Engineering System Security for a System of Systems Context

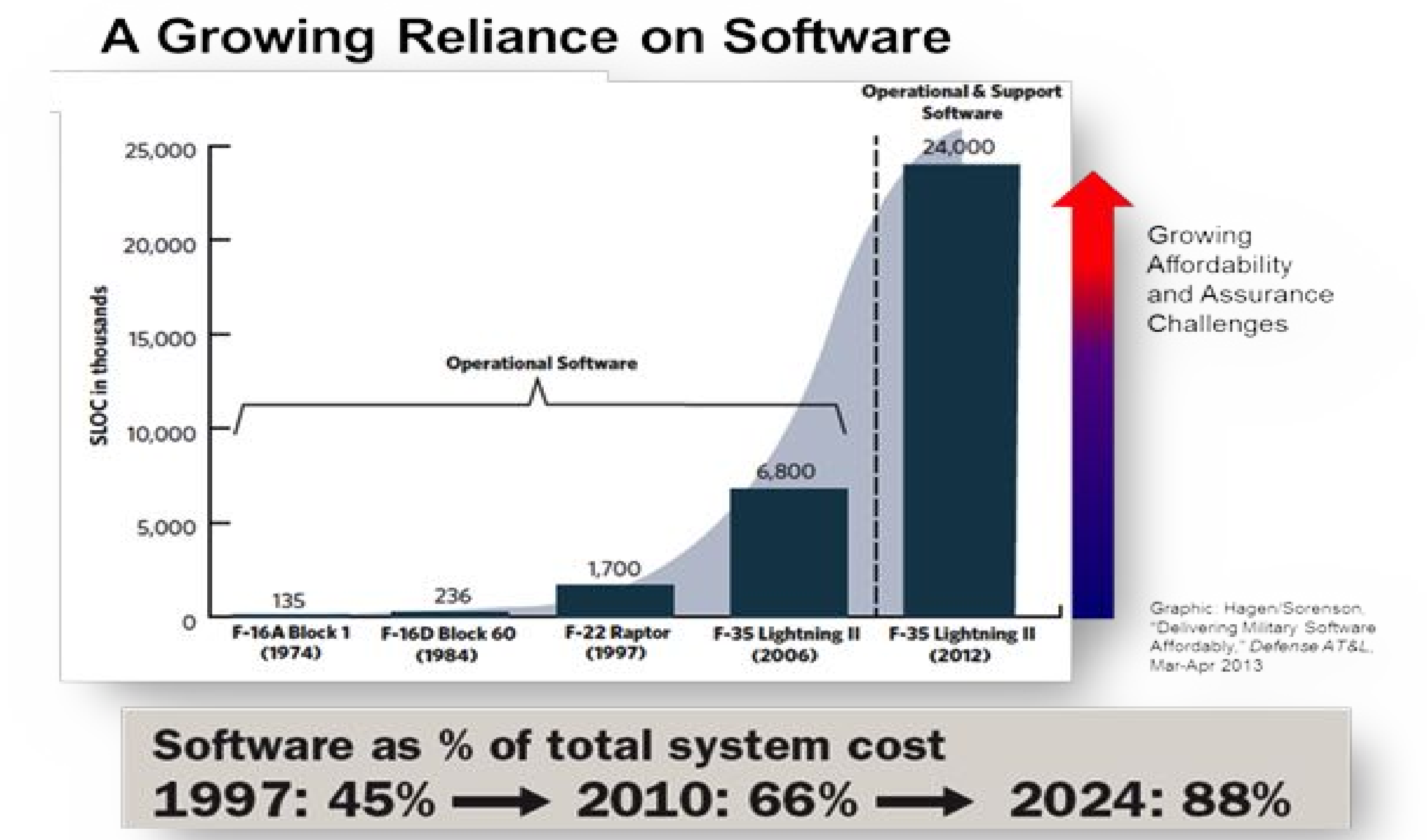
Carol Woody, Ph.D.
Technical Manager,
Cybersecurity Engineering

Software Engineering Institute (SEI)
Carnegie Mellon University (CMU)
Pittsburgh, PA 15213

DM-0002588

Security Challenges in a System of Systems Context

Software Reliance is Rapidly Expanding



Source: U.S. Air Force Scientific Advisory Board. *Sustaining Air Force Aging Aircraft into the 21st Century* (SAB-TR-11-01). U.S. Air Force, 2011.

Anyone Can Write Software

How To Raise The Next Zuckerberg: 6 Coding Apps For Kids

http://readwrite.com/2013/04/19/how-to-raise-the-next-zuck-6-coding-apps-for-kids/

TYNKER - We Empower KIDS to Become Makers

https://www.tynker.com/

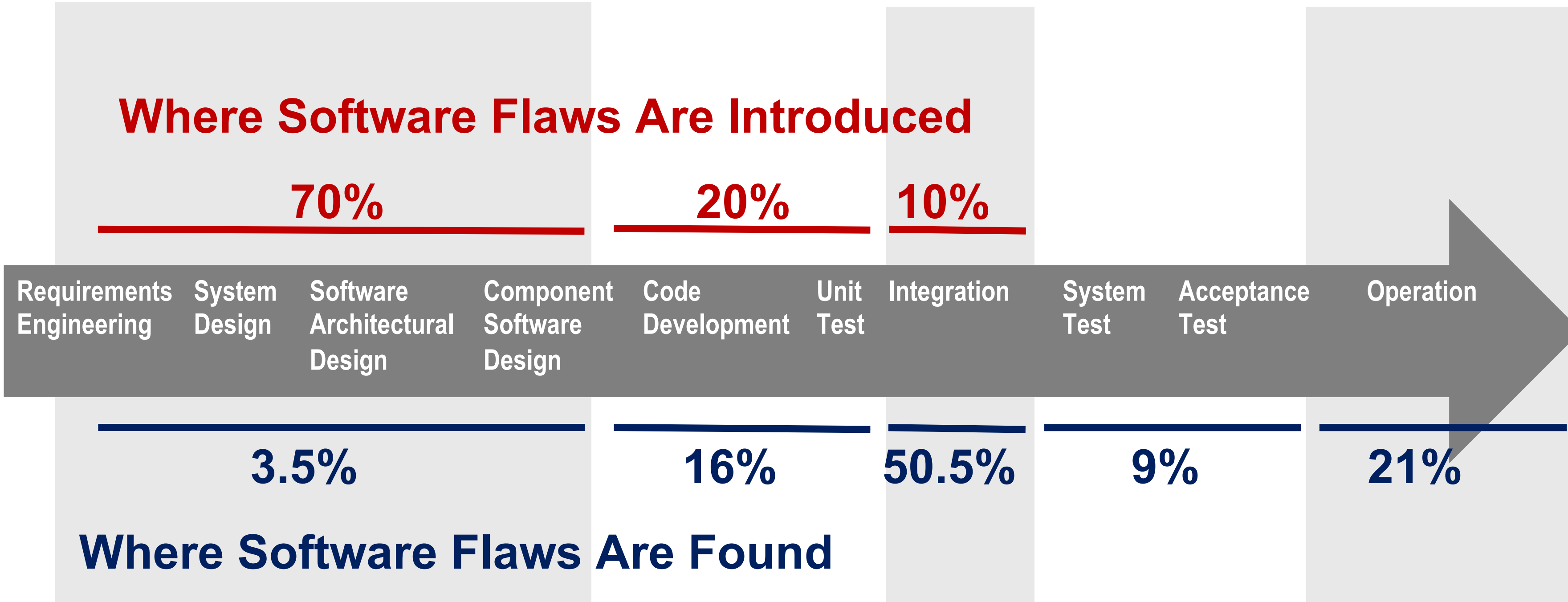
How and Why to Teach Your Kids to Code

http://lifehacker.com/how-and-why-to-teach-your-kids-to-code-510588878

From 1997 to 2012, software industry production grew from $149 billion to $425 billion

From 1990 to 2012, business investments in software grew at more than twice the rate of all fixed business investments; and from 2010 to 2012, software accounted for 12.2 percent of all fixed investment, compared to 3.5 percent for computers and peripherals

Measuring the Growing Defects in Software



Best-in-class code: <600 defects per MLOC
Very good code: 600 to 1,000 defects per MLOC
Average quality code: 6000 defects per MLOC

Up to 5% of defects are vulnerabilities

Sources: *Critical Code*; NIST, NASA, INCOSE, and Aircraft Industry Studies

Estimating Software Vulnerabilities

The **F-22** has 1.7 MLOC

- 1,020–10,200 defects (best – avg.)
- 51–510 vulnerabilities

The **F-35 Lightning II** has 24 MLOC

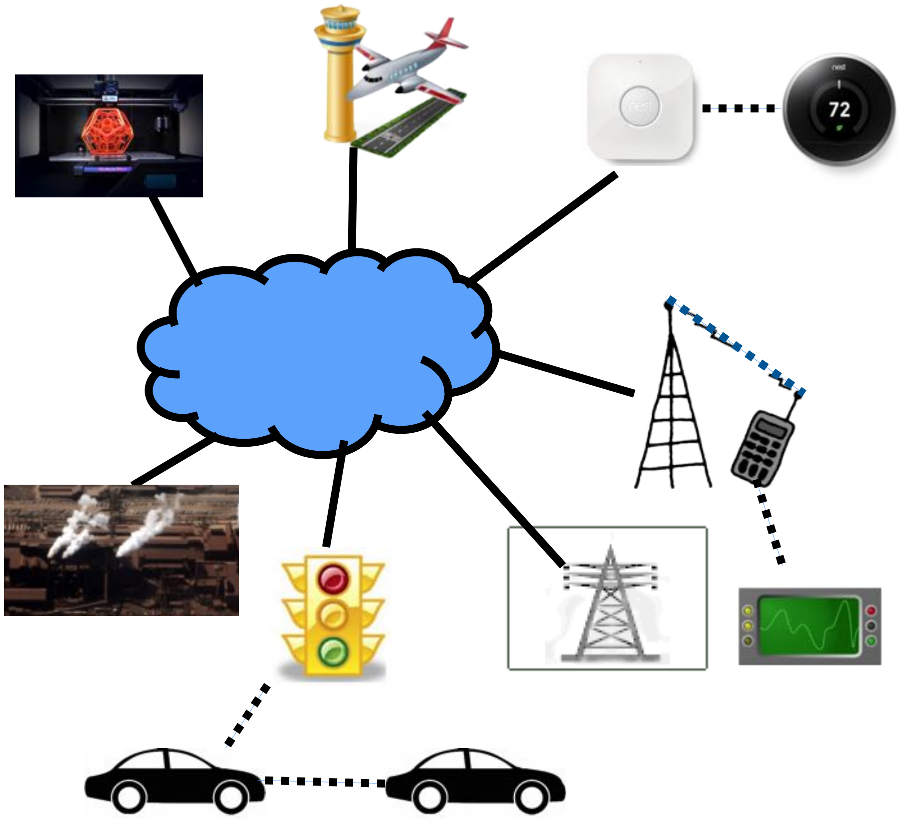
- 14,400–144,000 defects (best – avg.)
- 720–7,200 vulnerabilities

Best-in-class code:
<600 defects per MLOC

Very good code:
600 to 1,000 defects per MLOC
Average quality code: 6000 defects per MLOC.

5 % of defects are vulnerabilities.
Woody, Carol; Ellison, Robert; and Nichols, William. Predicting Software Assurance Using Quality and Reliability Measures. CMU/SEI-2014-TN-026. Software Engineering Institute, Carnegie Mellon University. 2014. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=428589)

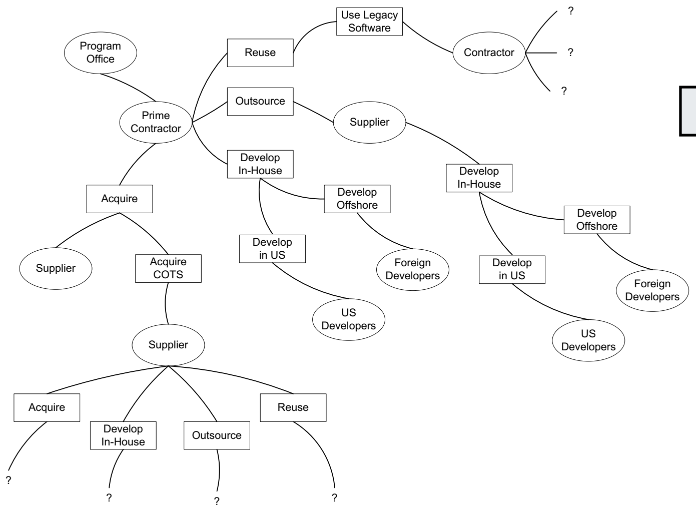
Software Connecting and Communicating Grows

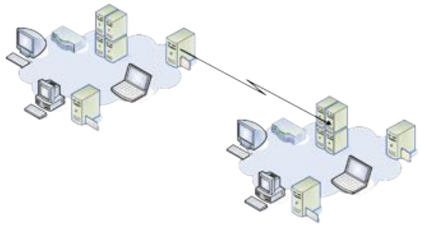


- Cellular
  - Main processor
  - Graphics processor
  - Base band processor (SDR)
  - Secure element (SIM)
- Automotive
  - Autonomous vehicles
  - Vehicle to infrastructure (V2I)
  - Vehicle to vehicle (V2V)
- Industrial and home automation
  - 3D printing (additive manufacturing)
  - Autonomous robots
  - Interconnected SCADA
- Aviation
  - Next Gen air traffic control
  - Fly by wire
- Smart grid
  - Smart electric meters
  - Smart metering infrastructure
- Embedded medical devices

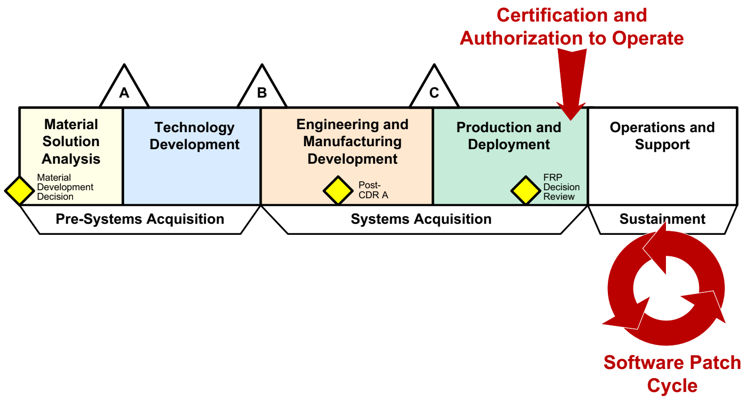
Systems are Built Independently by Many Hands

Software Supply Chain



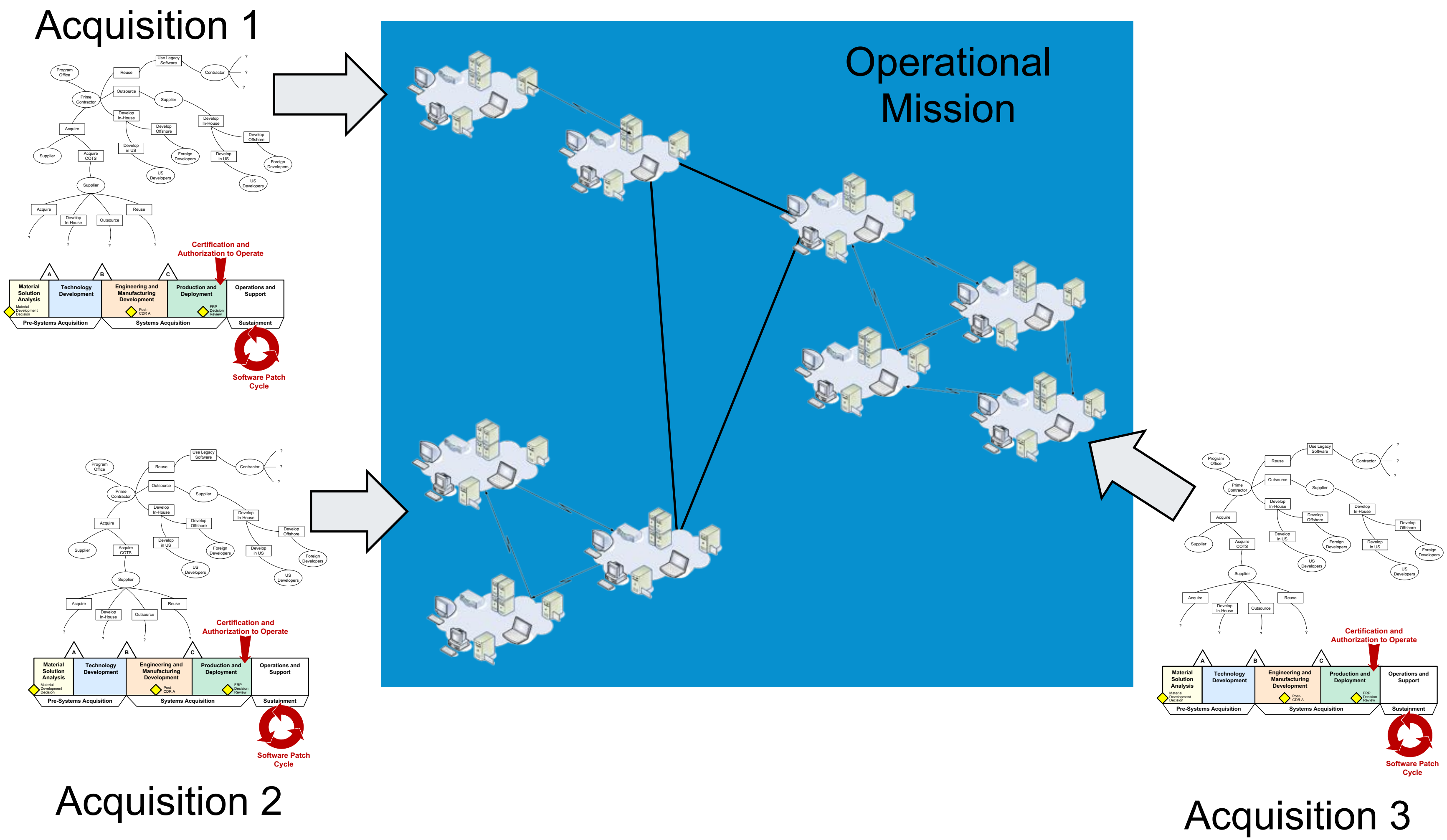


Integrated Software-Reliant System



Visibility and direct controls are limited to the top of the supply chain

Operational Missions Rely on Systems of Systems



Software in Systems of Systems - 1

SoS Characteristic (Maier 1998)	Growing Insecurity	Engineering Software to be Secure
Operational Independence	Acquirers/Integrators assemble software from many vendors to seamlessly deliver end-to-end mission capability	Acquirers must identify and mitigate vulnerabilities in software performing mission-critical functions
Managerial Independence	Vendors build and sell software for specialized niche markets (e.g. point-of-sales, printing, Cloud computing)	Acquirers select market dominants (costs more widely distributed, more resources for support, more functionality growth)
Evolutionary Development	Vendors release new functionality to capture market share and drop support of older versions	Acquirers must patch critical software quickly to reduce the attack potential

Software in Systems of Systems - 2

SoS Characteristic (Maier 1998)	Growing Insecurity	Engineering Software to be Secure
Emergent Behavior	Acquirer's focus on least cost and speed of delivery with extensive connectivity results in widespread vulnerability Vendors drive down costs through standardized interfaces (e.g. TCP/IP), reuse and push for early releases to dominate their niche markets; Vendor demand licenses that absolve them of liabilities	Acquirers must impose and monitor quality and security related requirements in their vendor contracts and ensure vendors manage their software supply chains effectively (increased costs and increased oversight)
Geographic Distribution	Vendors deliver insecure-by-default software (faster and easier)	Acquirer must impose secure-by-default requirements

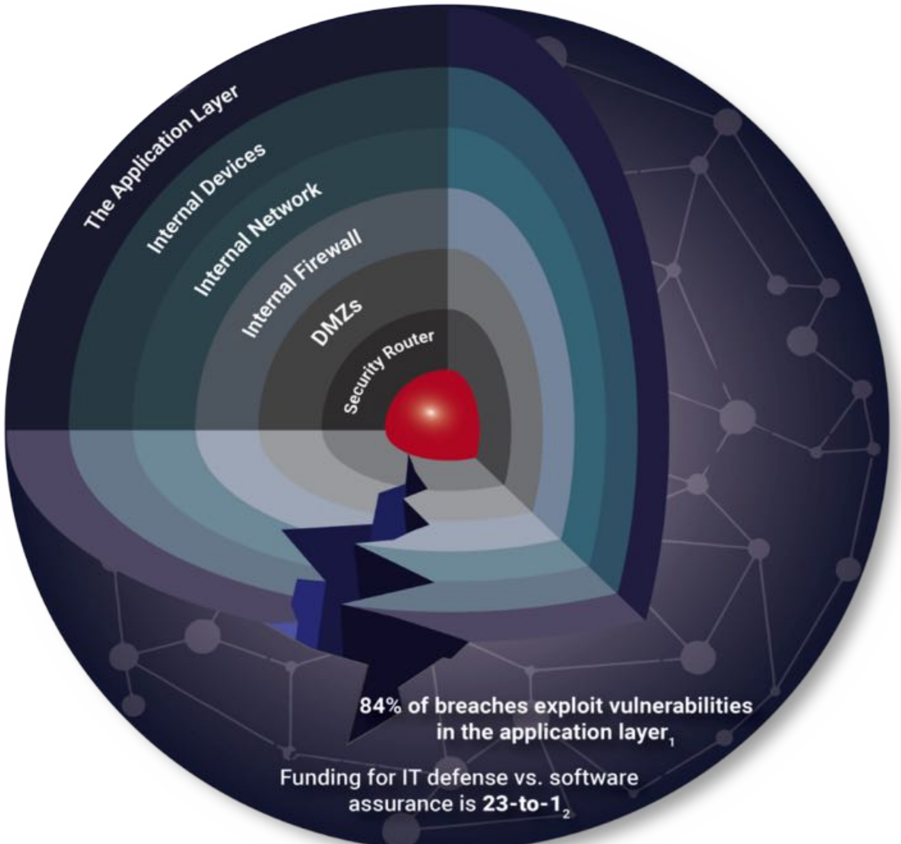
Security Vulnerabilities are Increasing

Security vulnerability is a weakness which allows an attacker to bypass security controls

Requires three elements:

- System susceptibility or flaw,
  - Millions of lines of software code, which contains defects, handling an ever increasing amount of functionality
  - Thousands of software vulnerabilities
  - Increased reliance on software not built for purpose (e.g. commercial and open source software)
- Attacker access to the flaw, and
  - Increased connectivity linking systems to other systems and connecting to new types of devices (Internet of Things)
  - Increased system and device remote communication capability
- Attacker capability to exploit the flaw
  - Access to the same tools and techniques used to build software
  - Reverse engineering capabilities for commercial and open source
  - Malware and attack platforms and frameworks

84% of Security Breaches Exploit the Software Applications



Security must be Engineered into the Lifecycle of Applications changing the way we build and buy technology

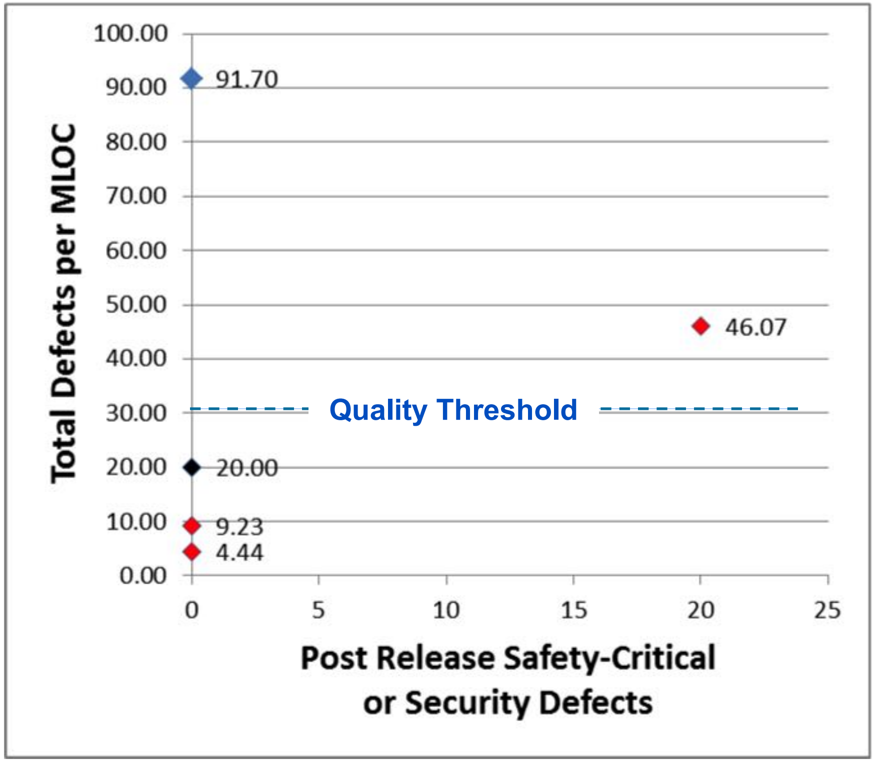
- "76% of U.S. developers use no secure application program process"[4]
- "More than 40% of software developers globally say that security isn't a top priority for them"[4]

1. Clark, Tim, *Most cyber Attacks Occur from this Common Vulnerability,* Forbes. 03-10-2015
2. Feiman, Joseph, *Maverick Research: Stop Protecting Your Apps; It's Time for Apps to Protect Themselves,* Gartner. 09-25-2014. G00269825
3. *Horvath, Mark, Neil MacDonald, Ayal Tirosh: Integrating Security Into the DevSecOps Toolchain, Gartner. 11-16-2017. G00334264*
4. Microsoft[1]– http://visualstudiomagazine.com/articles/2013/07/16/majority-of-us-devs-dont-practice-secure-coding.aspx

Software Quality Improves System Security

Data Shows Increased Quality can Reduce Security Risk

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



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

Woody, Carol et al. *Predicting Software Assurance Using Quality and Reliability Measures.* CMU/SEI-2014-TN-026. Software Engineering Institute, Carnegie Mellon University. 2014.
http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=428589

Semantic Gaps

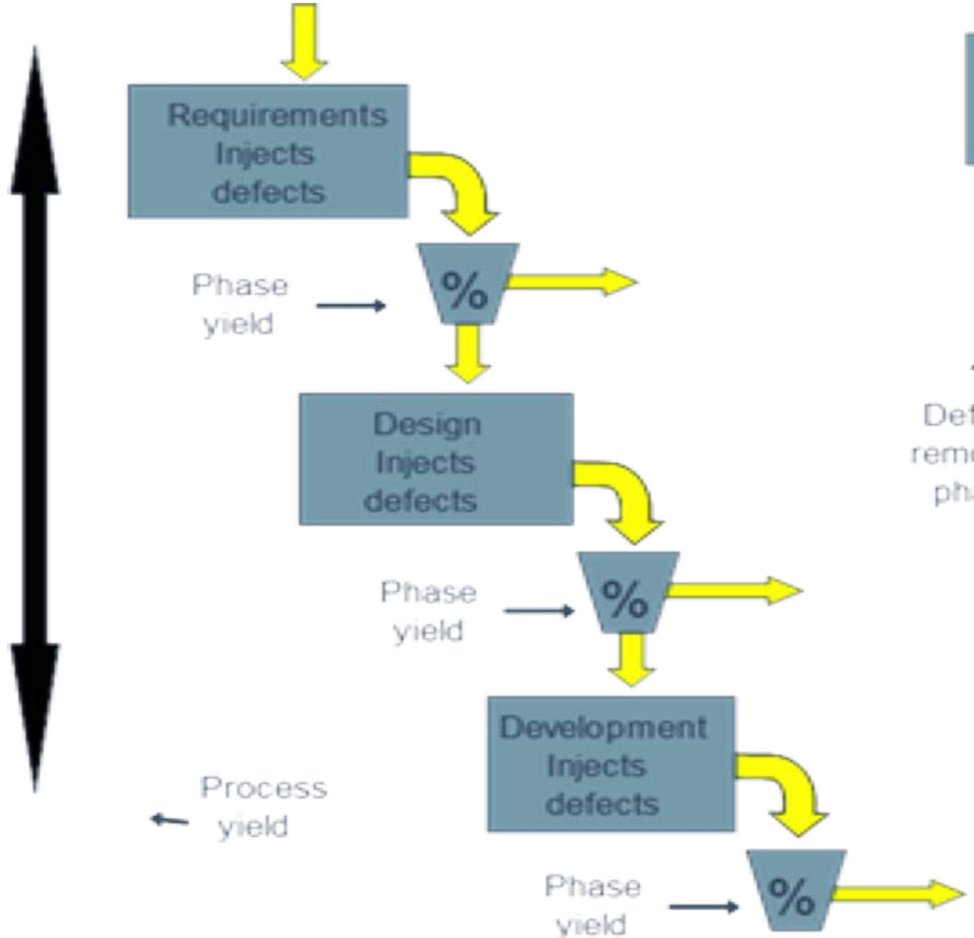
Quality tracks defects/faults (engineering and testing)

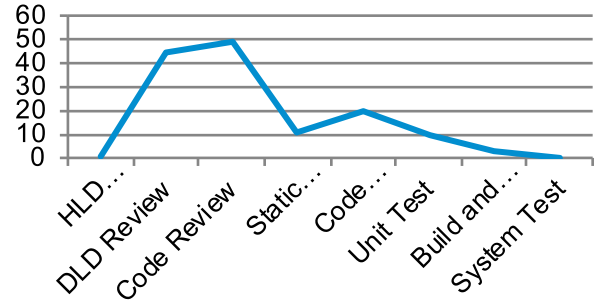
- **Defect:** non-fulfilment of intended usage requirements (ISO/IEC 9126) [essentially nonconformity to a specified requirement, missing or incorrect requirements]
- **Software fault:** accidental condition that causes a functional unit to fail to perform its required function (IEEE Standard Dictionary of Measures to produce reliable software 982.1, 1988)

Security cares about vulnerabilities (operations)

- **Information security vulnerability:** mistake in software that can be exploited by a hacker to gain access to a system or network (http://cve.mitre.org/about/terminology.html)

Quality Focuses on Defect Injection and Removal





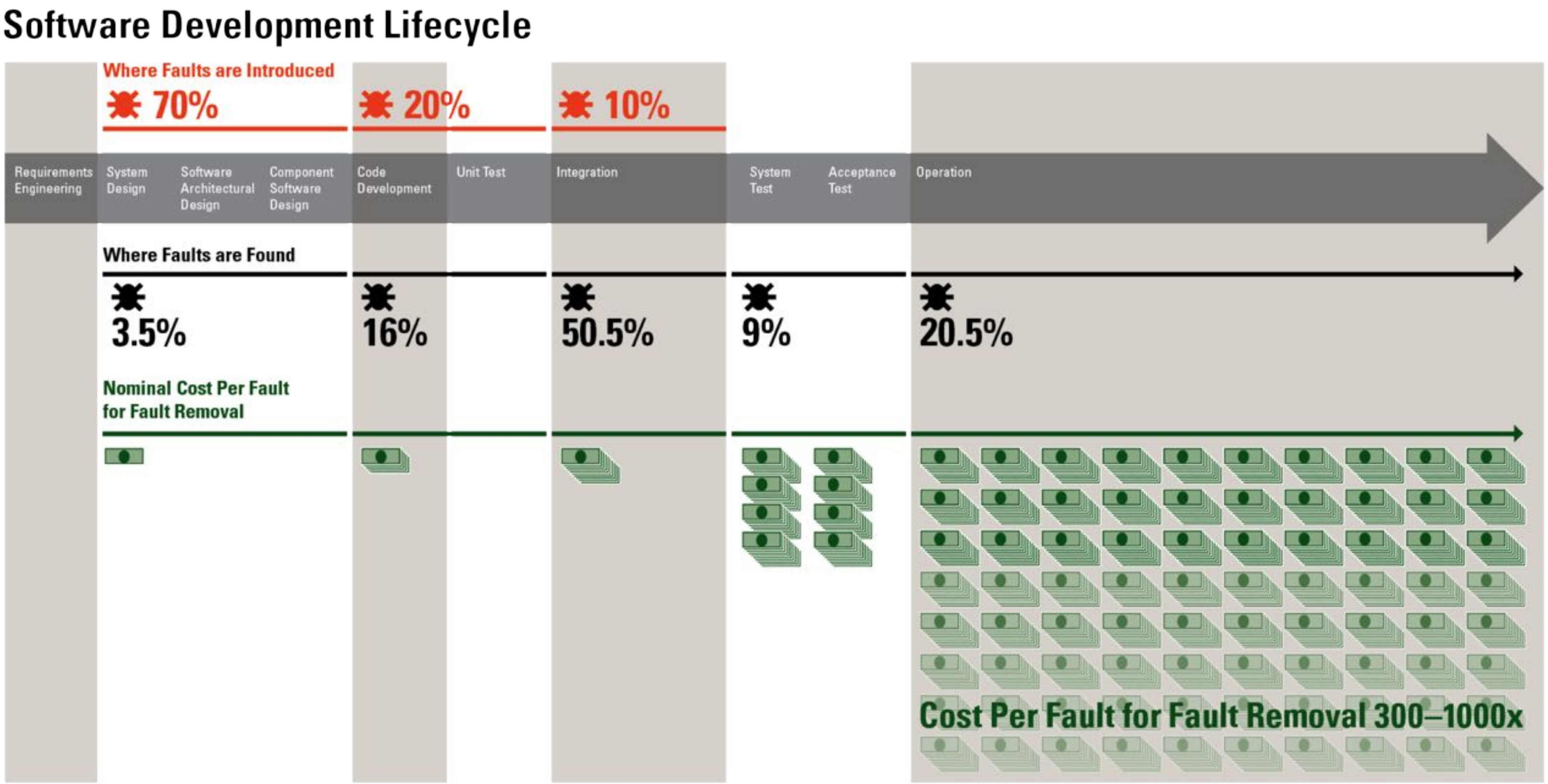
Poor quality does predict poor security

Effective quality focuses on defect removal at every step and provides cost-effective security results

Software Faults: *Introduction, Discovery, and Cost*

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

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



Security Engineering Risk Analysis (SERA) for a System of Systems

Security Engineering Risk Analysis (SERA)

What

- A systematic approach for analyzing complex security risks in software-reliant systems and systems of systems across the lifecycle and supply chain

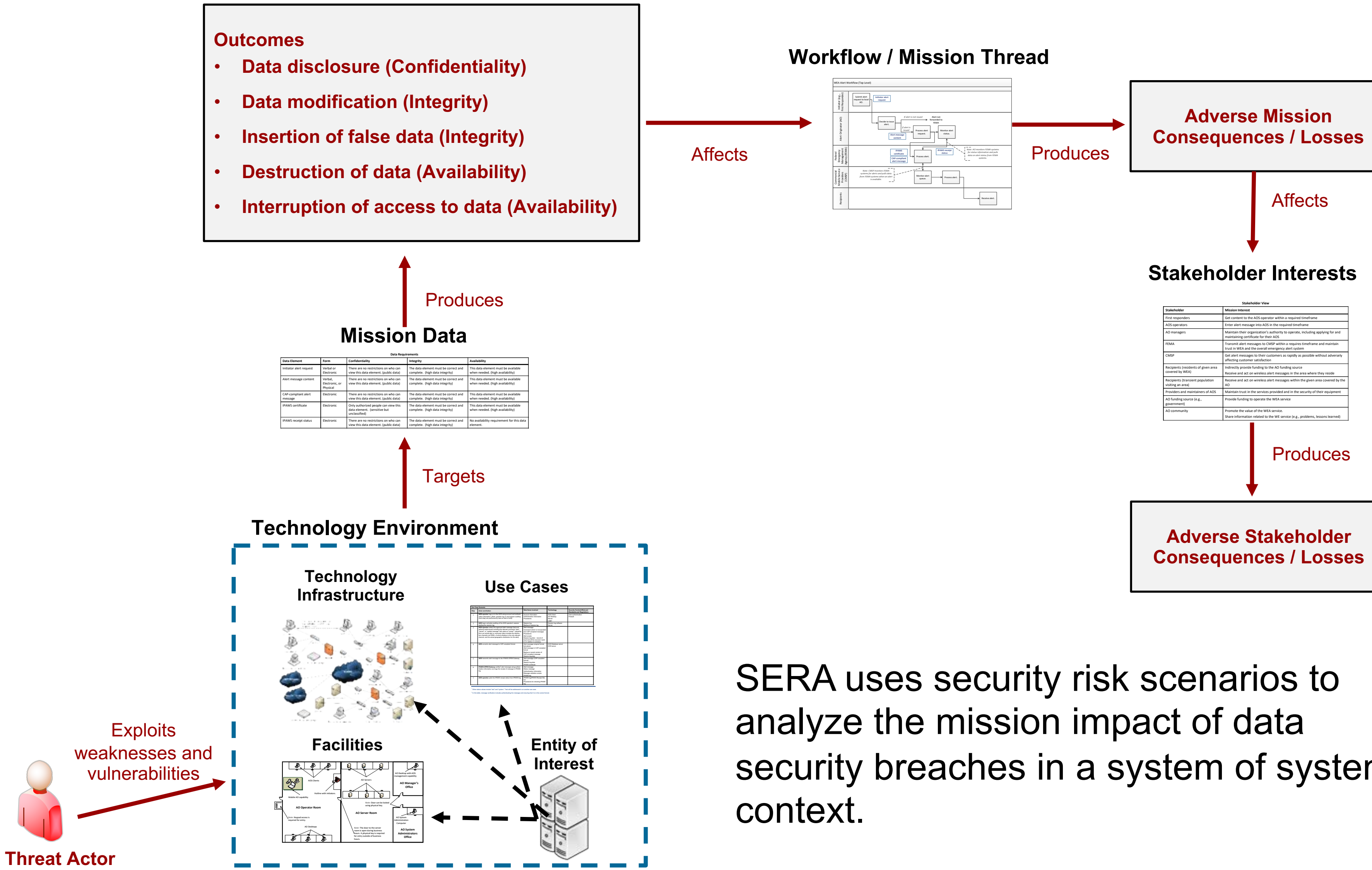
Why

- Build security into software-reliant systems by addressing design weaknesses as early as possible (e.g., requirements, architecture, design)
- Assemble a shared organizational view (business and technical) of system of system security risk

Benefits

- Identify and correct design weaknesses before a system is deployed
- Reduce residual cybersecurity risk in deployed systems

SERA Method: *Developing Mission Security Risk Scenarios*

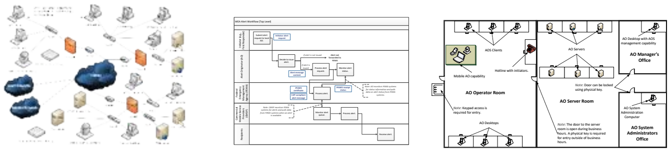


SERA uses security risk scenarios to analyze the mission impact of data security breaches in a system of systems context.

SERA Method: *Four Tasks*

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

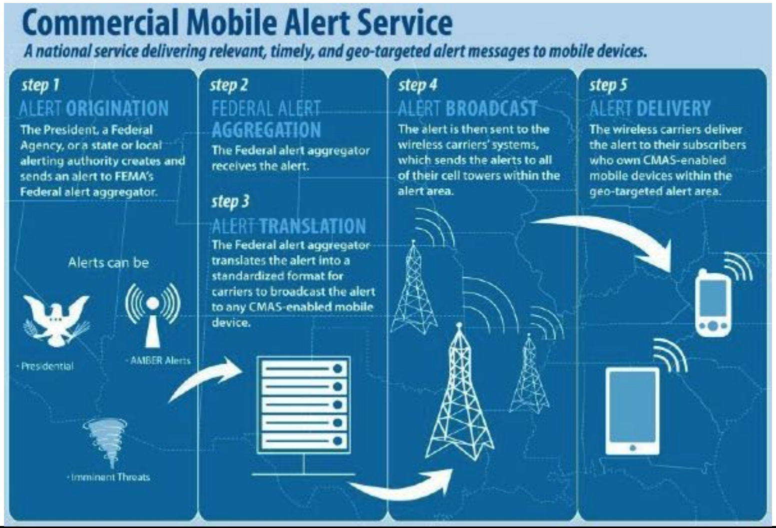
Control Approach Worksheet

Control Plan Worksheet

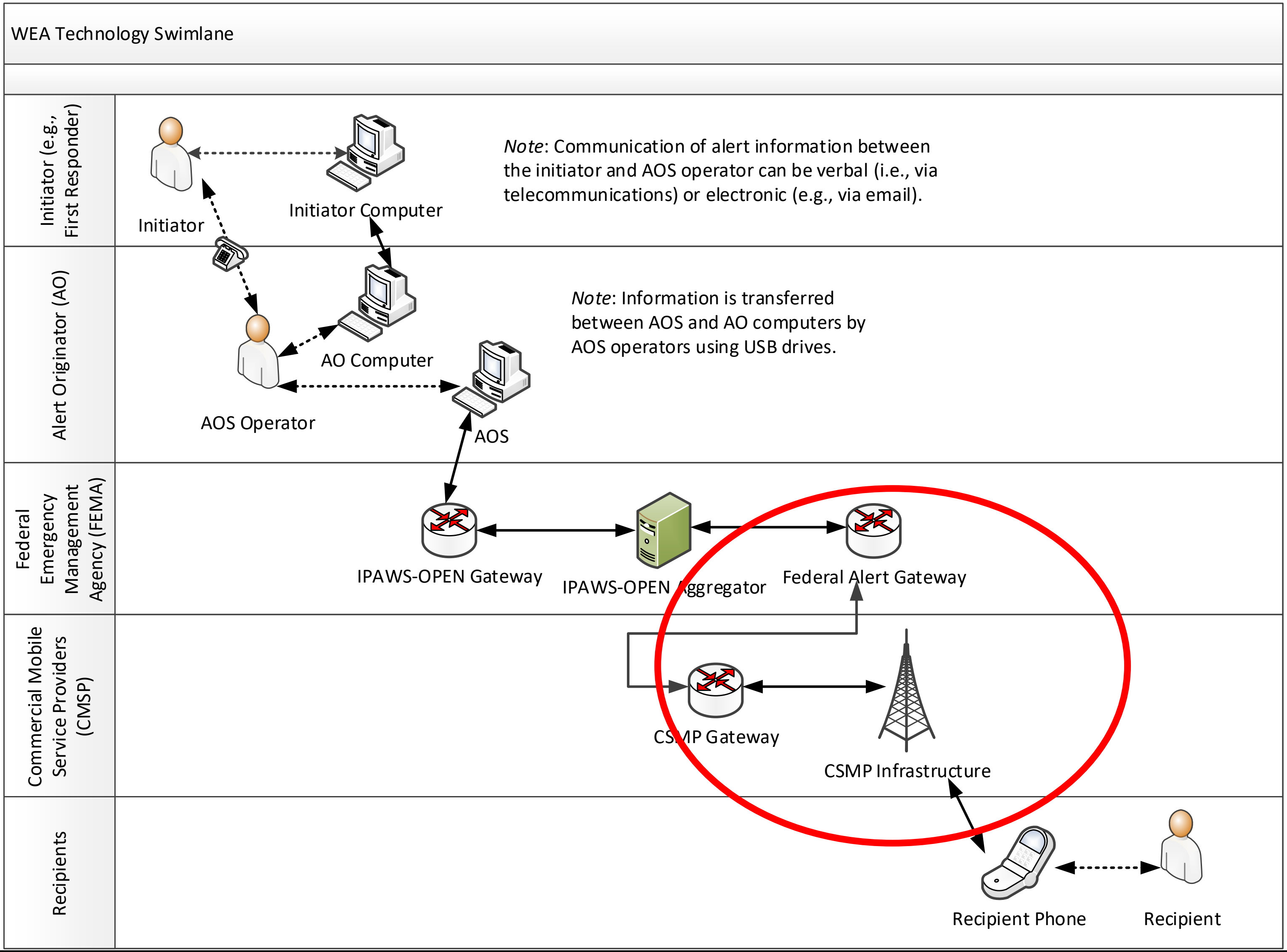
Example: *Wireless Emergency Alerts (WEA)*

WEA is a major component of the Federal Emergency Management Agency (FEMA) Integrated Public Alert and Warning System (IPAWS).

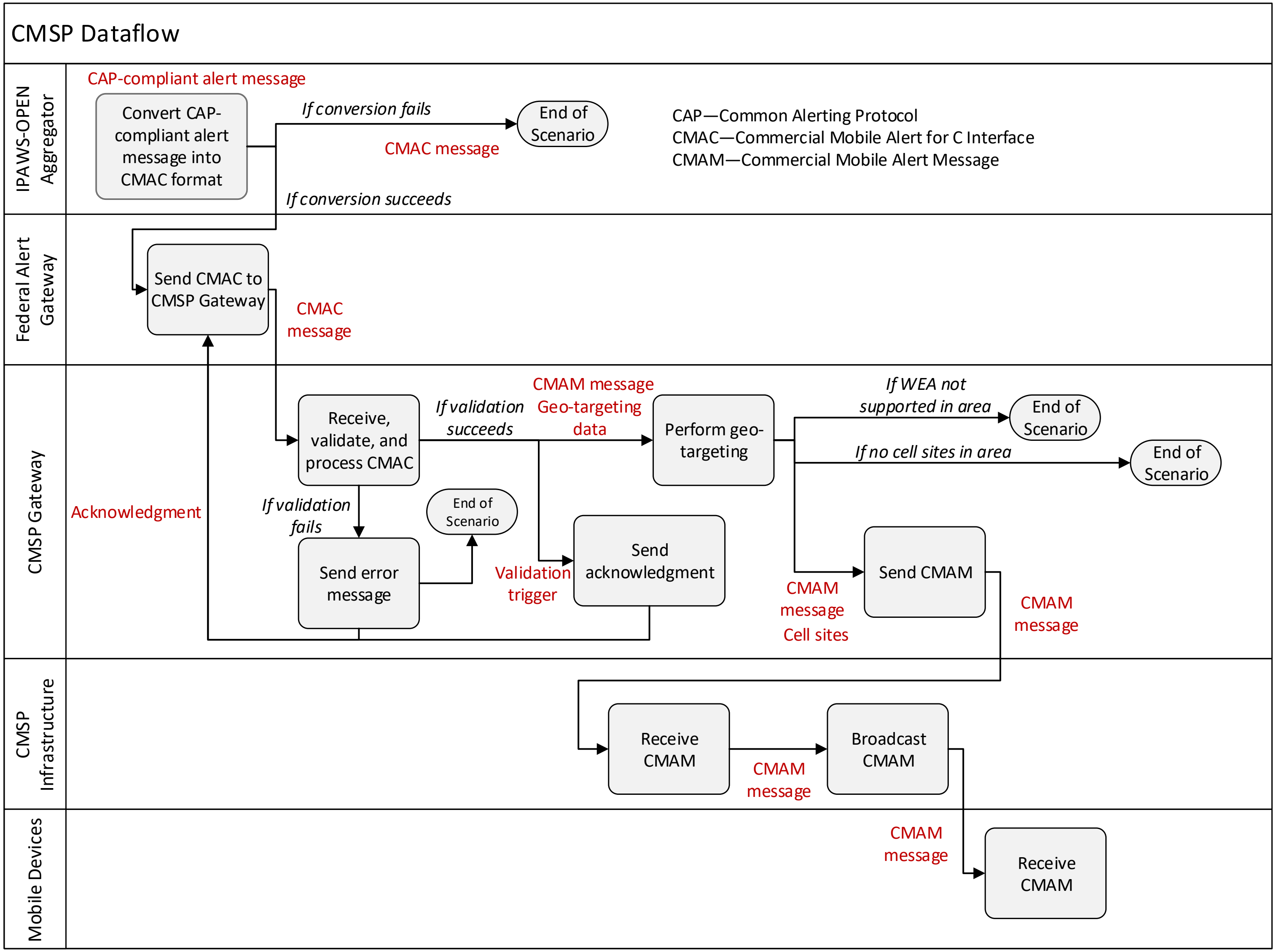
- Enables federal, state, territorial, tribal, and local government officials to send targeted text alerts to the public via commercial mobile service providers (CMSPs).
- Customers of participating wireless carriers with WEA-capable mobile devices will automatically receive alerts in the event of an emergency if they are located in or travel to the affected geographic area.



SERA Task 1: *WEA System of Systems*



SERA Task 1: *CMSP Dataflow*



SERA Task 2: *Elements of Security Risk Scenario*

Threat Components

- Actor – Motive – Goal – Outcome – Means – Threat Complexity

Threat Sequence

- Threat Step – Enabler(s)

Workflow Consequences

- Consequence – Amplifier(s)

Stakeholder Consequences

- Consequence – Amplifier(s)

SERA Task 2: *Security Risk Scenarios -1*

Example:

R1. Insider Sends False Alerts

- **IF** an insider with malicious intent uses the CMSP infrastructure to send nonsense alert messages repeatedly, **THEN** customers could become annoyed with the carrier; the carrier could incur considerable costs to recover from the attack; the carrier's reputation could be tarnished; and public trust in the WEA service could erode.

SERA Task 2: *R1 Threat Sequence*

T1. The insider is upset upon learning that he is not receiving a bonus this year and has been passed over for a promotion.

T2. The insider begins to behave aggressively and abusively toward his coworkers.

T3. The insider develops a logic bomb designed to replay a nonsense CMAM message repeatedly.

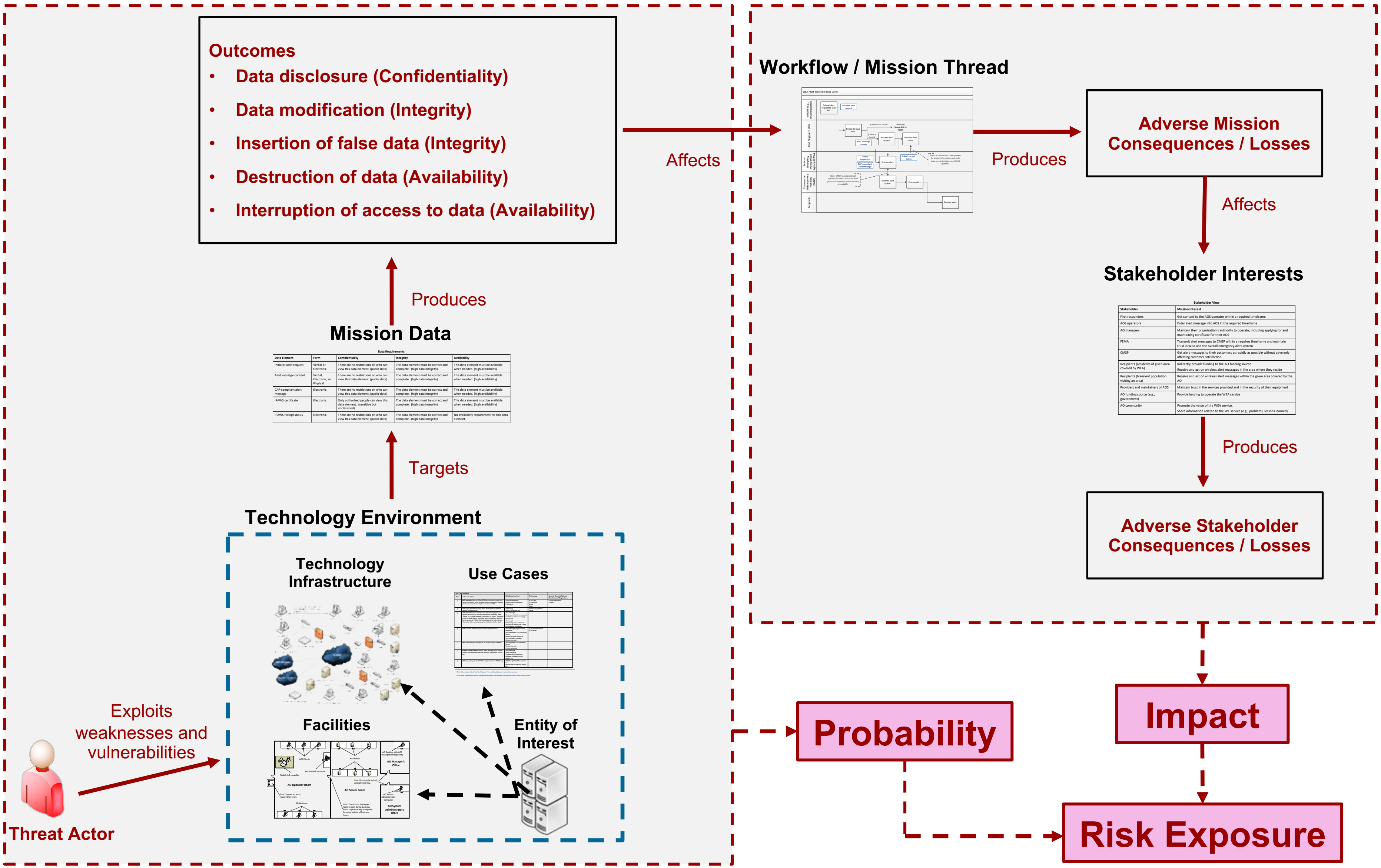
T4. The insider uses a colleague’s workstation to check-in the modified code with the logic bomb.

T5. Seven months later, the insider voluntarily leaves the company for a position in another organization.

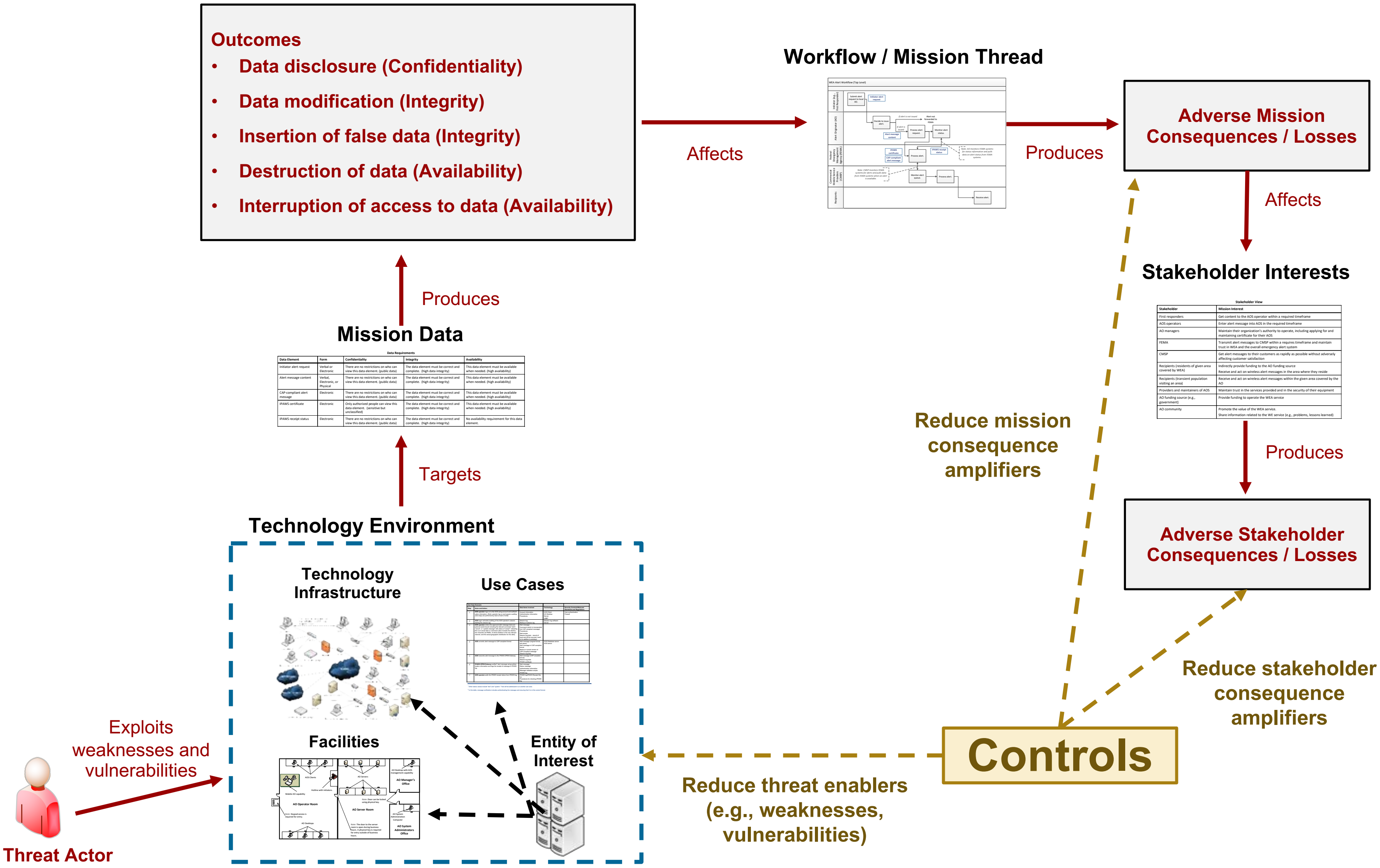
T6. Twenty-one days after the insider leaves the carrier, the logic bomb is activated automatically.

T7. The malicious code causes the carrier’s WEA service to send a nonsense WEA alert repeatedly to people across the country.

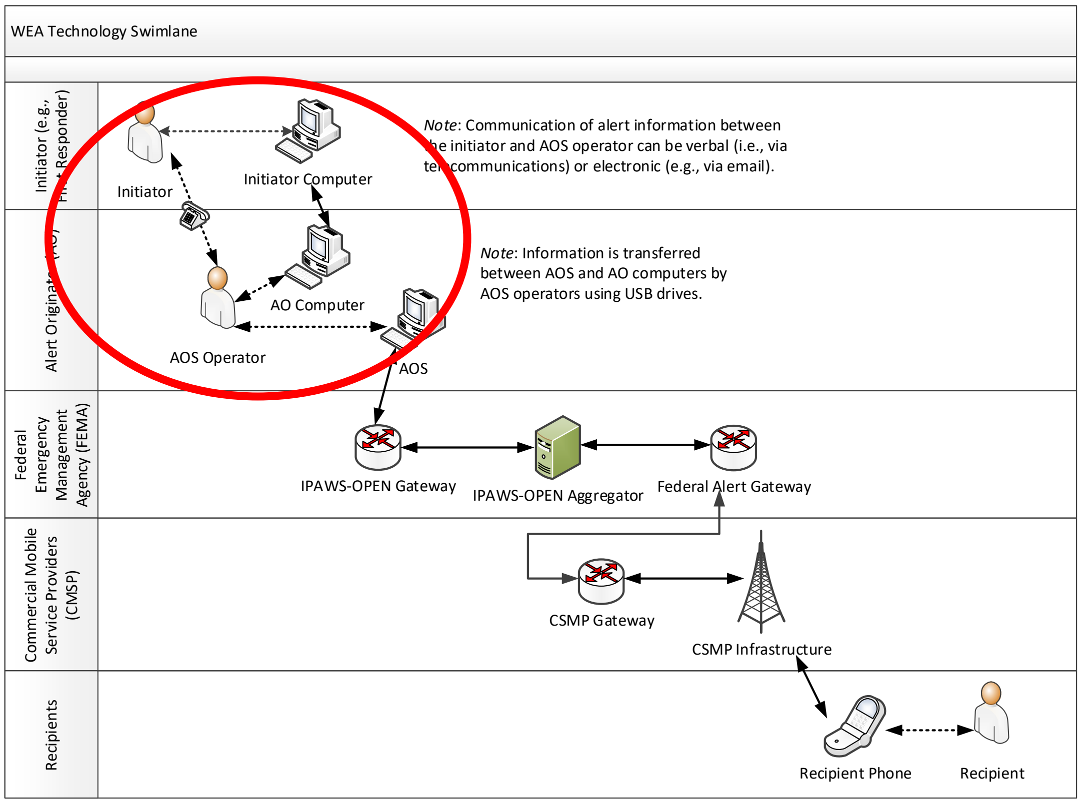
SERA Task 3: *Risk Measures*



SERA Task 4: *Controls*



Realized System of System Security Risk



Recent Hawaii incident involved sending a inaccurate public notice of a missile attack – the software did not require multiple confirming authorizations and a single bad actor created international havoc

Commercial Service Providers have no means of blocking messages from a legitimate source

Engineering Security into the System of Systems

Understand the Security Risks

Weak perceptions of security risk lead to poor security decisions

- Perceptions are primarily based on knowledge about successful attacks
  - the current state of security is largely reactive
  - successful organizations learn from attacks and figure out how to react and recover faster and be vigilant in anticipating and detecting attacks

Develop security attack scenarios using SERA to evaluate "what if" possibilities

Interconnections Expand Access to Software Vulnerabilities

Highly connected systems require alignment of risk across all stakeholders and systems to ensure critical security risks are not ignored

- Decisions for reduced quality and accepted software vulnerabilities in one system can increase security risks for other systems
- Security must also be balanced with other opportunities/needs (performance, reliability, usability, etc.)
- Interactions occur at many technology levels (network, security appliances, architecture, applications, data storage, etc.) and are supported by a wide range of roles

Collaborative choices for system security are needed for systems that must interrelate to address a mission

Attackers Do Not Respect System and Organizational Boundaries

There are no prefect protections against attacks.

There exists a broad community of attackers with growing technology capabilities able to compromise the confidentiality, integrity, and availability of any and all of your technology assets and the attacker profile is constantly changing.

- The attacker uses technology, processes, standards, and practices to craft a compromise (socio-technical responses).
- Attacks are crafted to take advantage of the ways we normally use technology or designed to contrive exceptional situations where defenses are circumvented

Security decisions need to include consideration of how the system and system of systems can be attacked

Additional Material Scheduled for Publication

Engineering Emergence

A Modeling and Simulation Approach

Part of the System of Systems Engineering Series
Mo Jamshidi – Series Editor

Editors/Affiliations

Larry B. Rainey, Integrity Systems and Solutions, Colorado, USA
Mo Jamshidi, University of Texas, San Antonio, USA

The book examines the nature of emergence in context of man-made (i.e. engineered) systems, in general, and system of systems engineering problems. It investigates emergence from a modeling and simulation perspective to interrogate or explore the domain space via modeling and simulation to facilitate understanding, detection, classification, and prediction of the phenomenon. The text is the first to address emergence from an engineering perspective. It uses the discipline of modeling and simulation to explore the phenomenon of emergence found in man-made systems, in general, and system of systems engineering applications, specifically.

SAVE 20%

Key Features

- Addresses emergence as it is found in man-made systems. Also, considers the environment for understanding emergence.
- Provides specific examples of how various modeling and simulation paradigms/techniques can be used to investigate emergence in an engineering context.
- Explains how modeling and simulation can be used to facilitate the detection, classification, prediction, and control of the phenomenon of emergence.

Selected Contents

Section I: Introduction and Overview. Introduction and Overview for Engineering Emergence: A Modeling and Simulation Approach. System of Systems Engineering: An Overview. Section II: Theoretical Perspectives. DEVS-based Modeling and Simulation Framework for Emergence in System-of-Systems. Sources for Emergence and Development of Systems-of-Systems. Leveraging Deterministic Chaos to Mitigate Combinatorial Explosions. Phenomenological and Ontological Models for Predicting Emergence. System of Systems Integration Process Model. Simulation Tool Requirements for Engineering Emergence. An Ontological Study of Emergence. Modeling and Validation Challenges for Complex Systems. Foundations for the Modeling and Simulation of Emergent Behavior Systems. Goal Oriented Requirements Engineering for Emergence in Self Adaptive System of Systems. Engineered to be Secure. Cyber Insecurity is Growing Exponentially. The Challenge of Performing Research Which Will Contribute Engineering Helpful Knowledge of Emergence. Section III: Theoretical Perspectives with Practical_Applications. Macroscopic Features of Emergence in Man-Made Systems. A Model Generation, Verification and Validation Method for Purging Negative Behaviors from a Design. A Model-Based Approach to Investigate Emergent Behaviors in Systems of Systems. InterDyne: A Simulation Method for Exploring Emergent Behavior Deriving from Interaction Dynamics. Verification Approaches for Complex Systems. Emergence in the Context of System of Systems. Section IV. Summary. Lessons Learned and the Proposed Way-Ahead.

ENGINEERING EMERGENCE
Larry B. Rainey
Mo Jamshidi

SAVE 20% when you order online and enter Promo Code **EEE17**
FREE standard shipping when you order online.

Catalog no. K33015
January 2019, 500 pp.
ISBN: 978-1-138-04616-0
$129.95 / £100.00

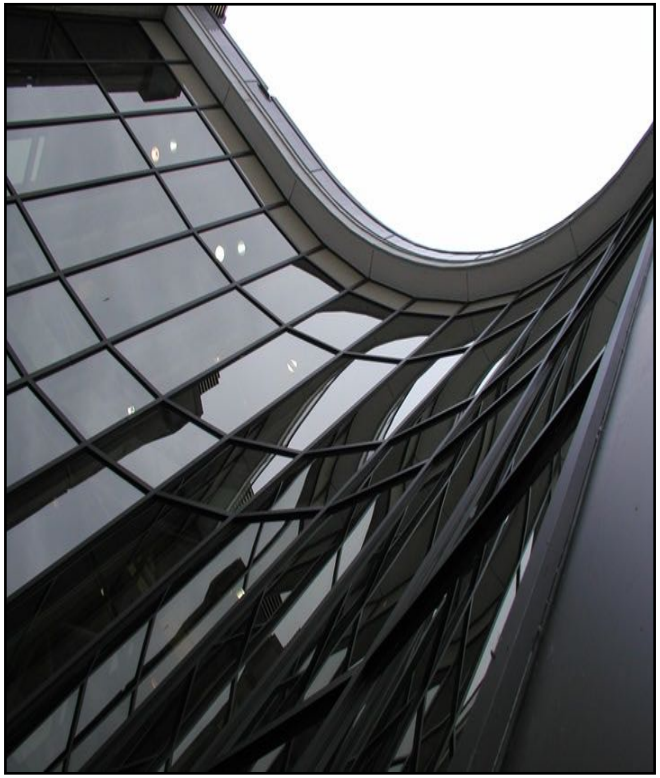
www.crcpress.com

CRC Press
Taylor & Francis Group

e-mail: orders@crcpress.com
1-800-634-7064 • 1-561-994-0555 • +44 (0) 1235 400 524

Two chapters are included in **Engineering Emergence** to be released January 2019 highlighting "Engineered to be Secure" and "Cyber Insecurity is Growing" for systems of systems.

Contact Information



Carol Woody, Ph.D.

cwoody@cert.org

Web Resources (SEI)

http://www.sei.cmu.edu/