

# Automated Continuous Estimation for Pipelines of Pipelines

## QuARP #2 Workshop

February 2022  
Bill Nichols  
Chris Miller  
Hasan Yasar  
Rob McCarthy  
Luiz Antunes  
David Shepard

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

# Document Markings

Copyright 2022 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](mailto:permission@sei.cmu.edu).

DM22-0485

# Quarterly Advisory Review Panel

ACE/CD completed in November 2021

We are beginning a 2-year program extending Automated Continuous Estimation to the multiple interacting pipelines common to DoD.

These meetings are to share progress and solicit feedback from Subject Matter Experts.

QuARP 1: Virtual workshop to review the pipeline of pipeline definitions, variants, and problems

QuARP 2: Virtual workshop to review pipeline-of-pipeline construction and simulation

QuARP 3: Closeout workshop, validate models and information dashboard displays

# Agenda

Topic	Time (min)
Introductions	5
Milestones and Accomplishments <ul style="list-style-type: none"><li>• ARS paper</li><li>• Operational Pipeline for Demonstration</li><li>• Integration of Hygeia</li></ul>	10
<b>Current Work</b> <ul style="list-style-type: none"><li>• Multi-Pipeline example</li><li>• Review of Pipeline-of-Pipeline programmatic problems</li><li>• Discussion</li></ul>	
Summarize Next Steps	5

# 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

# ACE/CD Summary

In ACE/CD we identified

Common scenarios including indicators for

- Evaluating Percent Complete, and probabilistic capability schedule estimates
- What if scenarios for change in resources or capabilities needed

Instrumentation points and available metrics

Supporting these needs included

- Data collection from pipeline tools directly,
- Data from the roadmap, and workflow management, wbs,
- Structuring of WBS (capabilities, features) and to support capabilities schedule
- Data storage options including warehousing and data lake alternatives

# FY22 Milestones — Status

Quarter	Milestone	Status
Q1	M1. Validate PoPs definitions and pipeline variations	today
Q2	Present at ARS PoPs in house prototype, integrate with Holocron M2. Complete PoPs simulated data instantiation	In progress
Q3	White Paper M3. Validate PoPs data for multiple scenarios	Planned
Q4	M4. Validate PoPs Dashboard with QuARP SME	Planned

# Accomplishments

Present at ARS Panel

Established an initial operational pipeline using a local project

Integrated Hygeia/Holocron

# Acquisition Research Symposium Paper and Panel

<https://www.nps.edu/web/acqnresearch/symposium>

## ***Automated Data for DevSecOps Programs***

<https://dair.nps.edu/bitstream/123456789/4583/1/SYM-AM-22-070.pdf>

Project teams monitor their work locally, but **what does the program see?**

Program

Insight



Progress



Projects

Insight



Progress



Insight



Progress



Insight



Progress



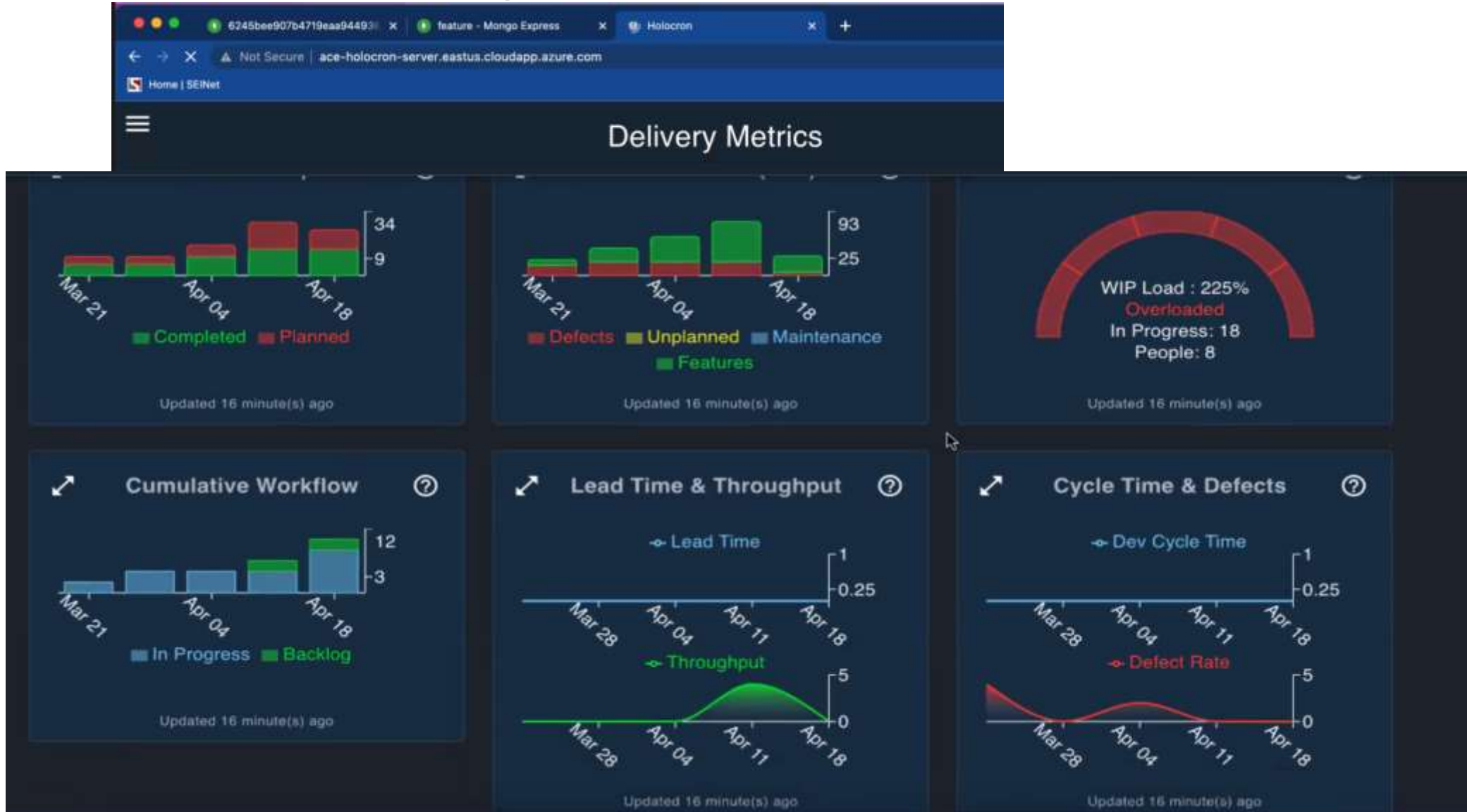
# Prototype Pipeline

Using local development for MDLAR research project

## Prototype Objectives

- Identify gaps in current implementation
- Learn specific features and capabilities of existing tools
- Learn constraints on how project work must be structured
- Specify additional data to be extracted
- Identify non-explicit assumptions
- Demonstrate successful use of WBS structure and/or labels to
  - Measure % Complete
  - Compare planned vs unplanned work

# Prototype Pipeline Hygeia and Holocron, out of the box



# Hygeia Demo

[Holocron \(azure.com\)](https://holocron.azure.com)

# Strengths and Gaps

GitLab supports hierarchy of epics to organize work

GitLab supports linked issues, for example 'blocked' or 'blocked by'.

GitLab has limited support for viewing work breakdown hierarchy, Jira is stronger

Milestones can be composed of issues, but

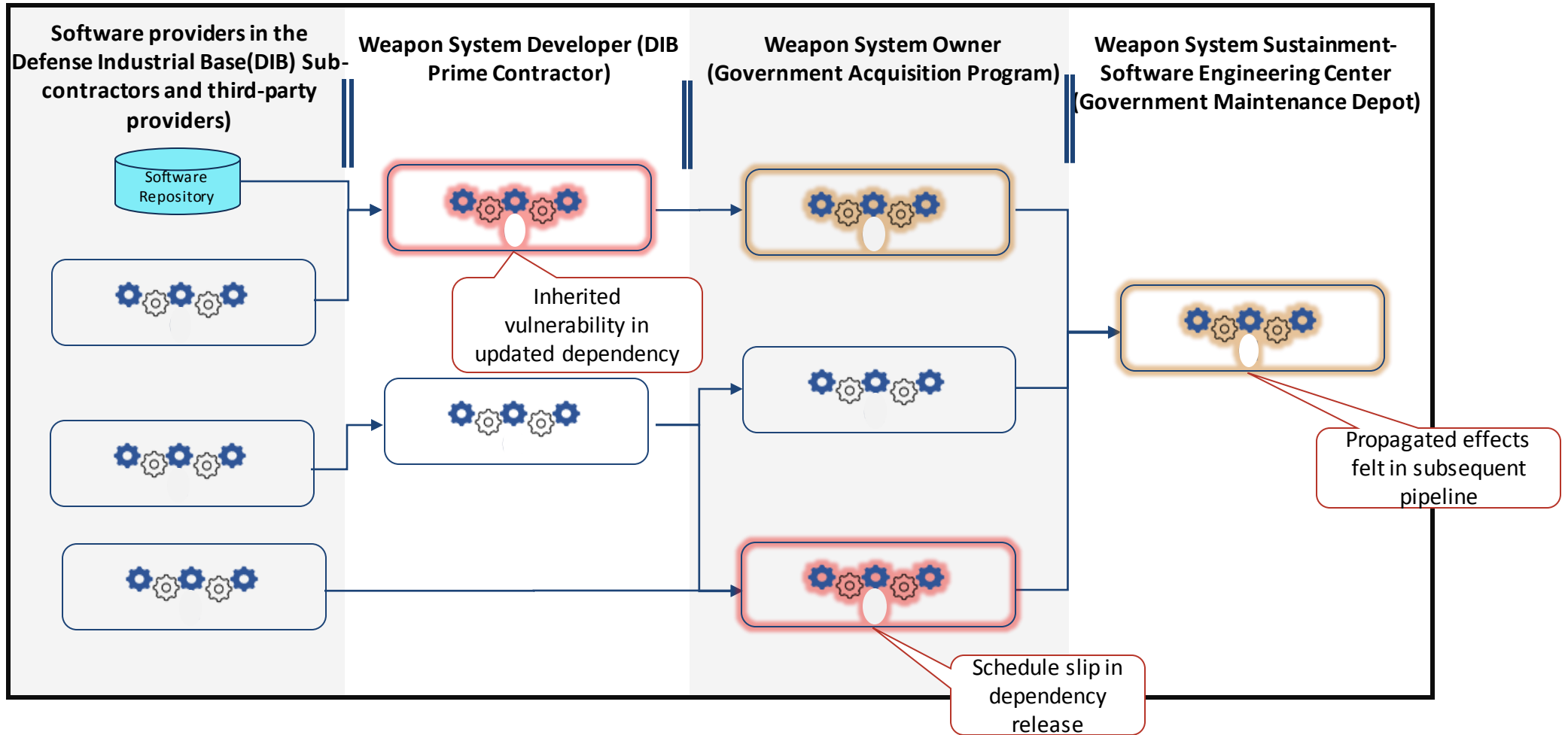
- Only supports % complete where
- behavior is not clearly defined when some but not all are estimated
- Currently does not support rates and schedule projections,
- no adjustments for estimation bias

# Work in Progress

Use local prototype to PM build dashboard

Develop the Multi-pipeline Simulation

Draft White Paper on Program Management using DevSecOps Data



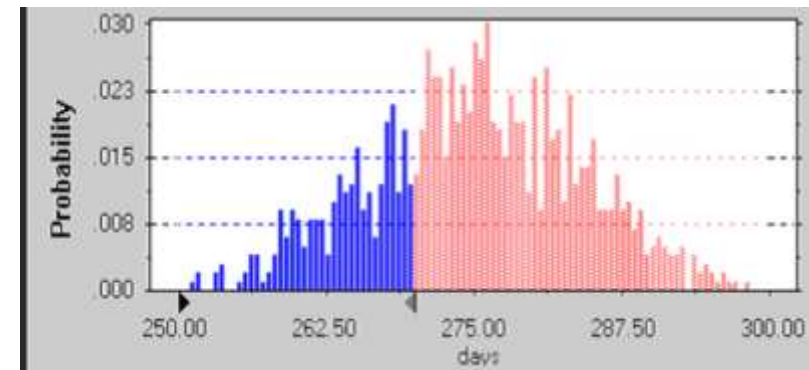
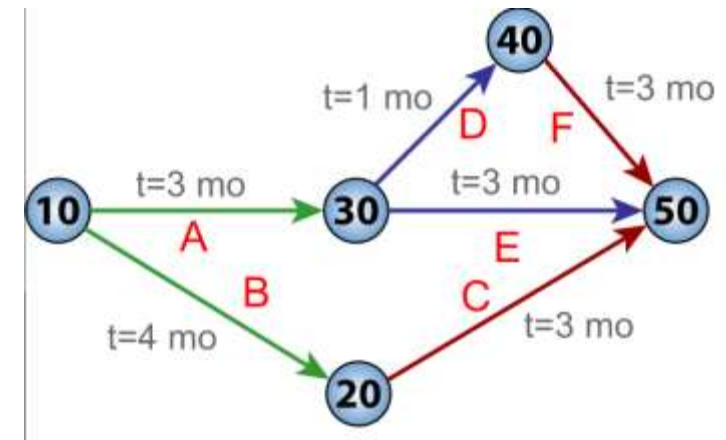
# Multi-Pipeline Projections: Approach

## Approach:

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

## Data needed

- Node structure (WBS, product dependencies)
- For each Pipeline
  - **Effort Rate** and **variation** (by skill ?)
  - **Production Rate** and **variation** by **work type**
  - **Primary work** and **Rework** by activity
  - Defect Rates



# Discussion for Multi-Pipeline Scenarios

## Scenarios

- 1) Identify a schedule slip at a node
  - determine schedule milestone effects
  - explore alternatives with shifting effort or work
- 2) Vulnerability introduced,
  - determine remediation effects after discovered
  - explore alternatives
- 3) What else is high priority?

# Multi-Pipeline Projections: Data and Challenges

## Challenges Include:

- Surrounding tools such as GitLab and Jira manage the tasking,
  - Use a front end (Hygeia) isolate the interface with issue tracking tools
- Confounders such as effort variation
  - may not be visible
  - Include **assumptions** such as full time-team member allocation
- Pipelines may have different iteration cadence
- What additional contextual information is required?
  - How do we select work building toward a capability?
    - Use labels and WBS hierarchy
  - Are there distinct work types (components, sub-domains, front-end/back-end) that affect work rates or variation?
  - How to separate primary work from rework

# White Paper On How The DevSecOps Pipeline Can Provide PMs With Actionable Data

Modern software development requires changes in the way PMs make decisions on their programs, especially large program with multiples pipelines

Currently working on a short (5-7 pages) White Paper

- Provide overview of data needed to support PM decisions in agile, multi-pipeline programs
- Provide insight into how pipeline data can support PM decisions

Current topics include:

- Cost
  - Schedule
  - Performance (includes Quality)
  - Risk
  - Scope
  - Organization / Staffing
  - Processes
  - Stability
  - Documentation (to include Government Baseline)
- Final product expected prior to next QUARP

# Emphasis: Program Decisions, example Scope

In modern software development scope is more flexible.

- As users' priorities for specific capabilities change, the scope can be adjusted to allow new priorities to be completed sooner
- New priorities can be swapped for less important capabilities or cost and schedule can be added

To help make the add a new capability that is high **priority**, the PM needs to understand:

- Any **dependencies** on priorities that are being moved to later
- The **relative effort** needed for the new capability versus those being moved to later
- **Availability** of any specific SMEs and **pipeline capacity** needed for the new capability
- Any **dependencies** for the new capability (either internal or external)

Pipeline data can help provide this kind of data to the PM

# What areas are most important to you?

What data would you use the most in making program decisions with respect to software?

- Cost
- Schedule
- Performance (includes Quality)
- Risk
- Scope
- Organization / Staffing
- Processes
- Stability
- Documentation (to include Government Baseline)

# PM Decision Example: Scope

In modern software development scope is more flexible.

- As users' **priorities** for specific capabilities change, the scope can be adjusted to allow new priorities to be completed sooner
- New priorities can be swapped for less important capabilities or cost and schedule can be added

To help make the add a new capability that is high priority, the PM needs to understand:

- Any **dependencies** on **capabilities**
- The **relative effort** needed for the new capability versus those being moved to later
- **Availability** of any specific **SMEs** and **pipeline capacity** needed for capability

Pipeline data can help provide this kind of data to the PM

# Next Steps

Demonstrate % complete and projections with

- Simulated multi-pipeline
- Prototype local project

Document indicator definitions

Demonstrate PoPs Dashboard using Hygeia

Complete White Paper

# Wrap-up

How well did this workshop accomplish our purpose?

## Next steps

- Complete data dictionary and pipeline of pipeline descriptions

# Thank You!

We are grateful for your time and commitment!

For any questions, information, or content please contact

- Bill Nichols [wrn@sei.cmu.edu](mailto:wrn@sei.cmu.edu)
- Hasan Yasar [hyasar@sei.cmu.edu](mailto:hyasar@sei.cmu.edu)
- Rob McCarthy [rmccarthy@sei.cmu.edu](mailto:rmccarthy@sei.cmu.edu)
- Luiz Antunes [lantunes@sei.cmu.edu](mailto:lantunes@sei.cmu.edu)
- Chris Miller [clmiller@sei.cmu.edu](mailto:clmiller@sei.cmu.edu)

Or

Akia Williams [awilliams@sei.cmu.edu](mailto:awilliams@sei.cmu.edu)