Managing Supply Chain Complexity with the Acquisition Security Framework (ASF)

Dr. Carol Woody (presenter)
Christopher Alberts
Charles Wallen

December 7, 2021

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.

DM21-0435

Topics

Describing the Context

Acquisition Security Framework Overview

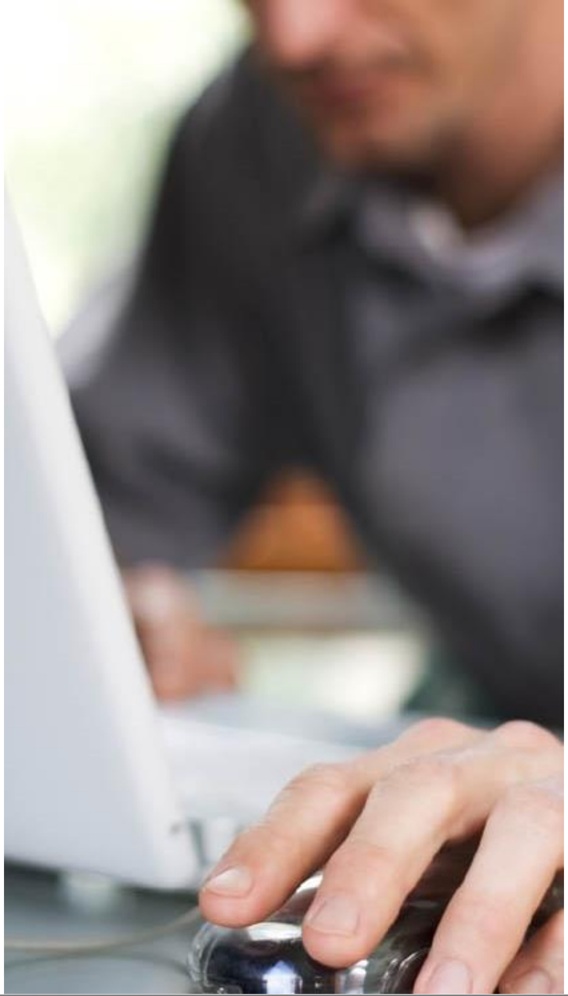
Current Framework Details

Applying the Framework

Summary

Acquisition Security Framework (ASF)

Describing the Context



Challenge: Software is Everywhere

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

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

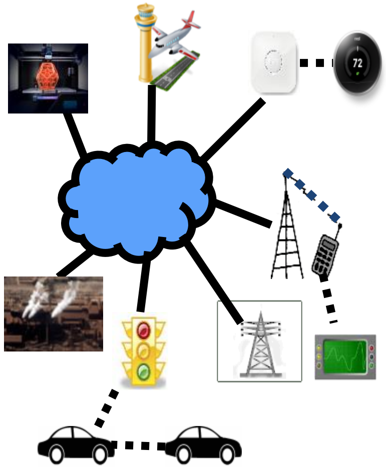
You are getting ***a software platform:***

- Software is a part of almost everything we use.
- Software defines and delivers component and system communication.
- Software is used to build, analyze and secure software.

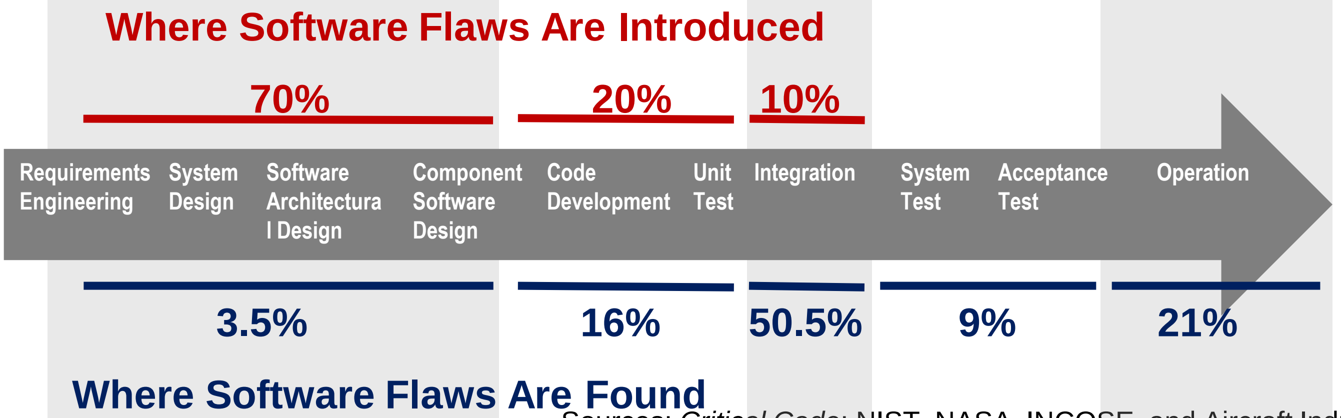
All software has defects:

- Best-in-class code has <600 defects per million lines of code (MLOC).
- Good code has around 1000 defects per MLOC.
- Average code has around 6000 defects per MLOC.

(based on Capers Jones research http://www.namcook.com/Working-srm-Examples.html)



Challenge: Most Software Defects Are Found Long After They Are Introduced



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

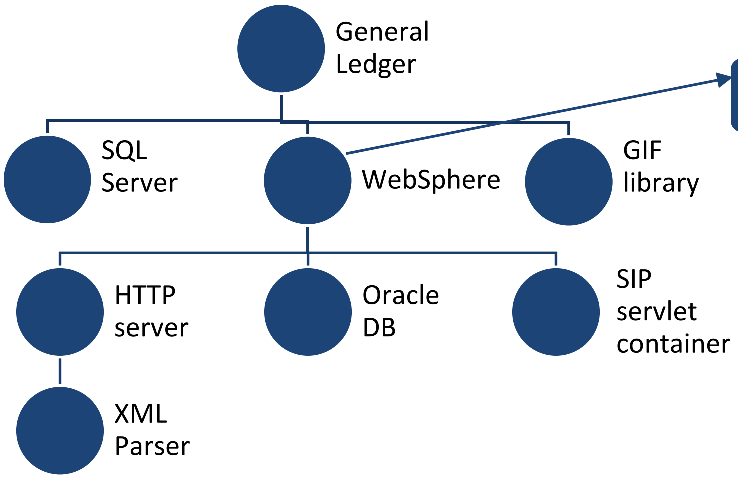
All software code contain defects; up to 5% are vulnerabilities

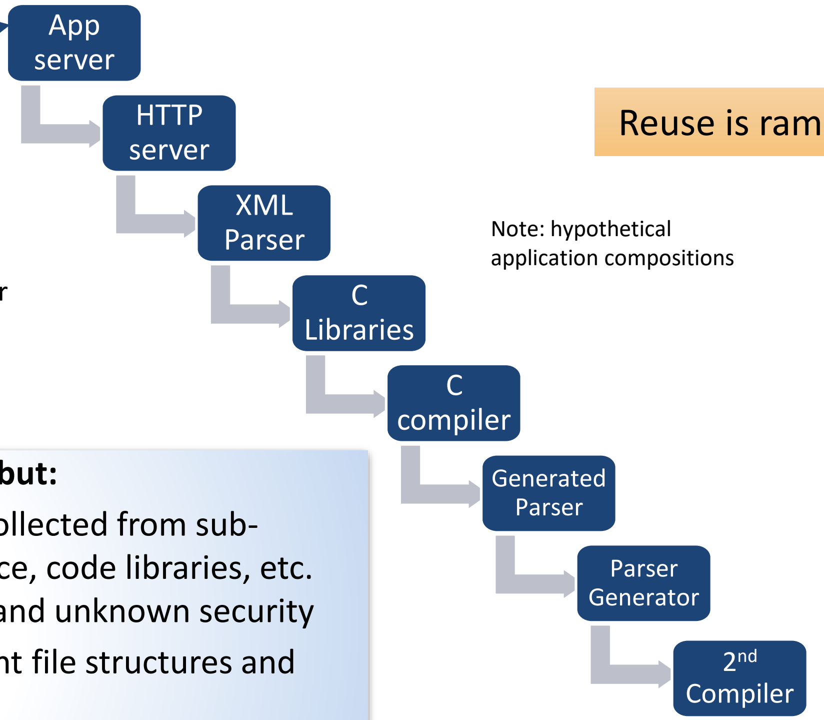
ref: Woody, Carol et al. *Predicting Software Assurance Using Quality and Reliability Measures* http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=428589)

Hundreds of thousands of known software vulnerabilities exist in operations

ref: NIST National Vulnerability Database, https://nvd.nist.gov/general/nvd-dashboard

Software Development is Now Module Assembly





Reuse is rampant!

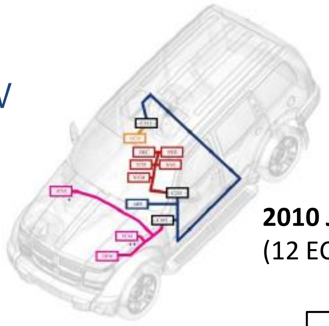
Note: hypothetical application compositions

Delivered product maps to desired functionality, but:

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

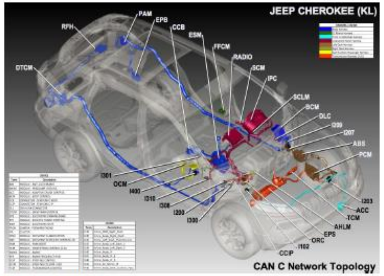
Assembly from 3rd Party Components Reduces Construction Cost/Schedule and Increase Flexibility

Example:
Vehicles are now Assembled from Engine Control Units (ECUs)



2010 Jeep Cherokee
(12 ECUs)

2014 Jeep Cherokee
(32 ECUs)



CAN-C Network – 2014 Jeep Cherokee

Supply Chain Risk Increases Exponentially

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

Modern high-end automotive vehicles have software and connectivity:

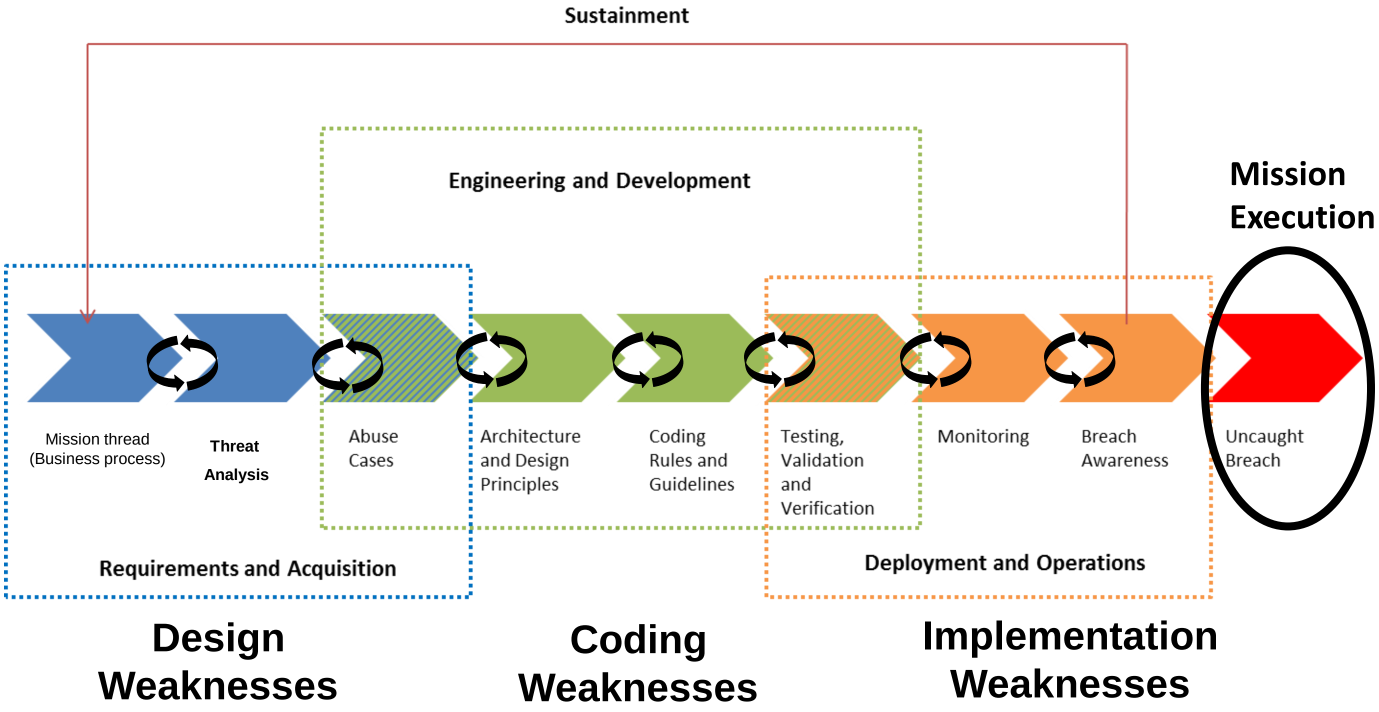
- Over 100 million lines of code
- Over 50 antennas
- Over 100 ECUs

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

Challenge: Major Shifts in Technology Adds Cybersecurity Risk

From…	To…
Hardware-based solution	Software-intensive system
Waterfall methodology	Agile at scale approach
Organization owned infrastructure	Shared infrastructure (e.g. Cloud)
Compliance verification upon completion before fielding (e.g. ATO)	Continuous integrated monitoring (e.g. cATO)
Systems developed from requirements and architectural designs	Systems assembled primarily from reused (often 3rd party) components that map to requirements
Development life cycle tailored to the system under development	DevSecOps Development Factory using 3rd party tools and automation

Cybersecurity and Supplier Risk are Lifecycle Concerns

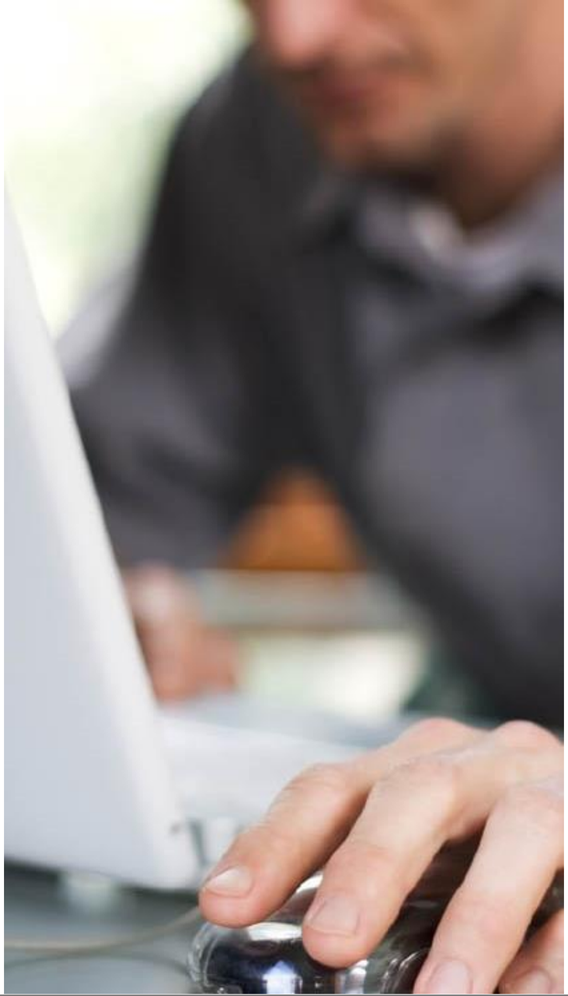


Key Gaps Impacting Cybersecurity and Supply Chain Risk

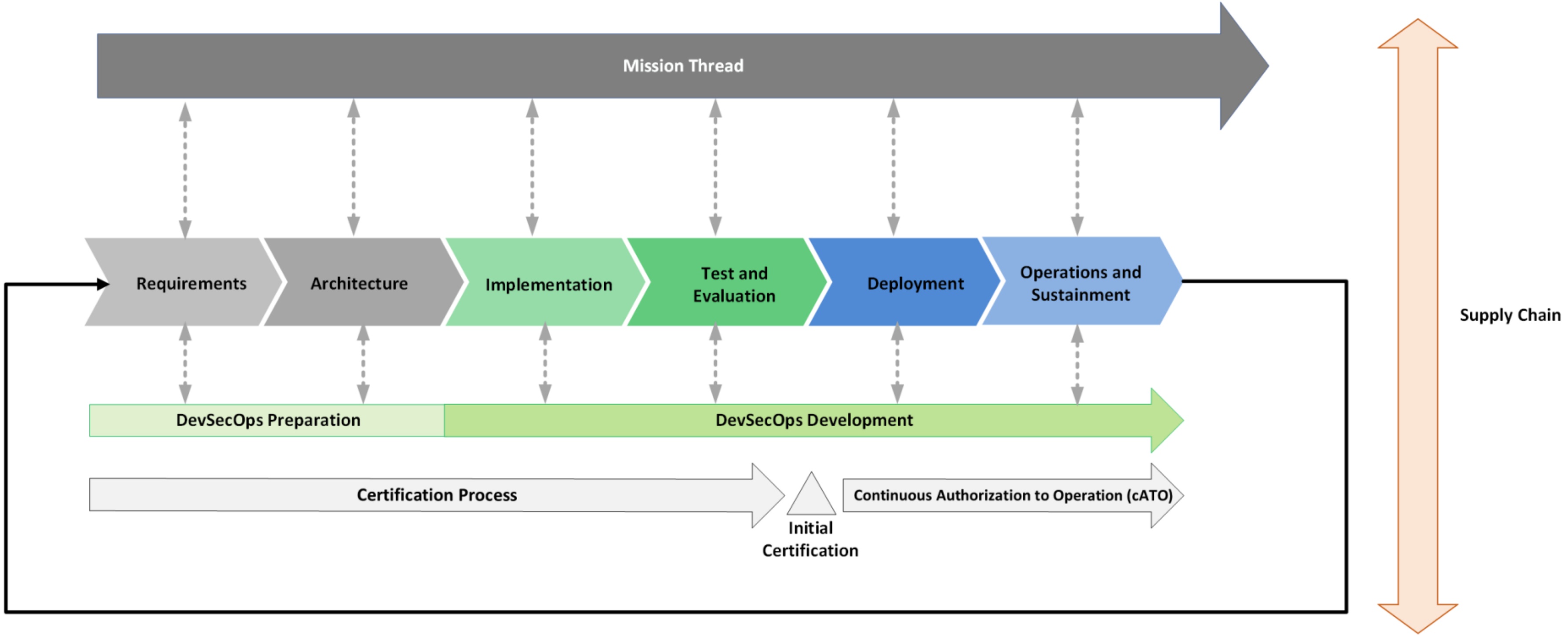
- **System Engineers** frequently decompose the system into its technology components and delegate risk and requirements management
- **Systems Engineers** are not learning from current operational experience
- **System Engineers** often accept risks without understanding the potential mission impacts over the system's lifecycle
- **Program Managers** have not focused on acquisition oversight in the face of growing third party service and product dependencies
- **Program Managers** can define acquisition requirements using standards, guidelines, and controls as a substitute for effective system security requirements

Acquisition Security Framework (ASF)

Acquisition Security Framework (ASF) Overview



Acquisition Security Framework (ASF) Problem Space



ASF Problem Space -2

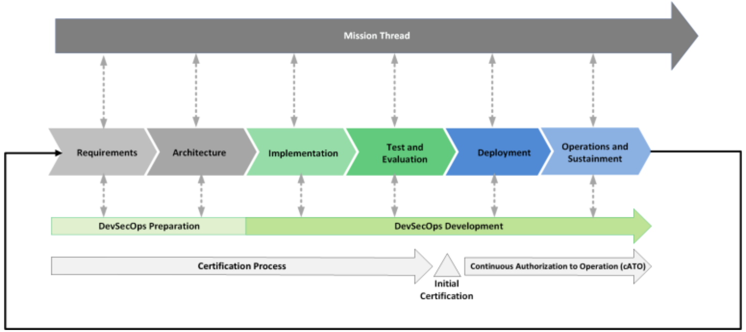
Cybersecurity practices need to be integrated with engineering activities across the systems lifecycle to

- Mitigate acquisition-related security risks
- Implement resilient architectures

Cybersecurity risks must be managed continuously during operations to ensure that evolving security and resilience requirements are met, effectively and efficiently.

- Update software, hardware, and firmware to address security vulnerabilities
- Manage operational security processes to produce consistent results over time

DevSecOps components must be integrated into the systems lifecycle via collaborative process management.



Integrated Security Risk Management

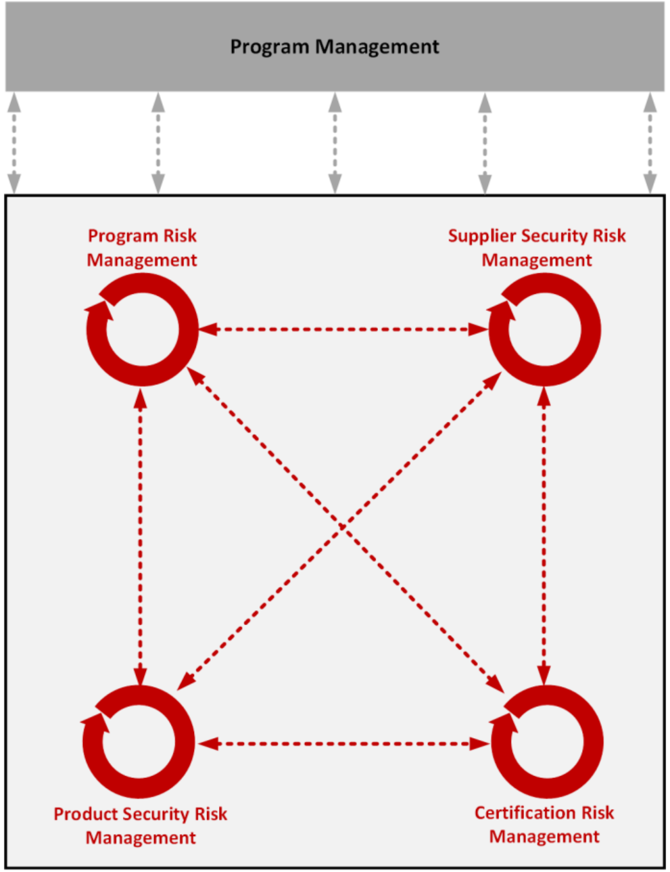
Security risk is managed from multiple perspectives across an acquisition program.

Leadership roles and coordination points change and evolve throughout the lifecycle.

Risk identification, prioritization, and escalation must be ongoing by all areas From all perspectives

The program's risk management strategy defines how

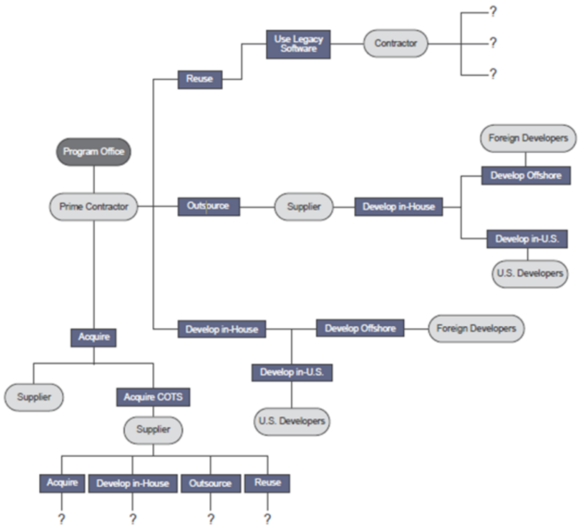
- Groups manage risk collaboratively
- Technology and security gates support security risk objectives



Aligning and Managing Security Objectives -1

Each organization/program unit addresses security from a different perspective:

- Mission Thread
  - Focus: Assuring mission success
- Acquisition and Development
  - Focus: Build security into the software-reliant system
- Operations and Sustainment
  - Focus: Protection and sustainment of the system
- Certification
  - Focus: Certify systems for deployment

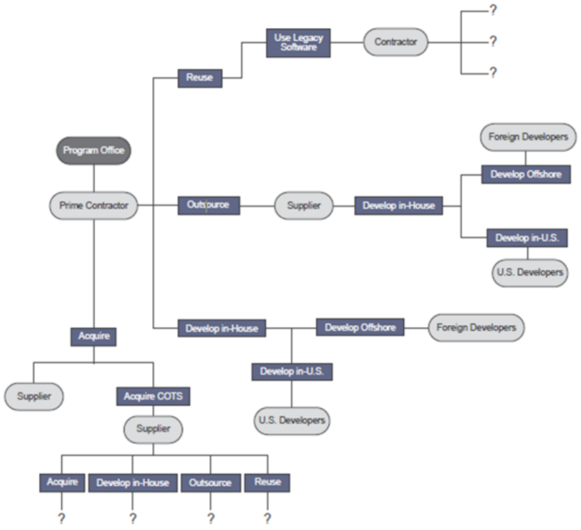


Security objectives across organizations/program units need to be aligned and managed.

Aligning and Managing Security Objectives -2

ASF facilitates the alignment of shared foundational program objectives to support:

- Governance of program management, suppliers, controls, compliance, and certification
- Process management and improvement to monitor
  - Ongoing changes in security posture
  - Program security effectiveness and efficiency
- Risk management and disposition strategies



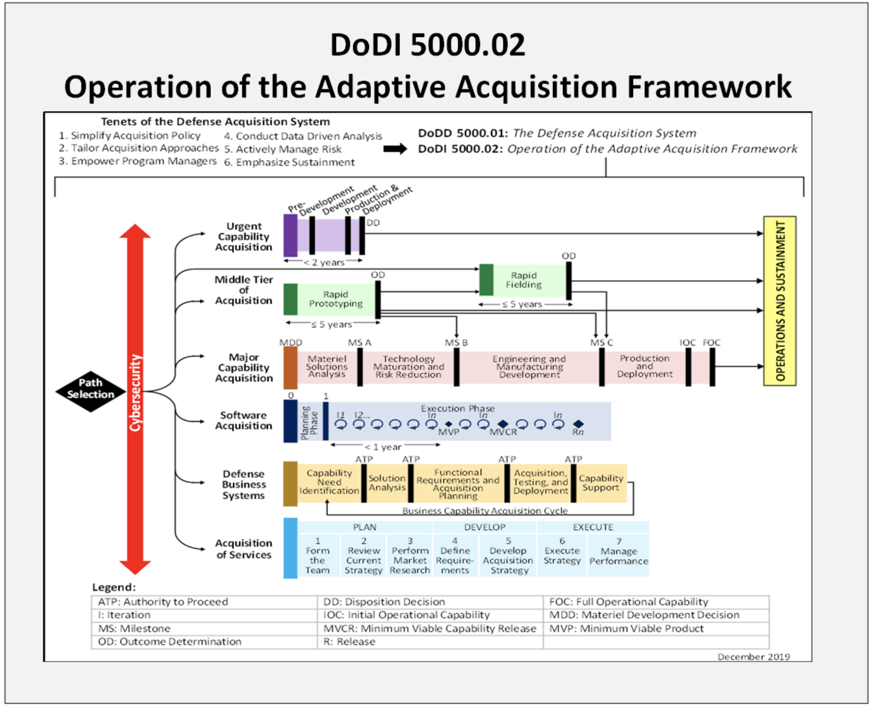
Integrating Security into Acquisition and Engineering

Security practices (management and technical) need to be integrated into a program's existing acquisition and engineering practices.

Security practices and processes (management and technical) need to scale to multiple types of acquisitions, including

- Major capability acquisition
- Software acquisition
- Defense business systems
- Acquisition of services

Security practices and processes must scale to specific development approaches, such as DevSecOps.



Process Management and Improvement

Higher degrees of process management translate to more stable environments that

- Produce consistent results over time
- Are able to achieve their missions during times of stress

Each organization/program unit must manage the maturity of its security practices.

Security practices do not need to be at a uniform level of maturity to be sufficient.

Acquisition Security Framework (ASF) Task: Goals

Integrate software security engineering practices into the acquisition lifecycle

- Define a risk-based framework that supports
  - Security engineering across the lifecycle and supply chain
  - Complexity through integrated process management
- Integrate lessons learned from successful supply chain attacks (e.g., malware, ransomware, denial of service)
- Incorporate DevSecOps concepts and principles[1]
- Adapt system and software engineering measurement activities to include security where appropriate, especially in early lifecycle activities
- Ensure consistency with DoD policies, such as DoDI 5000.02, *Operation of the Adaptive Acquisition Framework*.

1. As defined in Woody, C.; Chick, T.; Reffett. A.; Pavetti, S.; Laughlin, R.; Frye, B.; & Bandor, M. "DevSecOps Pipeline for Complex Software-Intensive Systems: Addressing Cybersecurity Challenges." *Journal of Systemics, Cybernetics and Informatics*. Volume 1. Number 5. (ISSN: 1690-4524) 2020. pp. 31-36.

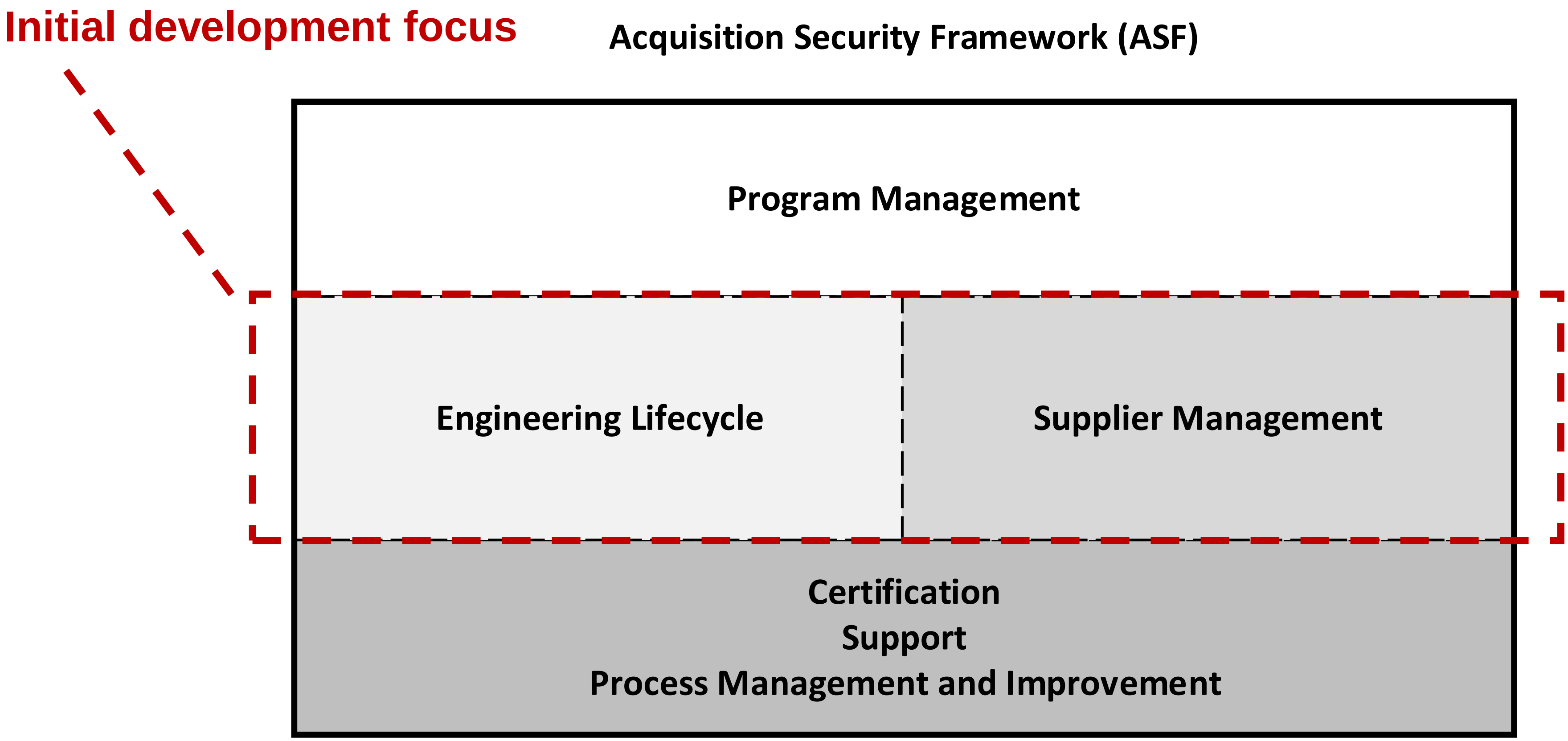
What is the ASF?

The ASF structures a collection of cybersecurity practices that an acquisition program should perform when acquiring a secure and resilient software-reliant system into the areas that need to ensure they are performed:

- Program Management
- Engineering Lifecycle
- Supplier Management
- Certification
- Support
- Process Management and Improvement

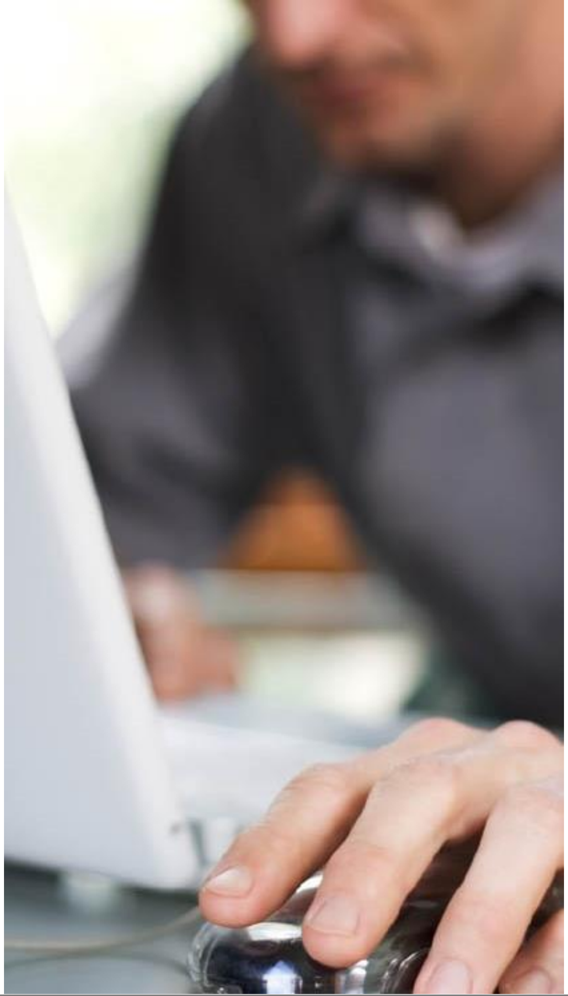
The framework enables programs to identify gaps when acquiring, engineering, and operating secure, resilient software-reliant systems.

ASF Structure



Acquisition Security Framework (ASF)

Current Framework Details



Engineering Lifecycle

Domain	Key Concepts
Engineering Infrastructure	Infrastructure Development Infrastructure Operation and Sustainment
Engineering Activities	Product Risk Management Requirements Architecture Third-Party Components Implementation Test and Evaluation Transition Artifacts Deployment Secure Product Operation and Sustainment

Example: Architecture

Goal—Cybersecurity risks in the architecture and design are identified and mitigated.

The purpose of this goal is to identify and mitigate security risks resulting from in the system's architecture and detailed design.

1. Is a process for performing security risk analysis of the architecture and detailed design defined?
2. Are identified security risks in the architecture and detailed design addressed?
3. Has an architecture tradeoff analysis of quality attributes, including security, been performed?
4. Have security risks resulting from architecture tradeoffs been communicated to stakeholders?
5. Has the architecture's attack surface been minimized based on the results of an attack-path analysis?
6. Is a cross check of the architecture and detailed design performed to resolve any issues or inconsistencies in security features?
7. Are security requirements updated periodically to reflect security changes to the architecture or detailed design?
8. Are reviews conducted with stakeholders to ensure that security risks in the architecture and detailed design are mitigated sufficiently?

Example: Third-Party Components

Goal—Security vulnerabilities in third-party components (TPCs) are identified and mitigated.
The purpose of this goal is to develop a bill of materials (BOM) for a product and ensure that operational security risks in the third-party software, firmware, and hardware are managed over time.

1. Are engineering relationships with third parties based on standards, guidelines, and policies?
2. Is an identification scheme that uniquely identifies each third-party component (TPC) implemented?
3. Is a repository to track TPC usage in products implemented and maintained?
4. Is a process defined for identifying the TPCs used in a product to create a bill of materials (BOM)?
5. Are suppliers evaluated and selected for their use of secure development practices?
6. Is a process defined for assessing a TPC's operational risk?
7. Are TPCs monitored for vulnerabilities and available patches?
8. Are TPCs prioritized for patch application based on operational risk?

Example: Implementation

Goal—Vulnerabilities in software code are identified, managed, and tracked.

The purpose of this goal is to identify and address vulnerabilities and security issues in the code base.

1. Is an appropriate suite of security tools integrated into the software development environment?
2. Are secure coding standards and practices applied?
3. Are code reviews (e.g., peer reviews) performed to identify weaknesses and vulnerabilities?
4. Is source code in critical components analyzed using white-box testing (e.g., static code analysis) during coding and unit testing to identify weaknesses and vulnerabilities?
5. Is software in critical components analyzed using black-box testing (e.g., dynamic code analysis, vulnerability scanning) during integration testing to identify weaknesses and vulnerabilities?
6. Are coding weaknesses and vulnerabilities remediated and tracked to resolution?

Supplier Dependency Management

Domain	Key Concepts
Relationship Formation	Establishing supplier relationships is planned Formal agreements include resilience requirements Supplier are evaluated Managing supplier risk
Relationship Management	Suppliers are identified and prioritized Supplier performance is governed and managed Supplier risk management is continuous Change and capacity management are applied to suppliers Supplier access to program or system assets is managed Infrastructure and governmental dependencies are managed Supplier transitions are managed
Supplier Protection and Sustainment	Disruption planning includes suppliers Planning and controls are maintained and updated Situational awareness extends to suppliers

Supplier Dependency Management – Example - 1

Domain 1. - Relationship Formation

Goal 1– Establishing supplier relationships is planned.

The purpose of this goal is to assess whether processes are in place to enter into relationships and formal agreements with suppliers.

1.	Does an established process exist for entering into formal agreements with suppliers? [EXD:SG3.SP3]*

Goal 2 – Formal agreements include resilience requirements.

The purpose of this goal is to assess whether supplier agreements include resilience/security requirements.

1.	Are resilience requirements included in formal agreements with suppliers? [EXD:SP3.SP4]

* References the CERT Resilience Management Model. The naming format is: Domain:Goal:Practice.

Supplier Dependency Management – Example - 2

Domain 3. - Supplier Protection and Sustainment

Goal 1 – Disruption planning includes suppliers.

The purpose of this goal is to assess whether the program or system accounts for suppliers as part of its incident management and service continuity processes.

2.	Have incident declaration criteria that support the program or system been established and communicated to relevant suppliers? [IMC:SG3.SP1, IMC:GG2.GP7]

* References the CERT Resilience Management Model. The naming format is: Domain:Goal:Practice.

Next Steps

Preparing two areas of practice for broader distribution and review:

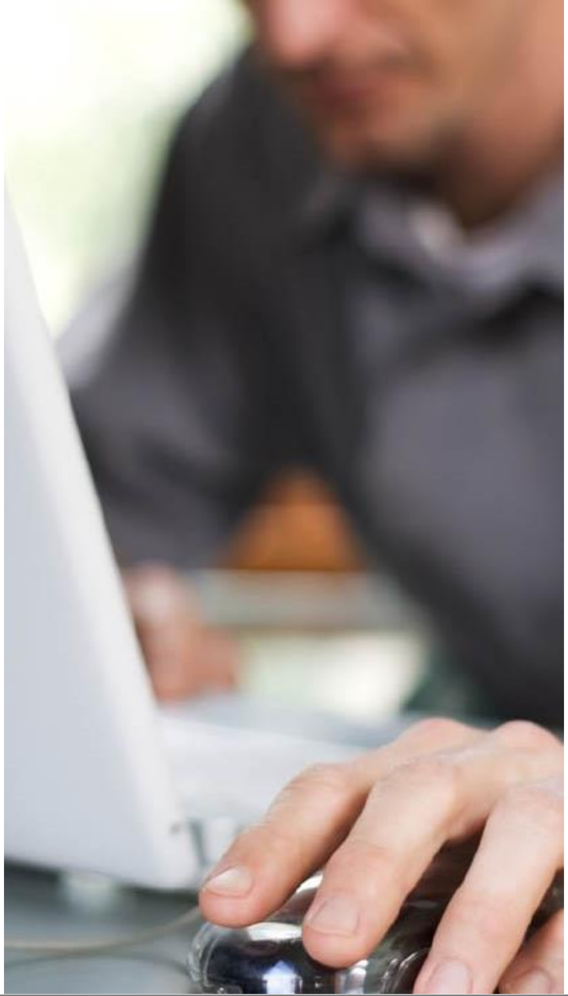
- Engineering Lifecycle
- Supplier Management

Drafting next area: Program Management

Exploring pilot opportunities with DoD programs to apply published practices for gap analysis

Acquisition Security Framework (ASF)

Summary



Barriers to Effective Management

Complexity

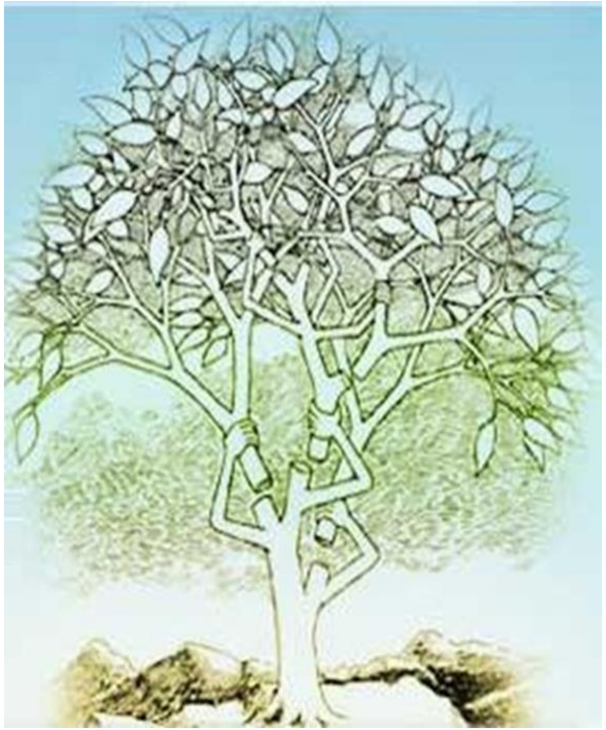
Siloed departments operating under different requirements

- Procurement/acquisitions
- Operations
- Incident management

Vagueness or limitations in formal agreements

Changing requirements across system lifecycles

Incomplete or narrow risk management processes



Acquisition Security Framework Approach

Integrate cybersecurity practices with engineering activities across the systems lifecycle to

- Mitigate acquisition-related security risks
- Implement resilient architectures

Continuously manage cybersecurity risks during operations

Integrate DevSecOps components into the systems lifecycle via consistent and collaborative process management

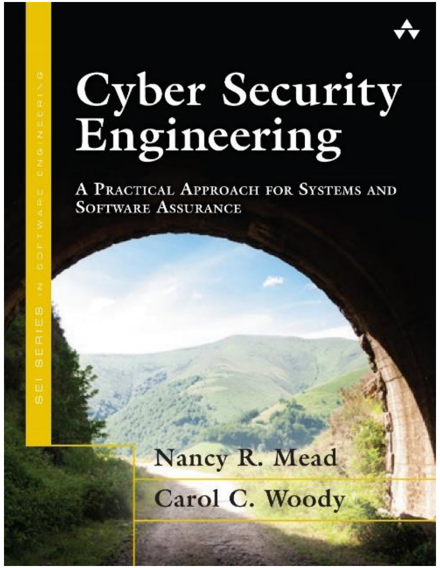
Ensure consistency with DoD policies, such as

- DoDI 5000.02, *Operation of the Adaptive Acquisition Framework*
- DoDI 5000.83, *Technology and Program Protection to Maintain Technological Advantage*

Opportunities to Learn More

Textbook

Cybersecurity Engineering



SEI Book Series

Professional Certificate

CERT Cybersecurity Engineering and Software Assurance

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.

cwoody@cert.org

Web Resources

Building security into application lifecycles

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

CMU SEI Home Page

https://sei.cmu.edu/