

Automated Continuous Estimation for Pipelines of Pipelines

QuARP FY 2023 #1 Workshop

January 2023

Bill Nichols
Chris Miller
Hasan Yasar
Brent Clausner
Luiz Antunes
David Shepard
Joe Yankel
Julie Cohen

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM23-0016

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 and simulate 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

Agenda

Topic	Time (min)
Introductions	5
Milestones and Accomplishments <ul style="list-style-type: none">• PM White Paper• ACE/PoPs Getting Started Document• Multi-pipeline simulation	30
Summarize Next Steps Industrial Pipeline prototype	10
Wrap-up	10

FY23 Milestones

Quarter	Milestone	Status
FY23 Q1	White Paper M3. Validate PoPs data for multiple scenarios	Complete
FY23 Q2	Specify initial on-Site PoPs Objectives and measurement plan	In Progress
FY23 Q3	Review initial pilot PoPs outputs	
FY23 Q4	Review Experience Report	

Accomplishments

- Presented at Boehm CCSE International forum on COCOMO
- Published White Paper on Program Management using DevSecOps Data
- Completed Getting Started description handout
- Begin initial industrial Pilot

Program Management White Paper

Program Managers—The DevSecOps Pipeline Can Provide Actionable Data

JANUARY 2023 • WHITE PAPER <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=890538>

[Julie B. Cohen](#), [William Richard Nichols](#)

paper describes the Automated Continuous Estimated for a Pipeline of Pipelines research project, which automates data collection to track program progress.

PUBLISHER:

Software Engineering Institute

SUBJECTS agile, devsecops, acquisition, measurement and analysis

Abstract

This paper describes how the Software Engineering Institute's (SEI's) Automated Continuous Estimation for a Pipeline of Pipelines (ACE/PoPs) research project can help program managers (PMs) leverage existing DevSecOps software development environments to automate data collection and integrate cost, schedule, and engineering performance. Using this information, PMs can track, forecast, and display program progress.

Getting Started with ACE/PoPs

Are you ready for the ACE/PoPs pilot program?

SINCE YOU EXPRESSED INTEREST IN PILOTING THE AUTOMATED COST ESTIMATION IN A PIPELINE OF PIPELINES (ACE/PoPs), we want to help you get started. This booklet helps your organization prepare to participate in the ACE/PoPs pilot program and set your expectations.

Prerequisites

To participate in the ACE/PoPs pilot program, your organization should have more than one DevSecOps pipeline applied to a common product. For the pilot, we also need

- a tool baseline that identifies each unique software development pipeline
- an inventory of tools used in the pipeline
- business rules that define how each tool performs specific software development functions
- explicit hand-offs and/or data transfers (i.e., interfaces) between the individual tools in each pipeline
- when available, tool logs and telemetry application programming interfaces (APIs)

Traceability from Capability to Software Acceptance

To measure if the program is on track or forecast when a capability will be deployed requires that your organization has well-defined product decomposition so that progress can be traced through product increments. For your pilot to be successful, you must include an initial set of need statements or requirements, use a work breakdown structure (or similar rubric) to provide traceability to a delivered capability, and follow a defined release strategy. The following figure from the Software Acquisition Pathway shows the decomposition of capabilities through the Roadmap and Backlogs phases.

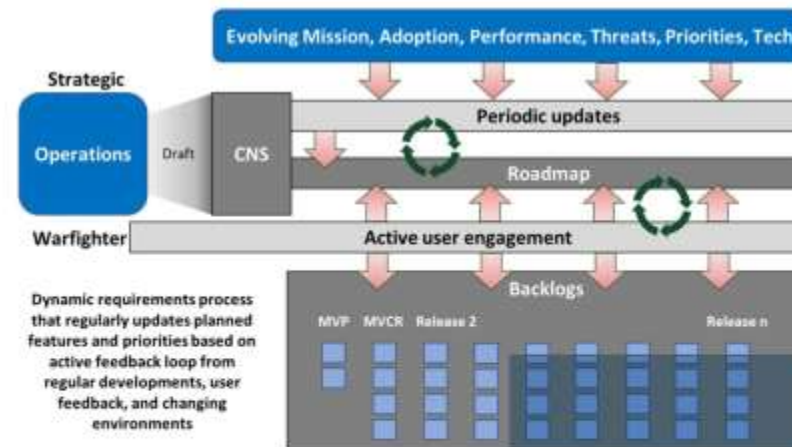


Figure 1: Software Acquisition Pathway—Defining Needs and Engaging Users

Nominal Plan and Schedule

You must establish a plan and schedule prior to beginning the pilot.

2 page description

- What to expect
- Prerequisites

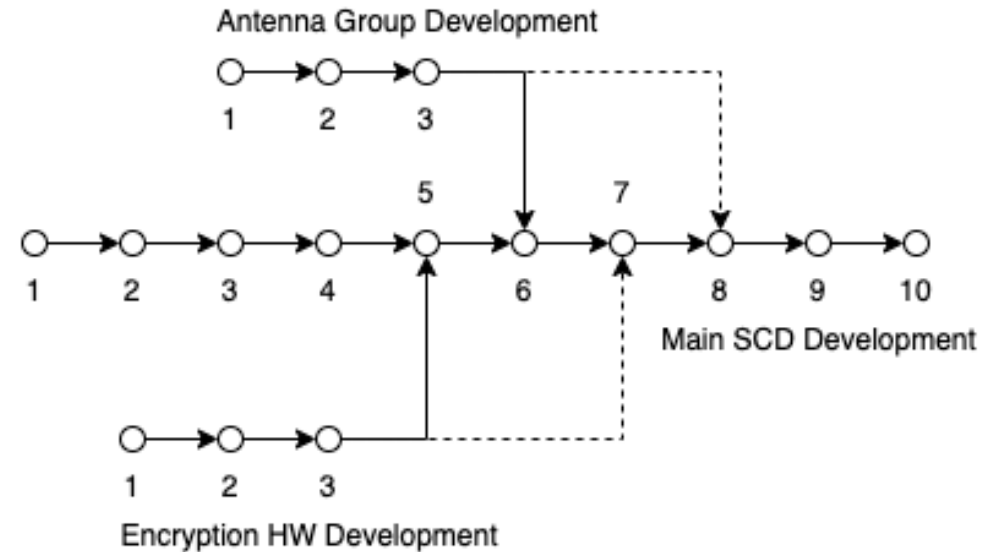
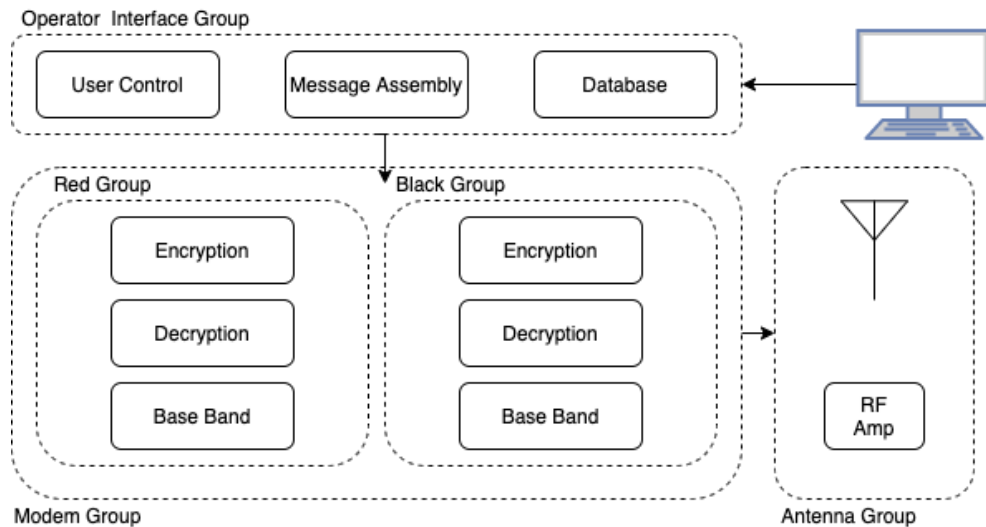
Multi-Pipeline Simulation Projections: Example

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.

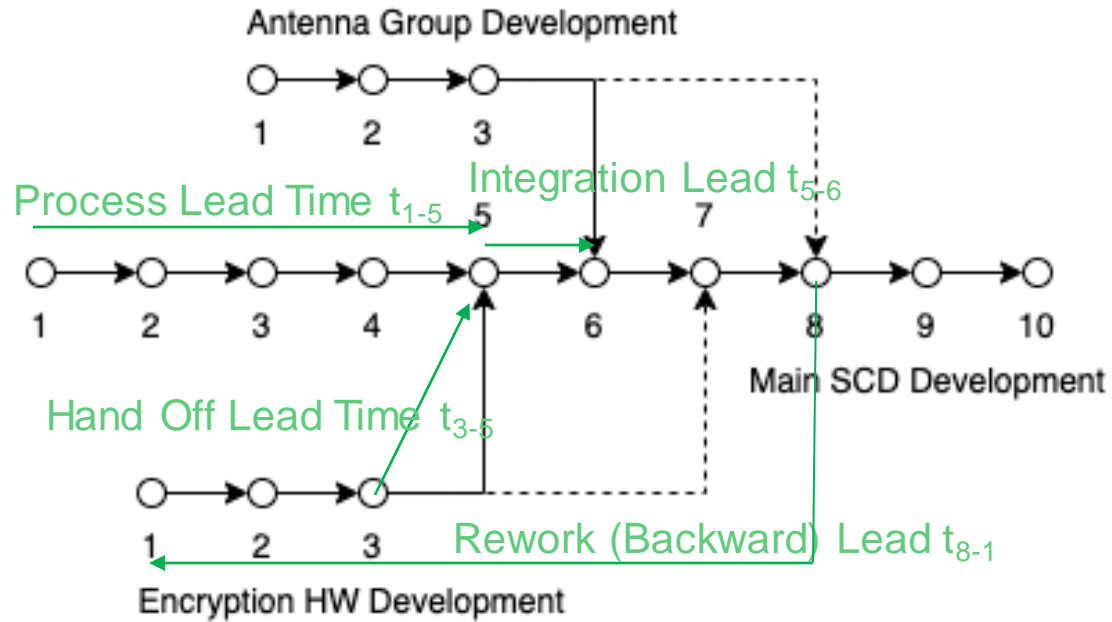
Satellite Communication Device



Multi-Pipeline Projections: Scenarios

1. **Baseline:** Ideal Scenario in which all dependencies are produced on time and with no issue propagation downstream. Main pipeline also executed on time and with no severe issues causing delays or rework.
2. **Dependency delay:** One dependency delays that affect main pipeline: One of the dependency pipelines has a delay in building its work products, which introduces a ripple into the main pipeline.
3. **Multiple dependency delays:** Both dependencies have issues that cause delays in building their work products, which propagate into the main pipeline.
4. **Integration issues:** Work products are **delivered** on time, but inconsistencies in API contract generates incompatibility, and rework, delaying the main pipeline.
5. **Test Issues:** Dependencies are delivered on time and are well integrated, but latent issues are discovered late in testing and force main pipeline to wait for another dependency release.
6. **Urgent change:** For example, a high severity vulnerability has been discovered, the patch must be made, propagate through the multi-pipeline, and perhaps cause other delays.

Multiple-Pipeline Topology



- t_{1-5} Lead time enter development to integration
- t_{5-6} Lead time for successful integration
- t_{3-5} Cross Pipeline lead time exit 3 to enter 5
- t_{8-1} Issue return lead time, identify in 9 to re-enter 1

Input 1: Planned Effort (three team example)

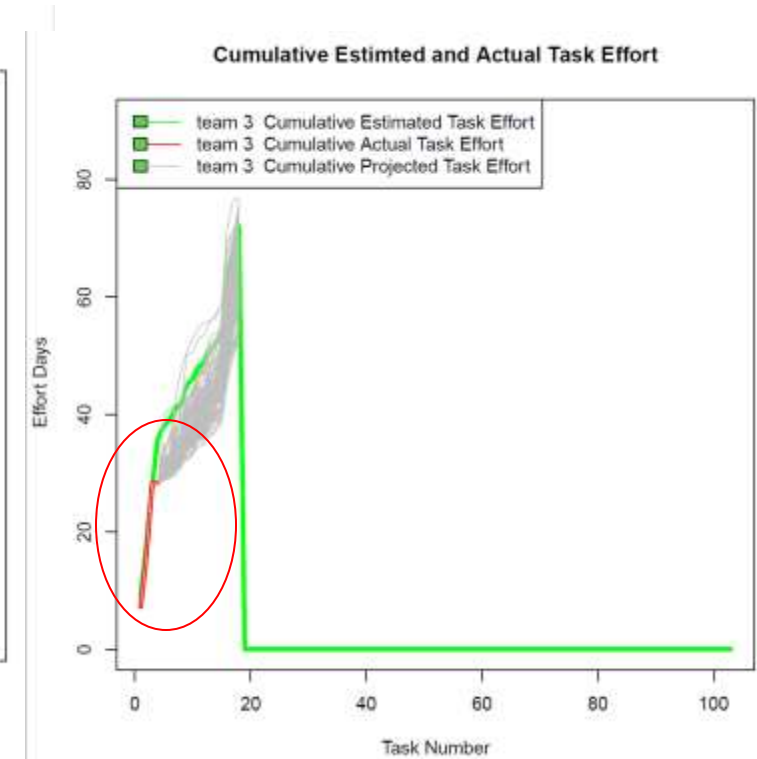
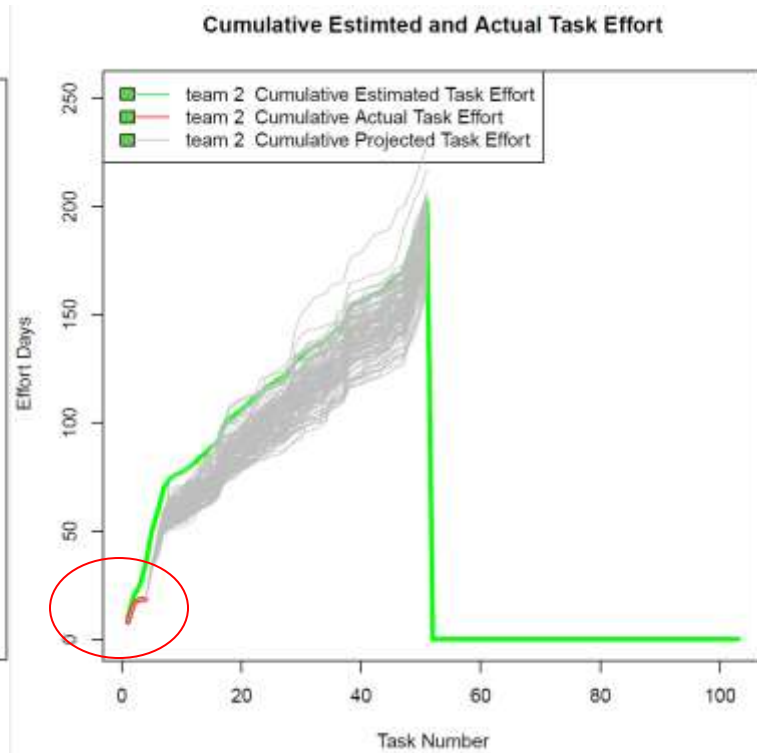
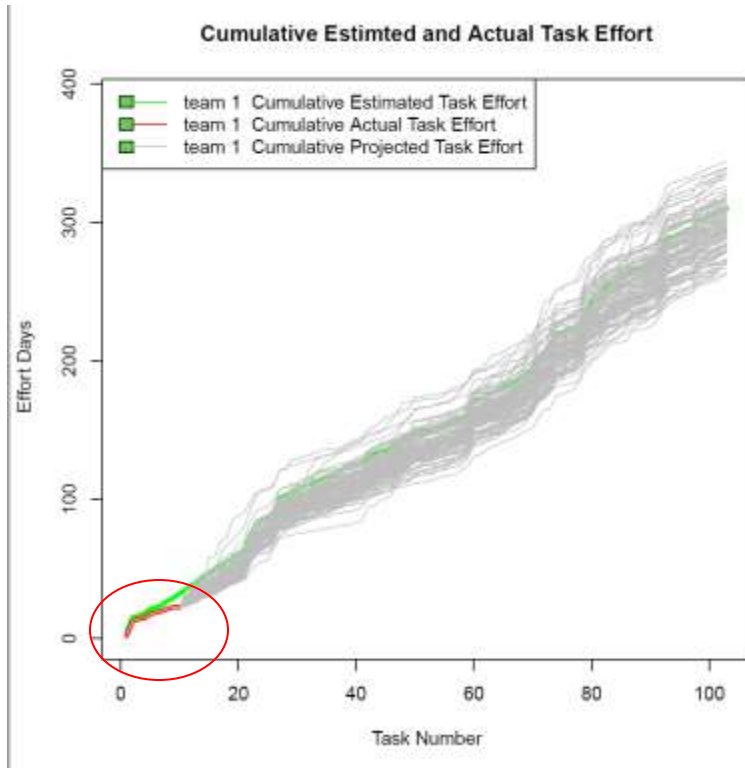
calendar.day	schedule.day	date	day.of.week	Team	team_members	daily.effort	cum.effort hr	Team	daily.effort	cum.effort hr	team_members	Team	daily.effort	cum.effort hr	team_members
22	16	10/3/2022	Monday	Ant	2	0	0	EnHw	3	43	5	Main	3	48	3
23	17	10/4/2022	Tuesday	Ant	2	0	0	EnHw	3	46	5	Main	3	51	3
24	18	10/5/2022	Wednesday	Ant	2	0	0	EnHw	3	49	5	Main	3	54	3
25	18	10/6/2022	Thursday	Ant	2	0	0	EnHw	3	52	5	Main	3	57	3
26	18	10/7/2022	Friday	Ant	2	0	0	EnHw	3	55	5	Main	3	60	3
27	19	10/8/2022	Saturday	Ant	2	0	0	EnHw	0	55	5	Main	0	60	3
28	20	10/9/2022	Sunday	Ant	2	0	0	EnHw	0	55	5	Main	0	60	3
29	21	10/10/2022	Monday	Ant	2	2	2	EnHw	3	58	5	Main	3	63	3
30	22	10/10/2022	Tuesday	Ant	2	2	4	EnHw	3	61	5	Main	3	66	3
31	23	10/10/2022	Wednesday	Ant	2	2	6	EnHw	3	64	5	Main	3	69	3
32	23	10/10/2022	Thursday	Ant	2	0	6	EnHw	3	67	5	Main	3	72	3
33	23	10/10/2022	Friday	Ant	2	0	6	EnHw	3	70	5	Main	3	75	3
34	24	10/10/2022	Saturday	Ant	2	0	6	EnHw	0	70	5	Main	0	75	3
35	25	10/10/2022	Sunday	Ant	2	0	6	EnHw	0	70	5	Main	0	75	3
36	26	10/10/2022	Monday	Ant	2	2	8	EnHw	3	73	5	Main	3	78	3

Input 2: Task List, Estimates, Milestones, Merge (example)

WBS	Activity	Task	Plan_Hours	Plan_Date	Plan_Week	Actual_Hours	Cum_Act_Hours	Actual_Date
MWS_UI	HLD	1.1.1.1.1. Research web frameworks that operate within server hardware	5	9/12/2022	1	1.785891	1.785891	9/12/2022
MWS_UI	HLD	1.1.1.2.1. Design mission data entry user interface	10	9/12/2022	1	10.387276	12.173167	9/16/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 1	1	9/19/2022	2	1.65089	13.824057	9/16/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 2	2	9/19/2022	2	0.33386	14.157917	9/16/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 3	3	9/19/2022	2	3.196781	17.354698	9/19/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 4	1	9/19/2022	2	1.12179	18.476488	9/20/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 5	2	9/19/2022	2	1.089861	19.566349	9/20/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 6	3	9/19/2022	2	1.665779	21.232128	9/21/2022
MWS_UI	CODE	1.1.1.2.2. Write UI templates for web application 6	2	9/19/2022	2	1.097377	22.329505	9/21/2022

This version marks merge points as tasks repeated across teams

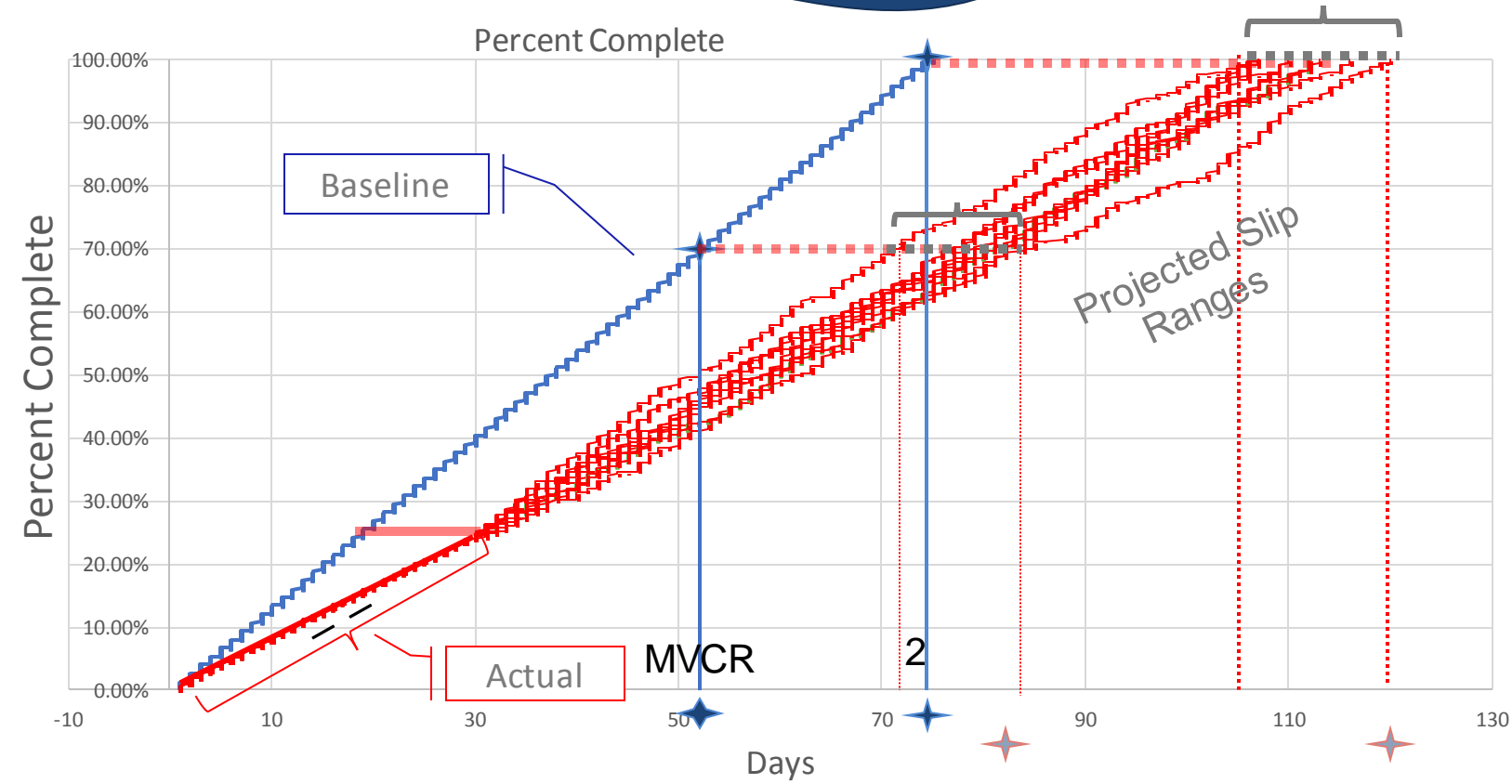
Input 3: Task Actuals



Single Team Range Confidence

Probabilistic Projections

Necessary Data and Information

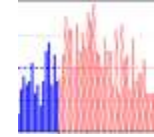


Distributions for
 $t_{lead}^i, t_{factory}^i, t_{deliver}^i$

- Staff Hours(t)
- Roadmap
- WBS
- IMS

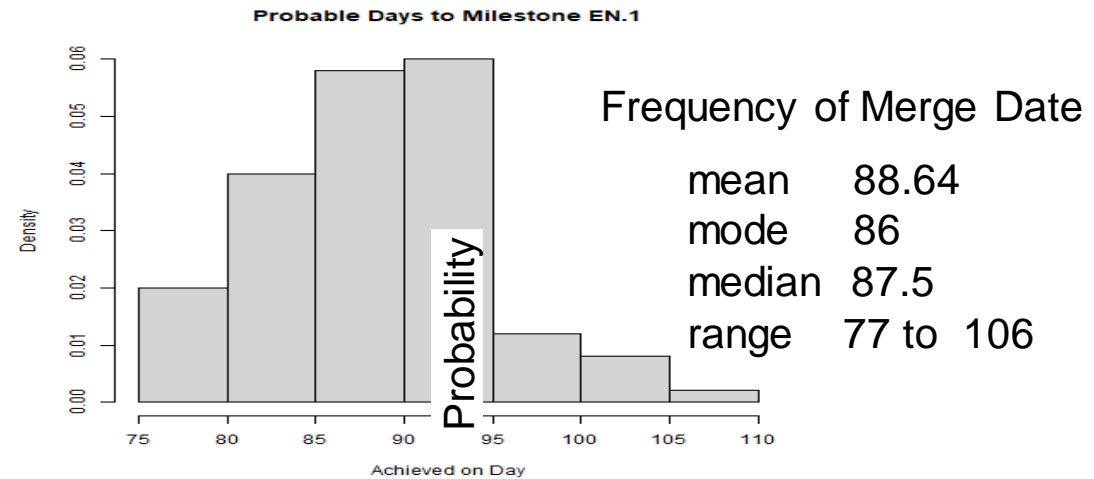
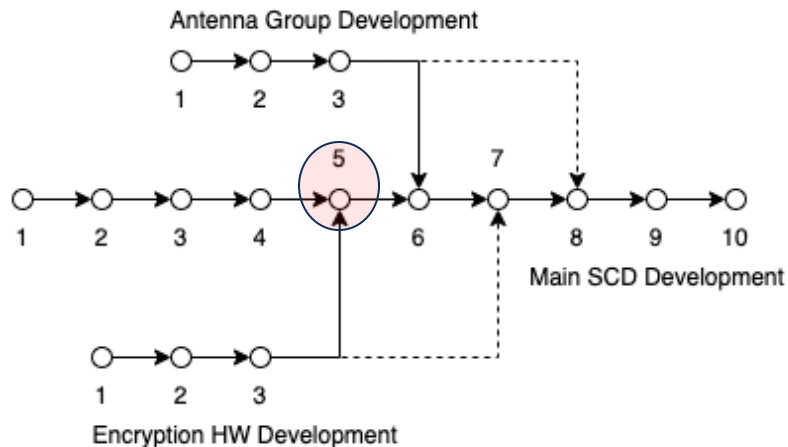
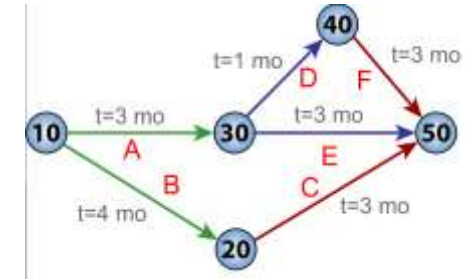
Using distributions to capture and analyze variation instead of using averages requires a not only maintaining a lot more data but also a lot more work!

Multipipeline, Merge Date Range



Approach:

- Assemble dependency network of work steps.
- Parameterize task durations.
- Use Monte Carlo to generate many simulated histories



Calendar Day Achieved

Initial Prototype Objectives

Defense Industrial Base Modernization Project

Identify problems and risks

- Track progress
- Predict date ranges for completion and merges
- Aggregate data from multiple distinct pipelines (convert to common units)
- Identify problem work items early

Improve data accuracy, reduce collection cost (automation)

- Automated collection of work state change timestamps
- Integrate Jira with pipeline tools

Stretch Goals

- Include quality defects and other quality indicators from Build and Test
- Localize issues by correlating discoveries with repository changes

Prototype Learning Objectives

Validate the information we can collect and correlate from the pipeline

Quantify the predictive value of the empirical data

- Parameterized vs non-parametric
- Stability of parameters over time
- Validate data relationships

Additional Prototypes

Working on getting a second prototype

We have time to begin another

Please consider distributing

- One Page Description
- Getting Started with ACE/PoPs
- **Program Managers—The DevSecOps Pipeline Can Provide Actionable Data**

Thank You!

We are grateful for your time and commitment!

For any questions, information, or content please contact

For any questions, information, or content please contact

Bill Nichols wrn@sei.cmu.edu

Hasan Yasar hyasar@sei.cmu.edu

Brent Clausner beclusner@sei.cmu.edu

Luiz Antunes lantunes@sei.cmu.edu

Chris Miller cmiller@sei.cmu.edu

Julie Cohen jcohen@sei.cmu.edu

David Shepard djshepard@sei.cmu.edu

Or

Akia Williams awilliams@sei.cmu.edu