

# Automated Continuous Estimation for Pipelines of Pipelines

## QuARP #1 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).

DM22-0138

# 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
Review of ACE/PoPs Goals	5
Overview of Pipeline-of-Pipelines definitions and variants	20
Review of Pipeline-of-Pipeline programmatic problems	5
Discussion	10
Summarize Next Steps	5

# Goal: 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
- Provide evidence for corrective actions

DSO programs make realistic schedule and cost commitments

Status and projection models available in real time

Model pipeline-of-pipeline systems using Monte Carlo.

Automate the Program Management Status report for DevSecOps pipelines.

Directly collect data from DevSecOps pipeline tool

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

# FY22 Milestones — Status

Quarter	Milestone	Status
Q1	M1. Validate PoPs definitions and pipeline variations	In Review, will continue into Q2
Q2	M2. Complete PoPs simulated data instantiation	Started
Q3	M3. Validate PoPs data for multiple scenarios	Planned
Q4	M4. Validate PoPs Dashboard with QuARP SME	Planned

# Definition of Key Terms

**Pipeline** - software or hardware design technique in which the output of one process serves as input to a second, the output of the second process serves as input to a third, and so on, often with simultaneity within a single cycle time [4]. Pipeline sometimes refers only to an automated portions of the workflow.

**Pipeline instance** is the realization of a single process workflow and the tools to execute the workflow for a specific software language and application type for a software component. As much of the pipeline process is automated as is practicable. [1, 5]

**Software factory** - includes one or more continuous integration / continuous delivery (CI/CD) pipelines, each producing unique artifacts. It is designed for multi-tenancy to leverage economies of scale and automates software production for multiple products.

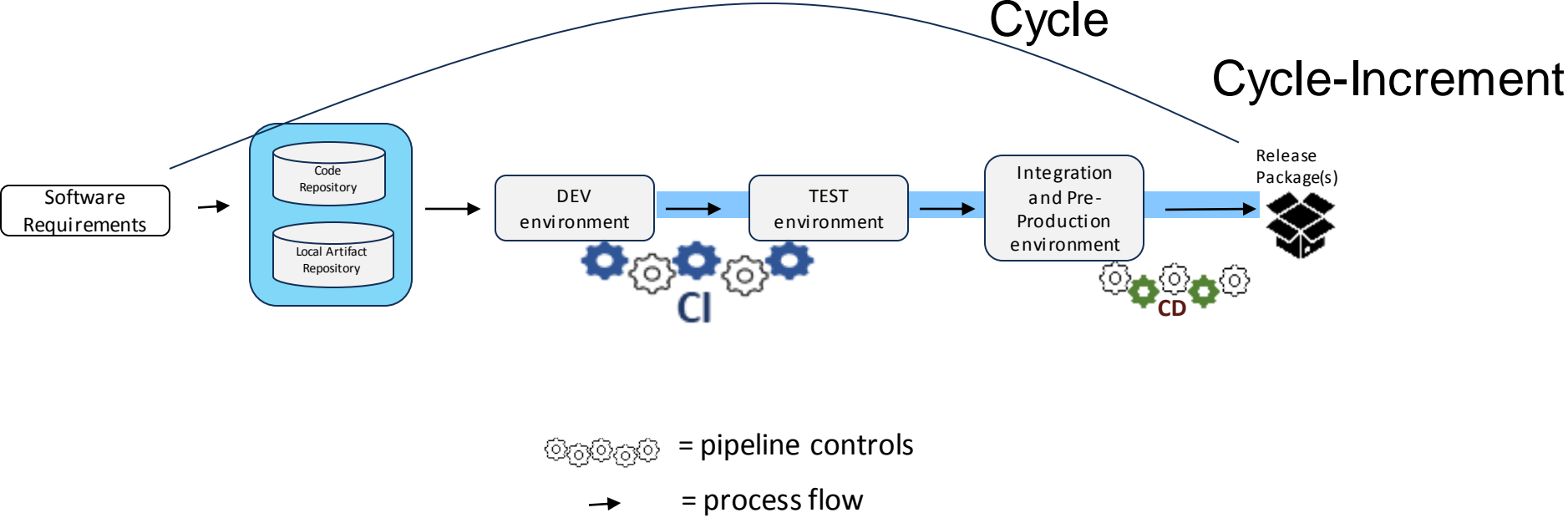
**Milestone** – an important event such as release of capability into test or deployment of a specified capability into the field.

**Release** - the interim state of software having completed integration and test, but not yet having been deployed.

**Deployment** – the state of software being available for use in the designated environment

**Throughput** - a measure of the pipeline completion rate, output per cycle.

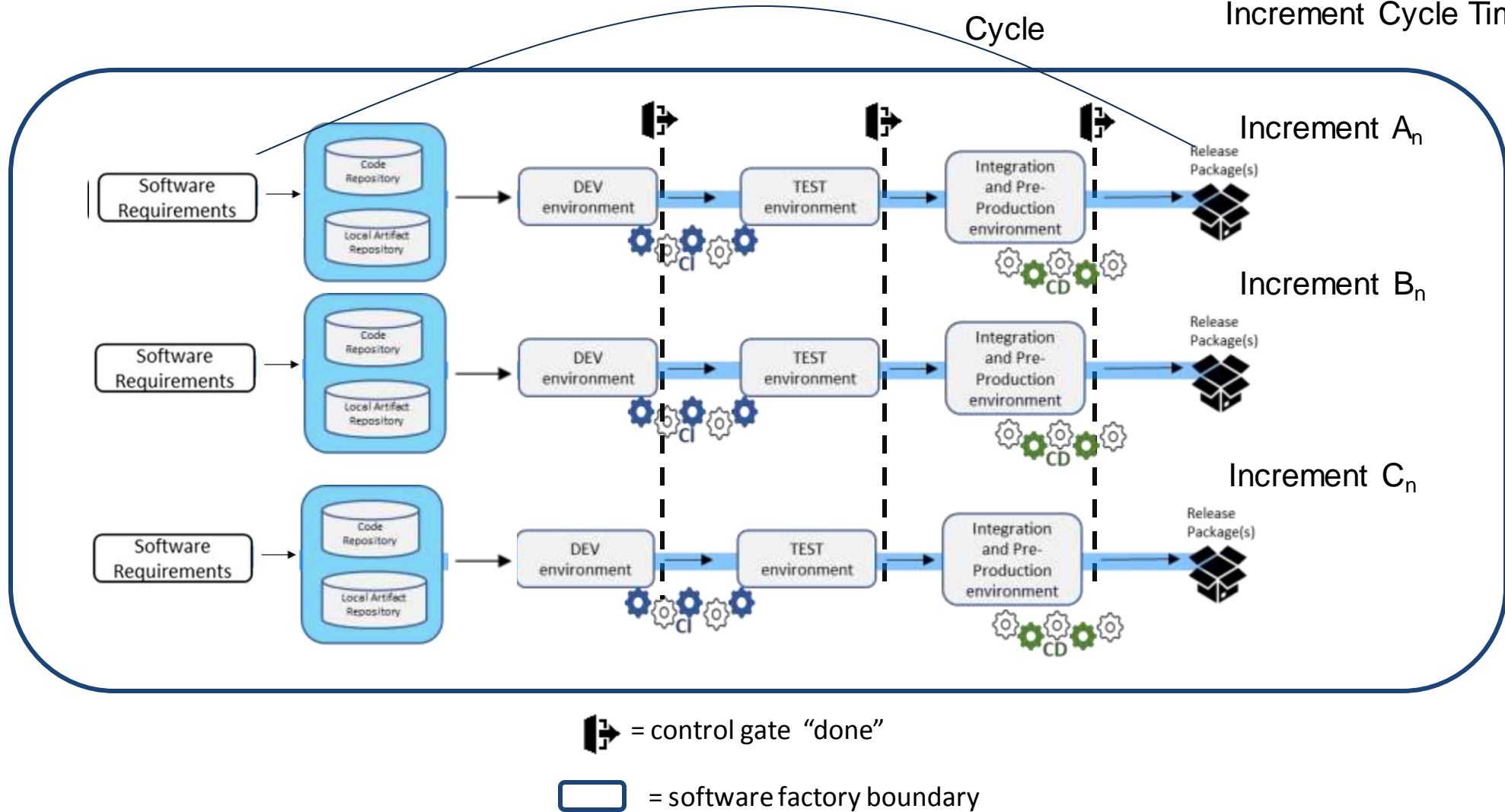
# Simplified Single Pipeline



This graphic consolidates the activities into three groups (DEV, TEST, and Integration). The gear icons represent the tools that make up the pipeline. Cycles can be defined for release and deployment. Not all pipelines deploy every release. Pipeline is overloaded to refer both to both a workflow with inputs and outputs and to the automated portions

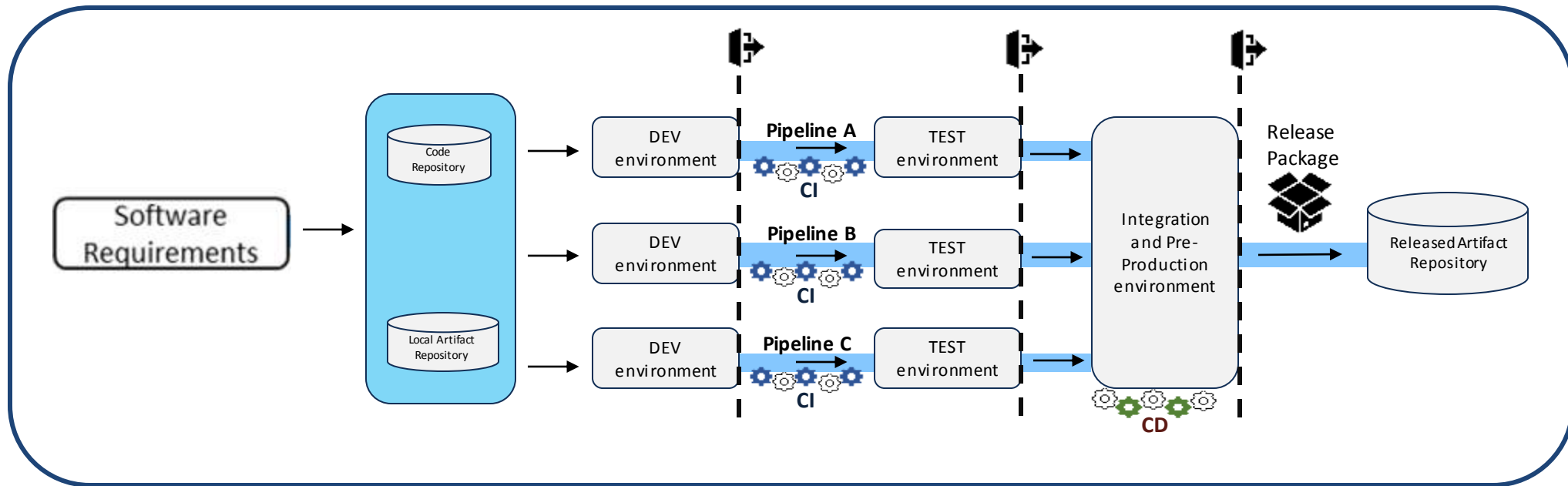
# Software Factory – Example Parallel Releases

Resources  
Throughput/Variation  
Increment Cycle Time



# Software Factory – Example (Pipelines in Parallel to Integration)

Derived from DevSecOps Reference Guide

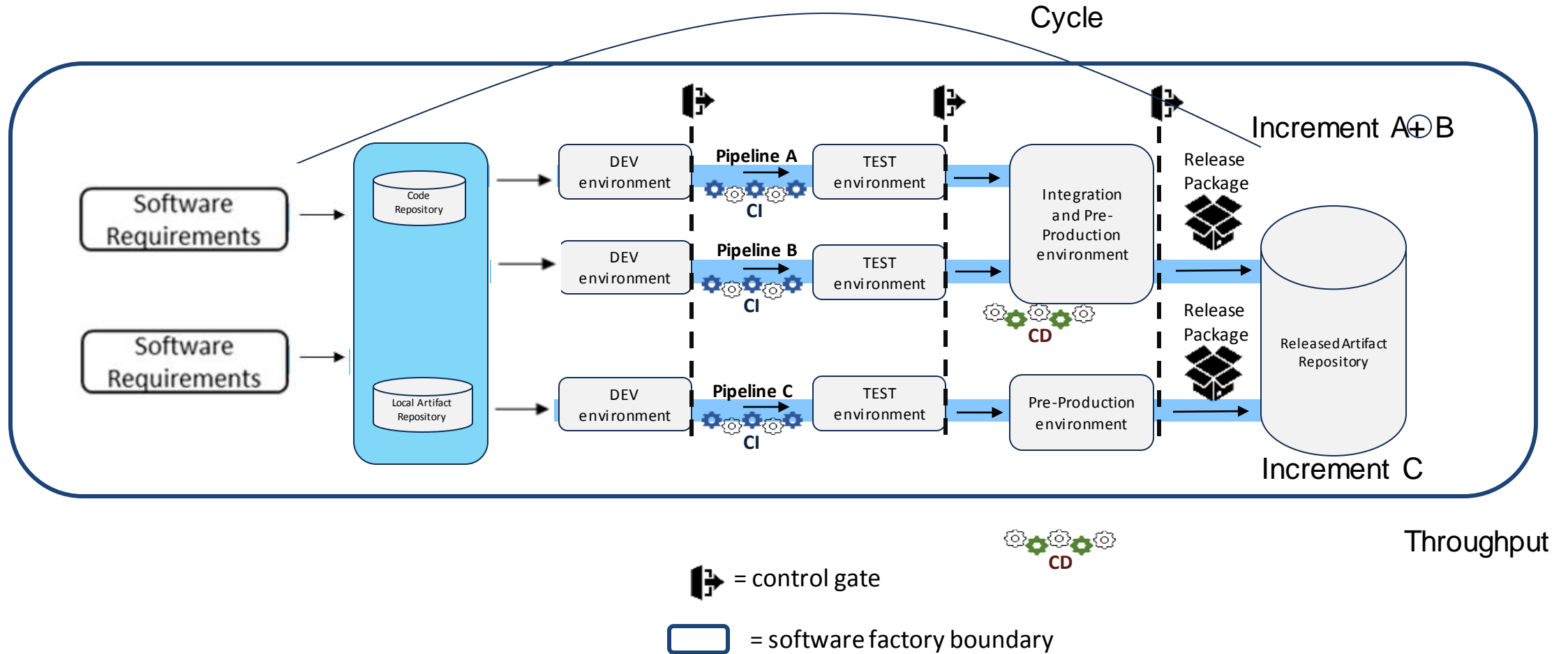


Control gates imply definition of done  
Release may not include deployment

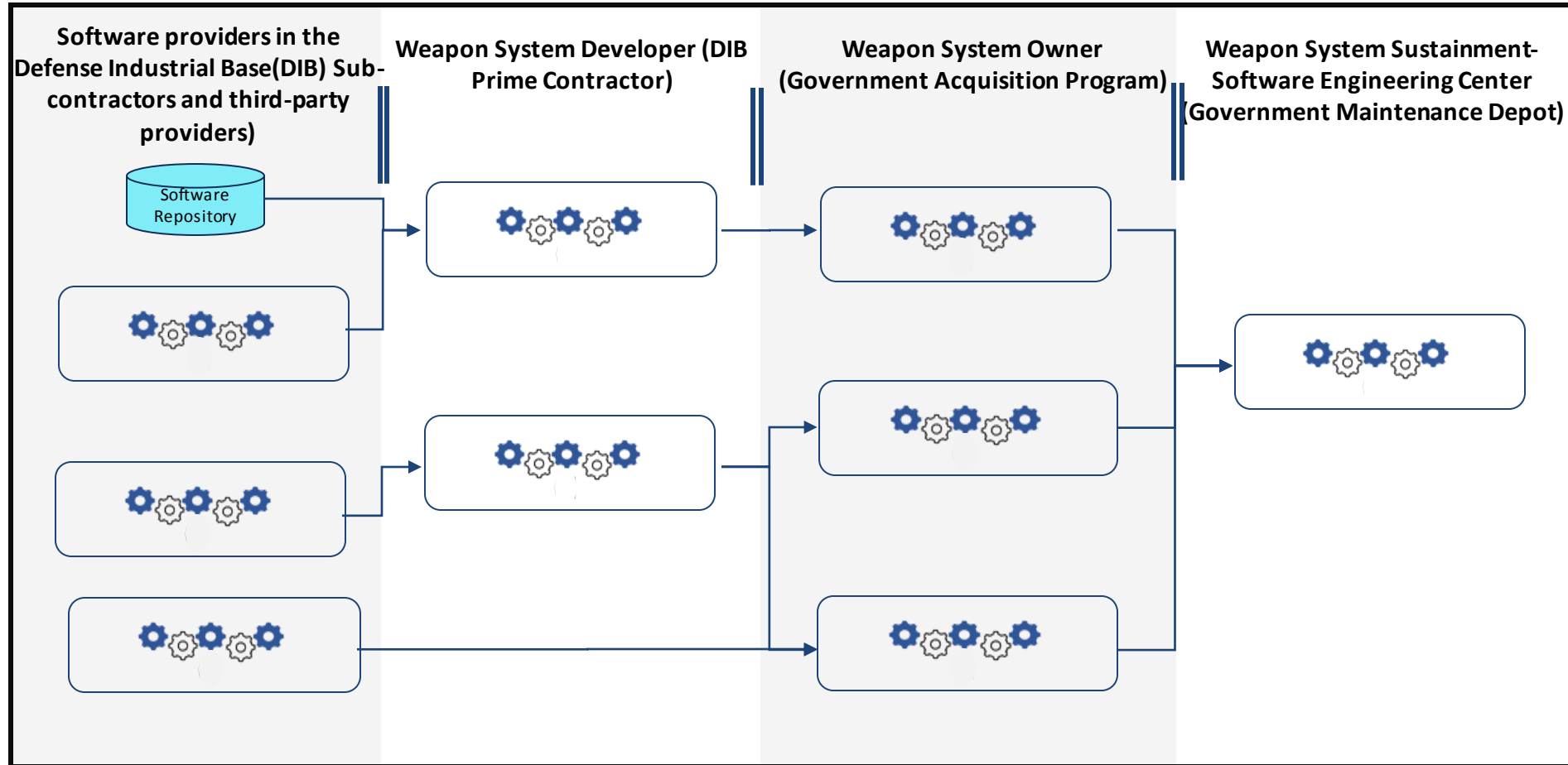
 = control gate

 = software factory boundary

# Software Factory – Example, Merge and Parallel release



# Pipeline of Pipelines (PoPs) for DoD Weapon System Acquisition and Sustainment



DevSecOps PoPs need to deal with pipelines in parallel and in series simultaneously  
All workflow have unautomated gaps  
Sequential pipelines include unautomated gaps

# Technical Approach: Pipeline Aggregation and Dependency

**Analyze network dependencies** with Monte Carlo (MC).

- Coordination is based on “has been delivered by” need date.
- MC needs parameterizations, such as **dependencies** and **ranges** for
  - Throughput/Production rate (for individual pipelines)
  - Problem/defect origin/find rates in test/integration (between pipelines)
  - Resolution lead time (affects multiple pipelines)

Pipelines could operate at different natural frequencies and produce different products.

- Calibrate schedule, cost, technical performance, quality for individual pipelines
- Correlate production to defined work along with production **variation**
- Define global percent completion and “long pole” in schedule

Unprecedented at this scale and frequency. Must be automated for PMs.

# Pipeline of Pipeline Issues

How do we predict milestone dates (or cycle) across multiple pipelines?

- Velocity has a different meaning for each pipeline. What can we know about the rates?
- What are successful practices for distributing work to pipelines?

How do we compare or normalize pipeline throughputs?

Can we focus on input/outputs or do we need internal pipeline insight?

What are the tradeoffs when compelling pipelines to synchronize? (local and global)

How do we characterize wait times or disruptions? (what if test fails and rework is needed?)

How do we recognize risk to milestone success? (what disruptions to simulate?)

Other issues remain; such as quality, software assurance, and delivered value.

# Discussion

Are these pipelines-of-pipelines representative?

Are these programmatic problems relevant?

What have we missed?

# 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)