Cybersecurity Engineering is Critical to Mission Success

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

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

Notices

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.

DM20-0787

Agenda

1 – What is Cybersecurity Engineering

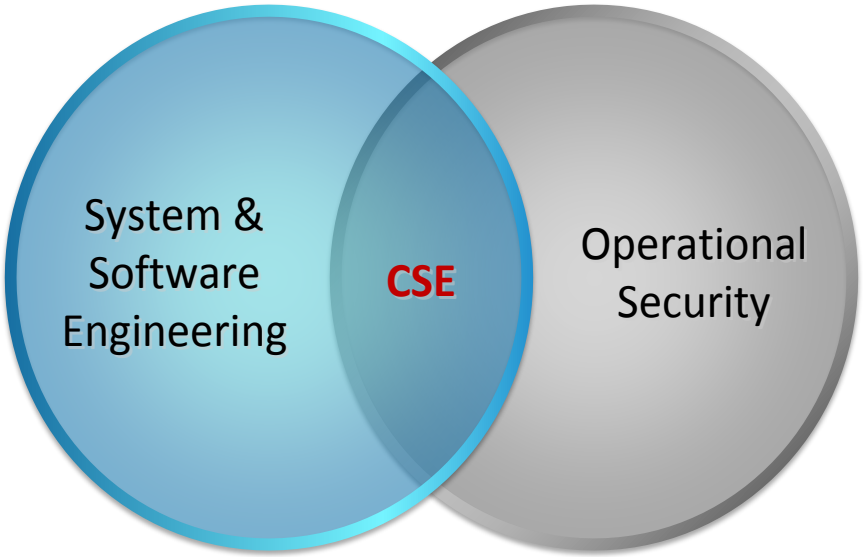
2 –Today's Technology Context

3 – Cybersecurity Engineering is Needed

4 – Summary

What is Cybersecurity Engineering

Cybersecurity Engineer



Preparing systems to operate securely – reliance on operational capability alone is insufficient in today's complex and highly integrated technology environments

Defines how a system's cybersecurity will perform in a system of systems context

- Plans and designs trust relationships
- Negotiates appropriate security requirements to ensure confidentiality, integrity, availability with monitoring in systems and software
- Plans and designs sufficient resiliency to recognize, resist, and recover from attacks
- Plans for operational security under all circumstances – designed in methods of denying critical information to an adversary to avoid/minimize mission impact
- Evaluate alternatives to determine the level of accepted cybersecurity risk

Cybersecurity Addresses Critical Needs

Key areas where system and software engineering are insufficient for building technology to operate in today's highly contested environments

- Risk Determination
- Defining and Monitoring System and Component Interactions
- Trusted Dependencies
- Attacker response
- Coordination of Security Throughout the Life Cycle
- Cybersecurity Education for Stakeholders, Developers, and Operators
- Planned and Dynamic Protection and Response
- Measurement for Cybersecurity Improvement

Reference:

"Principles and Measurement Models for Software Assurance", Nancy R. Mead, Dan Shoemaker (University of Detroit Mercy), Carol Woody, 2013 IJSSE Special Issue on Cybersecurity Scientific Validation, January 2013, http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=298843

Risk Determination

Cybersecurity engineering incorporates effective consideration of threats and mission risk

Perceptions of risk drive assurance decisions and lack of cybersecurity expertise leads to poor assurance choices

- Misperceptions are failures to recognize threats and impacts – "how could it happen to us?" or "it could not happen here!"
- Current perceptions are primarily based on knowledge about successful attacks and how threats could impact the operational mission
  - few acquisition and development leaders have shown expertise in this area
  - successful organizations learn from attacks (theirs and others) and figure out how to react and recover faster and be vigilant in anticipating and detecting attacks

Defining and Monitoring System and Component Interactions

Cybersecurity engineering considers risk to systems from the interaction among technology components and external systems

Highly connected systems require alignment of risk across all stakeholders and systems otherwise critical threats will be unaddressed (missed, ignored) at different points in the interactions

- Interactions must be designed to be assured
- There are costs to addressing assurance which must be balanced against the impact of the risk
- Risk 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

Trusted Dependencies

Cybersecurity engineering evaluates dependencies and inherited risk to ensure the appropriate level of trust is established

Your assurance may depend on other people's decisions and the level of trust you place on these dependencies (inherited risk):

- Each dependency represents a risk
- Dependency decisions should be based on a realistic assessment of the threats, impacts, and opportunities represented by an interaction
- Dependencies are not static and trust relationships should be reviewed to identify changes that warrant reconsideration
- Using many shared components (reuse, open source, collaboration environments) to build technology applications and infrastructure increases the dependency on other's assurance decisions

Attacker Response

Cybersecurity engineering will identify and plan for the types of attacks that are considered mission critical.

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

Coordination of Security Throughout the Lifecycle

Cybersecurity engineering is involved throughout the lifecycle to ensure coordination across all aspects.

Assurance requires planning for what might go wrong and effective risk coordination among all technology participants and their governing bodies

- Protection must be applied broadly across the people, processes, and technology because the attacker will take advantage of all possible entry points
- Authority and responsibility must be clearly established at an appropriate level in the organization to ensure effective participation and coverage

Cybersecurity Education for Stakeholders, Developers, and Operators

Cybersecurity engineering coordinate the knowledge levels needed from each participant in the lifecycle to ensure all participants have the appropriate level of competencies for their responsibilities

Your assurance will depend on the knowledge your resources bring

- Many hands touch the various parts of technology and each needs to have an appropriate level of cybersecurity expertise
- Data and decisions about how technology is acquired and used are spread across many parts of an organization
- Compromises can occur from:
  - Anyone on the organization's network clicking on a compromised email (Phishing attacks)
  - Anyone downloading compromised files (Malware attacks)
  - Anyone connecting a poorly protected device to the network
  - Developers writing vulnerable code and building APIs with insufficient protection
  - Leadership and vendors making poor technology investment and connectivity choices (Supply chain attacks)

Planned and Dynamic Protection and Response

Cybersecurity engineering focuses on preparation for the adversary in the operational context.

The capabilities to respond to a changing threat landscape must be designed into the system.

Implementation must represent a balance cybersecurity with competing qualities (i.e. safety, performance, and reliability) and adjust to changes in each of these areas

- Engineering challenge: Assurance cannot be added later; you must plan and build to the level of acceptable assurance that you need
- Continuous monitoring must be part of the planned response
- No one has resources to redesign systems every time the threat changes

Measurement for Cybersecurity Improvement

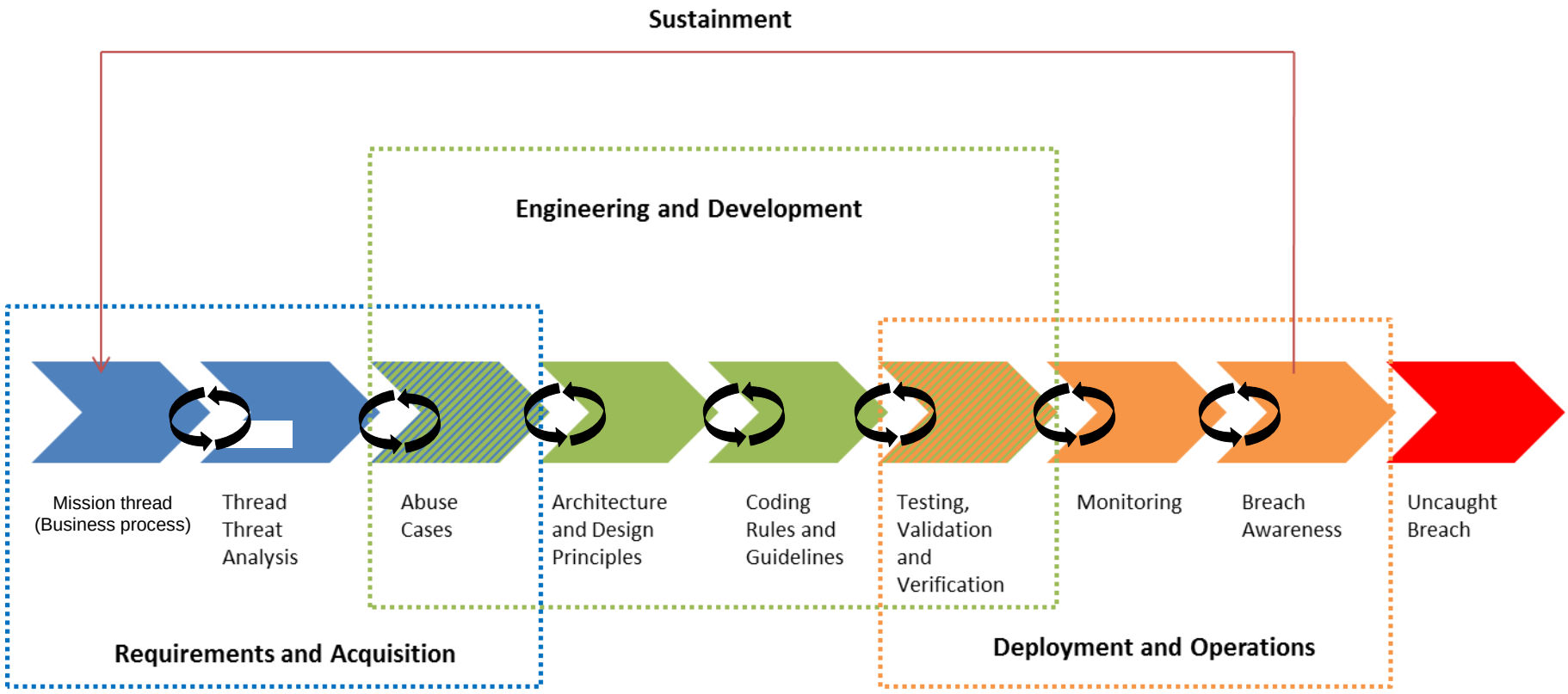
Cybersecurity engineering identifies data collected and analyzed to show accepted risk and identify opportunities for improvement

A means to measure and monitor assurance is critical to making it visible

- All elements of the socio-technical environment must tie together (practices, processes, procedures, products, etc.) and measurements must be consistent
- Effective measurement is well supported by sound engineering and organizational principles - well formed and consistently applied processes are critical to ensure an appropriate measurable response
- Measurement must be multi-faceted

Today's Technology Context

Cybersecurity Is a Lifecycle Challenge



Design Weaknesses

Coding Weaknesses

Implementation Weaknesses

Threat Response Needs to be Real-Time

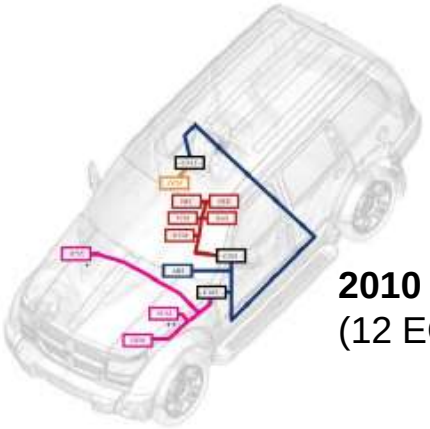
Why Automotive Cybersecurity Needs Real-Time Threat Detection

Though autonomous vehicles have been around for a while, the threat landscape they face has changed significantly in recent years. The focus used to be on securing these vehicles against vulnerabilities introduced during design or manufacturing stages. Now, as cybersecurity threats have become more sophisticated and adaptive, there is a growing consensus that automotive security needs to take the same step: away from a retroactive focus on eliminating security holes and towards real-time vulnerability scanning systems. In this article, we'll look at the changing threat landscape in the automotive sector and explain why the sector needs to make the transition to real-time threat detection.

https://www.rtinsights.com/why-automotive-cybersecurity-needs-real-time-threat-detection/

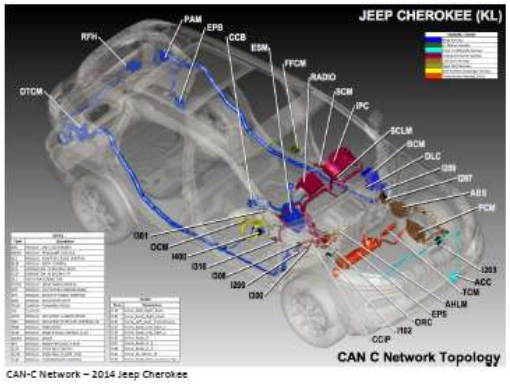
Distributed by Auto-ISACA **September 11, 2020**

Modularity is the Focus: Vehicles are now Assembled from Engine Control Units (ECUs)



2010 Jeep Cherokee
(12 ECUs)

2014 Jeep Cherokee
(32 ECUs)



ECUs are prefabricated, software driven components addressing select functionality and tailorable to a specific domain

Modern high end automotive vehicles have

- Over 100M lines of code
- Over 50 antennas
- Over 100 ECUs

Sources: Miller and Valasek, A Survey of Remote Automotive Attack Surfaces, http://illmatics.com/remote%20attack%20surfaces.pdf;
https://www.cst.com/webinar14-10-23~?utm_source=rfg&utm_medium=web&utm_content=mobile&utm_campaign=2014series
https://en.wikipedia.org/wiki/Electronic_control_unit

Software is Everywhere

All software is data and data is nothing more than 1s and 0s stored in defined patterns by software driven tools on some software controlled platform.

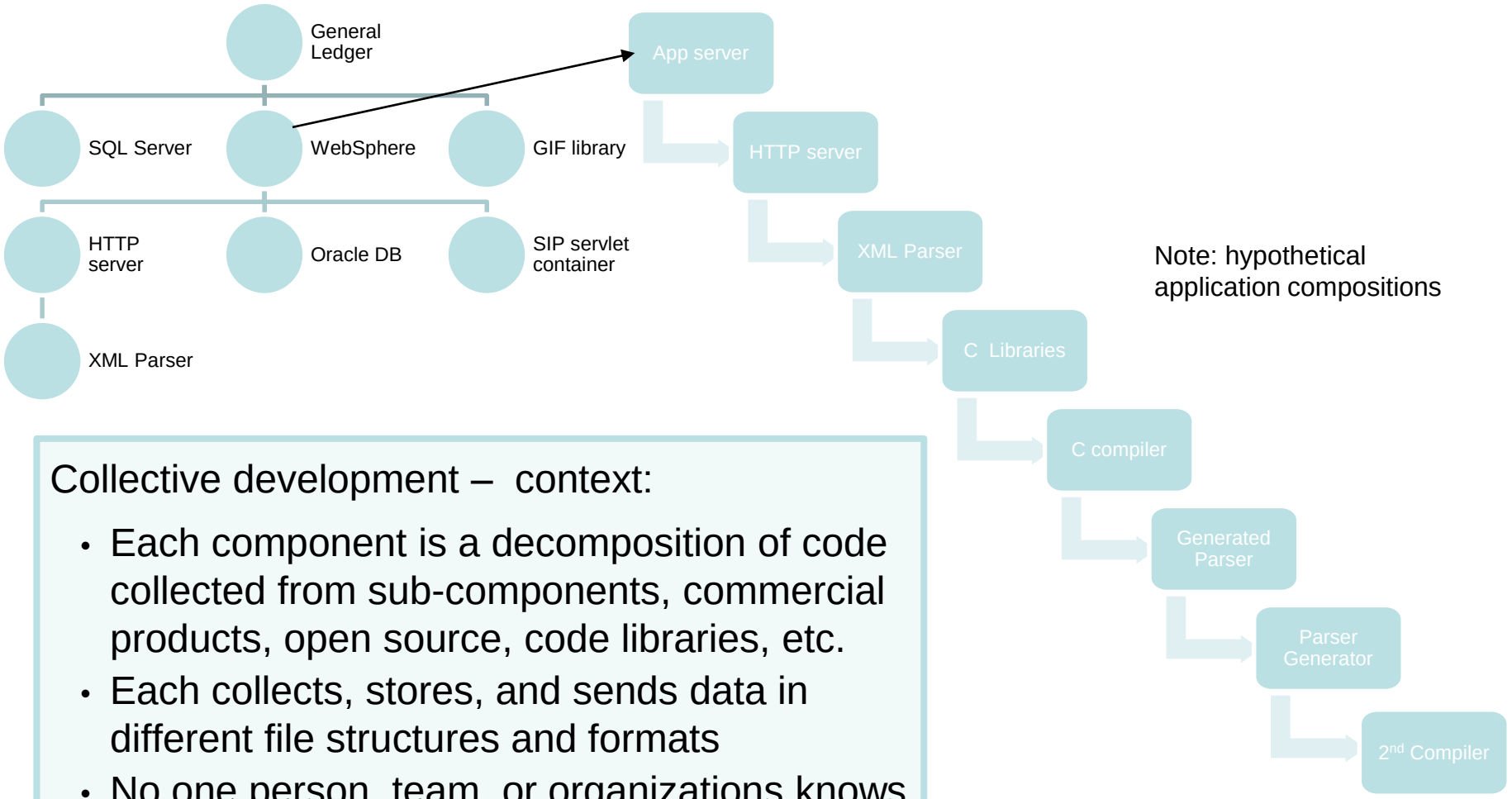
No matter what your focus, software is now a part of it and just about everything is a software platform.

Controls that limit the access and use of data and software are applied by software.

No software is perfect. The best written software still has 300 defects per million lines of code (MLOC). Average software has 6000 defects per MLOC.

(reference: Capers Jones, *sqgne.org/presentations/2011-12/**Jones**-Sep-2011.pdf)*

Software Development is now Assembly



Note: hypothetical application compositions

Collective development – context:

- Each component is a decomposition of code collected from sub-components, commercial products, open source, code libraries, etc.
- Each collects, stores, and sends data in different file structures and formats
- No one person, team, or organizations knows how all the pieces work

Reuse is rampant

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/

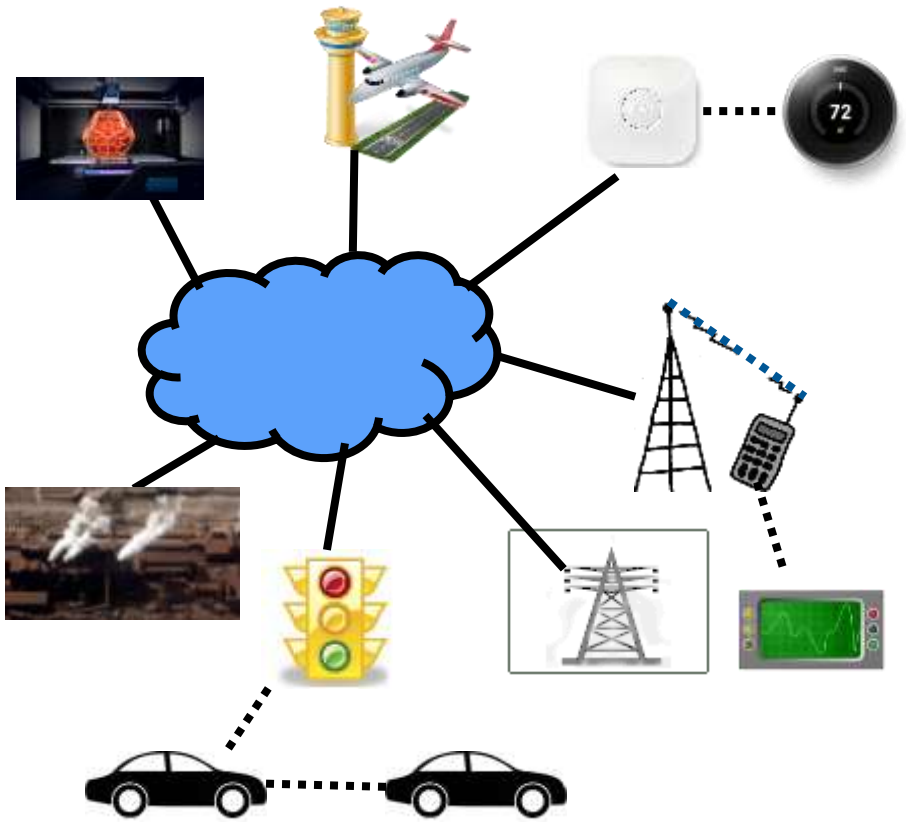
How and Why to Teach Your Kids to Code

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

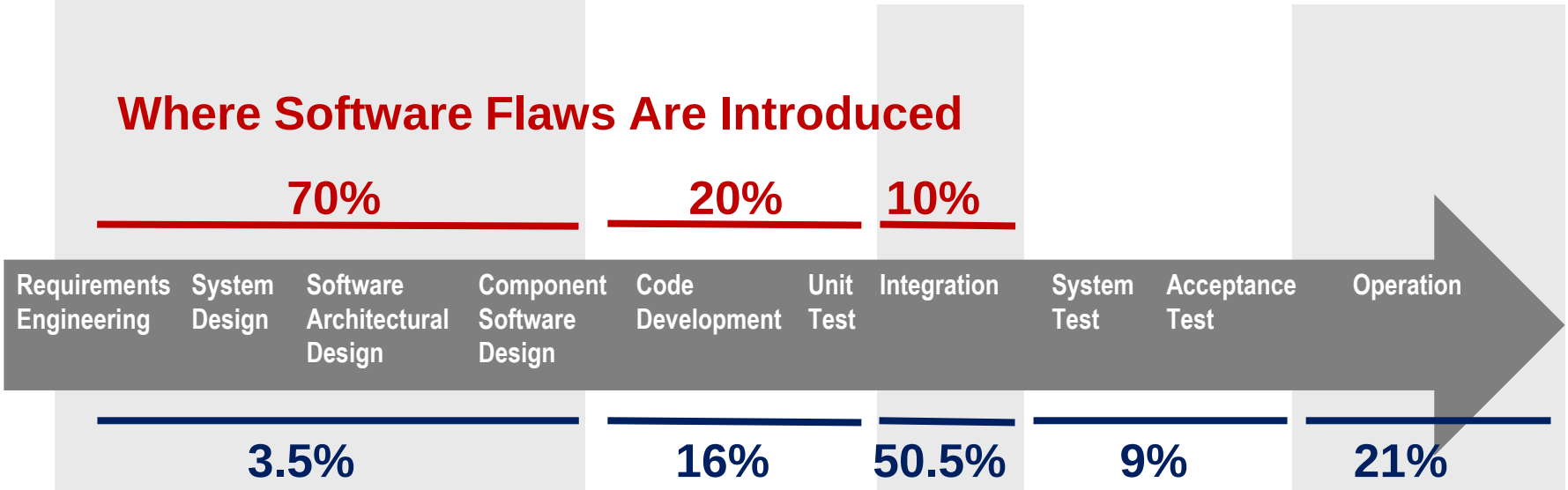
From 1990 to 2012, business investments in software grew at more than twice the rate of all fixed business investments; and from 2010 to 2012, software accounted for 12.2 percent of all fixed investment, compared to 3.5 percent for computers and peripherals

Software is Also Communicating to Other Systems/Devices



- 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

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
**Woody et al. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=428589)*

Chasing Flaws is Chronic

On “Patch Tuesday” August 11, 2020 Microsoft released updates to address at least 120 vulnerabilities in Windows and other products and services. Two of the flaws are being actively exploited: a memory corruption vulnerability in the scripting engine in Internet Explorer, and a spoofing flaw in Windows file validation that could be exploited to bypass security features. This is the 3rd in a row with over 100.

“The developers are pushing great maintenance cost onto the users but patching seems futile; the number of undetected flaws does not appear to be going down. The tools and processes that we are using for development are inadequate to the task.” Wm Hugh Murray; SANS NewsBites Vol. 20 Num. 065, August 17, 2018

NIST National Vulnerability Database (NVD) contains 148,248 known vulnerabilities – 11,723 were received so far this year [as of 8/15/20]

Examples of Cybersecurity Issues Abound

Academics Find Crypto Bugs in 306 Popular Android Apps, None Get Patched (reported September 8, 2020)

A team of academics from Columbia University has developed a custom tool to dynamically analyze Android applications and see if they're using cryptographic code in an unsafe way. Named **CRYLOGGER**, the tool was used to test 1,780 Android applications, representing the most popular apps across 33 different Play Store categories, in September and October 2019. Researchers say the tool, which checked for 26 basic cryptography rules, found bugs in 306 Android applications. Some apps broke one rule, while others broke multiple.

https://www.zdnet.com/article/academics-find-crypto-bugs-in-306-popular-android-apps-none-get-patched/

Current Development Practices Enable Attackers

The attacker needs 3 elements:

Availability of vulnerabilities

- Millions of lines of software code, which contains defects, 5% are potential vulnerabilities
- Thousands of known software vulnerabilities (see NIST National Vulnerability Database)

Attacker access

- Increased connectivity linking systems to other systems and connecting to new types of devices (IoT)
- Increased system and device remote communication capability with trusted relationships

Attacker capability to exploit

- Attackers have access to the same tools and techniques used to build software
- Reverse engineering can be applied to commercial and open source software to discover weaknesses

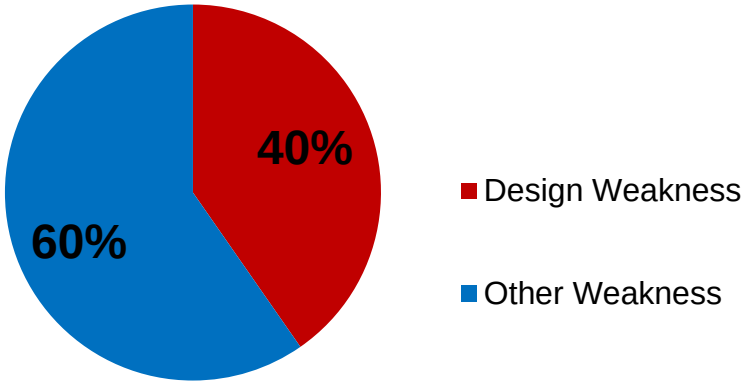
Today: Operations Plays Whac-A-Mole

Winning in features and effectiveness, but losing in defensibility, reliability, and stability

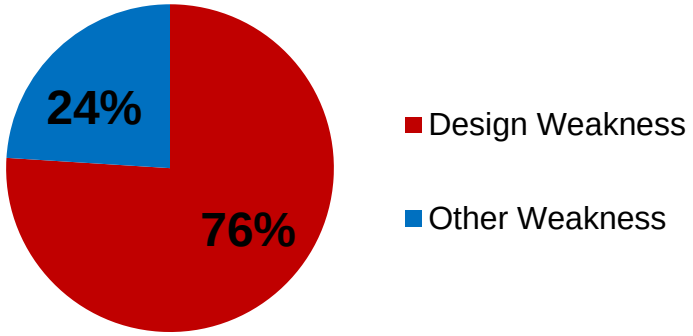
Cybersecurity Engineering is Needed

Design Weakness Challenge

940 Total CWEs*



Top 25 CWEs
(Most Dangerous)



Source: http://cwe.mitre.org/ as of Feb 9, 2014

Causes for design weaknesses:

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

Cybersecurity Requirements Challenges

Typical problems with cybersecurity requirements

- Stated as specific security practices (e.g. Authentication and Authorization) and not real requirements
- Focused narrowly on security mechanisms (e.g. Secure Socket Layer (SSL) for Web communication)
- Compliance mandates (e.g. Risk Management Framework (RMF)) and standards (e.g. ISO/IEC 27001) are substituted for cybersecurity requirements
- Ignored in requirements elicitation because no stakeholders are knowledgeable enough about cybersecurity impacts to state their cybersecurity requirements
- Ignored in trade-off selections and deferred to the selection of controls (compliance mandates) after designs are complete

Cybersecurity Engineering must Addressthe Requirements Gaps

Security Engineering Risk Analysis (SERA)

What is SERA

- Systematic method for analyzing complex cybersecurity risks in software-reliant systems and systems of systems
- Method for considering the operational aspects of a system to identify missing requirements, as well as design and architecture trade-offs with unacceptable mission impacts

Value in Using SERA

- Identifies cybersecurity weaknesses that, when addressed, improve the ability of a system to support mission success
- Assembles a shared operational view (mission and technical perspectives) to connect cybersecurity threats with mission risk

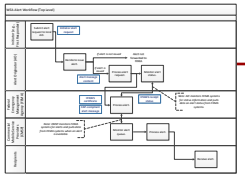
SERA: *Planning for Mission Security Risk*

Outcomes

- **Data disclosure (Confidentiality)**
- **Data modification (Integrity)**
- **Insertion of false data (Integrity)**
- **Destruction of data (Availability)**
- **Interruption of access to data (Availability)**

Affects

Workflow / Mission Thread

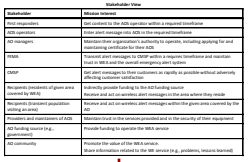


Produces

Adverse Mission Consequences / Losses

Affects

Stakeholder Interests

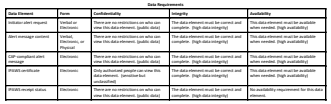


Produces

Adverse Stakeholder Consequences / Losses

Produces

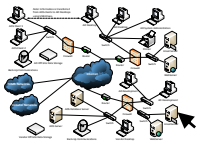
Mission Data



Targets

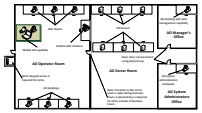
Technology Environment

Technology Infrastructure



Use Cases

Facilities



Entity of Interest

Exploits weaknesses and vulnerabilities

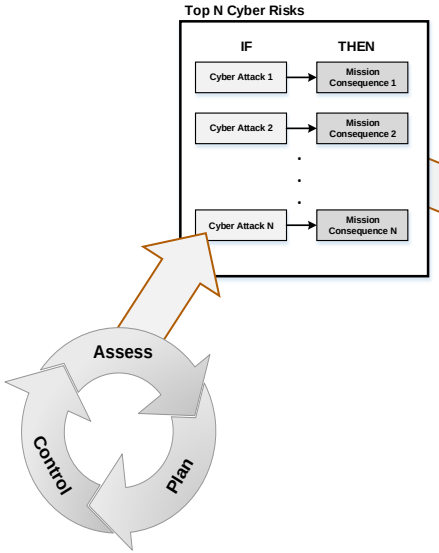
Threat Actor

In SERA, we develop security risk scenarios to analyze the mission impact of data security breaches.

Integration With Established Program Risk Management

Cybersecurity Engineering

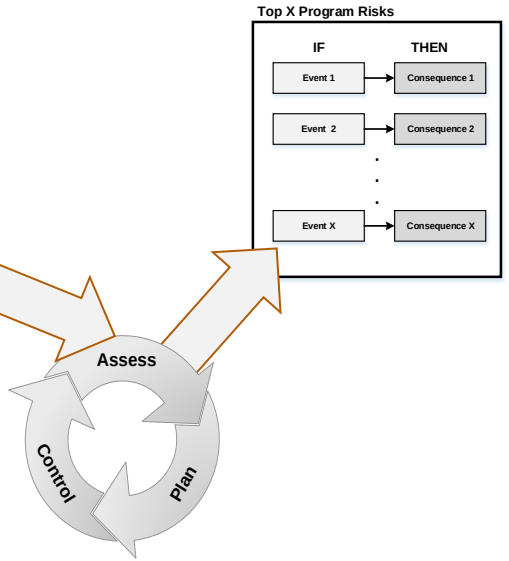
1. Cybersecurity Engineering assesses cyber risks.



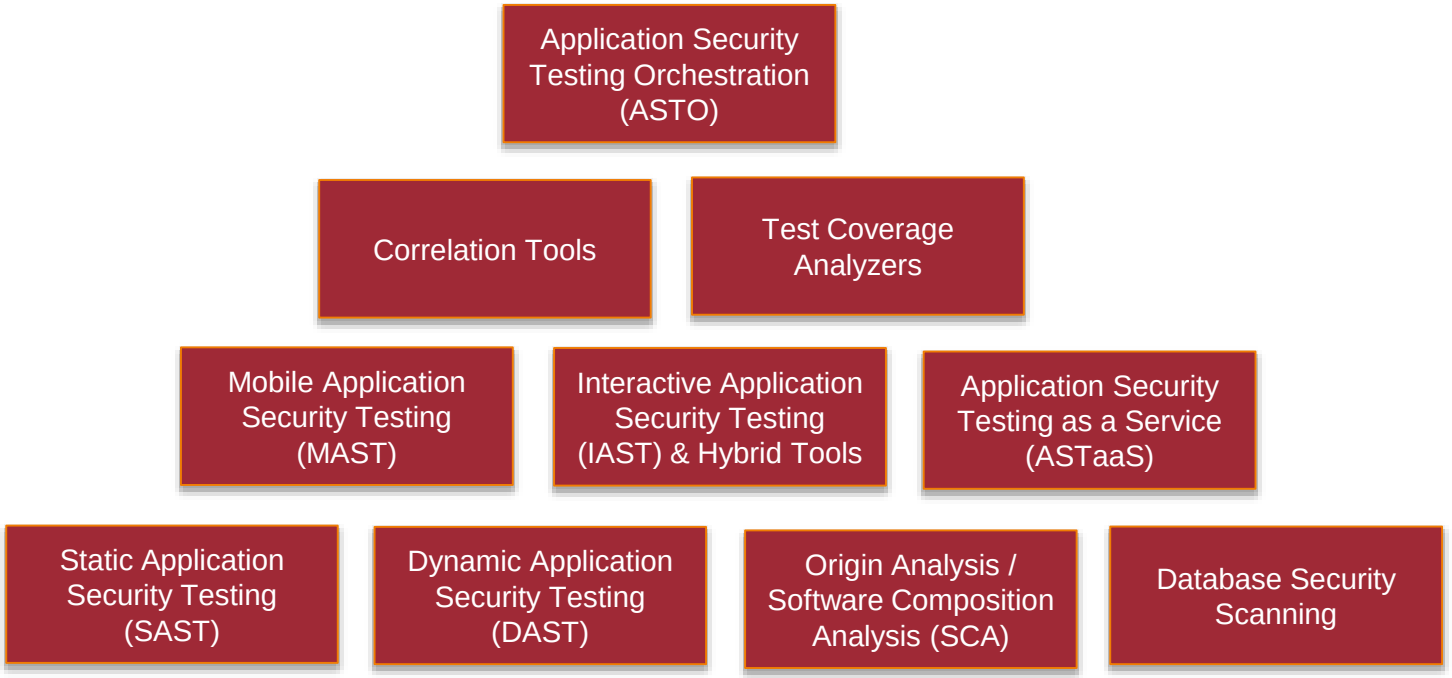
2. Cybersecurity Engineering escalates its top cyber risks to program management.

Program Management

3. Program management assesses cyber risks in relation to other program cost, schedule, and technical risks.



Coding & Implementation Weaknesses: Classes of Automated Security Testing Tools



Reference: State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation
(http://www.acq.osd.mil/se/docs/P-8005-SOAR-2016.pdf)

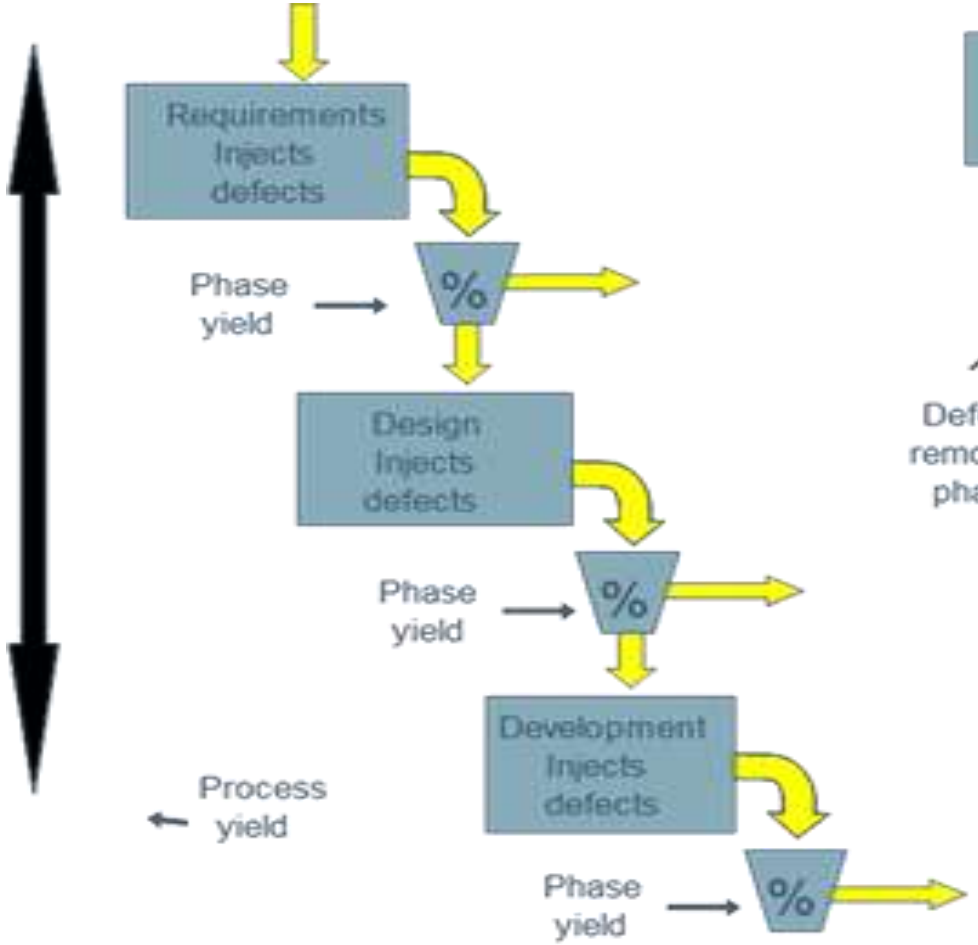
Assurance Measurement - Incremental Growth

Cybersecurity Engineering needs to establish the needed level of assurance (goal) and identify the measures that will be useful in monitoring

- Decompose the goal into sub goals that establish a practice, outcomes and possible metrics (use Goal Question Metric focus - ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf)
- Identify a subset of possible metrics that are already available or can be collected with minimal additional effort to begin assembling data for evaluation
- Add/adjust metrics building on what you are already doing

Woody, Carol; Ellison, Robert; & Ryan, Charles. *Exploring the Use of Metrics for Software Assurance.* CMU/SEI-2018-TN-004. Software Engineering Institute, Carnegie Mellon University. 2019. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=540881

Manage Defect Injection and Removal



Early Defect Removal Across Lifecycle

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

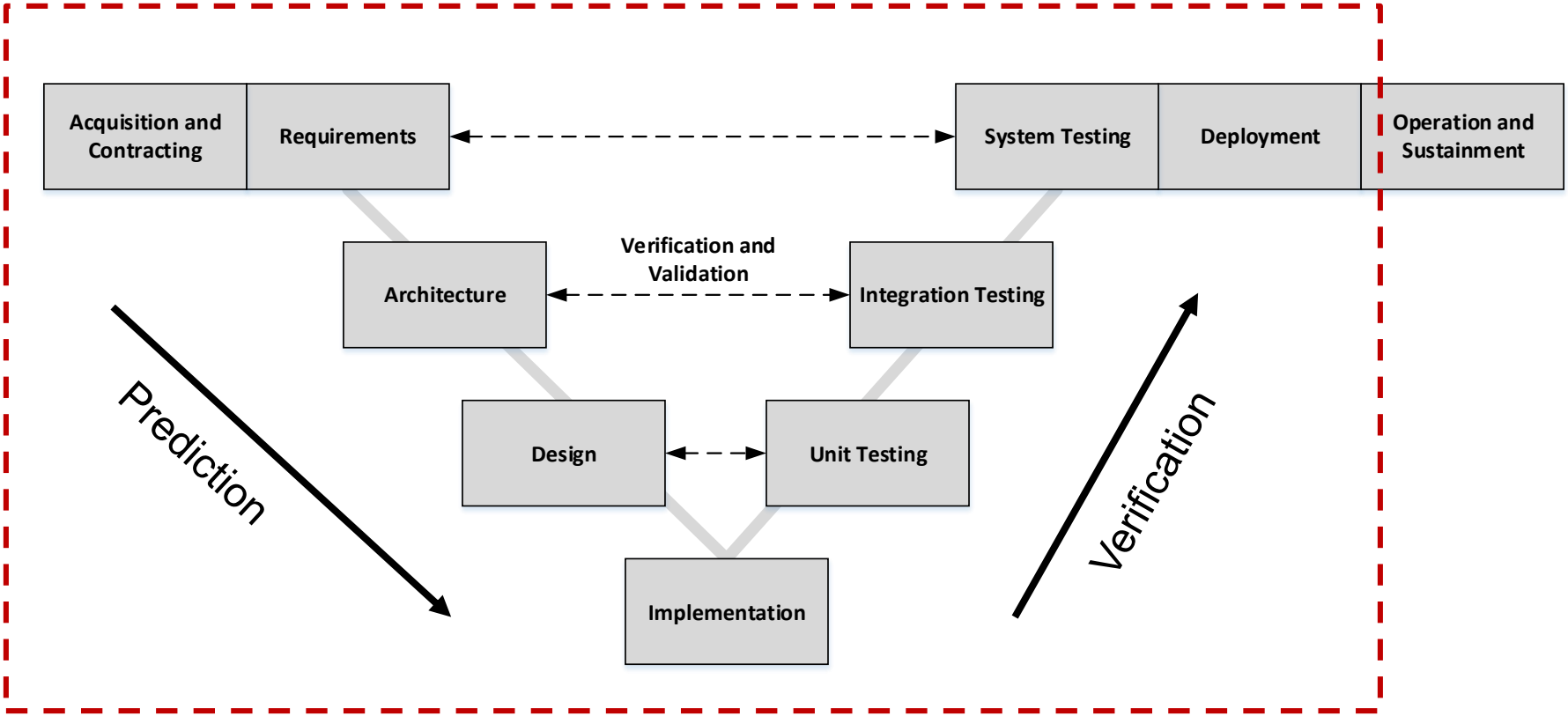
Establish Practices to Support the Goal with SAF

Engineering Best Practices for Software Security

- **Requirements.** Does the program/project define and manage software security requirements?
- **Architecture.** Does the program/project appropriately address security in its software architecture and design?
- **Implementation.** Does the program/project minimize the number of vulnerabilities inserted into the code?
- **Testing, Validation, and Verification.** Does the program/project test, validate, and verify security in its software components?
- **Support Tools and Documentation.** Does the program/project develop tools and documentation to support secure configuration and operation of software components?
- **Deployment.** Does the program/project consider security during the deployment of software components?

Summary

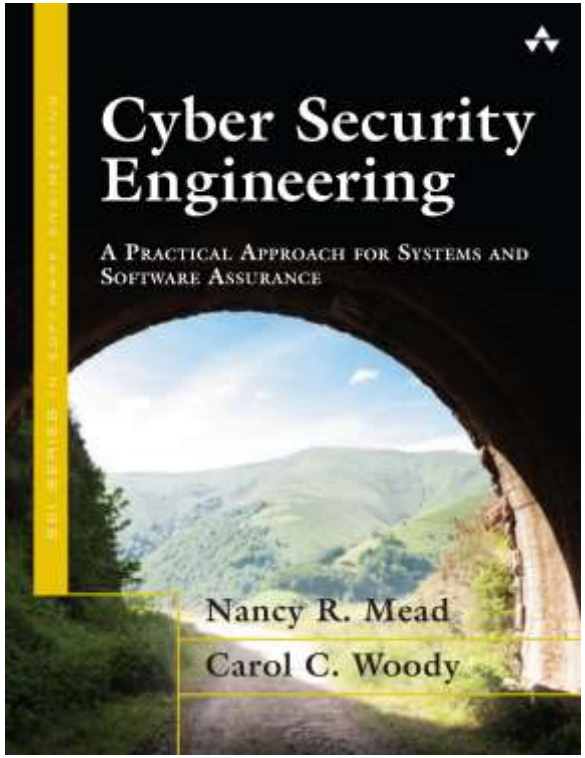
Continuous Focus on Cybersecurity Risk Across the Lifecycle



Current focus: Will the project finish on time and within budget?

Expanded focus: Will the deployed system have sufficient quality and security?

Additional Materials



Released November 2016 as part of the SEI Book Series

CERT Cybersecurity Engineering and Software Assurance Professional Certificate

Released Spring 2018
https://sei.cmu.edu/education-outreach/credentials/credential.cfm?customel_datapageid_14047=33881

Online training in five components

- Software Assurance Methods in Support of Cybersecurity Engineering
- Security Quality Requirements (SQUARE)
- Security Risk Analysis (SERA)
- Supply Chain Risk Management
- Advanced Threat Modeling

Contact Information

Carol Woody, Ph.D.
Software Engineering Institute

cwoody@cert.org

Web Resources

https://sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=48574

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