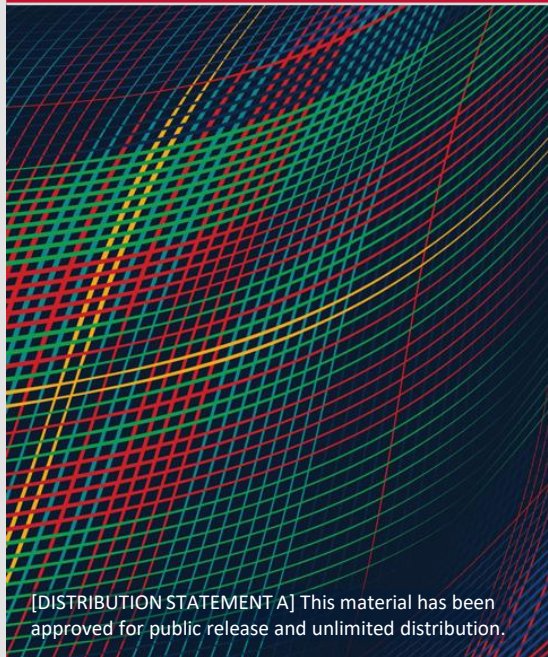
Acquisition Software Management Planning (SMP) Is Critical to Mission Success

MAY 7, 2026

Michael Bandor, presenter
Carol Woody, Ph.D.

Carnegie Mellon University
Software Engineering Institute



Document Markings

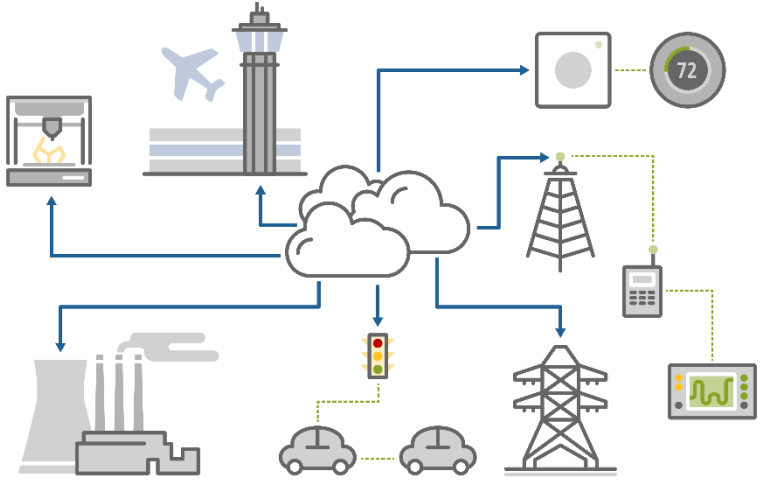
This material is based upon work funded and supported by the Department of War under Air Force Contract Nos. FA8702-15-D-0002, and FA870225DB003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM26-0225

Software is Everywhere

You think you're building (or buying, or using) a product

car or truck	satellite	mobile phone	development tools
home security system	aircraft	pacemaker	security tools
home appliance	financial system	bullets for a gun	



You're getting ***a software platform:***

- Software is a part of almost everything we use
- Software defines and delivers communication
- Software is used to build, analyze and secure software
- Software controls connections among products

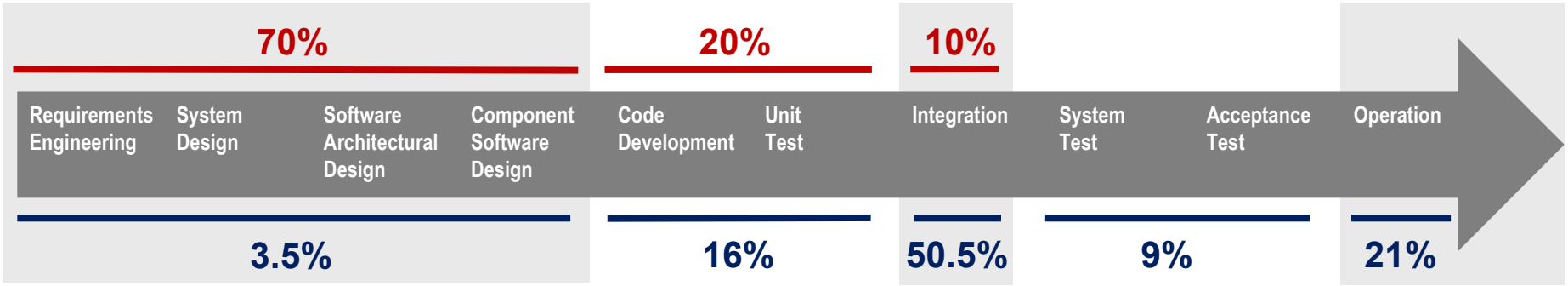
Software is not Visible and Does Not Wear Out

"The B-52 lived and died on the quality of its sheet metal. Today our aircraft will live or die on the quality of our software." - Air Force General

Quote: "Delivering Military Software Affordably," Defense AT&L, March-April 2013

All Software Has Defects and Potential Vulnerabilities

Where Software Defects Are Introduced



Where Software Defects Are Found

Best-in-class results: <600 defects per million lines of code (MLOC)
Very good code: 600 to 1,000 defects per MLOC
Average quality code: 6,000 defects per MLOC

5% of these defects are potential cyber vulnerabilities

Source Woody, Carol; Ellison, R; & Nichols, W. *Predicting Software Assurance Using Quality and Reliability Measures*. CMU/SEI-2014-TN-026. SEI. 2014. https://doi.org/10.1184/R1/6582113.v1

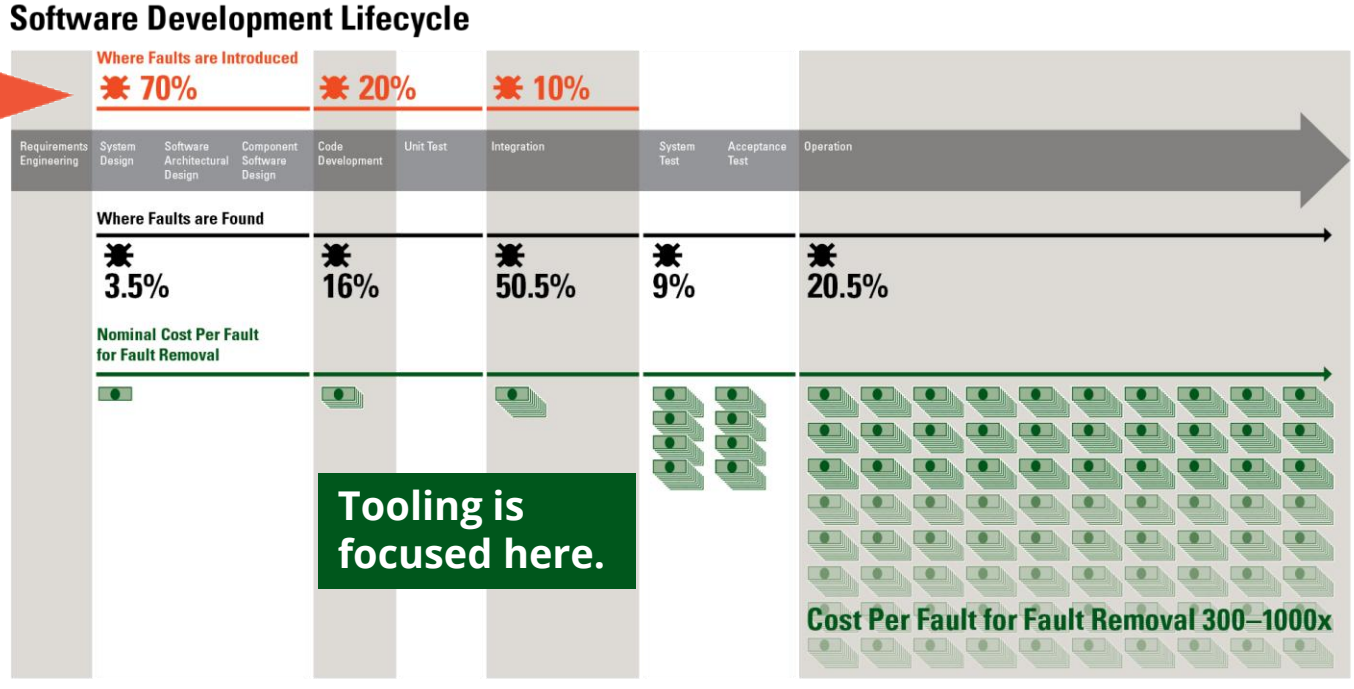
Software Defects: *Introduction, Discovery, and Cost*

Carnegie Mellon University Software Engineering Institute

Defects account for 30–50% percent of total software project costs.

- Most are introduced before coding (~70%).
- Most are discovered at system integration or later (~80%).

Opportunities to reduce the largest volume of vulnerabilities at the lowest cost are too frequently lost

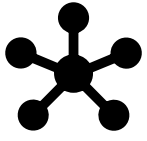


Risk from Software Is Increasing

Carnegie Mellon University Software Engineering Institute

Three aspects to increasing risk exist:

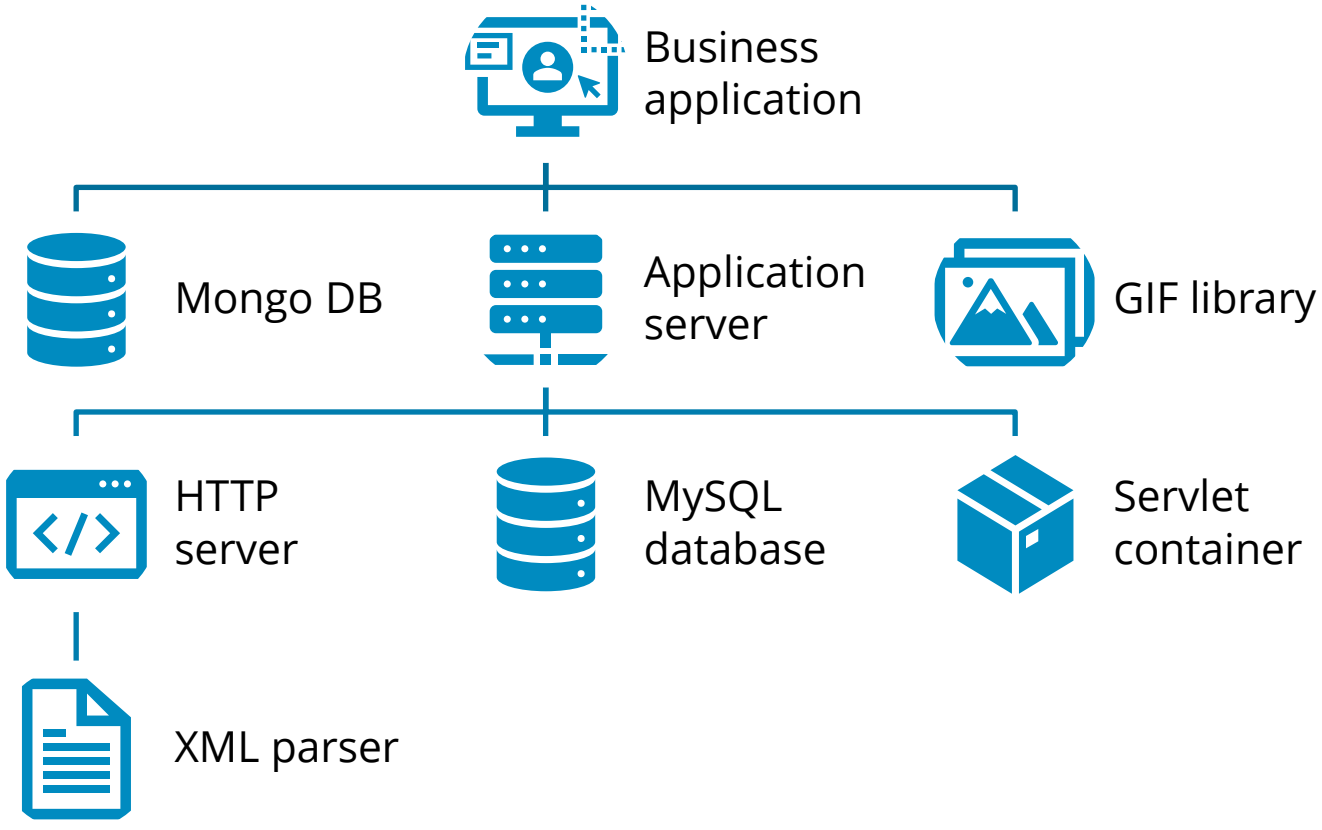
1. Growing Reliance on Software
   - Increased reliance on commercial and open source software (OSS) built using unknown processes
   - Millions of lines of software code handling an ever-increasing amount of functionality, bringing thousands of potential software vulnerabilities
   - Use of software in more systems, including cyber-physical systems, increases severity of impacts



2. Growing Connectivity
   - Increased connectivity linking systems to other systems and connecting to new types of devices (e.g., Internet of Things) allows additional access points and opportunities for attacker to pivot
   - Increased system and device remote communication capability leads to additional attack surfaces
   - Unexpected system responses can trigger a failure condition in another system

3. Growing Attacker Capabilities
   - Growing number of offensive tools (including repurposed defensive tools)
   - Reliance on open source provides attackers with additional information to develop exploits with

System Development Is Primarily Software Assembly



Components are assembled from new code, reused code, language libraries, and external services.

90% of organizations rely on open source software (OSS)

Github Website. October 17, 2025 [accessed]. https://octoverse.github.com/2022/

Estimated OSS value of ~9 Trillion USD

Hoffmann, Manuel; Nagle, Frank; & Zhou, Yanuo. "The Value of Open Source Software." Harvard Business School Working Paper, No. 24-038, January 2024.

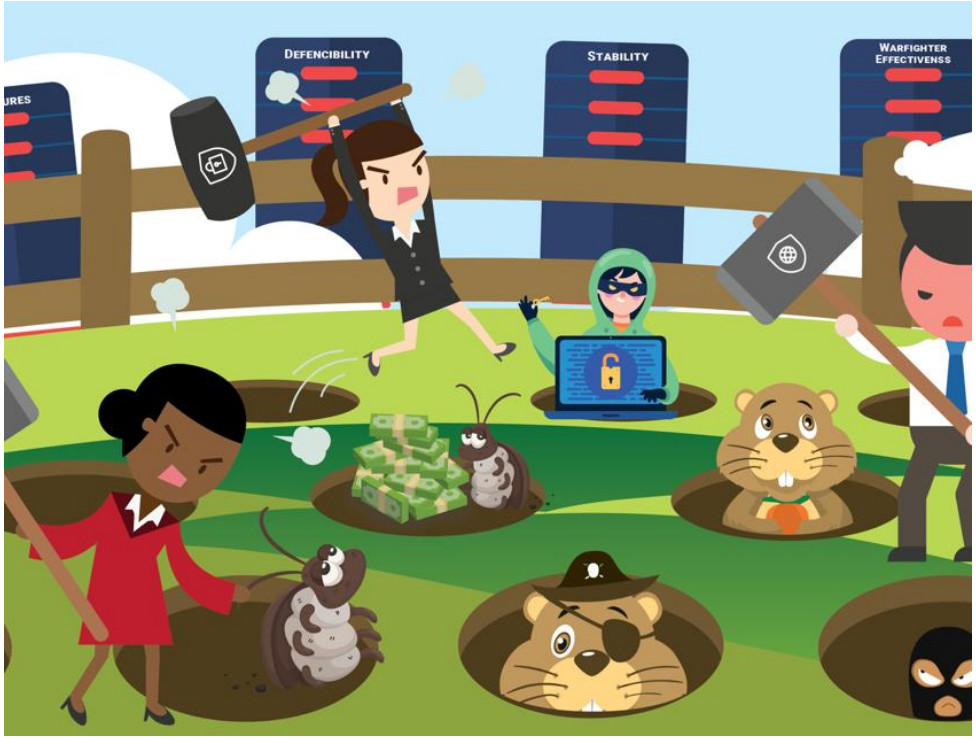
Software Supply Chain Attack History

In a survey of more than 230,000 organizations, 98% have a relationship with a third party that has already been breached within the last two years.

https://www.securityweek.com/98-of-firms-have-a-supply-chain-relationship-that-has-been-breached-analysis/

- Heartland Payment Systems (2009)
- Silverpop (2010)
- Epsilon (2011)
- New York State Electric and Gas (2012)
- Target (2013)
- Lowes (2014)
- AT&T (2014)
- HAVEX / Dragonfly attacks on energy industry (2014)
- DoD OPM breach (2015)
- Equifax (2017)
- Marriott (2018)
- SolarWinds (2020)
- Log4j (2021)
- Medibank (2022)
- MOVEit (2023)
- CrowdStrike (2024)
- AI Enabled Attacks (2025)

Avoiding Whack-A-Mole



It is extremely important to have a well-defined Software Management Plan (SMP) before implementing and managing software to avoid costly mistakes.

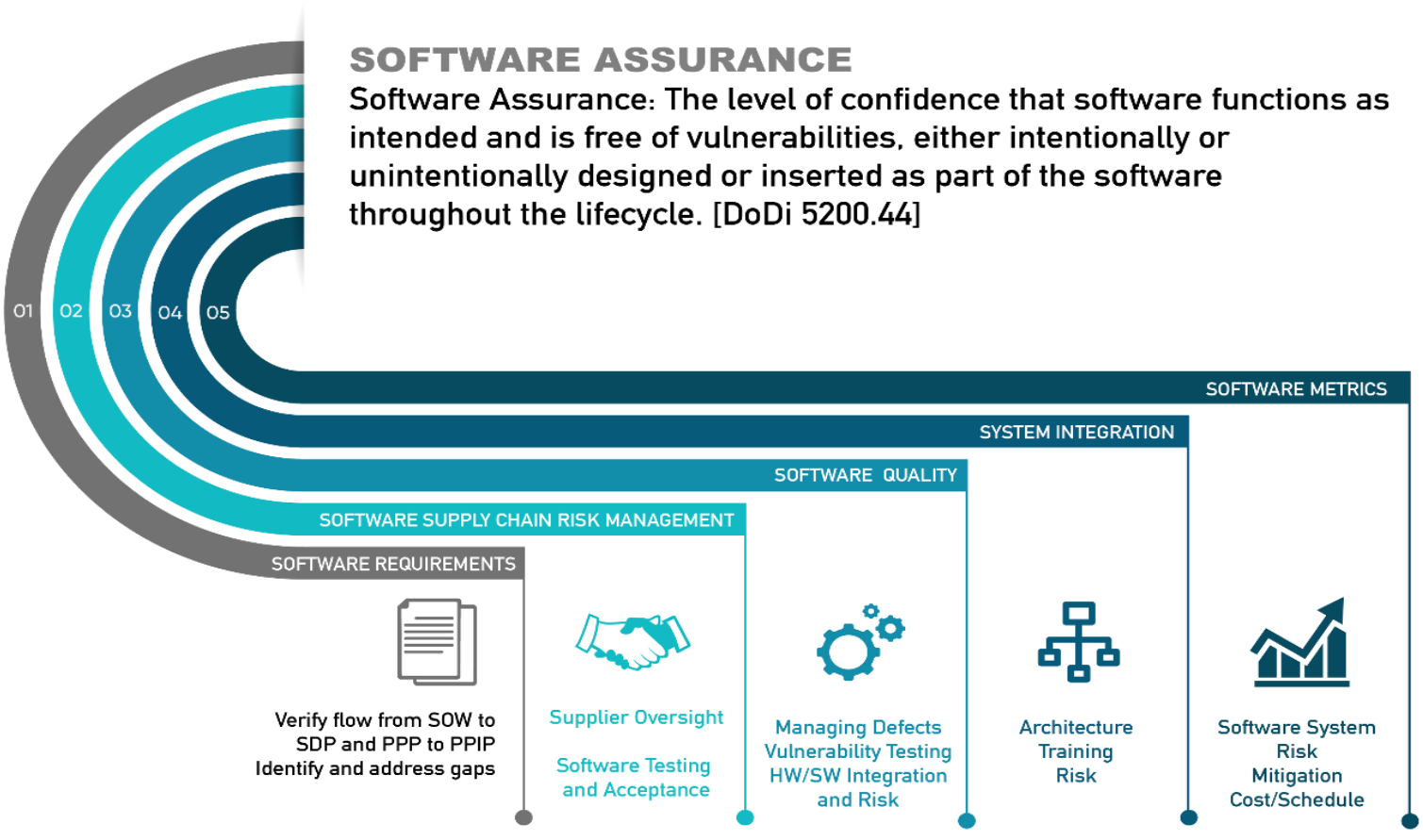
- Each organization must analyze the impact of software on system mission including the following:
  - Budget constraints
  - Solution space
  - Supplier relationships
- The end state must consider the perspective of all stakeholders, including software architects, developers, system administrators, test and quality-assurance engineers, security officials, management, end users, etc.

Addressing Software Assurance for Sentinel

Integrating Software Management Planning (SMP) Into the Acquisition Lifecycle

Carnegie Mellon University
Software Engineering Institute

Five Key Software Assurance Pillars and Effective SMP



https://www.sei.cmu.edu/blog/the-five-pillars-of-software-assurance-in-system-acquisition/

(1) Software Requirements for SwA

Definition:

Required functionality assigned by the system design to the software/firmware components of the system to perform their allocated capabilities within acceptable constraints and quality to support the system mission.

Acquisition Connections:

- Requirements relevant to SwA can be extracted from the Statement of Work (SOW) and mapped to the Software Development Plan (SDP).
- Cybersecurity requirements relevant to SwA are identified in the Program Protection Plan (PPP) and mapped to the Program Protection Implementation Plan (PPIP).
- Deliverables expected to address SwA need to be identified in the SOW.

(2) Software Supply Chain Risk Management for SwA

Definition:

Governance and monitoring of people, processes, and technology needed to acquire, manage, and sustain third-party software/firmware to ensure it meets system requirements and mission needs throughout its implementation.

Acquisition Connections:

- Use of third-party software offers cost and schedule advantages but can create cybersecurity and lifecycle risks that must be managed.
- Constant patching, technology refresh integration, and other mitigation techniques for vulnerability management are a reality of software use that must be implemented and managed across the lifecycle of all third-party software.

(3) Software Quality for SwA

Definition:

Building confidence that the people, processes, and practices used to create and acquire software are delivering needed mission results within acceptable levels of rigor and risk.

Acquisition Connections:

- Software quality reviews must focus both on the processes and the product to ensure it has sufficient assurance to meet the mission.
- Confidence for software quality must be established early in the lifecycle and confirmed with contractor reporting.

(4) System Integration

Definition:

Combining software components, hardware components, or both into an overall system that meets mission requirements. Each of these components can be assured within their development. Additionally, the integration of these components must be assured.

Acquisition Connection:

- Software expertise is needed at the governance level, system level, and within development and implementations teams to identify and address key software assurance gaps in processes and practices.
- Milestone reviews should be used to verify that the training and practices are in place to support software assurance success.

(5) Software Metrics for SwA

Definition:

Identification and analysis of metrics to demonstrate effective lifecycle software assurance to meet mission needs. These metrics provide objective data to evaluate progress, process, and quality of delivered software.

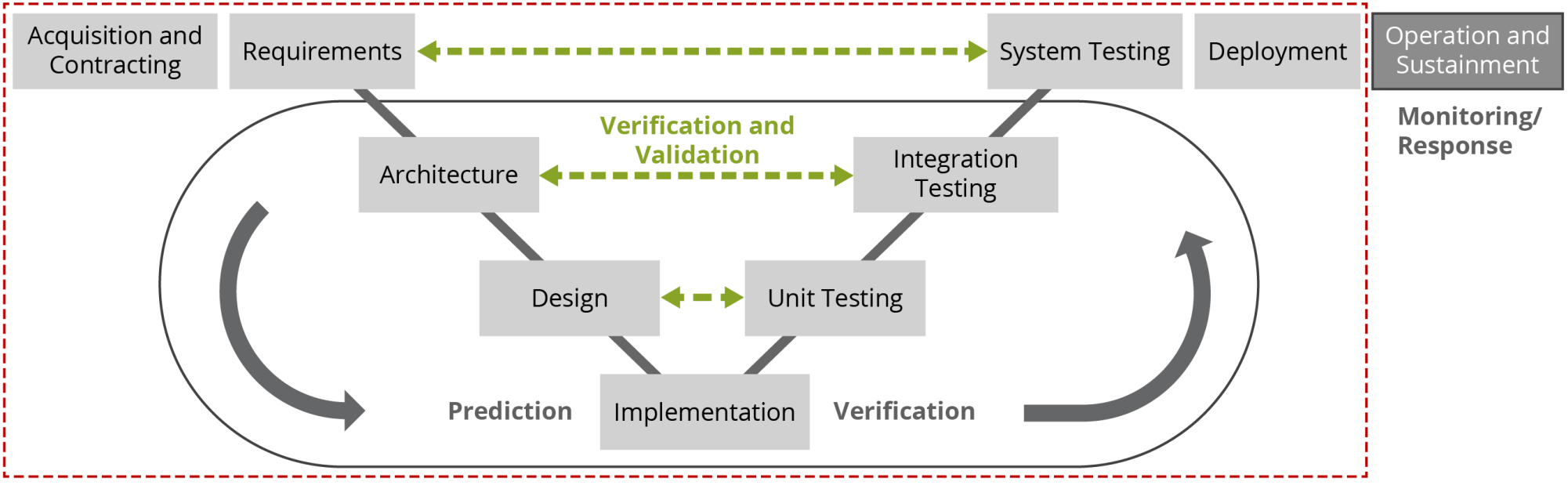
Acquisition Connection:

- Effective monitoring provides an opportunity for early identification and correction of software assurance gaps before they create an impact to cost and schedule.
- Metrics provide an objective way to evaluate progress, process, and quality of the delivered software.

SMP Prepares the Software for Mission Success

Carnegie Mellon University
Software Engineering Institute

Software Assurance Needs to be Throughout the Lifecycle



Next Steps

Carnegie Mellon University
Software Engineering Institute

Understand and Address Your Acquisition Oversight for Software Assurance (AOSA)

Carnegie Mellon University Software Engineering Institute

Plan for Potential Software Attacks

- Identify mission critical capabilities relying on software
- Identify/train acquisition stakeholders to participate in SwA collaboration

Perform Initial Assessment(s) to Establish

- Diagram your technology context(s)
- Analyze them to identify potential cyber attack risks
- Prioritize risks
- Identify mitigation options and plan for implementation
- Determine supply chain attack risks and implement mitigation options

Establish ongoing monitoring to address potential risk changes

Contact Information

Michael Bandor
mbandor@sei.cmu.edu

Carol Woody, Ph.D.
cwoody@sei.cmu.edu

Web Resources
https://sei.cmu.edu/

Email:
info@sei.cmu.edu