Fall 2014
SEI Research Review
FY14-03 Software Assurance Engineering

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carol Woody, Ph.D.
October 28, 2014

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
18 OCT 2014	**N/A**	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Software Assurance Engineering	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Woody /Carol	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**18**	

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0001767

Software Assurance Engineering

Mission success for software-reliant systems requires software assurance

> Software assurance: implementing software with a "level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the life cycle…" **Section 933 of National Defense Appropriation Act 2013**

Engineering software assurance into the acquisition and development life cycle

- Task 1: Analyzing Security Risk Early in the Software Life Cycle
- Task 2: Applying Software Quality Models to Software Assurance

Task 1 Analyzing Security Risk Early in the Software Life Cycle

Three main causes of operational security vulnerabilities:

- Design weaknesses
- Implementation/coding vulnerabilities
- System configuration errors

Design weaknesses are not easily addressed during operations.

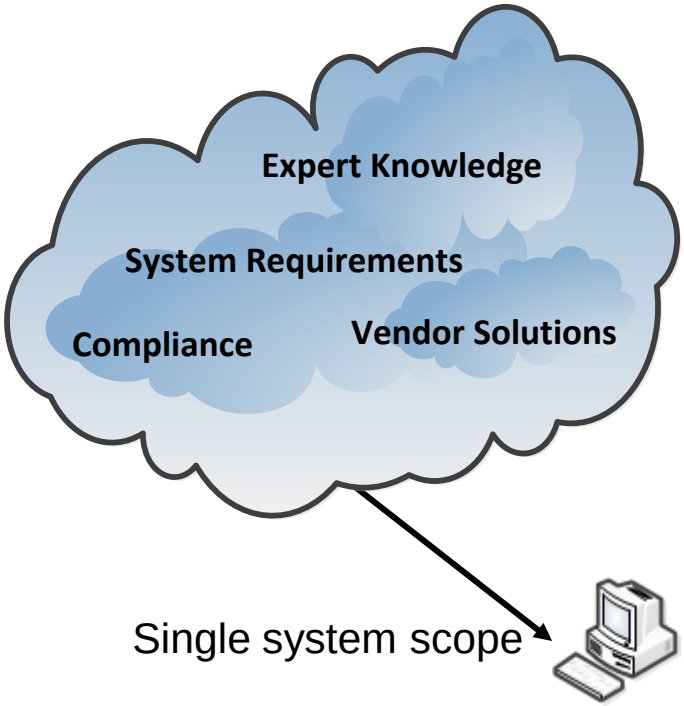
- 379 of the 940 common weakness enumerations (CWEs) are design weaknesses (http://cwe.mitre.org/)
- 19 of the top 25 are linked to design weaknesses(http://cwe.mitre.org/top25/)

Causes for design weaknesses:

- Poor security requirements
- Limited understanding of the impact of security risk on mission success

Goal: Reduce Security Design Weaknesses With Improved Requirements

Task 1: *Limitations in Current Security Design Practices*



Identification techniques are ad hoc

- Notation for expressing a security event/risk is incomplete
- Approaches rely on analysts' tacit knowledge of operational context
- Security controls address compliance not risk

Risk analysis (if done) is focused on a single system

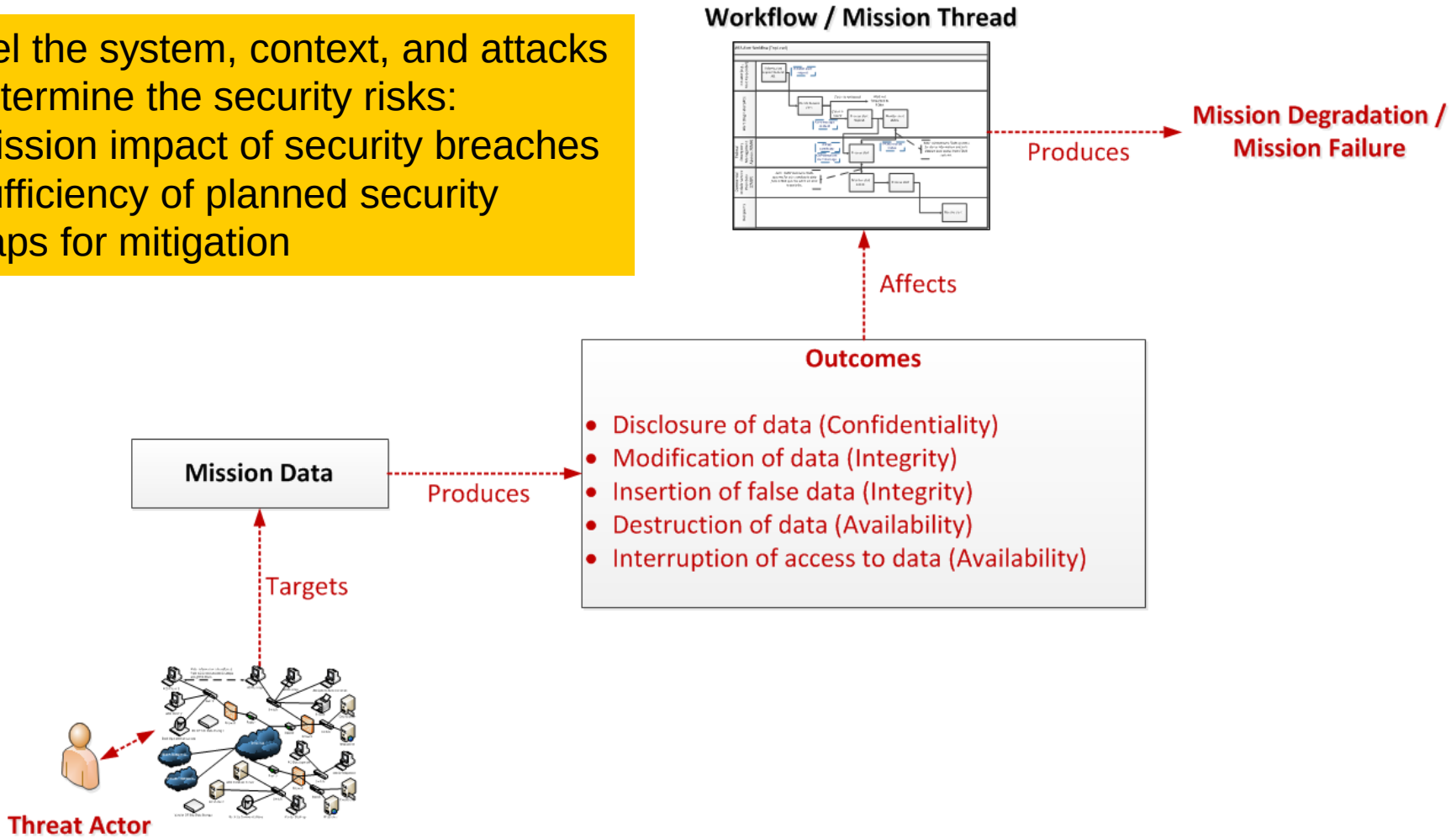
- Standalone (i.e., single system) models have been developed
- Risk analysis considers the exploit of an individual vulnerability within a single system

- Security experts need to communicate risk effectively with system and software engineers and acquisition experts
- Attacks frequently come from other trusted systems
- Complex attacks need to be included in security risk evaluation

Task 1: *Security Engineering Risk Analysis (SERA) Framework*

Model the system, context, and attacks to determine the security risks:

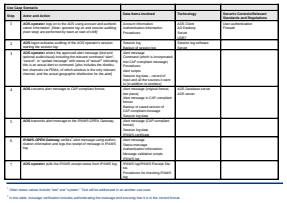
- mission impact of security breaches
- sufficiency of planned security
- gaps for mitigation



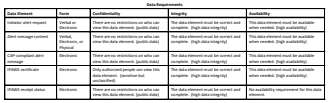
Task 1: *SERA Approach*

Threat Identification Models

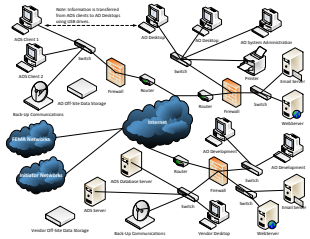
Use-Case View



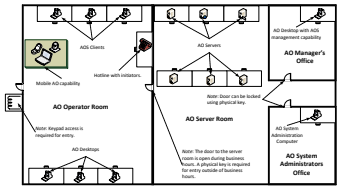
Data View



Network View

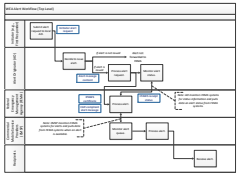


Physical View

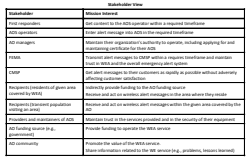


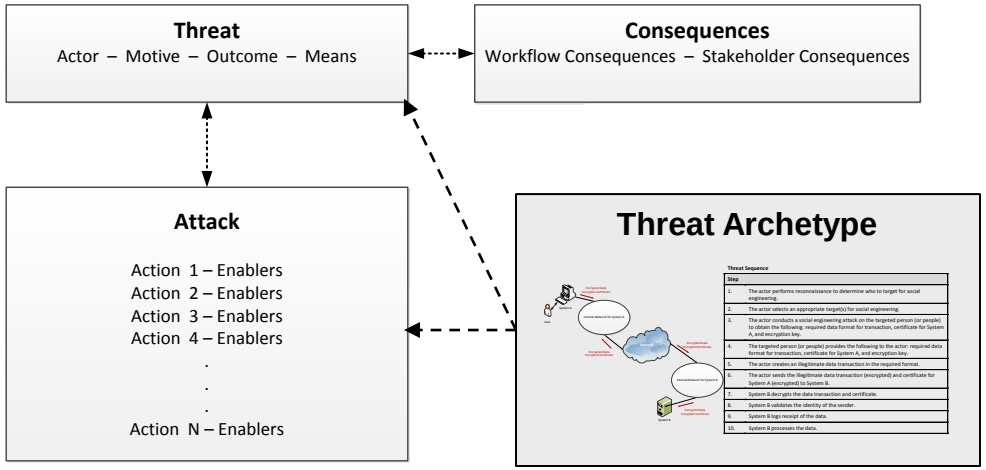
Consequence Analysis Models

Workflow View



Stakeholder View





Defined semantics for expressing a security event/risk

Library of threat archetypes to support threat identification

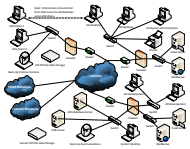
Models to support threat identification

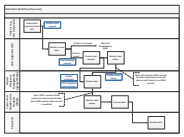
Models to support consequence analysis

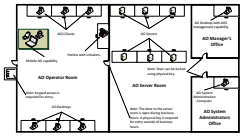
Task 1: *SERA 4-Step Process*

1. **Establish operational context.**
2. **Identify risk.**
3. **Analyze risk.**
4. **Develop control plan.**

Modeling Techniques







Risk Identification Worksheet

Risk Evaluation Criteria

Risk Analysis Worksheet

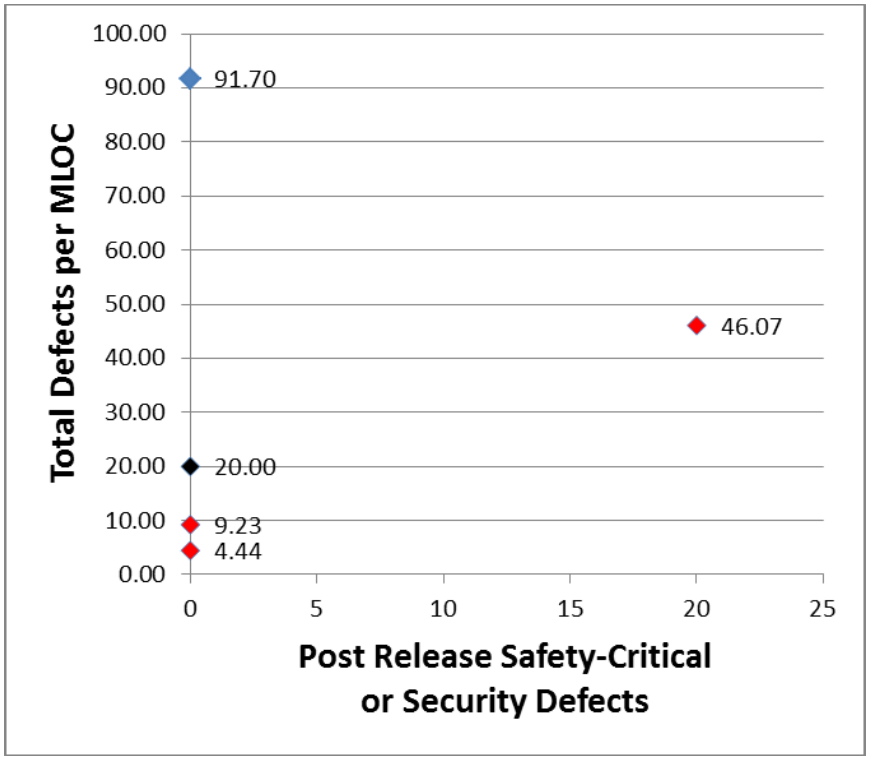
Control Approach Worksheet

Control Plan Worksheet

Task 2 Applying Software Quality Models to Software Assurance

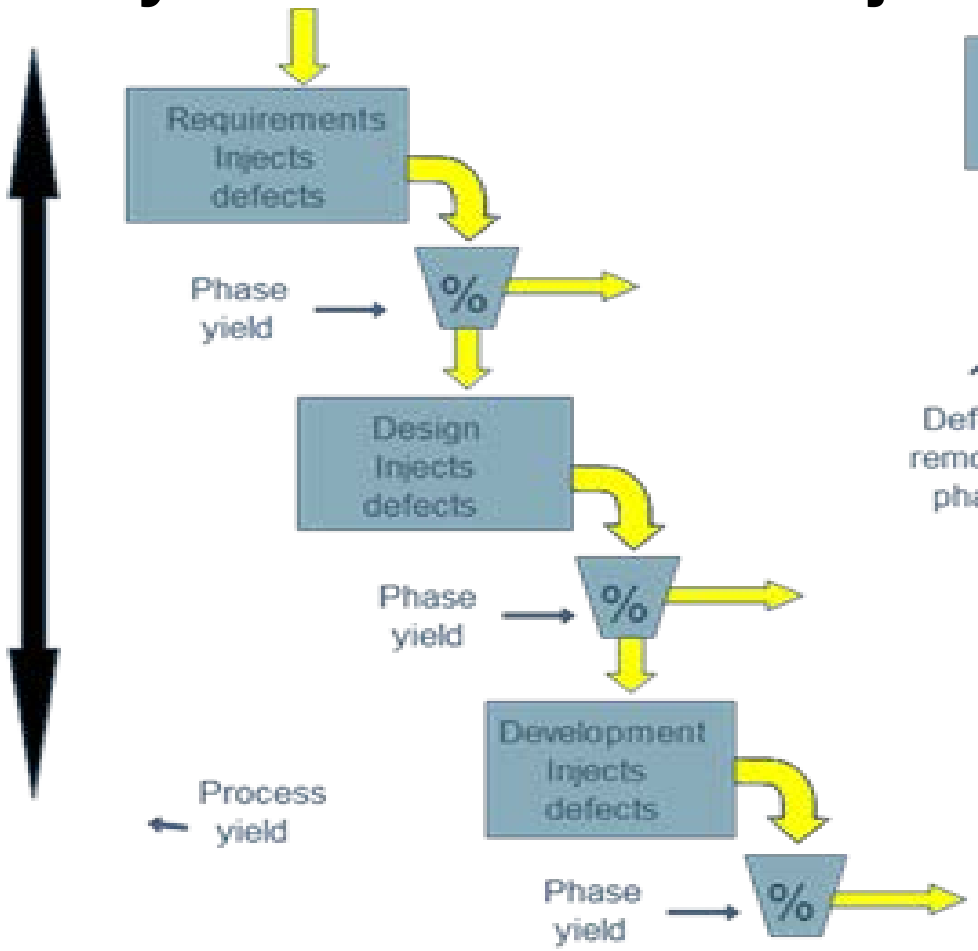
The SEI has quality data for over 100 Team Software Process (TSP) development projects used to predict operational quality.

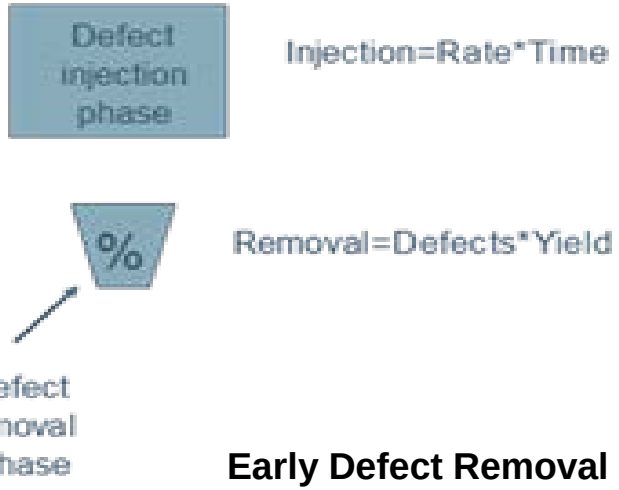
Data from five projects with low defect density in system testing reported very low or zero safety critical and security defects in production use.



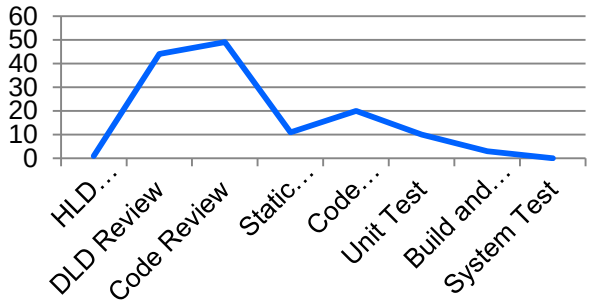
Goal: Use Predictions of Quality to Inform Security Risk Predictions

Quality Focus on Defect Injection and Removal





Early Defect Removal across Life Cycle



Vulnerabilities are defects

- 1-5% of quality defects are vulnerabilities (confirmed)
- 50-70% of security vulnerabilities are quality defects (unconfirmed)

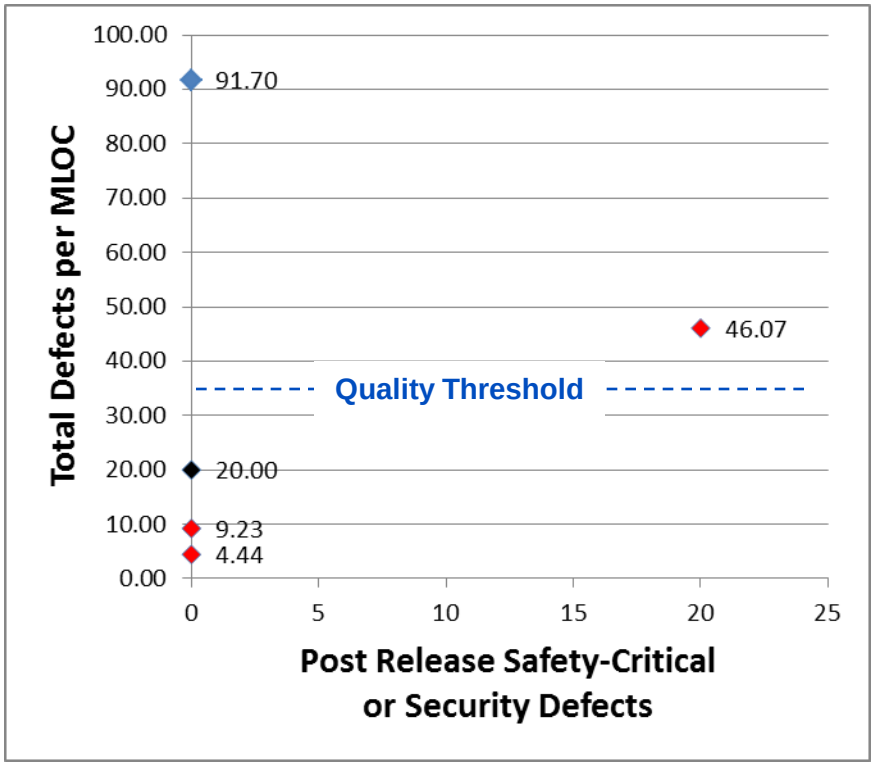
Task 2: Evidence Available for Quality Analysis

The data below describe successful results of systems in operation for at least a year from three organizations that are of interest to identify patterns appropriate for security.

Org.	Project	Type	Secure or Safety Critical Defects	Defect Density	Size
D	D1	Safety Critical	20	46.07	2.8 MLOC
D	D2	Safety Critical	0	4.44	.9 MLOC
D	D3	Safety Critical	0	9.23	1.3 MLOC
A	A1	Secure	0	91.70	.6 MLOC
T	T1	Secure	0	20.00	.1 MLOC

With one exception, projects implemented below 20 defects per MLOC had no reported operational security or safety-critical defects.

The exception utilized specialized defect removal practices for secure systems.



Task 2: *How Could Quality Help Security?*

Good quality will ensure proper implementation of specified results

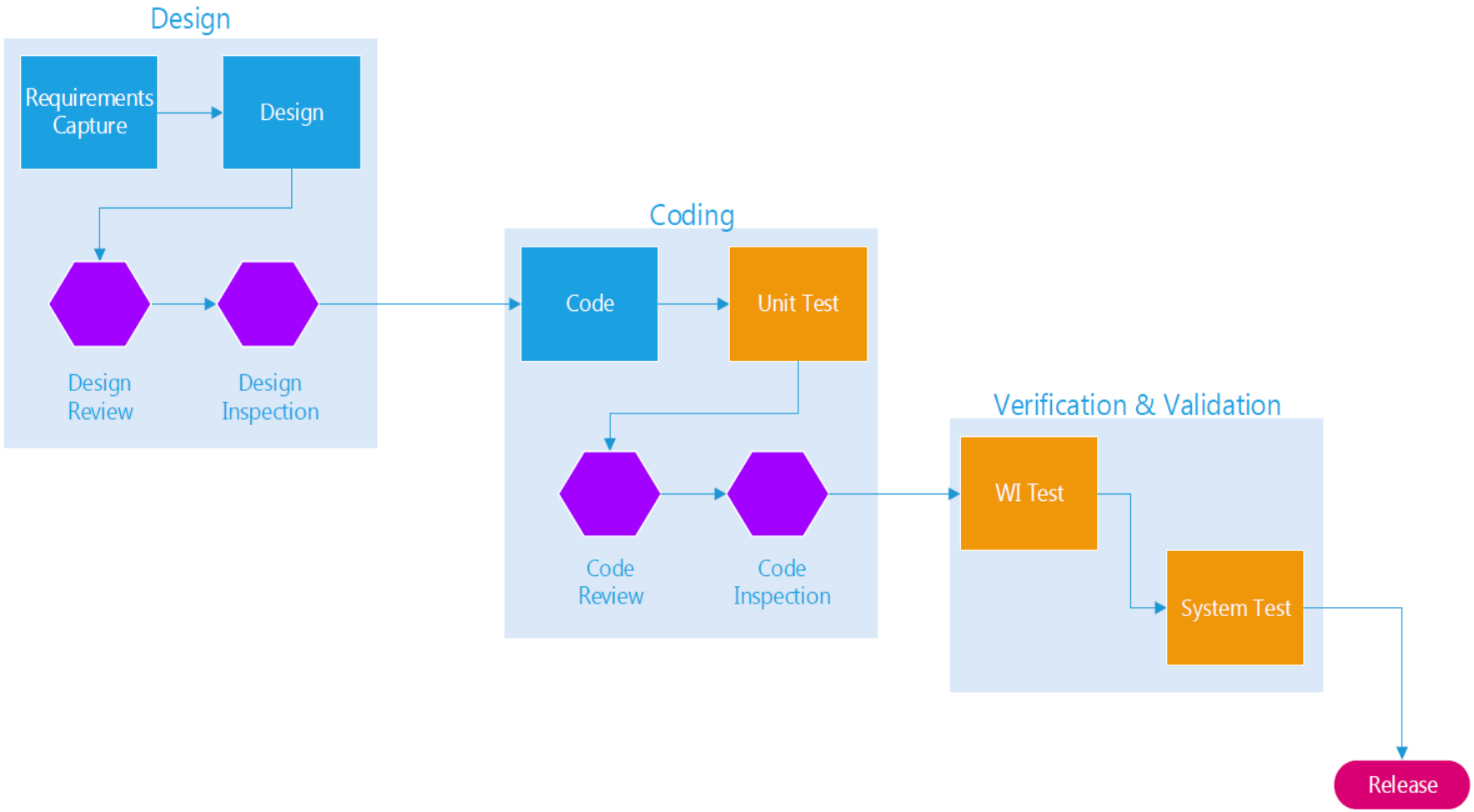
- Effective code checking will identify improper implementations of specifications (11 of SANS Top 25)

- Effective design reviews will identify missing requirements (12 of SANS Top 25)
  - **if** appropriate security results are considered in the development of requirements
  - **if** requirements are effectively translated into detail designs and code specifications to support the required security results

SANS Top 25: SysAdmin, Audit, Network, Security Top 25 Most Dangerous Programming Errors (http://cwe.mitre.org/top25)

Security Requirements Must be Properly Specified

Successful Projects Embed Quality and Safety/Security Inspection at Each Lifecycle Step



Successful Projects Use Metrics Extensively

Development Metrics

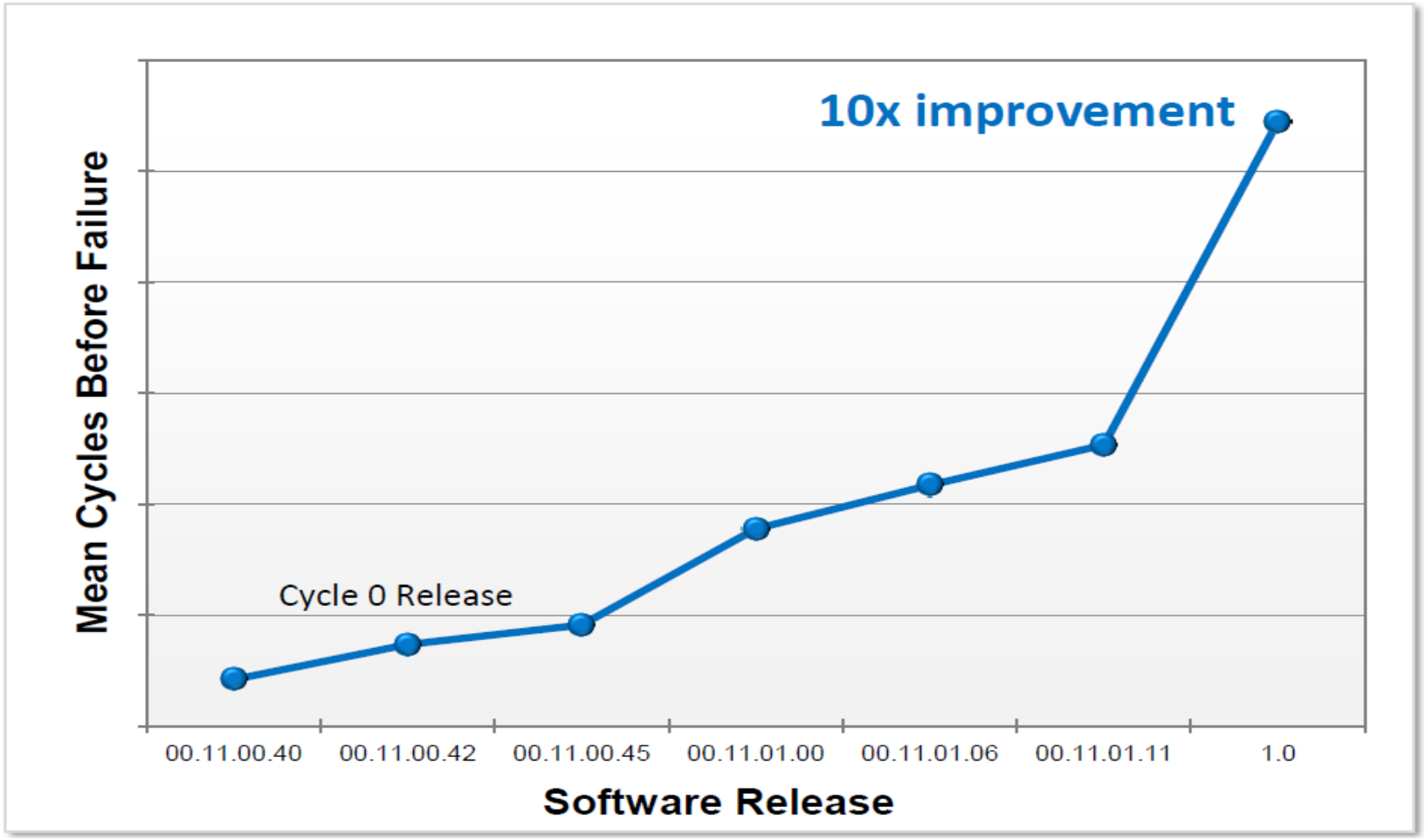
- Incoming/week
- Triage rate
- % closed
- Development work for cycle
- Software change request per developer per week
- # developers
- Software change request per verifier & validator per week
- # verification persons

Software Change Metrics

- Fixed work per cycle
- Deferred planned work per cycle

Testing and tools are used to confirm results NOT find problems

Successful Projects Show Improved Reliability



FY 14-03 Deliverables

Publications:

- CrossTalk September/October 2014 "Evaluating Security Risks Using Mission Threads" provides a summary of the SERA framework with an example
- SEI Special Report CMU/SEI-2013-SR-018 "Wireless Emergency Alerts (WEA) Cybersecurity Risk Management Strategy for Alert Originators" released March 2014 http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=70071
- Two technical notes (drafts are in review) to be released in November

Conferences:

- Presentation: "Modeling Early Life Cycle Security Risk" International Conference on Conceptual Modeling 2014 (ER2014) http://2014.erconference.org/
- Tutorial: "Security Risk Management using the Security Engineering Risk Analysis (SERA) Method" 2014 Annual Computer Security Applications Conference (ACSAC) https://www.acsac.org/2014/
- Proposed tutorial for Ground Systems Architecture Workshop (GSAW) 2015
- Abstract approved for IEEE International Symposium for Homeland Security 2015

FY14-03 Collaborations

Carnegie Mellon University funded collaboration with Travis Breaux, Assistant Professor of Computer Science, and his doctoral student Hunan Hibshi for academic review and two publications:

- H. Hibshi, T.D. Breaux, M. Riaz, L. Williams (2014) "Towards a Framework to Measure Security Expertise in Requirements Analysis," In Proc. IEEE 1st International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE), pp. 13-18.
- H. Hibshi, T.D. Breaux, M. Riaz, L. Williams (2014) "Discovering Decision-Making Patterns for Security Novices and Experts," In Submission: International Journal of Secure Software Engineering.

SERA Framework review by Ranjit Singh Mann, PE Software Engineering IPT Lead, Indirect Fire Protection Capability Increment 2-Intercept (IFPC Inc 2-I) Product Office (PdO)

Research Review Workshop, August 6, 2014 in Ballston with 16 participants from DoD, NSA, academia, and industry including the DoD Software Assurance Community of Practice (SwA COP) Metrics Team

Contact Information

Carol Woody, Ph.D.
Technical Manager
CERT/CSF/CSE
Telephone: +1 412-268-5800
Email: info@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257