Continuous Verification & Validation of Critical Software via DevSecOps

Hasan Yasar

Technical Director, Adjunct Faculty Member

Software Engineering Institute | Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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.

DM22-1111

Definitions (IEEE Std 1012™-2016)

Verification: The process of providing objective evidence that the system and it products

- Conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all activities during each life cycle process
- Satisfy standards, practices, and conventions during life cycle processes
- Successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities

Validation: The process of providing evidence that the system and its products

- satisfy requirements at the end of each life cycle activity
- Solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions)
- Satisfy intended use and user needs in the operational environment.

Definitions (IEEE Std 1012™-2016)

Verification: The process of providing objective evidence that the system and it products

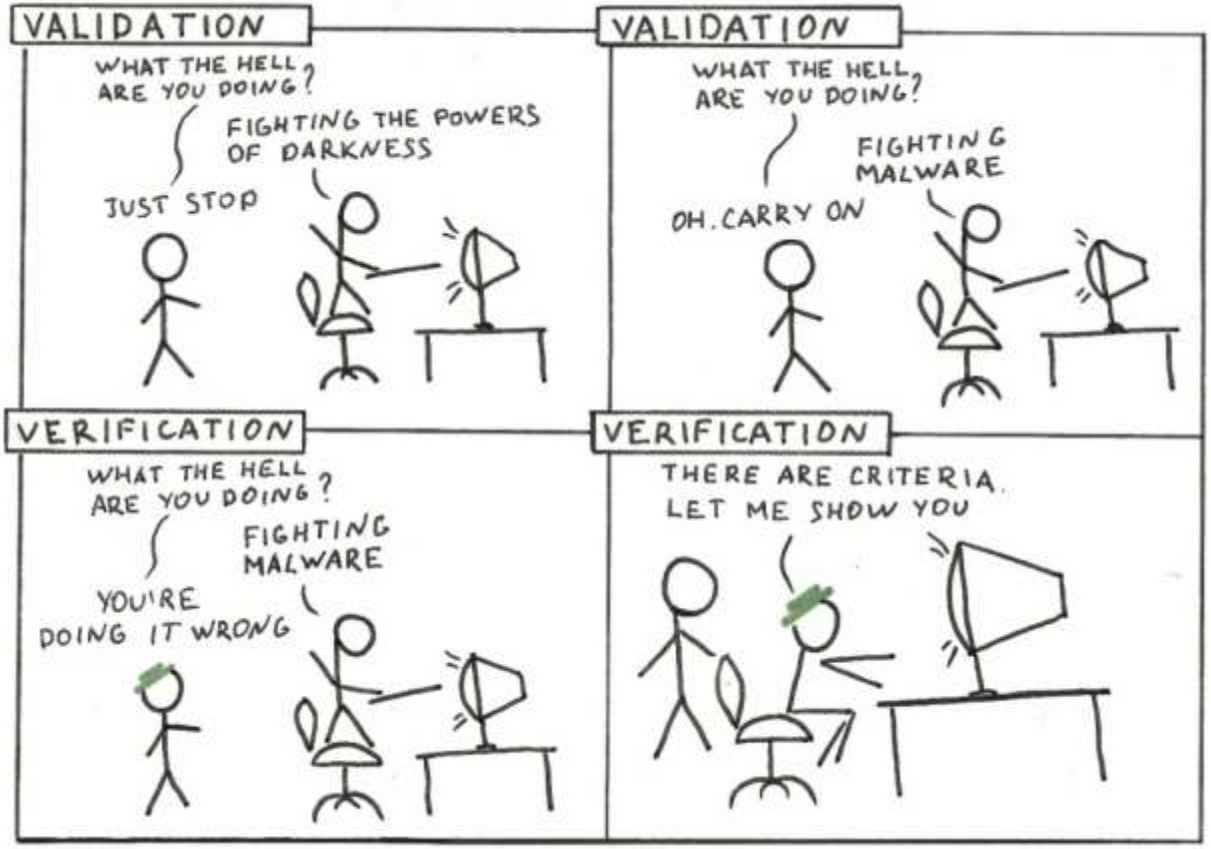
- Conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all activities during each life cycle process
- Satisfy standards, practices, and conventions during life cycle processes
- Successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities

Builds the product correctly

Validation: The process of providing evidence that the system and its products

- Satisfy requirements at the end of each life cycle activity
- Solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions)
- Satisfy intended use and user needs in the operational environment.

Builds the correct product



* https://securelist.com/the-power-of-vv/72615/

Why V&V?

Successful V&V process result

- Capture early detection and correction of any anomalies
- Engage with management insight into system lifecycle process
- Conformance to program performance, schedule and budget
- Early performance assessment
- Objective evidence
- Improve product quality from acquisition to operations
- Improve development and maintenance process

Verification Analysis

- Process:

  Conformance of developing product according to the specification

- Requirement:

  Architecture, Design, Code, SRS(System Requirement Specification), SDD (System Design Document)

- Activities:

  Reviews, Inspections, communication, code review, walkthroughs

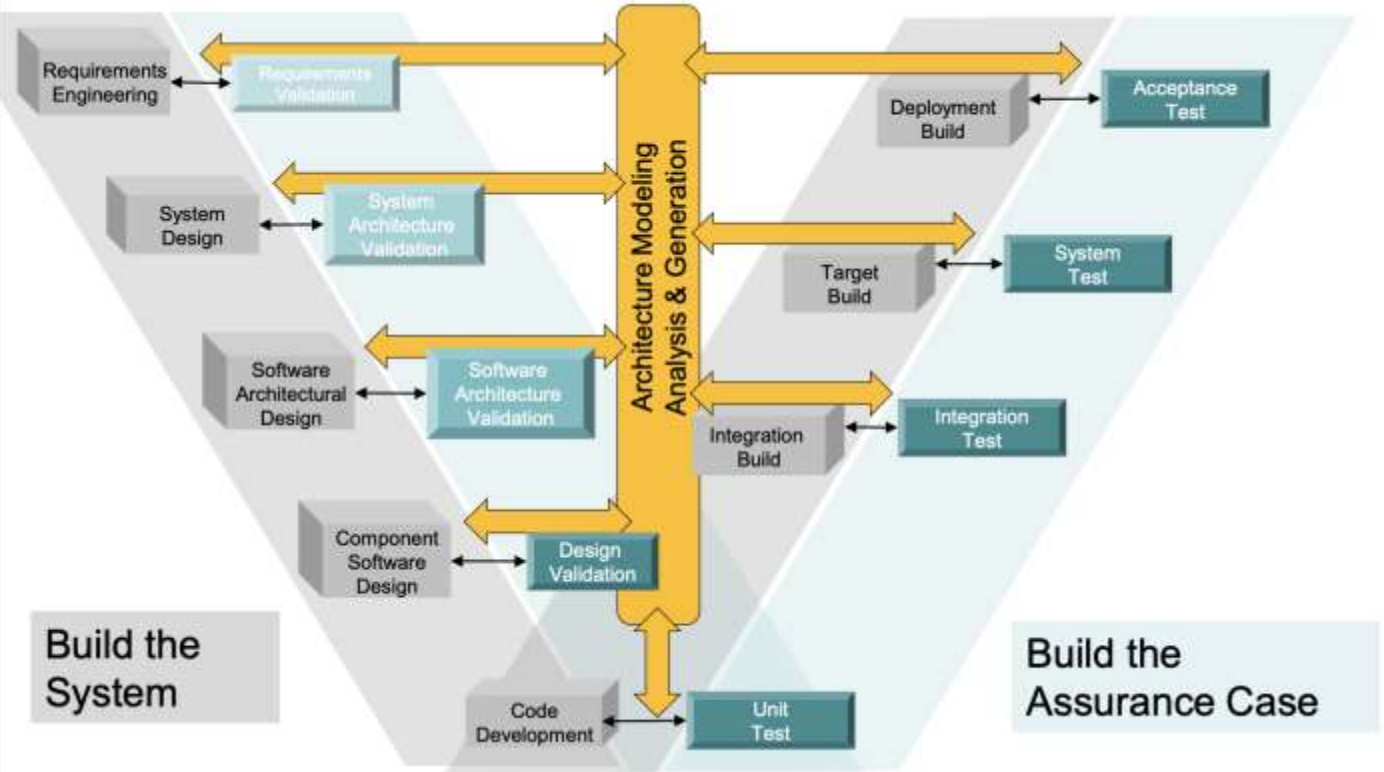
- Methods:

  Static Methods of checking documentations and code

Validation Analysis

- Process:
  - *Testing and validation of the developed product*
- Requirement:
  - *Actual product*
- Activities:
  - *Various level of testing, (unit, functional/ non-functional, acceptance) – Code execution*
- Methods:
  - *Dynamic process of testing the actual product*

Architecture-centric Validation & Verification



* http://fm.csl.sri.com/LAW/2010/law2010-slides-Lewis.pdf

V&V Activities - 1

- Concept Documentation Evaluation
- Requirements Allocation Analysis
- Requirements Evaluation
- Design Evaluation
- Interface Analysis
- Traceability Analysis
- Criticality Analysis
- Software Component Test and Design Plan V&V
- Software Integration Test and Design Plan V&V
- Hazard Analysis
- Security Analysis
- Software Qualification Test Plan V&V
- Software Acceptance Test Plan V&V

V&V Activities - 2

- Risk Analysis
- Source Code and Source Code Documentation Evaluation
- Software Integration Test Execution V&V
- Software Qualification Test Execution V&V
- Installation Configuration Audit
- Installation Checkout
- Evaluation of New Constraints
- Operating Procedures Evaluation
- VVP Revision
- Anomaly Evaluation
- Migration Assessment
- Retirement Assessment
- Software Disposal Evaluation

Main Activity - Hazard Analysis

- Analyze the potential hazards to and from the conceptual system.
- Identify the potential system hazards.
- Assess the consequences of each hazard.
- Assess the probability of each hazard.
- Identify mitigation strategies for each hazard.

Main Activity - Security Analysis

- Review the system owner's definition
- Analyze the system concept from a security perspective
- Identify potential security risks with respect to CIA triad.
- Include an assessment of the sensitivity of the information/data to be processed.
- Analyze self introduced the security risks

Main Activity - Risk Analysis

- Review and update risk analysis using prior task reports.
  - Previous test results
  - Identify new risks
  - Hazard and Security uses cases

- Provide recommendations to eliminate, reduce, or mitigate the risks.
  - Assess and evaluate hazard driven security analysis
  - Integrate back to early lifecycle

Continuous Verification and Validation of Critical Software

Current SW Development Process

DevOps / Agile

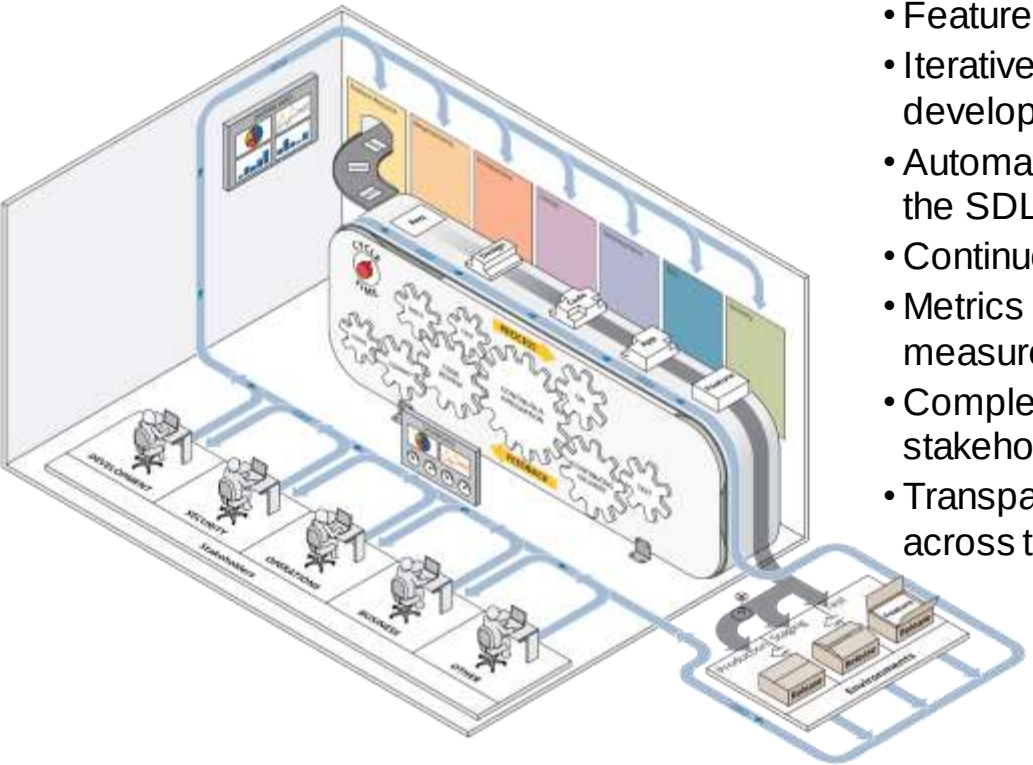
DevOps is a set of principles and practices emphasizing collaboration and communication between software development teams and IT operations staff along with acquirers, suppliers, and other stakeholders in the lifecycle of a software system[1]

Four Fundamental Principles

1. *Collaboration:* between all stakeholders
2. *Infrastructure as code (IaC):* assets are versioned, scripted, and shared
3. *Automation*: deployment, testing, provisioning, any manual or human-error-prone process
4. *Monitoring*: any metric in development or operation that can inform priorities, direction, and policy

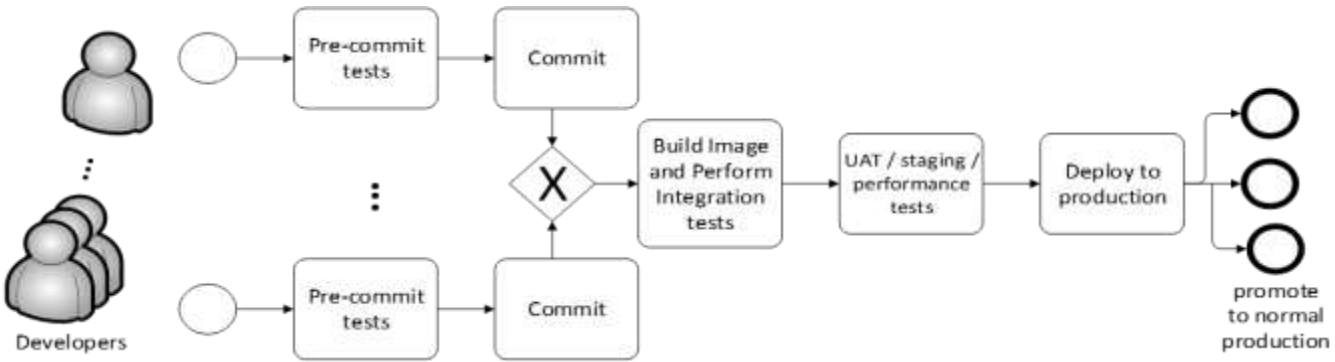
[1] IEEE 2675 DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package and Deployment

DevSecOps Software Factory Concept



- Feature to deployment
- Iterative and incremental development
- Automation in every phase of the SDLC
- Continuous feedback
- Metrics and measurement
- Complete engagement with all stakeholders
- Transparency and traceability across the lifecycle

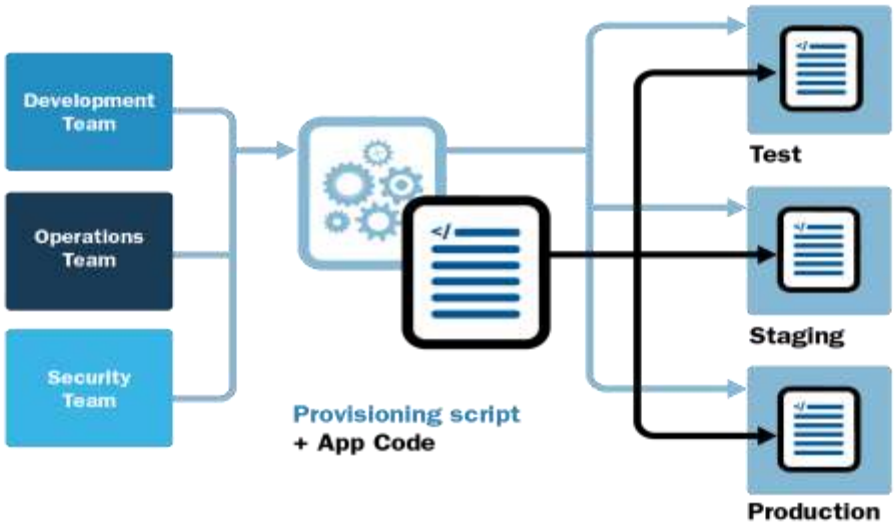
Multiple Environments in SDLC



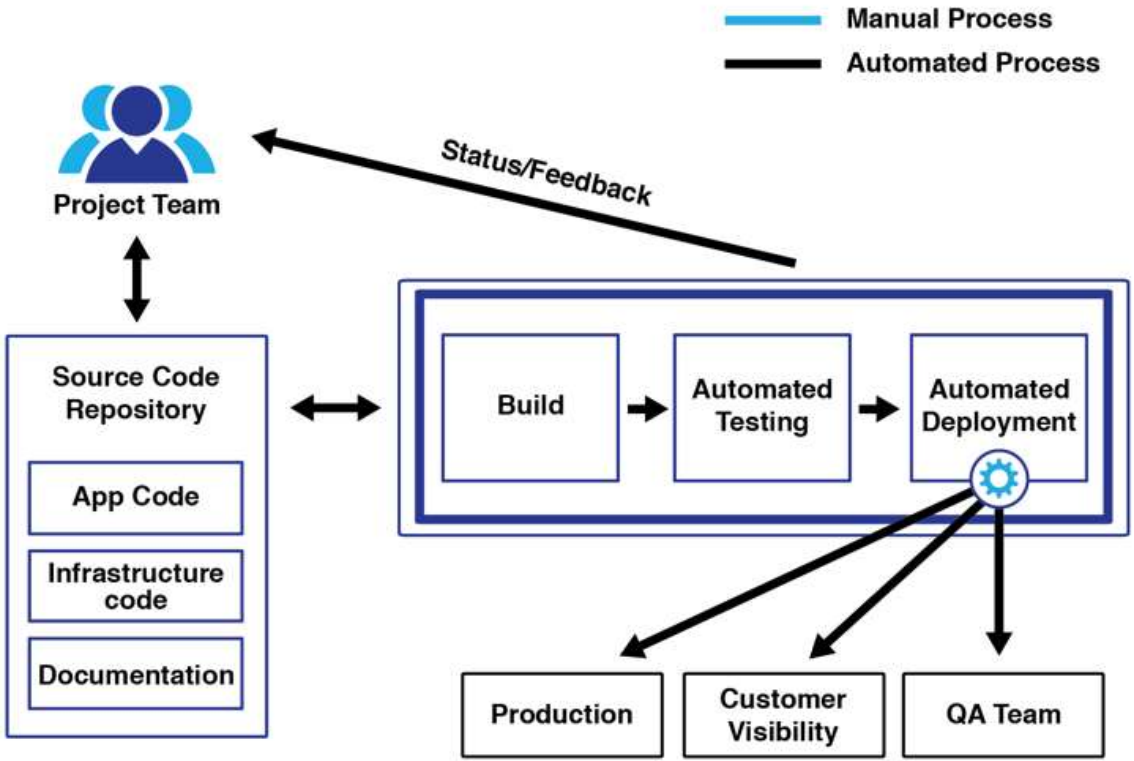
- Development environment
- Integration environment
- Staging environment
- Production environment

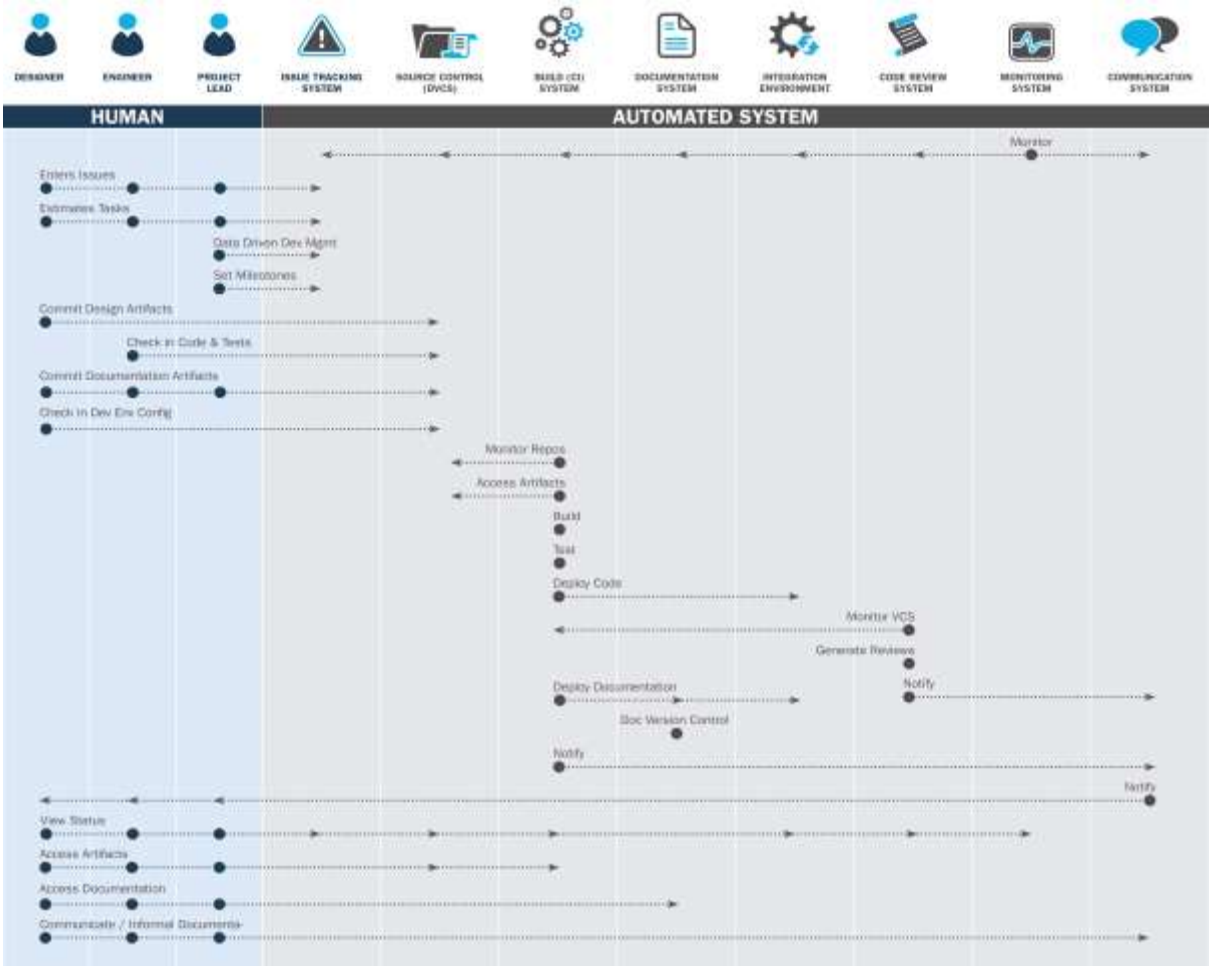
Infrastructure as Code

A program that creates infrastructure



Continuous Integration (CI) Model

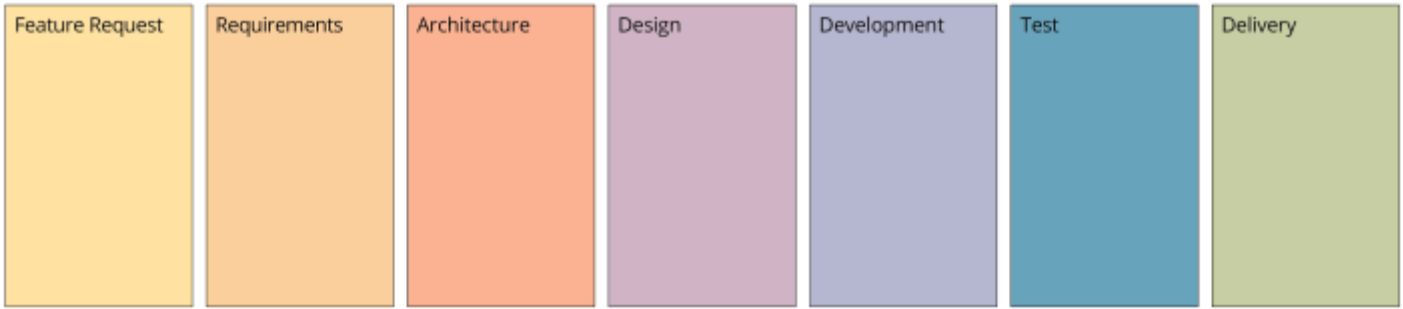


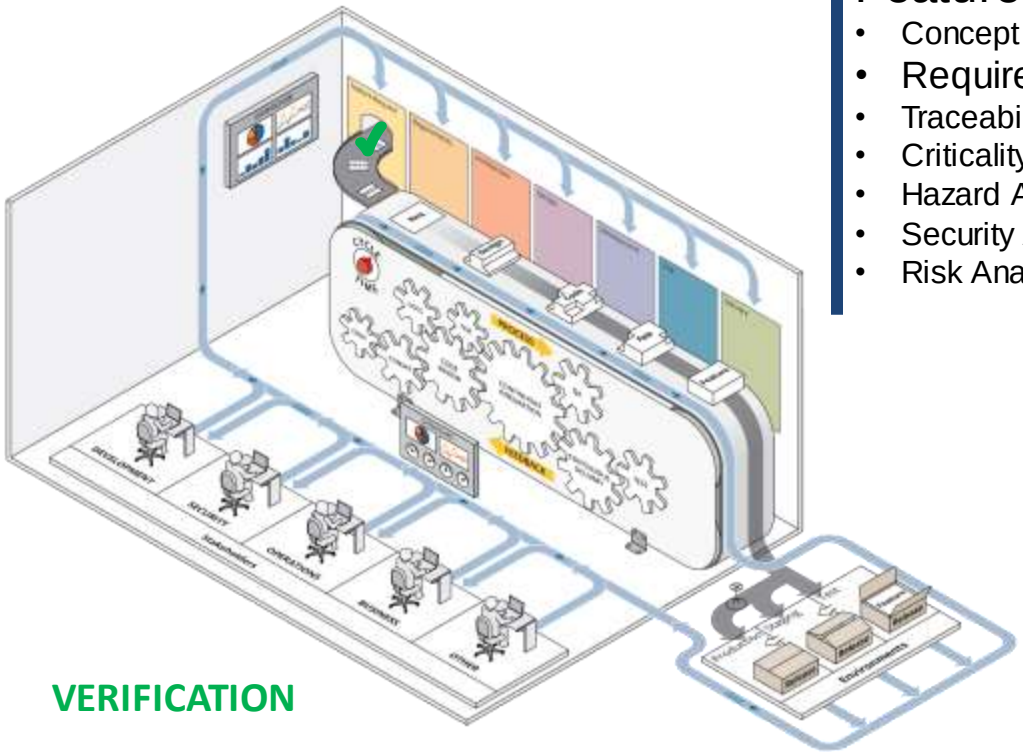


Continuous Verification and Validation of Critical Software

V&V Activities Across DevSecOps

Modern Software Development Phases

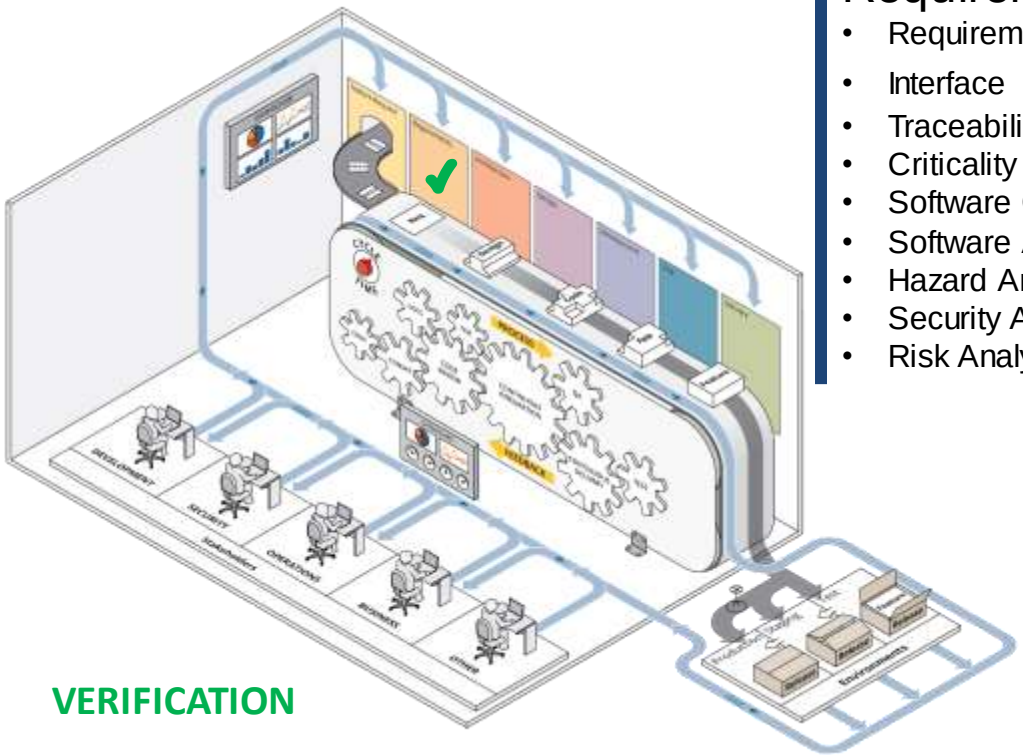




VERIFICATION

Feature Request/Concept

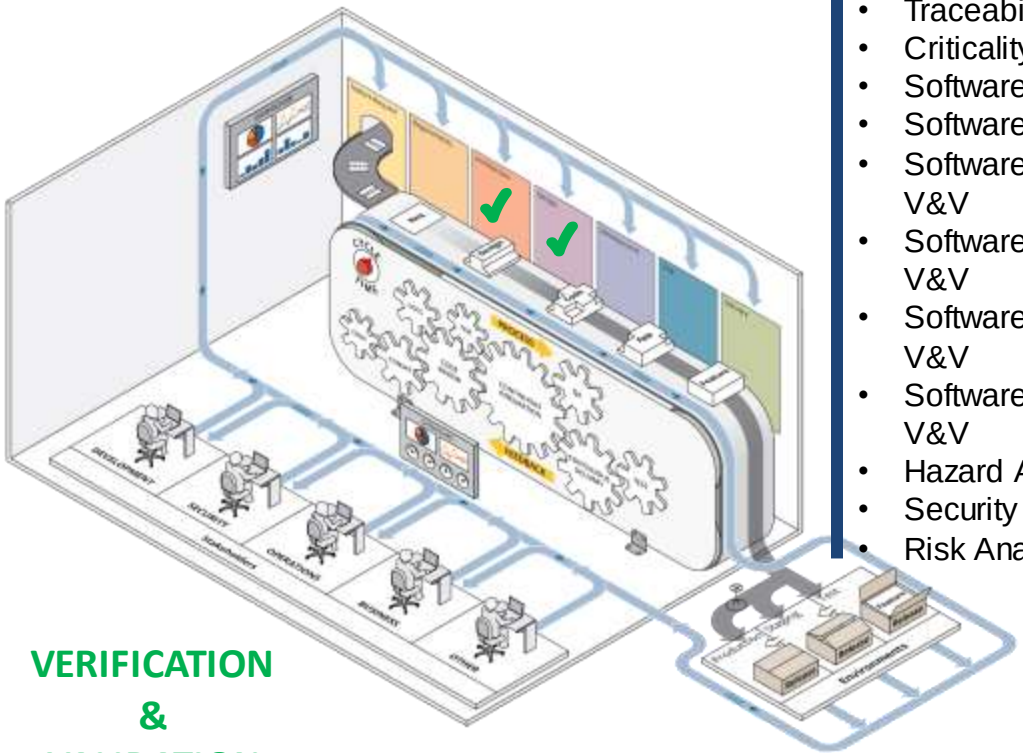
- Concept Documentation Evaluation
- Requirements Allocation Analysis
- Traceability Analysis
- Criticality Analysis
- Hazard Analysis
- Security Analysis
- Risk Analysis



VERIFICATION

Requirements

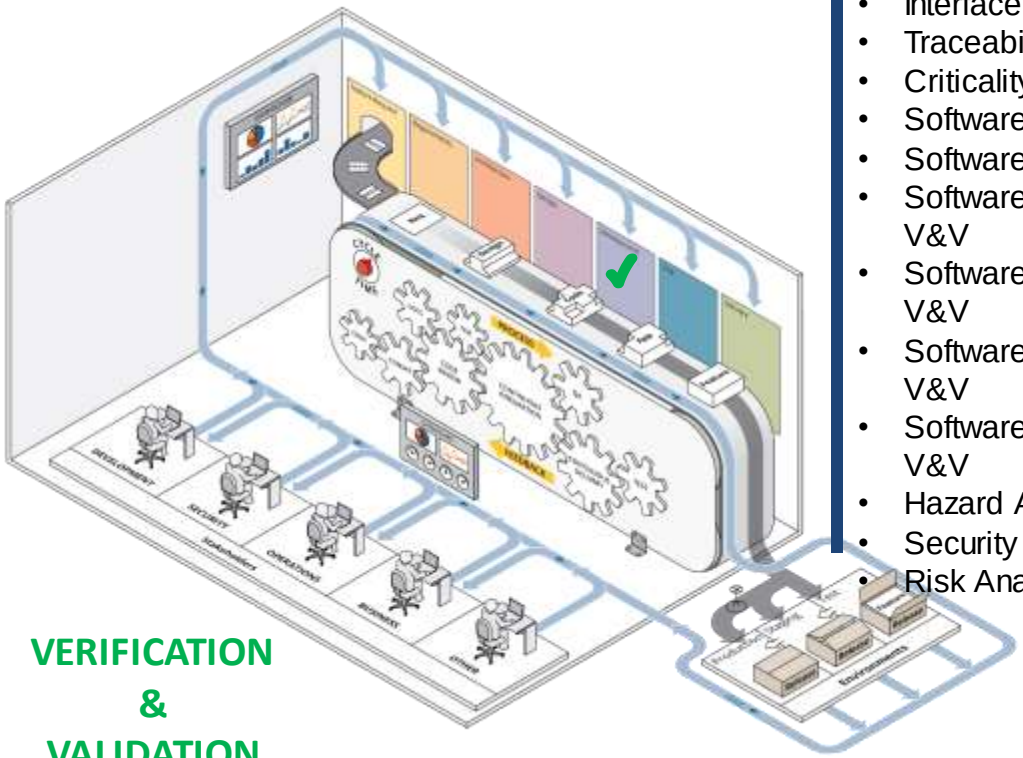
- Requirements Evaluation
- Interface Analysis
- Traceability Analysis
- Criticality Analysis
- Software Qualification Test Plan V&V
- Software Acceptance Test Plan V&V
- Hazard Analysis
- Security Analysis
- Risk Analysis



VERIFICATION & VALIDATION

Architecture & Design

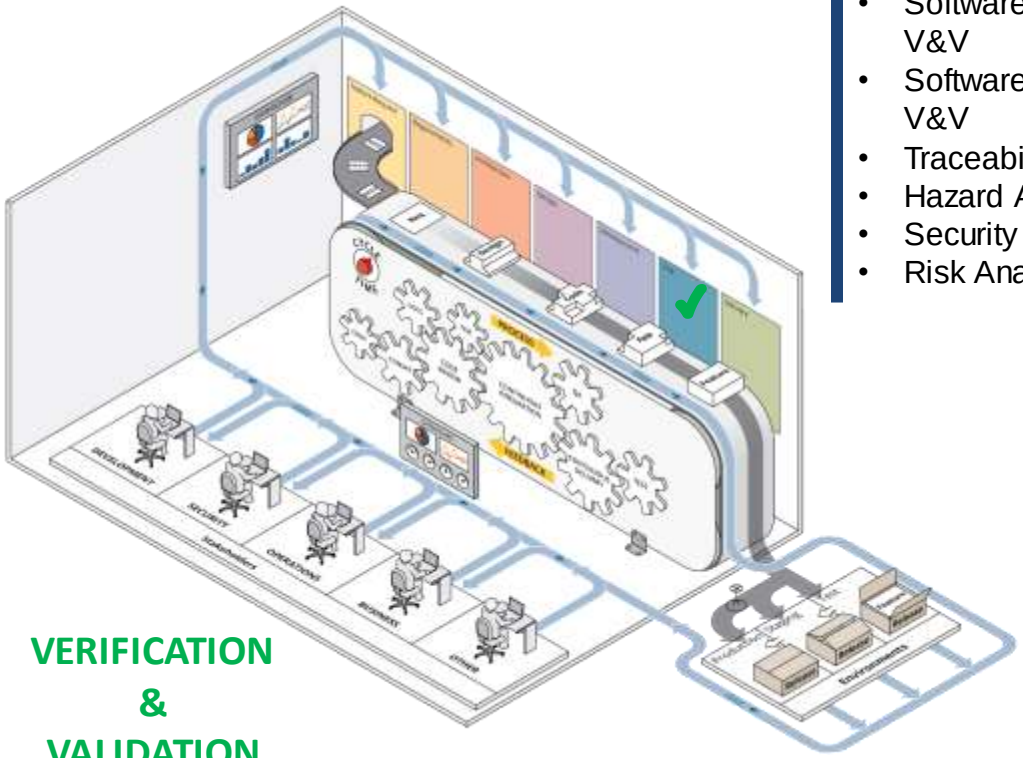
- Design Evaluation
- Interface Analysis
- Traceability Analysis
- Criticality Analysis
- Software Component Test Plan V&V
- Software Integration Test Plan V&V
- Software Component Test Design V&V
- Software Integration Test Design V&V
- Software Qualification Test Design V&V
- Software Acceptance Test Design V&V
- Hazard Analysis
- Security Analysis
- Risk Analysis



VERIFICATION & VALIDATION

Development

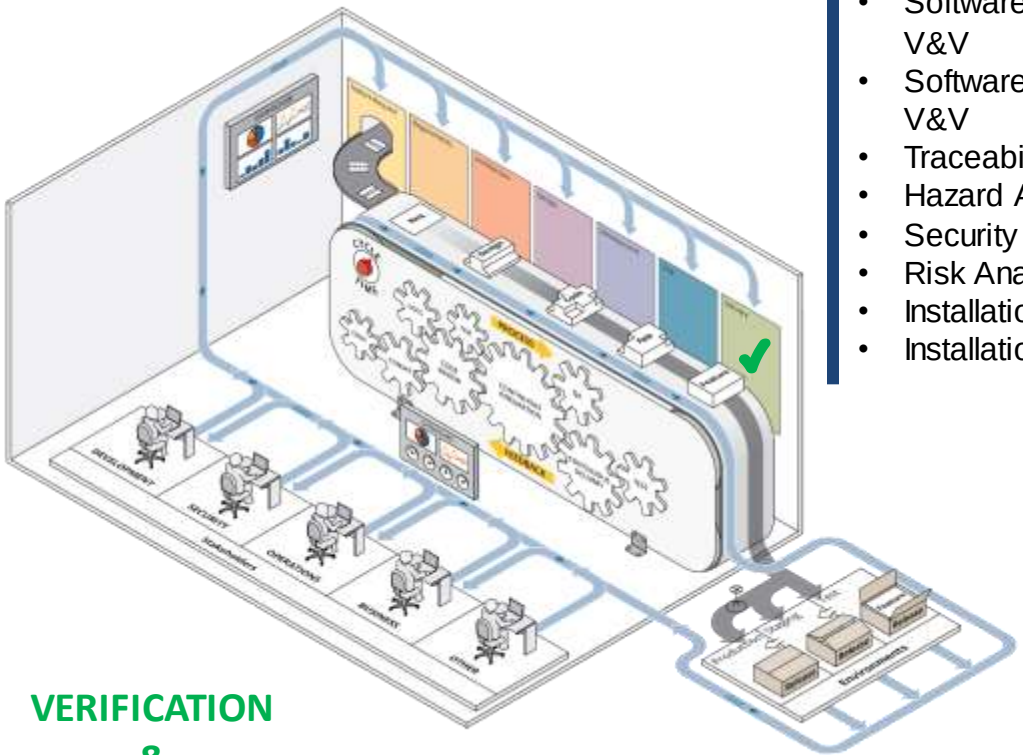
- Source Code and Source Code Documentation Evaluation
- Interface Analysis
- Traceability Analysis
- Criticality Analysis
- Software Component Test Plan V&V
- Software Integration Test Plan V&V
- Software Component Test Design V&V
- Software Integration Test Design V&V
- Software Qualification Test Design V&V
- Software Acceptance Test Design V&V
- Hazard Analysis
- Security Analysis
- Risk Analysis



VERIFICATION & VALIDATION

Testing/Integration

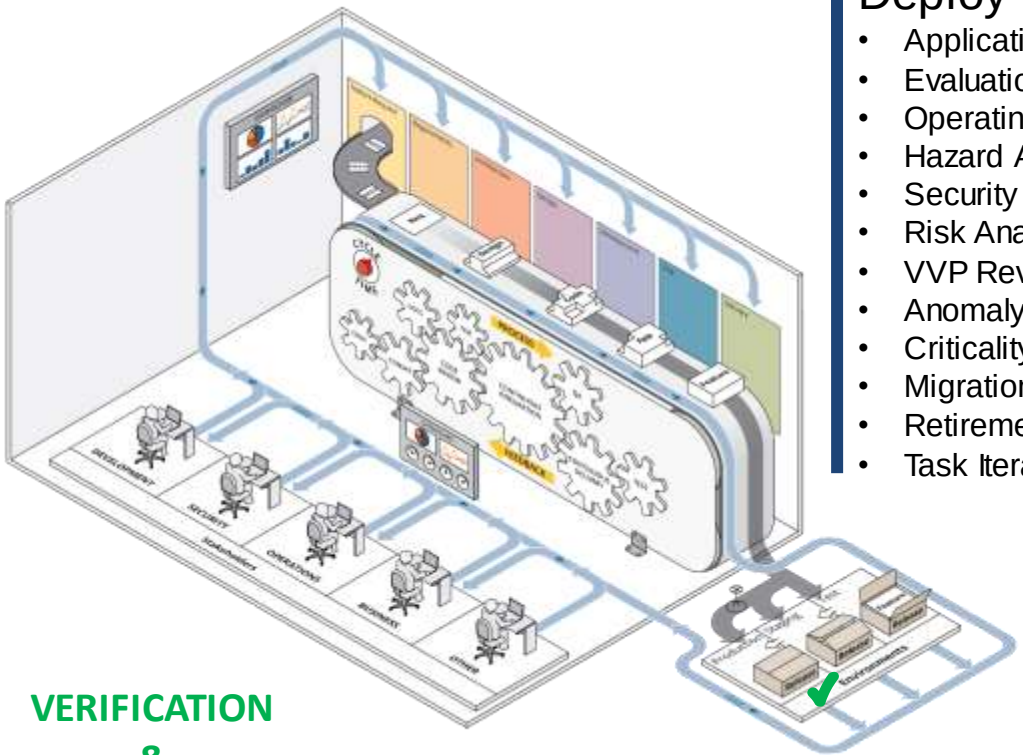
- Software Qualification Test Execution V&V
- Software Acceptance Test Design V&V
- Software Integration Test Execution V&V
- Traceability Analysis
- Hazard Analysis
- Security Analysis
- Risk Analysis



VERIFICATION & VALIDATION

Delivery

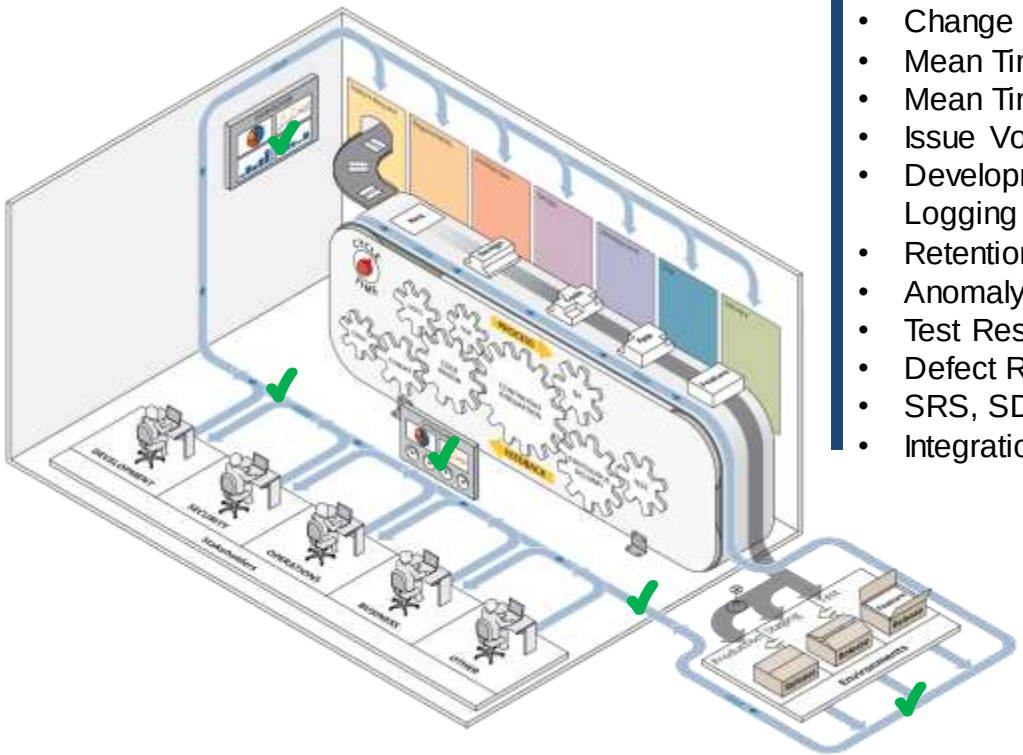
- Software Acceptance Test Procedure V&V
- Software Acceptance Test Execution V&V
- Traceability Analysis
- Hazard Analysis
- Security Analysis
- Risk Analysis
- Installation Configuration Audit
- Installation Checkout



Deploy

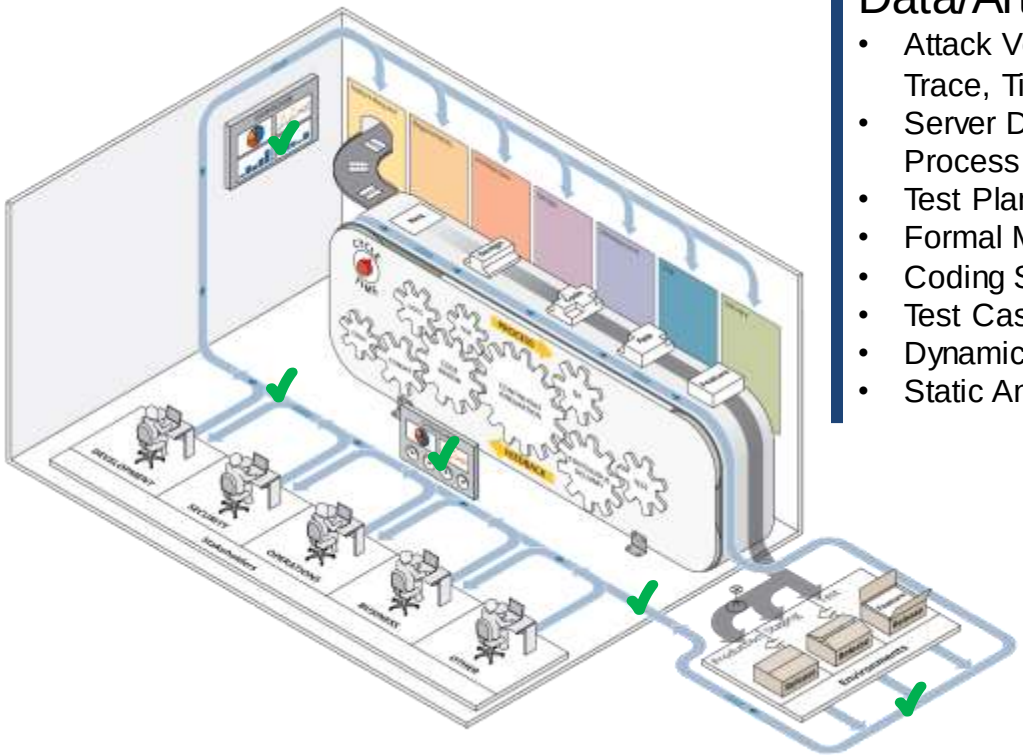
- Application Security Monitoring
- Evaluation of New Constraints
- Operating Procedures Evaluation
- Hazard Analysis
- Security Analysis
- Risk Analysis
- VVP Revision
- Anomaly Evaluation
- Criticality Analysis
- Migration Assessment
- Retirement Assessment
- Task Iteration

VERIFICATION & VALIDATION



Data/Artifact

- Change Failure Rate
- Mean Time To Recovery (MTTR)
- Mean Time to Detection (MTTD)
- Issue Volume and Resolution Time
- Development and Application Logging Availability
- Retention Control Compliance
- Anomaly reports
- Test Results
- Defect Rate
- SRS, SDD
- Integration results

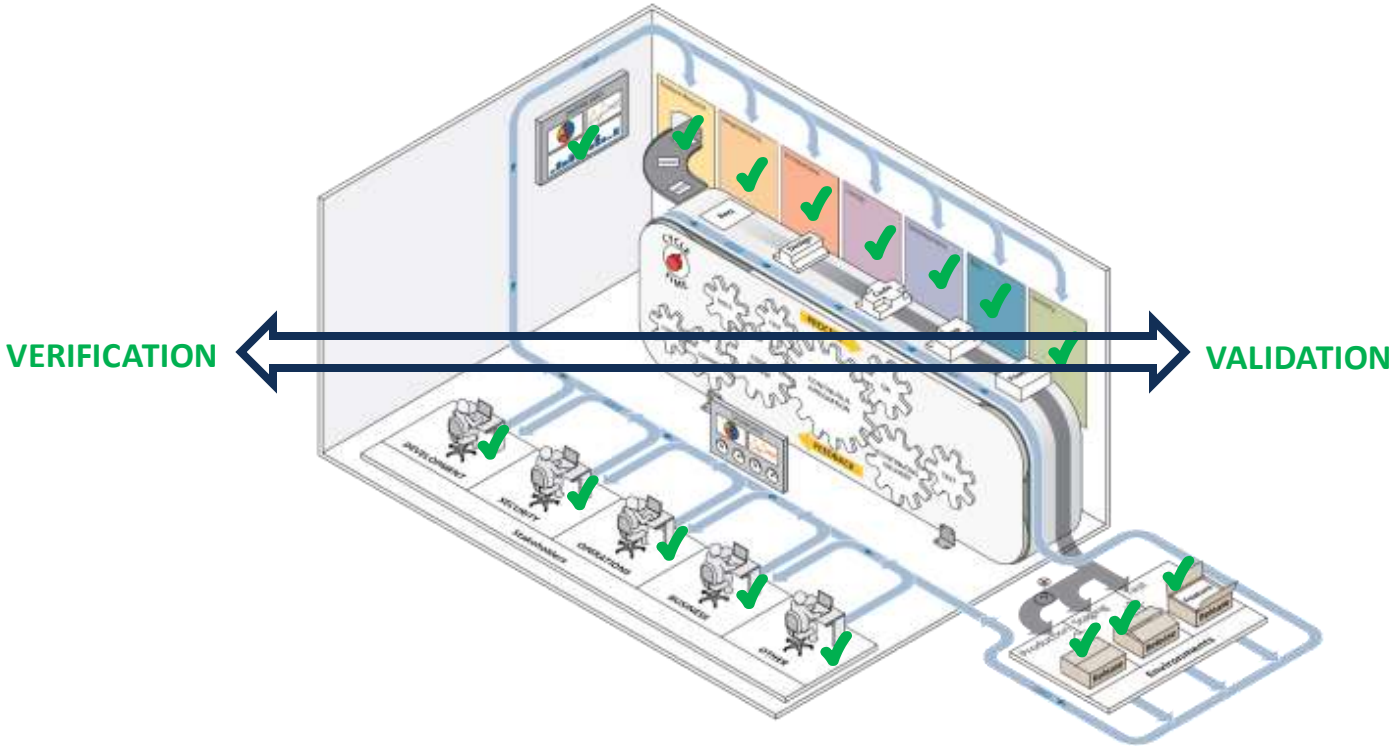


Data/Artifact

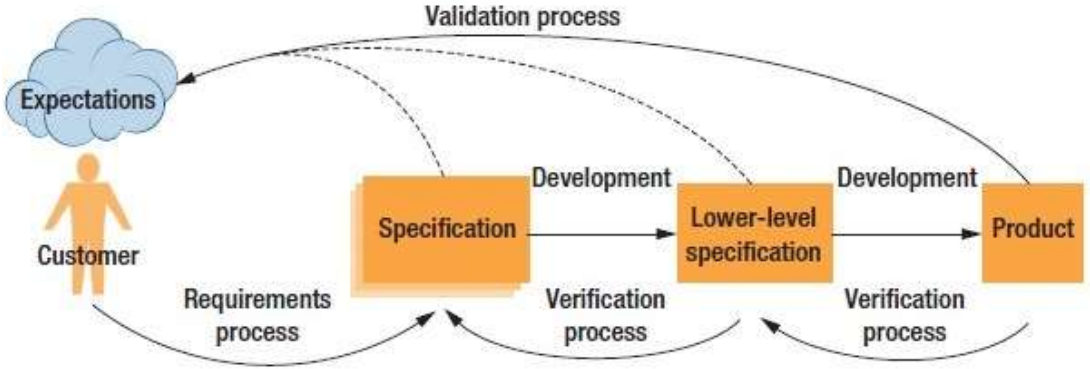
- Attack Vector Details (IP, Stack Trace, Time, Rate of Attack, etc)
- Server Disk Space, Load and Process Monitoring
- Test Plan
- Formal Methods
- Coding Standards
- Test Cases
- Dynamic Analysis
- Static Analysis

VERIFY & VALIDATE

Continuous V&V on every phases of lifecycle



V&V workflow



*https://www.infoq.com/articles/ieee-verification-and-validation-for-software-systems

For more information…

DevOps: https://www.sei.cmu.edu/go/devops

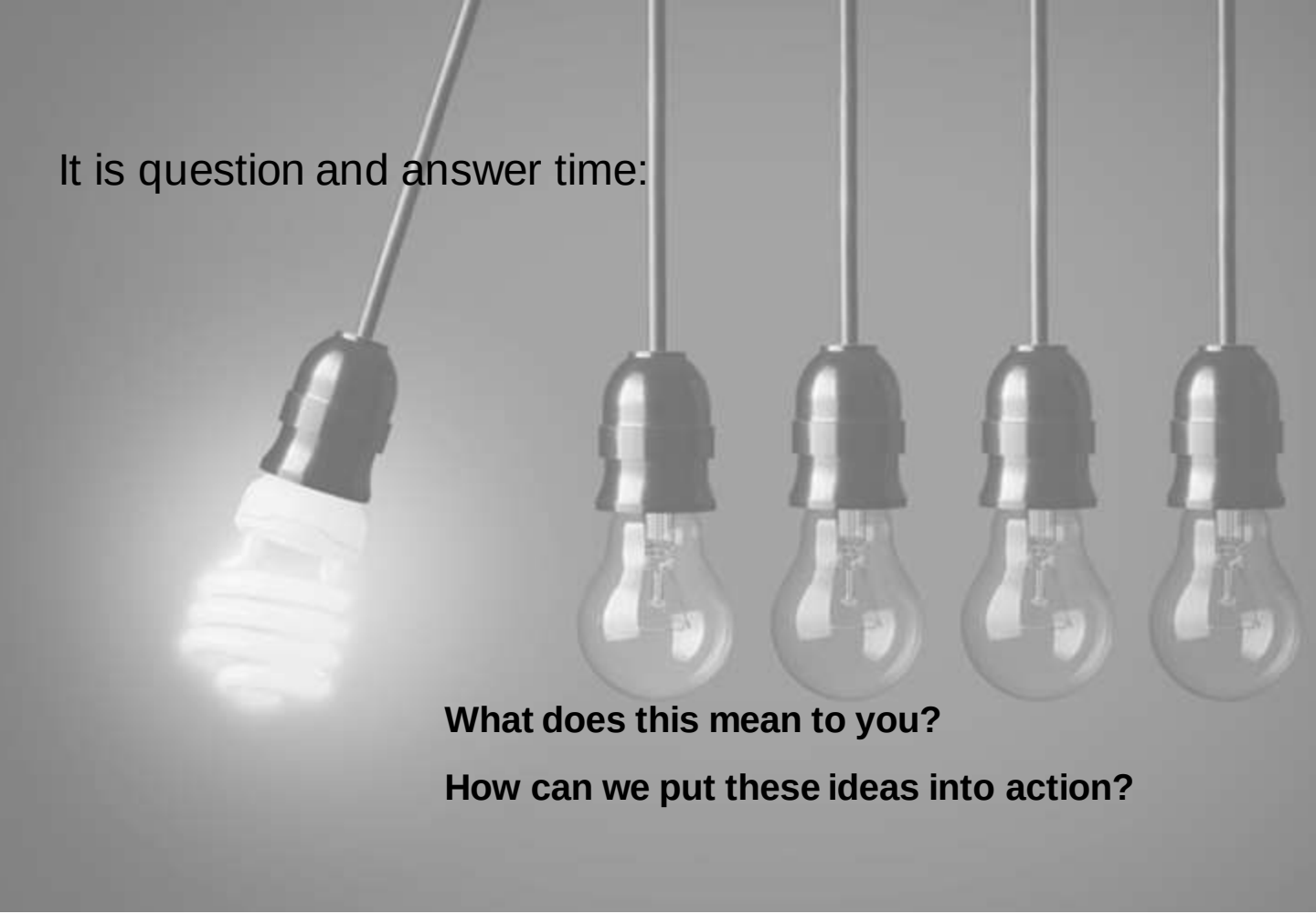
DevOps Blog: https://insights.sei.cmu.edu/devops

Webinar : https://www.sei.cmu.edu/publications/webinars/index.cfm

Podcast : https://www.sei.cmu.edu/publications/podcasts/index.cfm

Thank You

Hasan Yasar
Technical Director, Adjunct Faculty Member
Continuous Deployment of Capability
hyasar@sei.cmu.edu
@securelifecycle



It is question and answer time:

What does this mean to you?

How can we put these ideas into action?