Automating DevSecOps Development Data

Program Management at the Speed of Relevance

William Nichols, Christopher Miller

Luiz Antunes, Rob McCarthy

Hasan Yasar

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

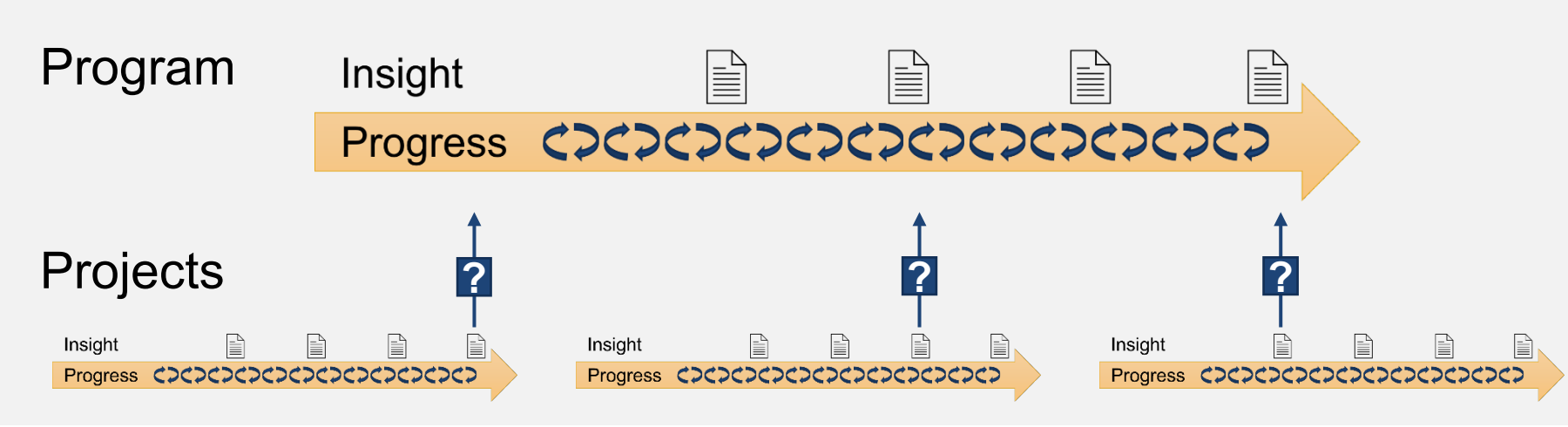
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0305

DoD Program Management Challenged by Increased Pace

Theoretically, DevSecOps provides constant opportunities for feedback and **corrective actions,** but **controlling a program** is different than **controlling a pipeline**.

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



Research Questions

What are the information gaps for DoD program managers in fast moving continuous delivery environments using DevSecOps?

What program information is needed for prediction and actionable decisions?

➔What data can we extract from the **pipeline(s)** or **other sources**?

How does data for program management differ from pipeline management? Is there an opportunity for reuse?

Can we gather and analyze that data to support real time (daily?) decisions?

➔How should the data be joined, transformed, and labeled to retain context?

How should we present information indicators to decision makers?

Approach

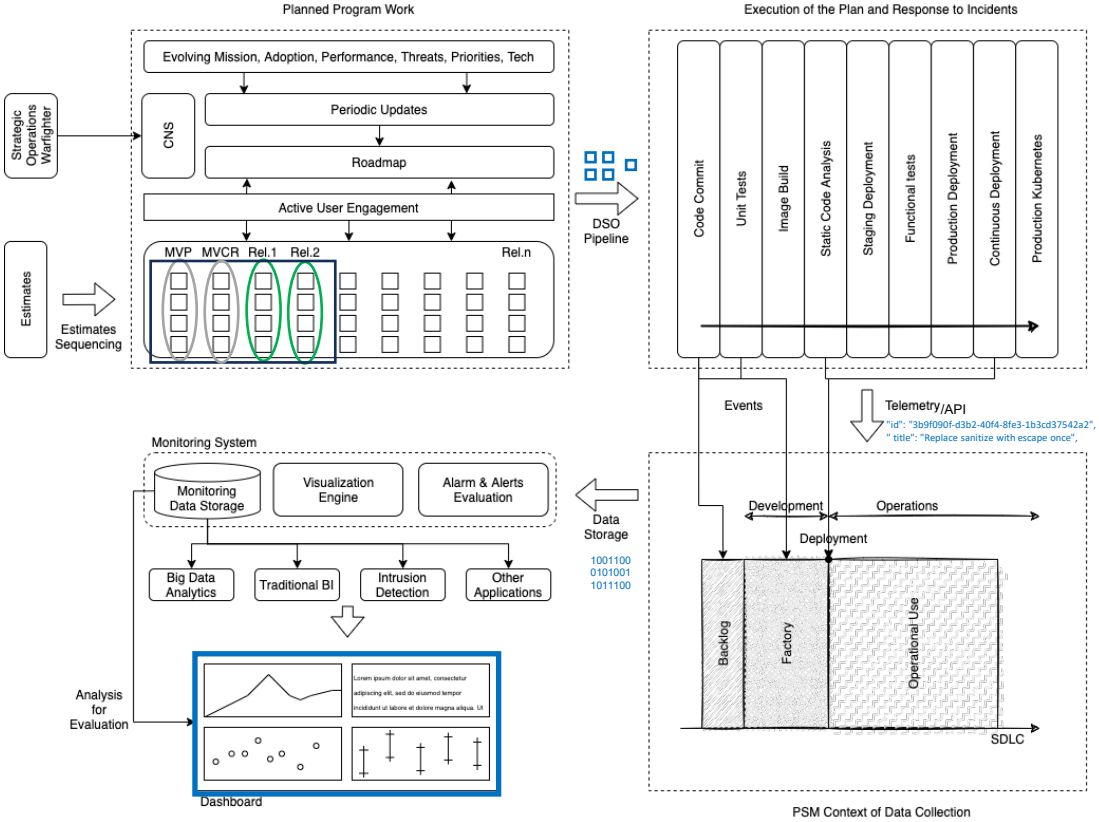
- Identify subject matter experts (SME) for our Quarterly Review Panel.
- Establish and validate key scenarios for Program Management (establish scope).
- Construct prototype pipeline for information modeling and prototyping.
- Hypothesize program management indicators.
- Model the pipeline for data collection points and data available (tool, type of event, timestamp, work item, etc.).
- Prototype **data** collection, storage, and indicator generation with **synthetic** data.
- Analyze the DevSecOps (DSO) pipeline in program context. (Identify needs and opportunities.)
- Validate results with the Quarterly Review Panel.

Problem Scenarios

The program must track progress toward important events (e.g., Minimum Viable Capability Release [MVCR]).

Scenario 1: Status and Projections	Scenario 2: "What if?"
Will we make the schedule commitment? **Where are we now?** **What is our completion rate?** • Actual effort applied • Which items are complete? • Which items remain for a capability? • % Complete overall/capability?	**Can we accept a change?** **What if we reduce scope?** **What if we add resources?** • What is the required effort? • How will our completion rate change? • How are capability commitments affected?
When will we finish current work? • Projection to complete (schedule/cost) • Projection to complete capability • Confidence range of estimates	**If we add effort, how long will it take?** • New projection to complete • New projections for capability complete • Confidence range
• Completion rate • Rework rates	• Estimation bias and variation

DevSecOps Measurement



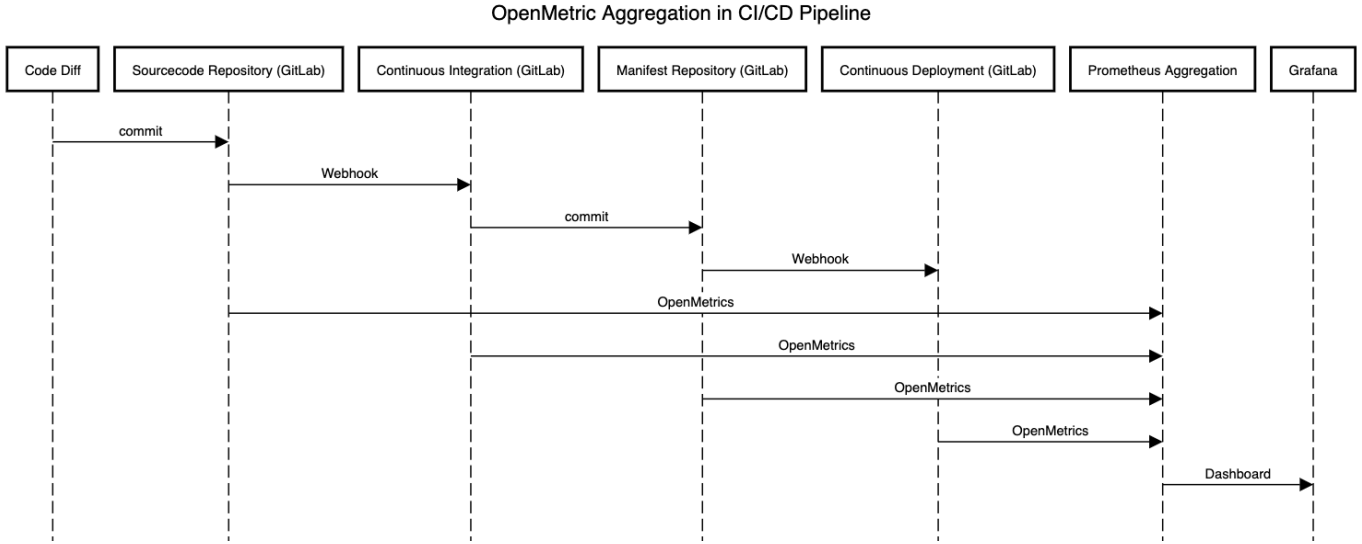
Planned work includes the work breakdown structure (WBS), work packages, work sequencing, and estimates.

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.

Combine and **Transform** the Data in Context



Model the pipeline.

Use labels to trace WBS, roadmap, and backlog to work packages.

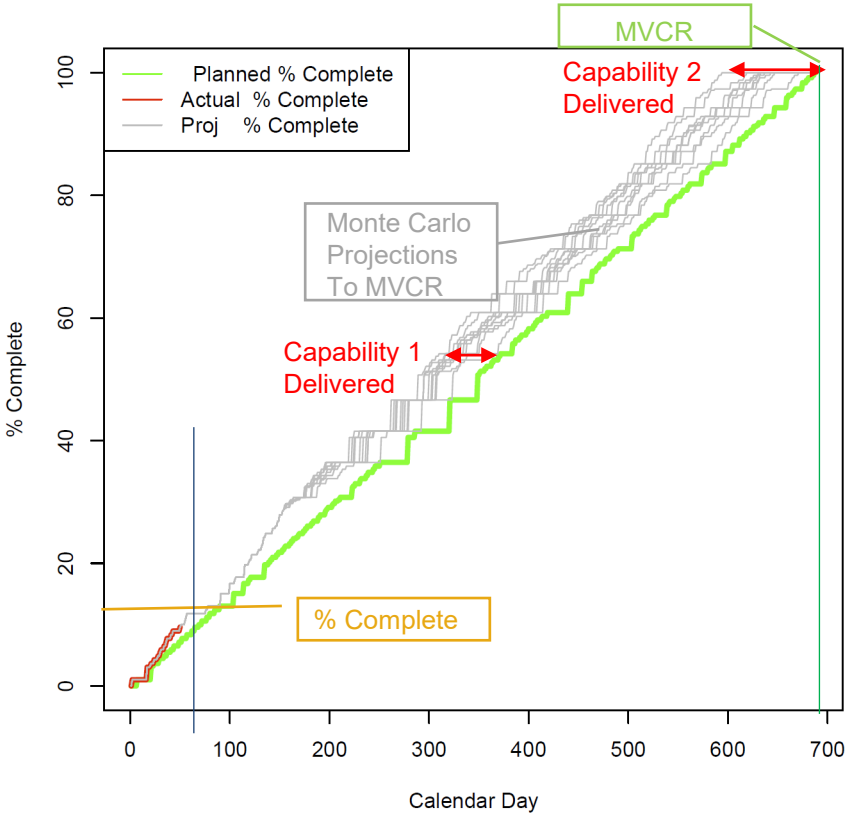
Data is collected from key events.

Precisely define data.

- Actual times
- Lead times
- Estimated dates

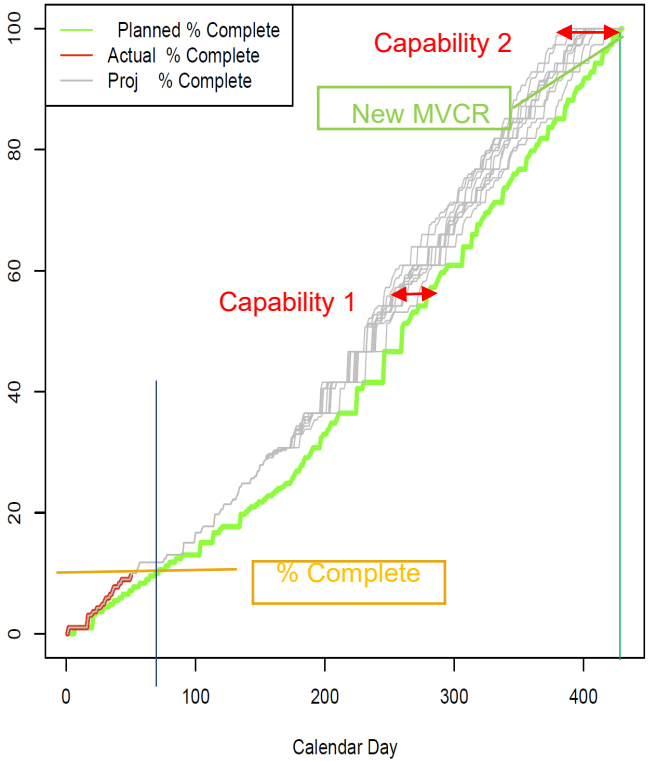
What-if Analysis, Percent Complete with Simulated data

Status and Projection to Complete



What if need date changes priorities?

What can we commit to with early action?



Add development staff.

Brings in schedule by 250 days

Halves the expected range

Summary and Future Work

Research Results

- Determined that the pipeline data must be supplemented with external sources
- Demonstration and proof of concept of DevSecOps automated development performance measurement and management-oriented visualization
- Completed a starter set of management issues and supporting DevSecOps measures, with their operationalization within a pipeline
- Demonstrated data storage, transformation, and data relationship

Future work

- Extend to multiple interacting pipelines, continuous Authority to Operate (cATO)
- Expand the types of questions and analysis supported
- Apply ML/AI to pipeline data for predictive and causal analysis
- Innovate new displays to make progress views consistent throughout and across programs