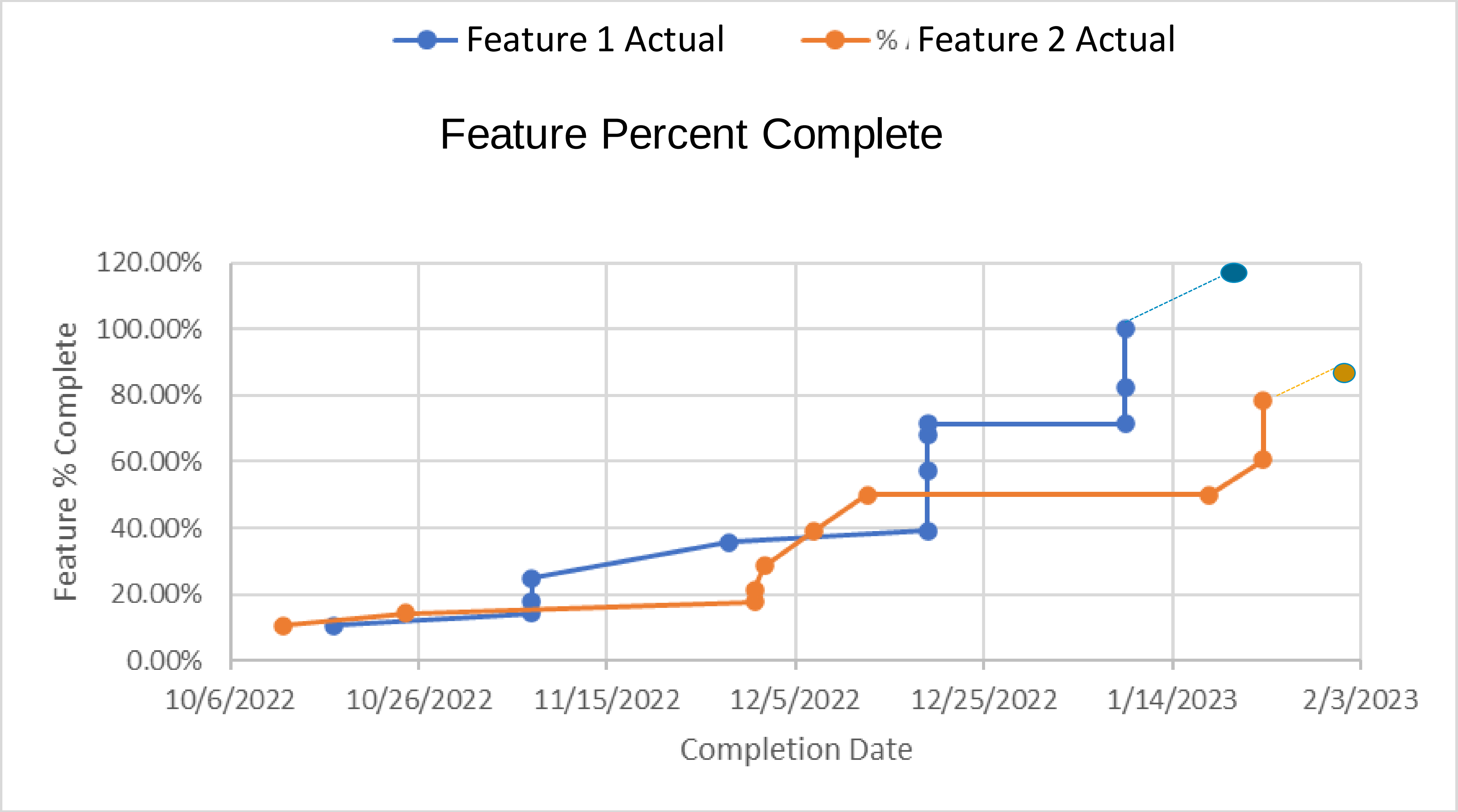
Growth = 1.10

Cost Rate = 1.15

Schedule Rate = .87

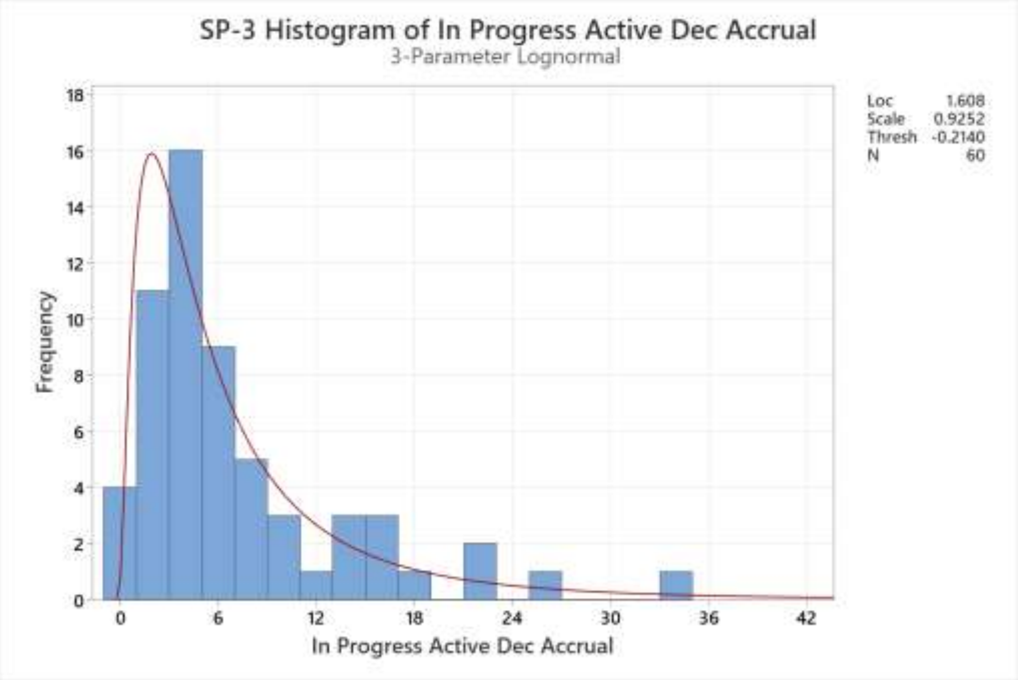
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Extract Feature (related Stories)



Measure Estimation Accuracy and Precision

Calibrate for each pipeline.
Determine that work is predictable **within a range**.
Estimate all work: program planning, road mapping, program increments, sprints.
Measure the work as it passes through the planning, backlog and pipeline.
Compare estimates with actuals.



While the estimation is stable, use the ranges to make future predictions.

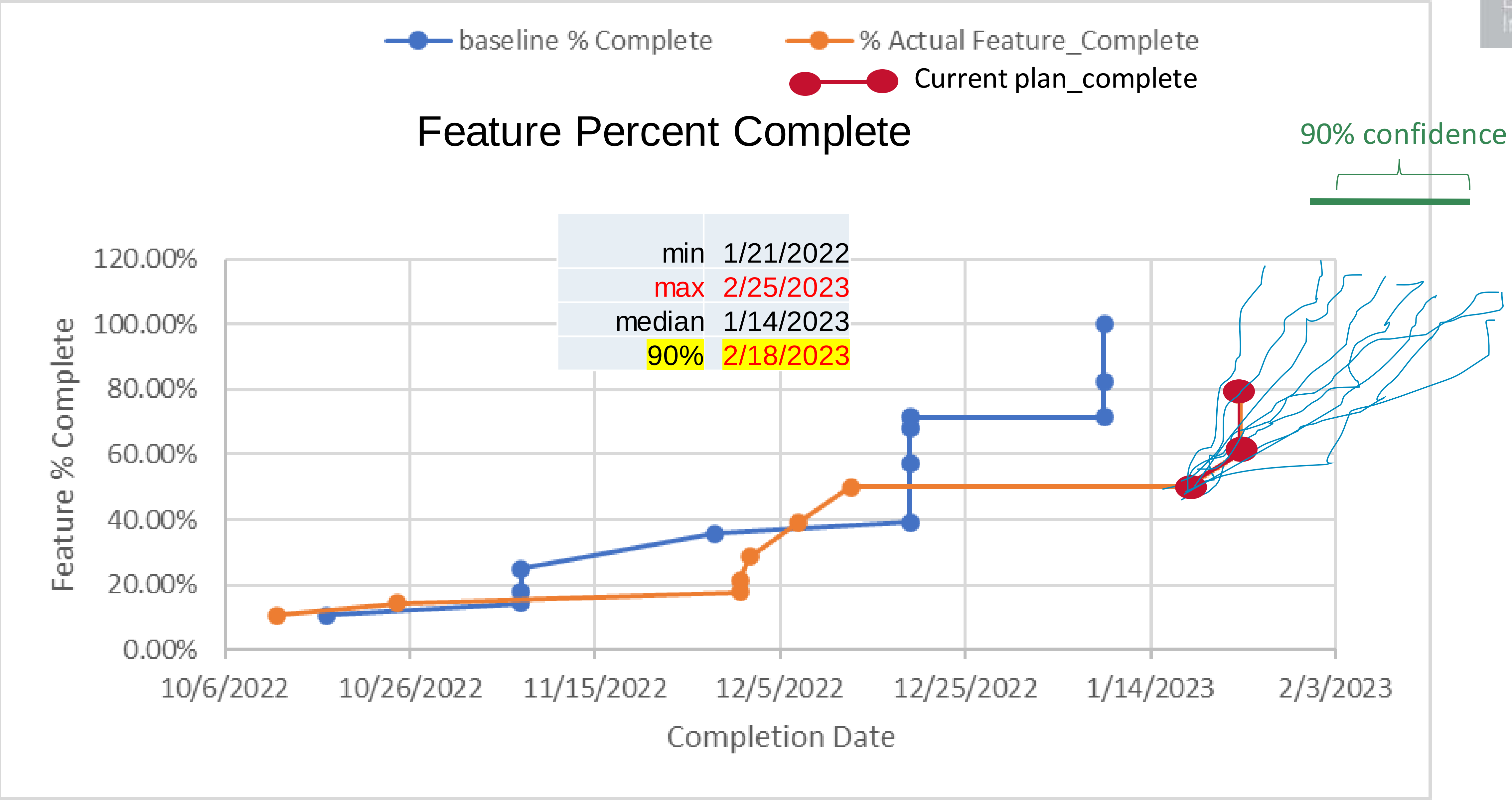
Statistics

Total Count	Mean	SE Mean	StDev
60	7.185	0.883	6.839

Minimum	Q1	Median	Q3	Maximum
0.335	2.583	4.599	8.965	33.44

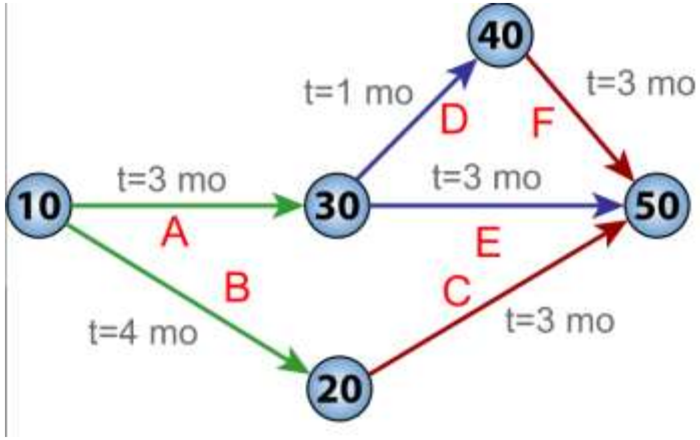
Mode	IQR
8	6.382

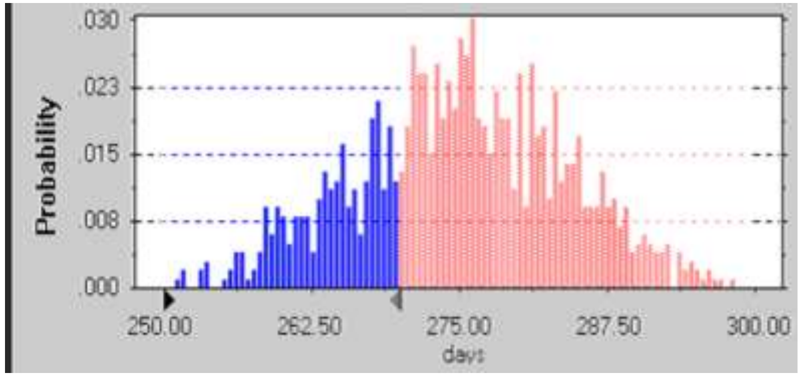
Feature Completion



min	1/21/2022
max	2/25/2023
median	1/14/2023
90%	2/18/2023

Extend Multi-Pipelines with Networks





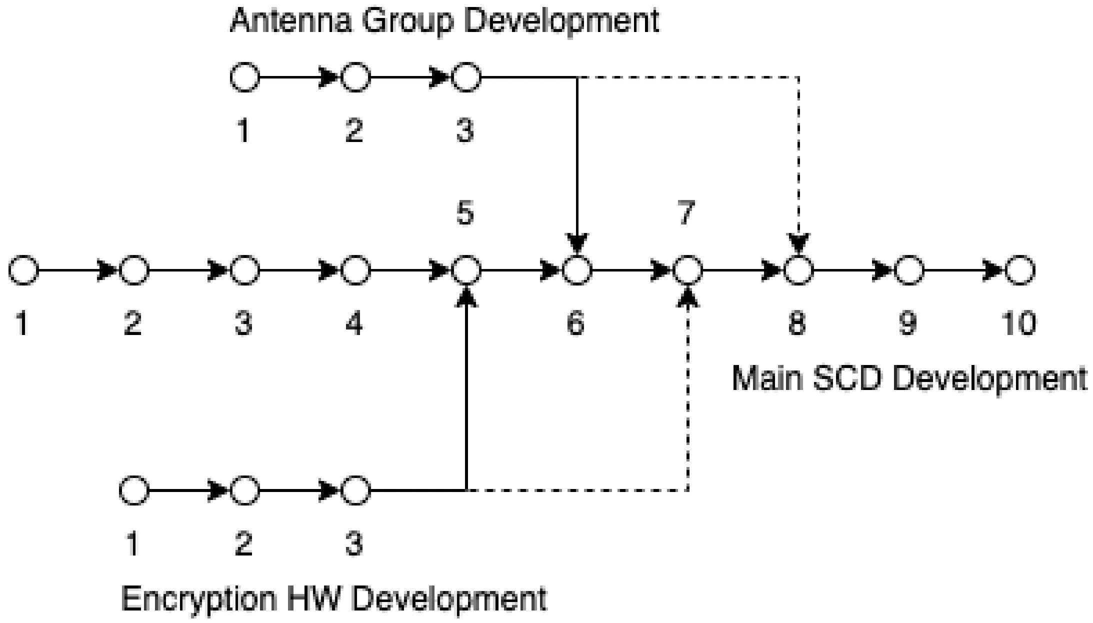
Approach

- Trace work item through development steps
- Identify blockers and integration points
- Probability of completion date

Data from DSO pipeline and other sources

- Product state node structure (capability based WBS, product dependencies, workflow)
- For each Pipeline obtain empirical data for
  - **Effort Rate** and **variation** (by skill?)
  - **Production Rate and variation** by **work type**
  - **Primary work** and **Rework** by activity
  - Defect Rates and fix latencies (build, test)

Pipeline of Pipelines PoPs Workflow Network Example



Model a fictitious device that captures characteristics of a real project dependencies between hardware and software capabilities.

Different pipelines produce dependencies used to model schedule, cost, and technical performance risks resulting from production variation, accumulated variance, and rework.

All nodes are pipeline activities, arrows are lead times. Nodes 5,6,7, and 8 are integration or test points.

Little’s Law assumptions are strongly violated except for **some** linear pipeline segments.

Typical Flow Metrics do not accommodate rework, merges, or multiple entry points

Observations and Lessons

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Lessons Learned

Measurement tools are siloed, making it hard to work together.

Different types of work need to be accounted for (product, bugs, research …). They have different characteristics, estimation accuracy, and outputs.

Averages alone don't support information needs high priority changes or statistical modeling. We need distributions.

We need more specific lead time measures for process steps and baselines for zero rework lead times. (total time until test completes is a candidate quality proxy).

Typical flow metrics don't appear to apply to the pipeline-of-pipelines because of branching and other assumptions violations.

Measuring staff availability remains unresolved.

Stay out of the SWAMP (SoftWare Analysis Metrics Pool)

Automation requires precise definitions.

Every measurement has precise meaning in a known context.

Every metric supports an information need.

Disciplined work decomposition (WBS) connected to product deliverables

Categorize and estimate work items

Automation support for workflow (e.g. Jira/GitLab) and technical implementation (DevSecOps)

Consistent workflow with start and finish

Minimize humans in the loop for data collection

Next Steps

Complete GQIM indicator Templates

Analyze projection accuracy

Complete analysis of gaps identified during Year 1

Experience Package

- Report
- Scenarios
- GQIM Measurement Indicators Examples

Life Cycle Measurement Gap Report

Prepare for a Transition, search for transition partners

Call to Action

Would you benefit from continuous updates to status and projections?

Are you using DevSecOps tool chains, issue trackers, and workflow management?

Can you share process data and discuss results?

Will you participate in our quarterly research review?

We can help!

- Share out Program Management Measurement White Paper
- Specify information, data, and displays for your program management
- Recommend approaches and tools to get started
- Evaluate your results for effectiveness

Links

Guidance for Program Mangers
Program Managers—The DevSecOps Pipeline Can Provide Actionable Data
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=890538

https://insights.sei.cmu.edu/blog/actionable-data-from-the-devsecops-pipeline/

How to get started
Getting Started with ACE/PoPs
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=890665

Coming
Starter packs

ACE/PoPs Team

William Nichols
Principal Engineer
Telephone: +1 412.268.1727

Brent Clausner
DevOps Engineer

Luiz Antunes
DevOps Engineer

Julie Cohen
Senior Engineer

Email: info@sei.cmu.edu

ACE/PoPs Team

Brigid O’Hearn
Senior Engineer

Chris Miller
Senior Researcher

Anandi Hira
Senior Researcher

Michael Bandor
Senior Engineer