A Cybersecurity Engineering Strategy for DevSecOps

Carol Woody, Ph.D.
Principal Researcher

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Topics

Challenges for Cybersecurity in DevSecOps

DevSecOps Pipeline Supports Critical Cybersecurity Requirements

Managing Supply Chain Risk for DevSecOps

Cybersecurity Strategy is Key to Success

Final Thoughts

A Cybersecurity Engineering Strategy for DevSecOps

Challenges for Cybersecurity in DevSecOps

Major Shifts in Technology Will Add 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

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	

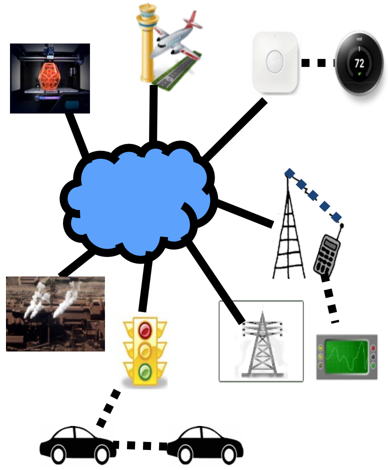
Actually you're getting ***a software platform:***

- 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.

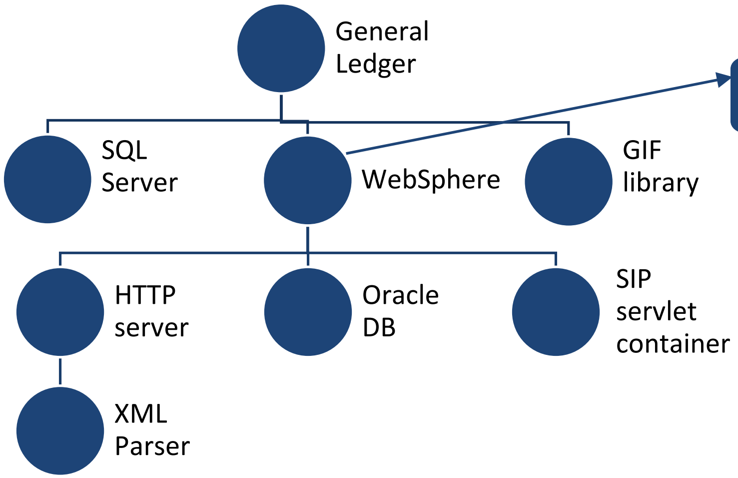
All software has defects:

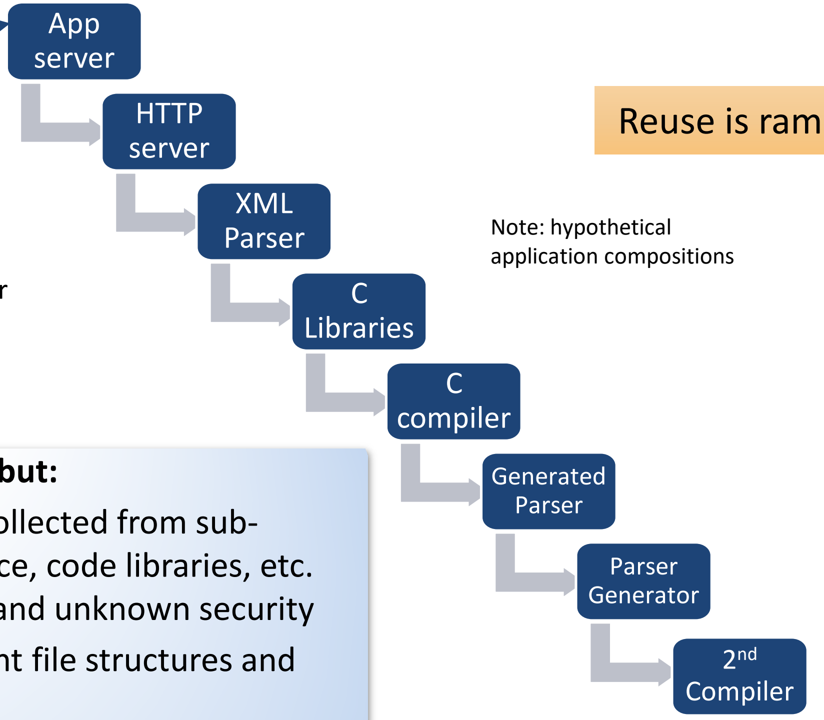
- Best-in-class code has <600 defects per million lines of code (MLOC).
- Good code has around 1000 defects per MLOC.
- Average code has around 6000 defects per MLOC.

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



Software Development is Now Module Assembly





Reuse is rampant!

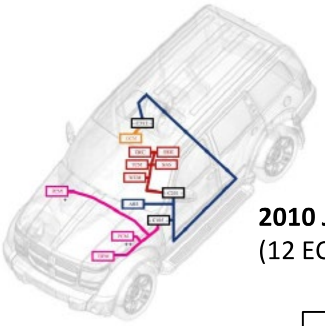
Note: hypothetical application compositions

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

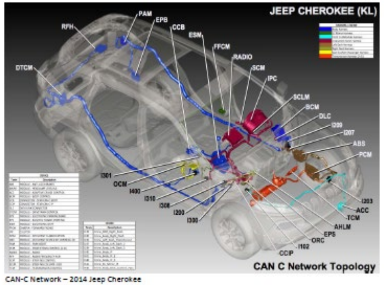
Assembly from 3rd Party Components Reduces Construction Cost/Schedule and Increase Flexibility

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



2010 Jeep Cherokee
(12 ECUs)

2014 Jeep Cherokee
(32 ECUs)



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

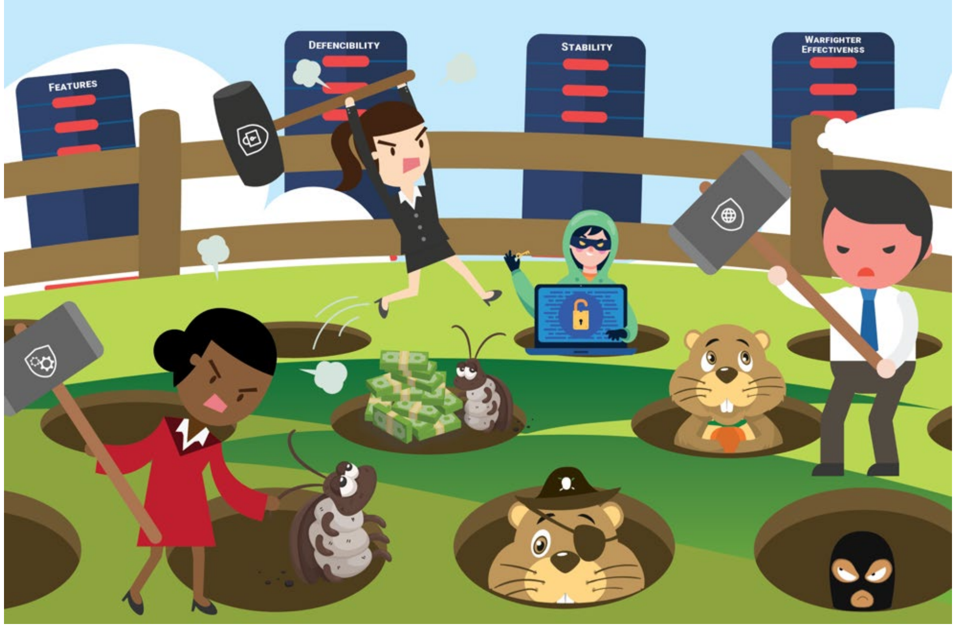
Sources: Miller and Valasek, A Survey of Remote Automotive Attack Surfaces, http://illmatics.com/remote%20attack%20surfaces.pdf; https://www.cst.com/webinar14-10-23~?utm_source=rfg&utm_medium=web&utm_content=mobile&utm_campaign=2014series
https://en.wikipedia.org/wiki/Electronic_control_unit

Chasing Vulnerabilities is a Chronic Activity for 3rd Party Code

The National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) contains **172,822 known vulnerabilities** – NVD received **16,190 new vulnerabilities in 2021** (as of 10/23/21).

- **Nearly Three-Quarters of Organizations Victimized by DNS Attacks in Past 12 Months** Domain name system (DNS) attacks are impacting organizations at worrisome rates. According to a new survey from the Neustar International Security Council (NISC) conducted in September 2021
- **North American Orgs Hit with an Average of 497 Cyberattacks Per Week**
  https://www.darkreading.com/attacks-breaches/north-american-orgs-experience-497-attacks-per-week-on-average-currently
- **Surge in Ransomware Incidents** Allianz Global Corporate & Specialty (AGCS) report analyzes the latest risk developments around ransomware. there was a 62% increase in ransomware incidents in the US in the same period that followed an increase of 20% for the full year 2020. https://www.helpnetsecurity.com/2021/10/18/five-ransomware-trends/

Today, Operations Plays Whack-a-mole Chasing Attacks

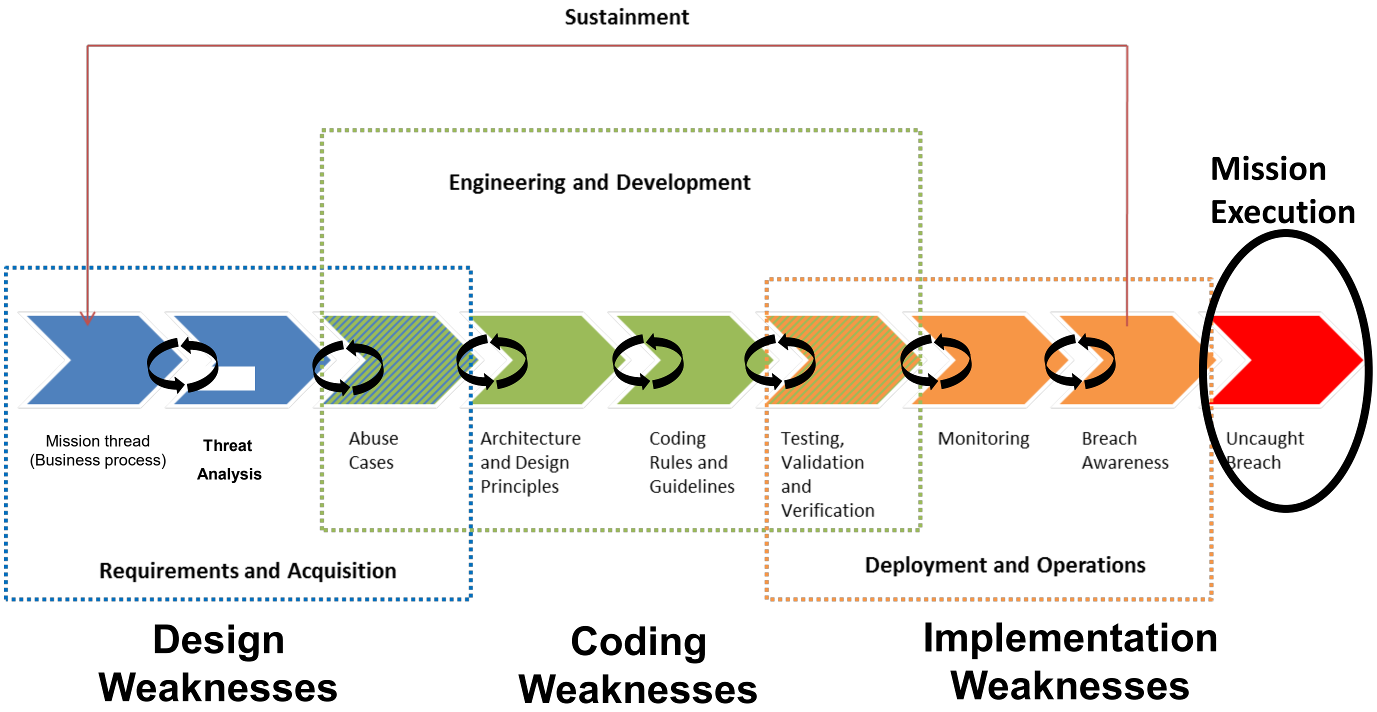


Rapid delivery of features is prioritized over defensibility, reliability, and stability.

Operational missions are jeopardized by weak designs that allow attackers to leverage the many vulnerabilities.

Once software's in an operational system, vulnerabilities can be difficult (or impossible) to mitigate.

Cybersecurity Should be a Lifecycle Effort

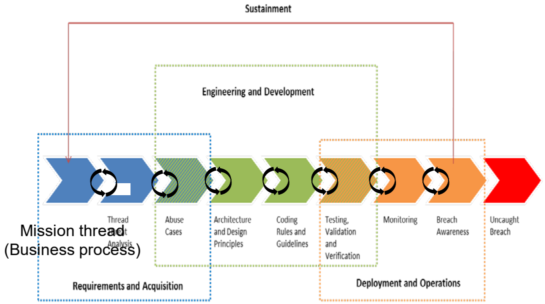


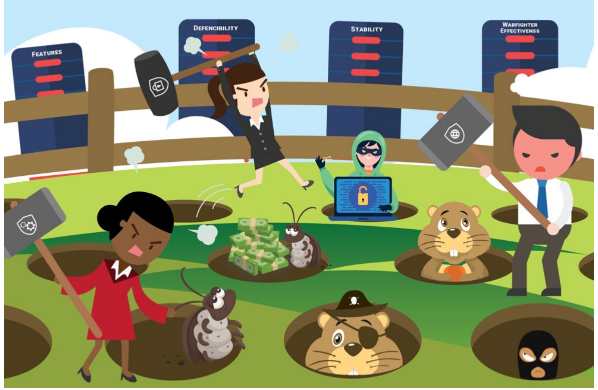
Testing (incomplete at best) verifies requirements and tools (costly with limited capabilities) look for weaknesses and vulnerabilities

Emerging Critical Needs

How can we confirm the DevSecOps pipeline is meeting our cybersecurity needs?

How can we effectively manage the supply chain risks that 3rd party code introduces?



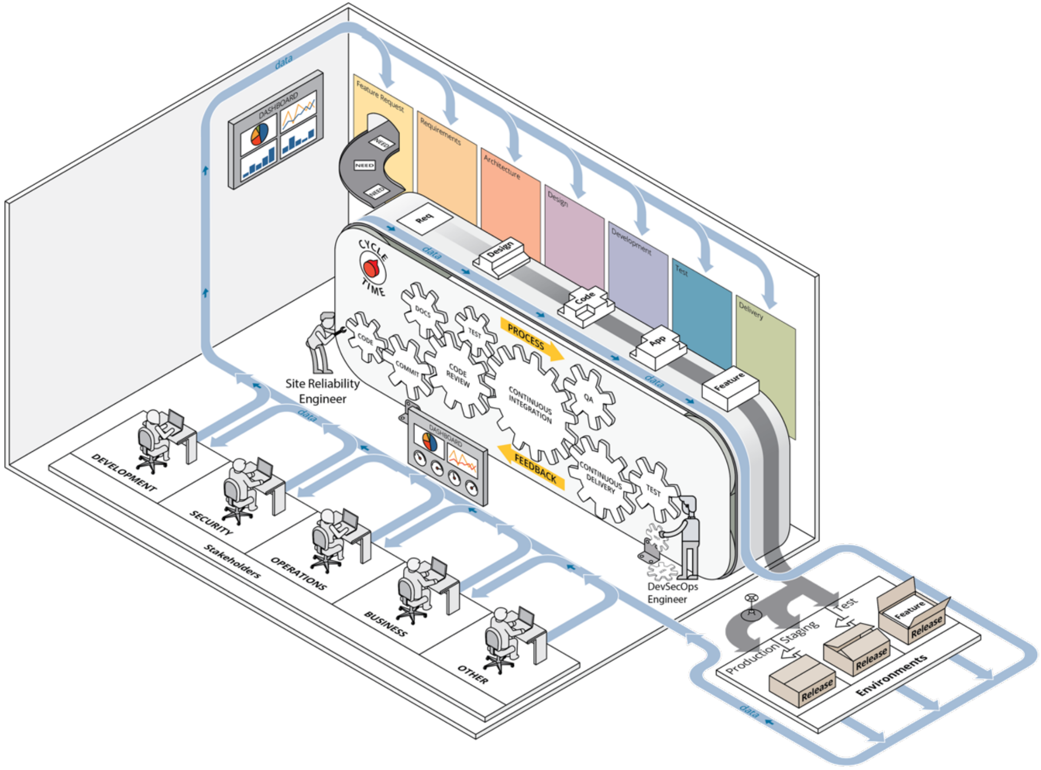


A Cybersecurity Engineering Strategy for DevSecOps

DevSecOps Pipeline Supports Critical Cybersecurity Requirements

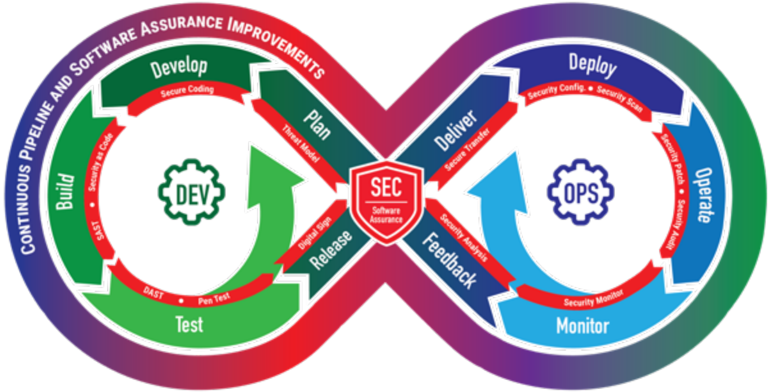
What is DevSecOps?

A cultural and engineering practice that breaks down barriers and opens collaboration between development, security, and operations organizations using automation to focus on rapid, frequent delivery of secure infrastructure and software to production. It encompasses intake to release of software and manages those flows predictably, transparently, and with minimal human intervention/effort [1].



[1] DevSecOps Guide: Standard DevSecOps Platform Framework. U.S. General Services Administration. https://tech.gsa.gov/guides/dev_sec_ops_guide. Accessed 17 May 2021.

A DevSecOps Pipeline is a System that Must be Engineered



The DevSecOps pipeline (DSO) is a socio-technical system composed of both software tools and processes. As the capability matures, it can seamlessly integrates three traditional factions that sometimes have opposing interests:

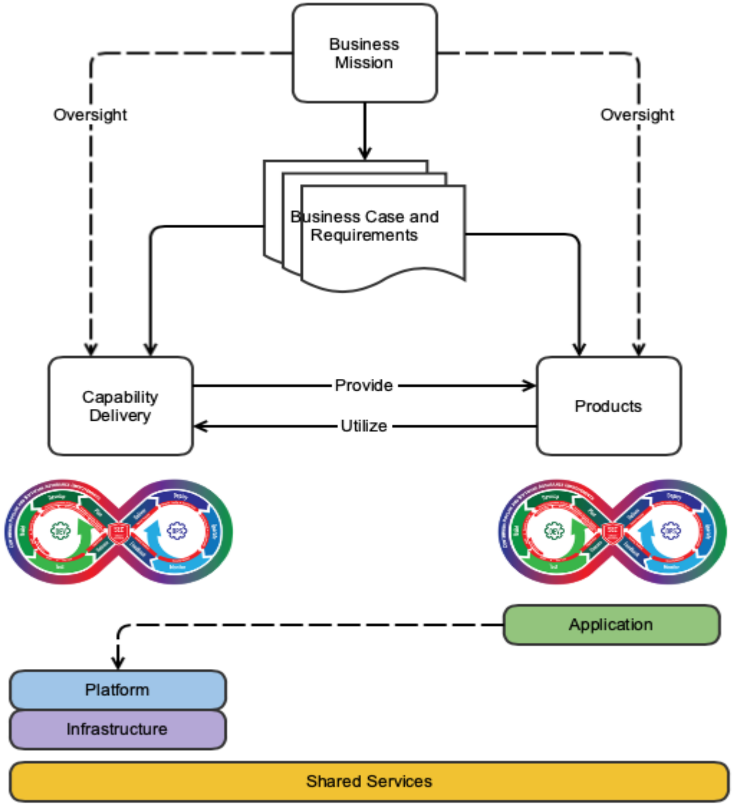
- development; which values features
- security, which values defensibility
- operations, which values stability

A DevSecOps pipeline emerges when continuous integration of these three factions is used to meet organizational, project, and team objectives and commitments.

DevSecOps Maturity Levels

Term	Documentation
Maturity Level 1	Performed Basic Practices: This represents the minimum set of engineering, security, and operational practices that is required to begin supporting a product under development, even if only performed in an ad-hoc manner with minimal automation, documentation, or process maturity. This level is focused on minimal development, security, and operational hygiene.
Maturity Level 2	Documented/Automated Intermediate Practices: Practices are completed in addition to meeting the level 1 practices. This level represents the transition from manual, ad-hoc practices to the automated and consistent execution of defined processes. This set of practices represents the next evolution of the maturity of the product under development's pipeline by providing the capability needed to automate the practices that are most often executed or produce the most unpredictable results. These practices include defining processes that enable individuals to perform activities in a repeatable manner.
Maturity Level 3	Managed Pipeline Execution: Practices are completed in addition to meeting the level 1 and 2 practices. This level focuses on consistently meeting the information needs of all relevant stakeholders associated with the product under development so that they can make informed decisions as work items progress through a defined process.
Maturity Level 4	Proactive Reviewing and Optimizing DevSecOps: Practices are completed in addition to meeting the level 1-3 practices. This level is focused on reviewing the effectiveness of the system so that corrective actions are taken when necessary, as well as quantitively improving the system's performance as it relates to the consistent development and operation of the product under development.

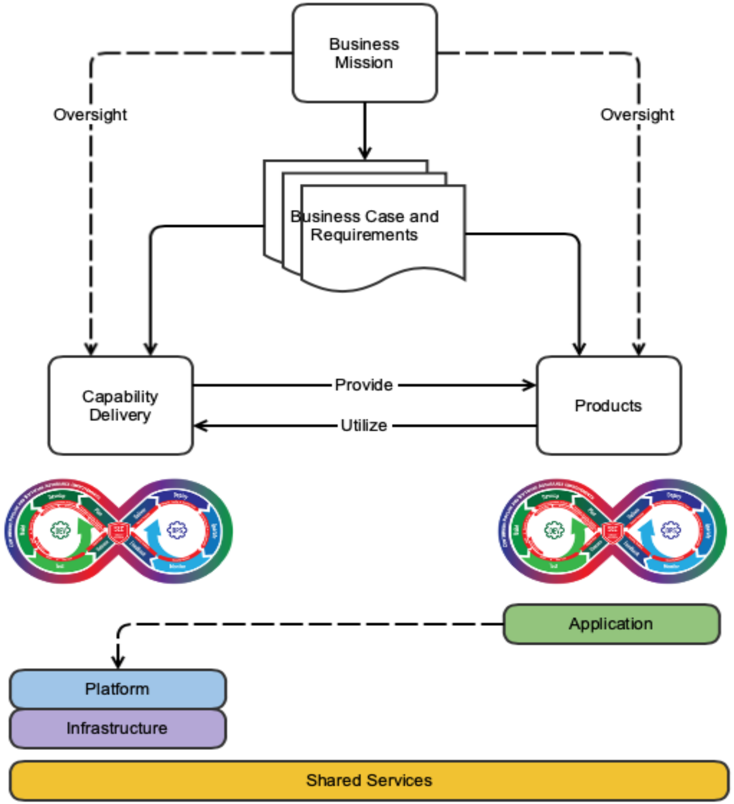
Challenge 1 for DSO: connecting process, practice, & tools



Creation of the DevSecOps (DSO) pipeline for building the product is not static.

- Tools for process automation must work together and connect to the planned infrastructure
- Everything is software and all pieces must be maintained but responsibility will be shared across multiple organizations (Cloud for infrastructure, 3rd parties for tools and services

Challenge 2 for DSO: cybersecurity of pipeline and product

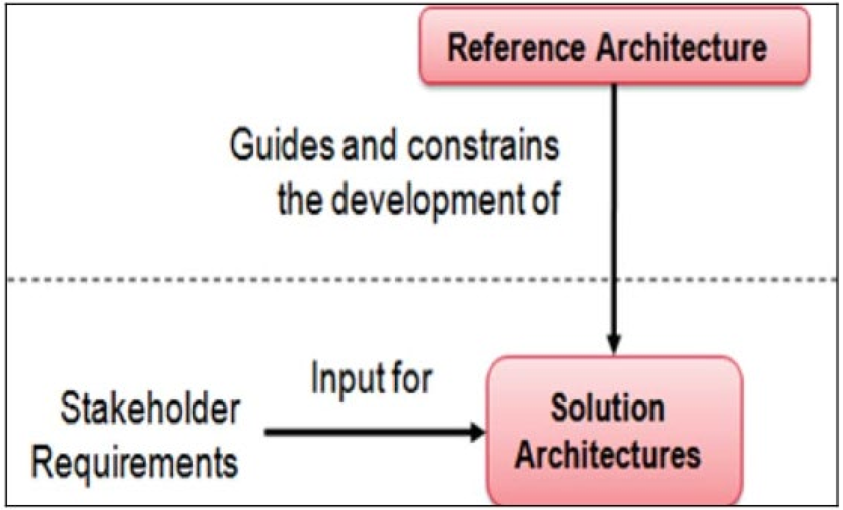


Managing and monitoring all of the various parts to ensure the product is built with sufficient cybersecurity and the pipeline is maintained to operate with sufficient cybersecurity is complex. Cybersecurity demands effective governance to address:

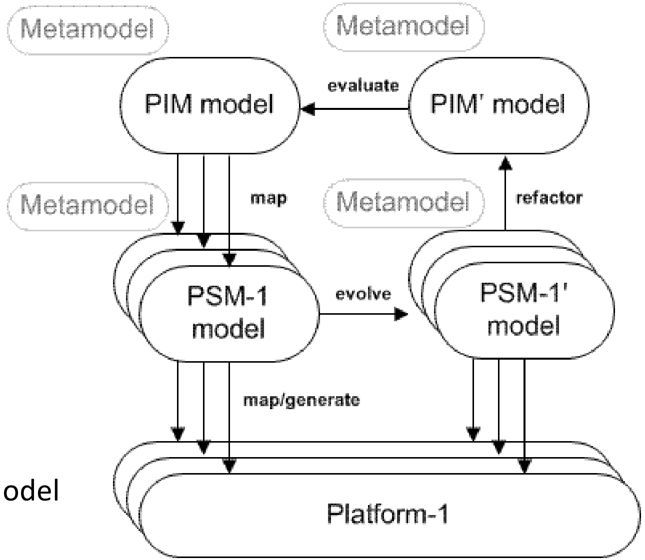
- What trust relations will be acceptable, and how will they be managed?
- What flow control and monitoring are in place to establish that the pipeline is working properly? Are these sufficient for the level of cybersecurity required?
- What compliance mandates are required? How are they addressed by the pipeline? Is this sufficient?

Reference Architecture/Platform Independent Model (PIM)

A **Reference Architecture** is an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions [2].



A PIM is a general and reusable model of a solution to a commonly occurring problem in software engineering within a given context, and is independent of the specific technological platform used to implement it.

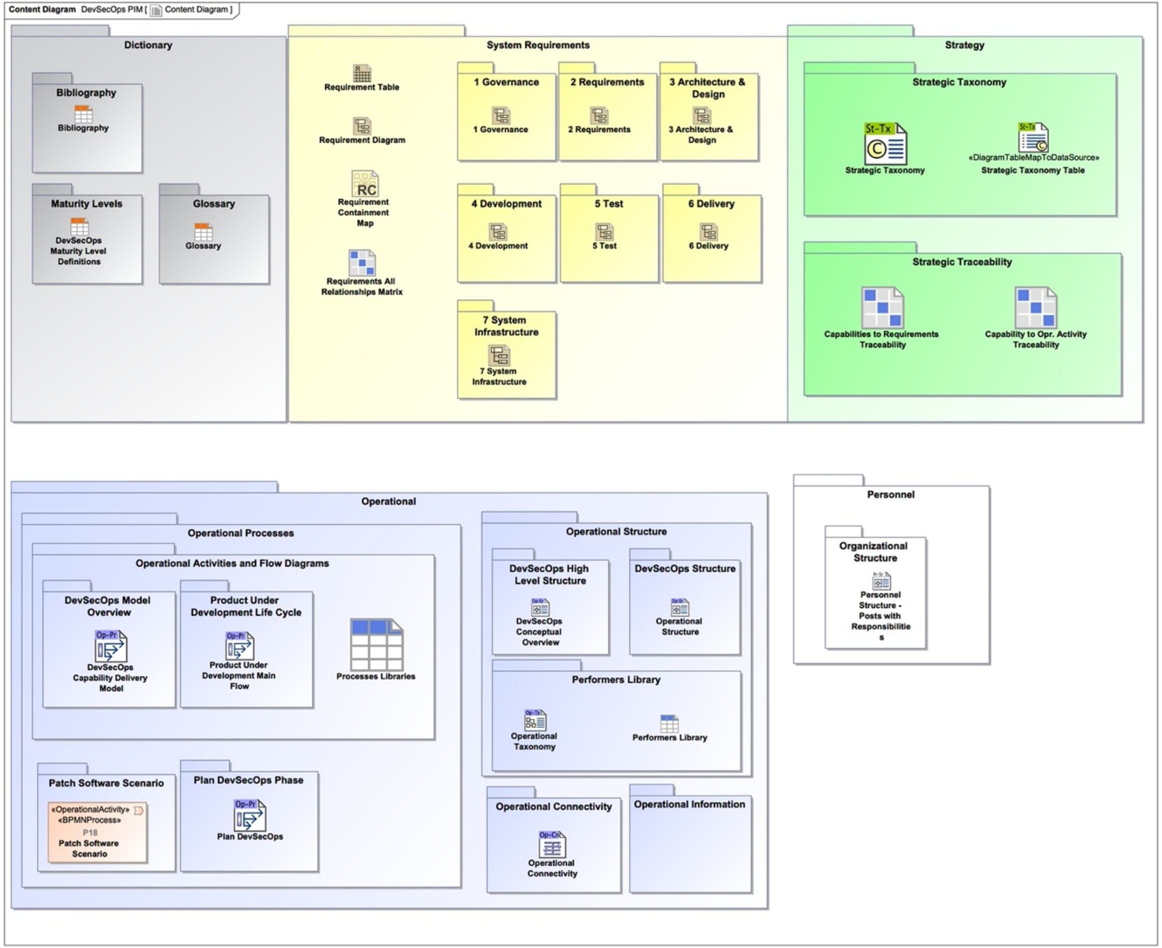


NOTE: PSM = Platform Specific Model

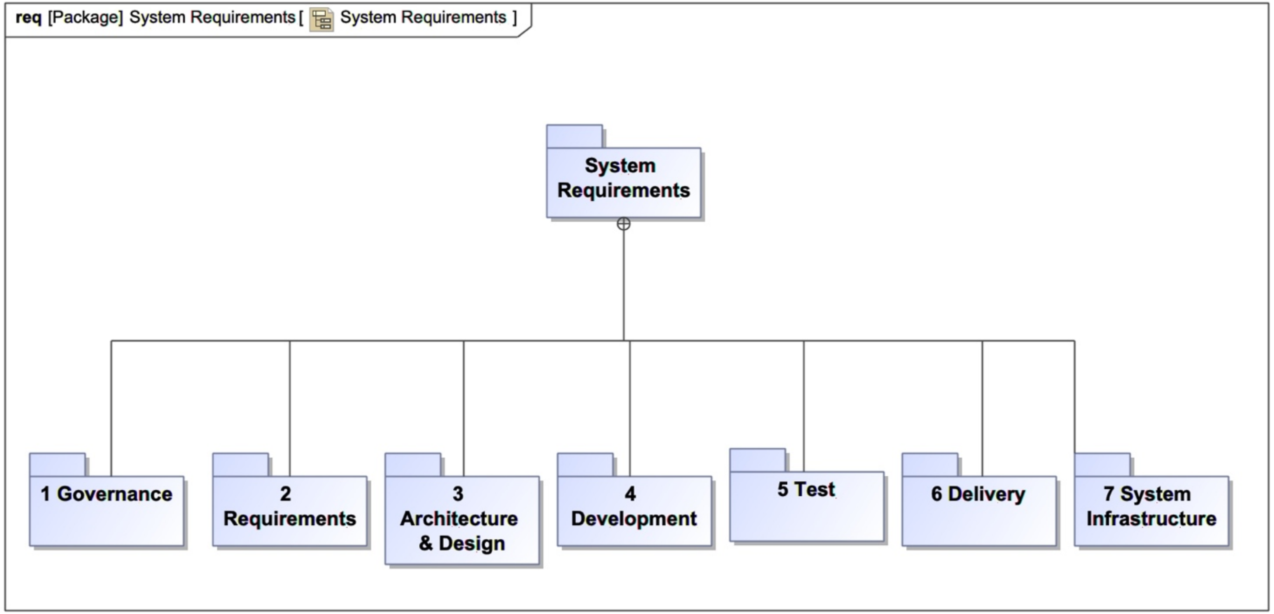
[2] DoD Reference Architecture Description, https://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf

PIM Content

- System Requirements
- Capabilities
- Operational Processes & Structures
- Roles
- Glossary
- Maturity Levels
- Bibliography



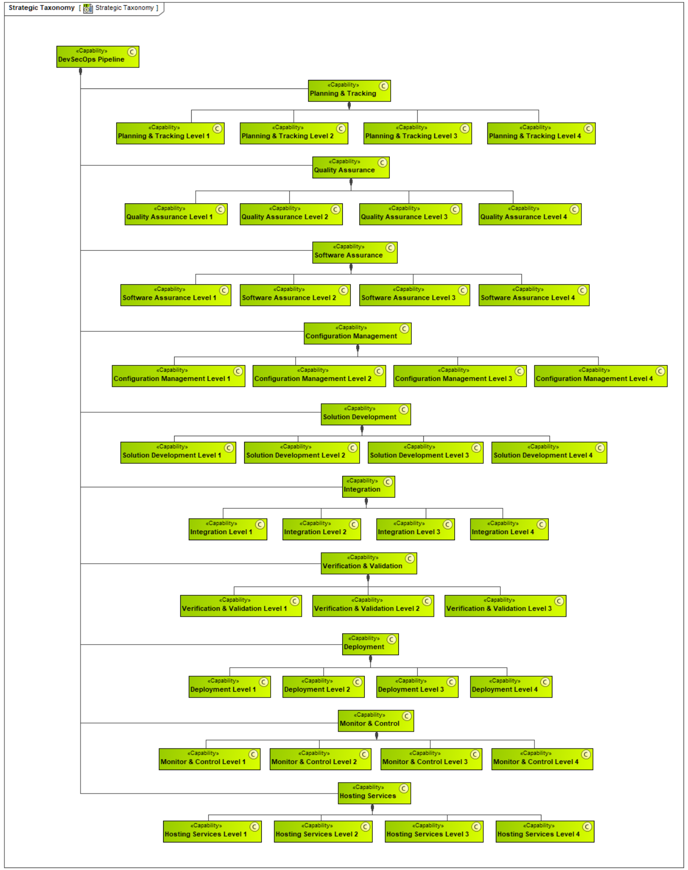
DevSecOps Requirements Map to Maturity



Legend
↗ Trace

	Maturity Levels	Maturity Level 1	Maturity Level 2	Maturity Level 3	Maturity Level 4
System Requirements		55	75	51	3
1 Governance		22	33	19	
Gov_1 Track Changes Associated to Requirements	1		↗		
Gov_2 Track Progress with Scrum/Kanban Boards		1	5	3	
Gov_3 Task Creation		2	4		
Gov_4 Metrics			4	3	
Gov_5 Knowledge Management		15	9	3	
Gov_6 System Assurance		2	4	10	
Gov_7 Defect and Issue Tracking			2		
Gov_8 Noncompliance Tracking			3		
Gov_9 Document and Manage Identified Risks		2	1		
2 Requirements		8	6	2	
Req_1 Document Requirements		4	6		
Req_2 Requirements Abstraction Layers	1	↗			
Req_3 Requirements Prioritization	1	↗			
Req_4 Requirements Validation	1			↗	
Req_5 Change Management of Requirements		1		1	
Req_6 Requirements Authorization	1	↗			
3 Architecture & Design		4	1	5	
Arc_1 Requirement Mapping	1	↗			
Arc_2 Implementation Mapping	1		↗		
Arc_3 MBSE				2	
Arc_4 System Assurance Design		3		3	
4 Development		13	18	5	
Dev_1 Mapping to Requirements	1	↗			
Dev_2 Mapping to Architecture	1	↗			
Dev_3 Mapping to Tests	1	↗			
Dev_4 Secure Software Development			7		
Dev_5 Code Reviews		1	1		
Dev_6 Orchestration			1	2	
Dev_7 Configuration Management		8	9	2	
Dev_8 Integrated Development Environment (IDE)	1	↗			
Dev_9 Development Information Radiator	1			↗	
5 Test		5	6	4	
Tes_1 Manual Testing		4			
Tes_2 Requirement Association	1		↗		
Tes_3 Automated Testing		1	2	2	
Tes_4 Code Coverage	1			↗	
Tes_5 Penetration and Fuzz Testing	1			↗	
Tes_6 Testing Information Radiator	1		↗		
Tes_7 Multi-phase Testing			2		
6 Delivery		2	1	4	
Del_1 Release Management	1			↗	
Del_2 ITSM Service Desk	1			↗	
Del_3 Continuous Delivery				2	
Del_4 Product Recovery	1	↗			
Del_5 System Recovery	1	↗			
Del_6 Configuration Item Integrity	1		↗		
7 System Infrastructure		1	10	12	3
Sys_1 System's Nonfunctional Requirements	1	↗			
Sys_2 Automated Provisioning	1			↗	
Sys_3 System Maintenance			2	1	2
Sys_4 Communication	1			↗	
Sys_5 Information Management			6	5	
Sys_6 Infrastructure Configuration Management			2	1	
Sys_7 Automated Patch Management				3	1

As a DevSecOps System Matures, so will its Capabilities

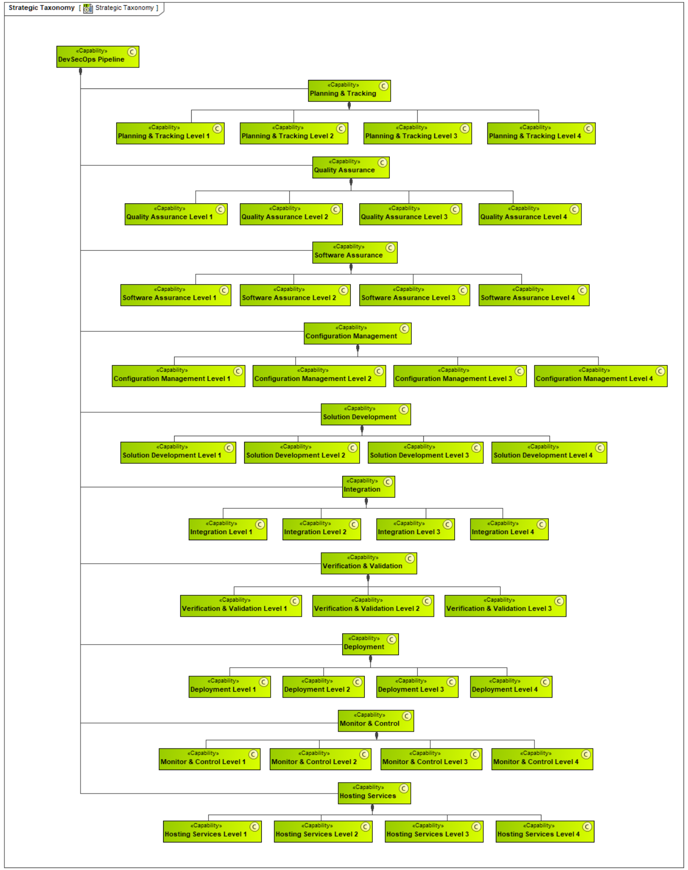


Legend

↗ Trace

Strategic Taxonomy / DevSecOps Pipeline \ System Requirements	1 Governance	2 Requirements	3 Architecture & Design	4 Development	5 Test	6 Delivery	7 System Infrastructure
Configuration Management	3	2		15	3	1	4
Deployment						6	1
Hosting Services	2	1			1		26
Integration				4			
Monitor & Control	27			3	3		12
Planning & Tracking	27	3				1	1
Quality Assurance	11			3	1		
Software Assurance	6	1	6	15	2	1	9
Solution Development	3	9	10	13			1
Verification & Validation	1	1		3	12		1

As a DevSecOps System Matures, so will its Capabilities

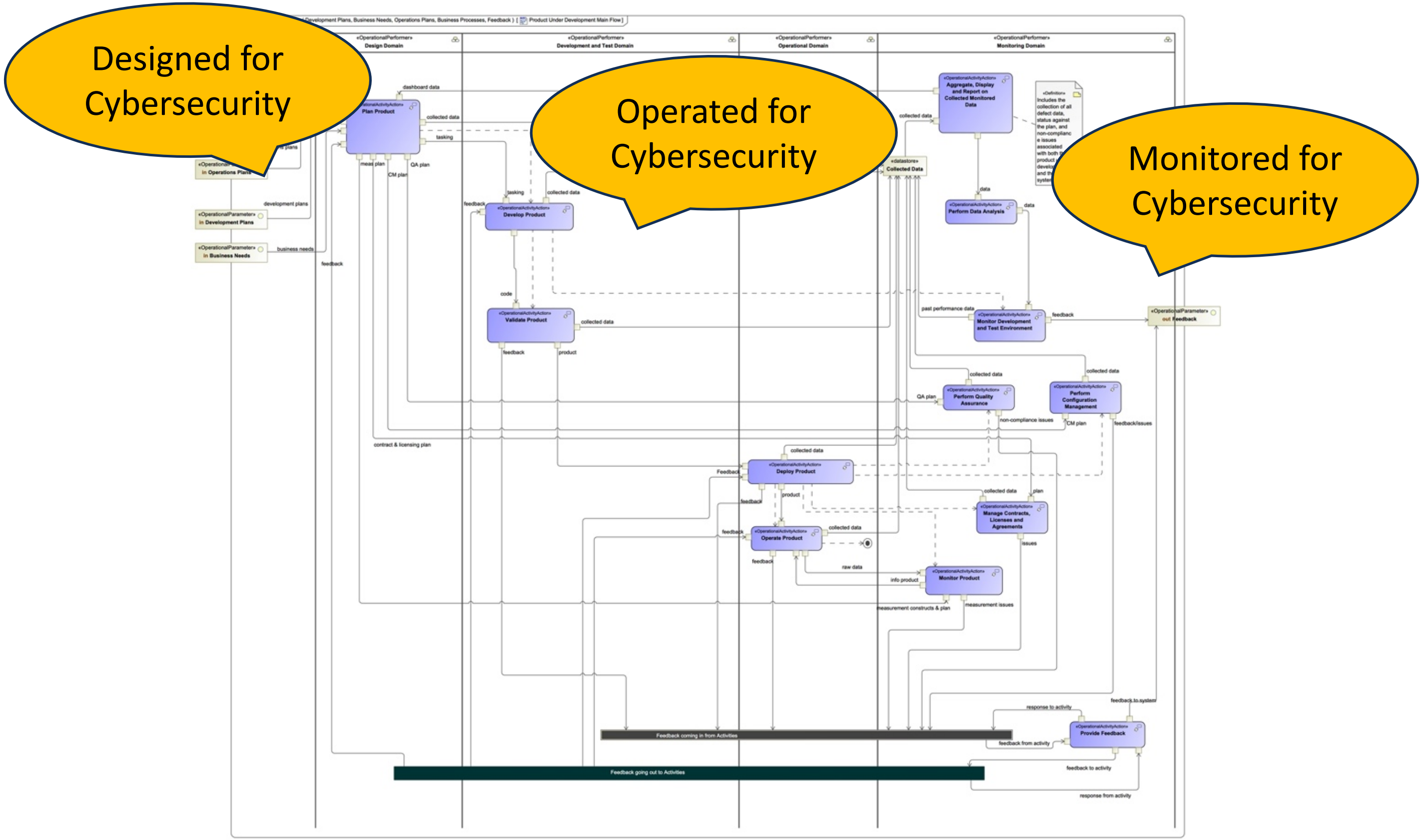


Legend
Trace

Strategic Taxonomy / DevSecOps Pipeline	1 Governance	2 Requirements	3 Architecture & Design	4 Development	5 Test	6 Delivery	7 System Infrastructure
Configuration Management	3	2		15	3	1	4
Deployment						6	1
Hosting Services	2	1			1		26
Integration				4			
Monitor & Control	27			3	3		12
Planning & Tracking	27	3				1	1
Quality Assurance	11			3	1		
Software Assurance	6	1	6	15	2	1	9
Solution Development	3	9	10	13			1
Verification & Validation	1	1		3	12		1

System Requirements

DevSecOps Pipeline Delivers Key Cybersecurity Requirements

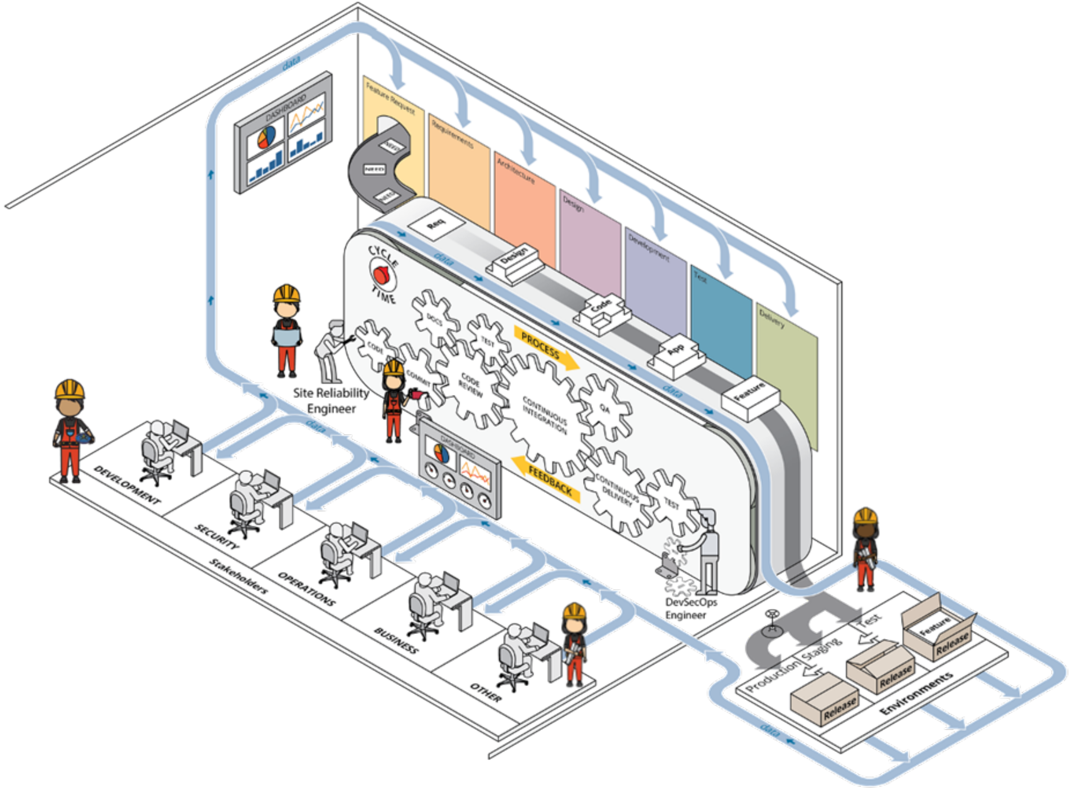


Planning and Monitoring is Critical

Is the pipeline performing as expected?

- Are the right things measured?
- Does this match the results?

How can performance be improved?



A Cybersecurity Engineering Strategy for DevSecOps

Managing Supply Chain Risk for DevSecOps

Types of Supply Chains Impacting Systems

Hardware Supply Chains

- Conceptualize, design, build, and deliver hardware and systems
- Includes manufacturing and integration supply chains

Service Supply Chains

- Provide services to acquirers, including data processing and hosting, logistical services, and support for administrative functions

Software Supply Chains

- Produce the software that runs on vital systems
- Comprise the network of stakeholders that contribute to the content of a software product or that have the opportunity to modify its content
- Language libraries and open source used in development

Acquisition Strategies

Formal Acquisition and Contracting

- Request for Proposal (RFP) response
- Negotiated outcomes bounded by cost and schedule

Commercial Off the Shelf

- Purchase of existing 3rd party product
- May include continuing service agreement for updates and fixes

Informal Selection

- Download from open source library
- Code extracted from prior versions or similar projects

Most organizations use all of these depending on the level of rigor needed to meet requirements

Supply Chain Risk: Example Incidents

- Heartland Payment Systems (2009)
- Silverpop (2010)
- Epsilon (2011)
- New York State Electric and Gas (2012)
- California Department of Child Support Services (2012)
- Thrift Savings Plan (2012)
- Target (2013)
- Lowes (2014)
- AT&T(2014)
- HAVEX / Dragonfly attacks on energy industry (2014)
- DOD TRANSCOM contractor breaches (2014)
- Equifax (2017)
- Marriott (2018)
- SolarWinds (2020)

Complexity: Aligning and Managing Security Objectives Across the Supply Chain

Mission View

- Focus: Assuring mission success

Infrastructure View

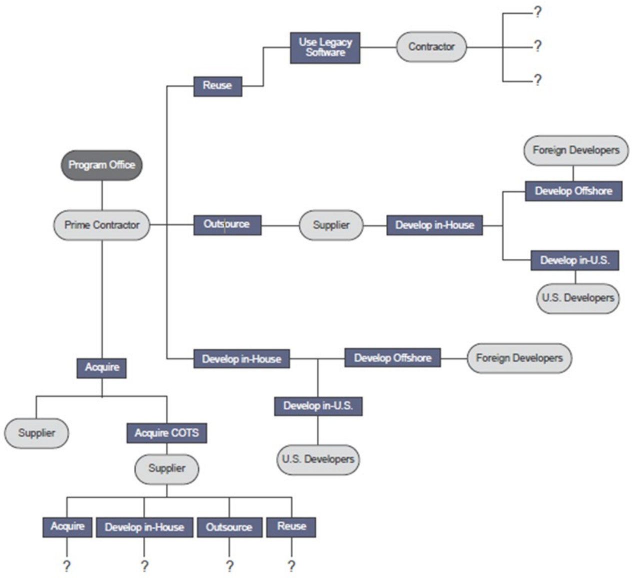
- Focus: Protection and sustainment of the infrastructure

Acquisition and Development View

- Focus: Build security into systems

Certification View

- Focus: Certify systems for deployment



Each organization/program unit addresses security from a different perspective (e.g., mission, infrastructure, acquisition and development).

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

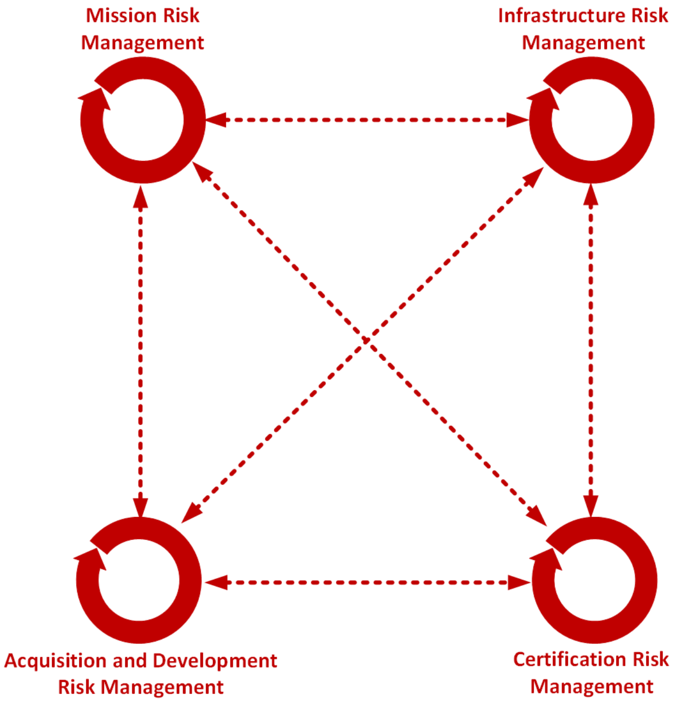
Complexity: Managing Security and Supply Chain Risk Across Organizations

Managed by multiple organizations/program units

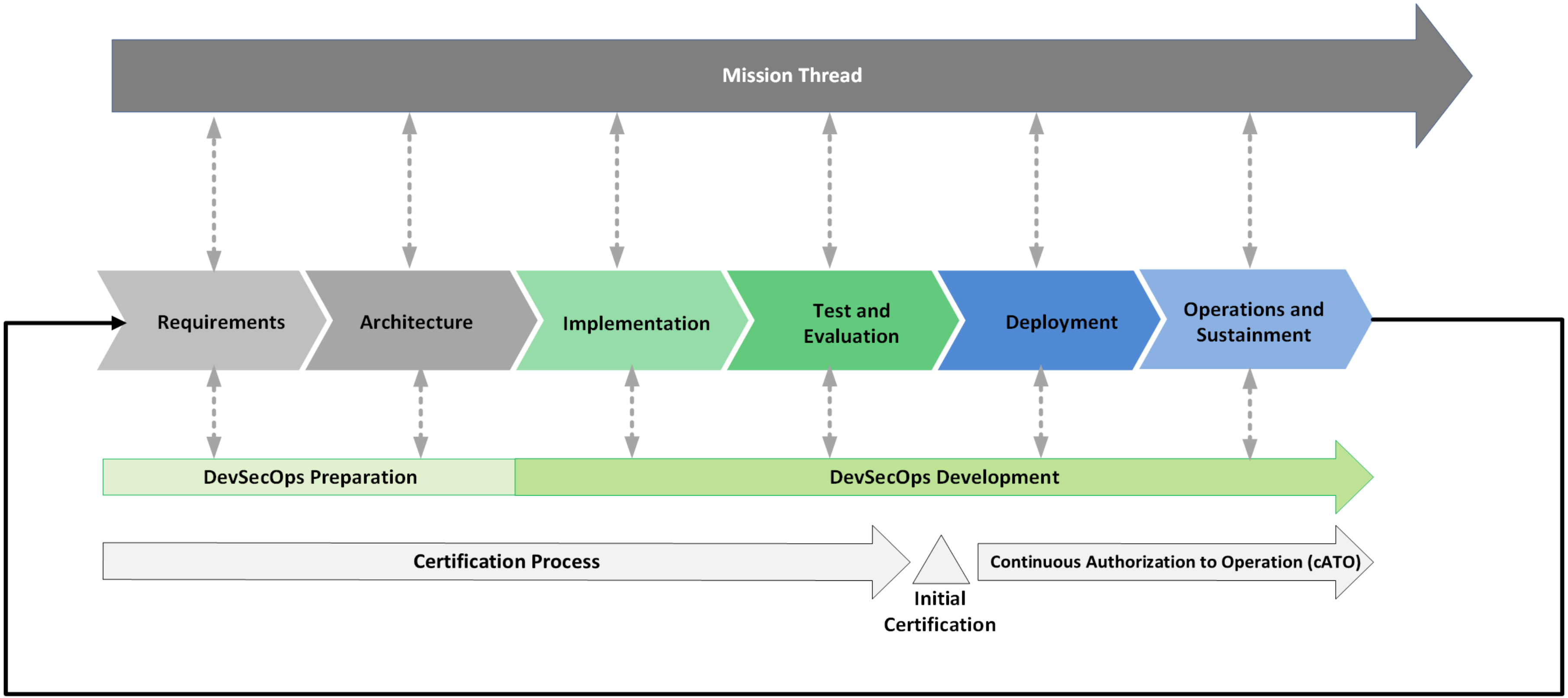
Activities, practices, and controls must align to keep overall security risk within an acceptable tolerance.

- Acquisition and development risk
- Certification risk
- Mission risk
- Infrastructure risk

Various participants lack clear reporting lines



DevSecOps Supply Chain Problem Space -1



DevSecOps Supply Chain Problem Space -2

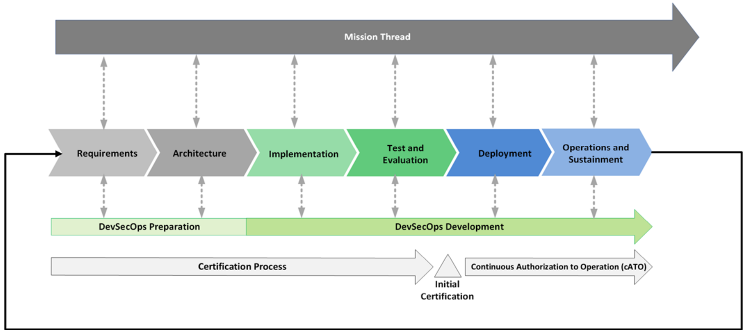
Cybersecurity practices need to be integrated with engineering activities across the systems lifecycle to

- Mitigate acquisition-related security risks
- Implement resilient architectures

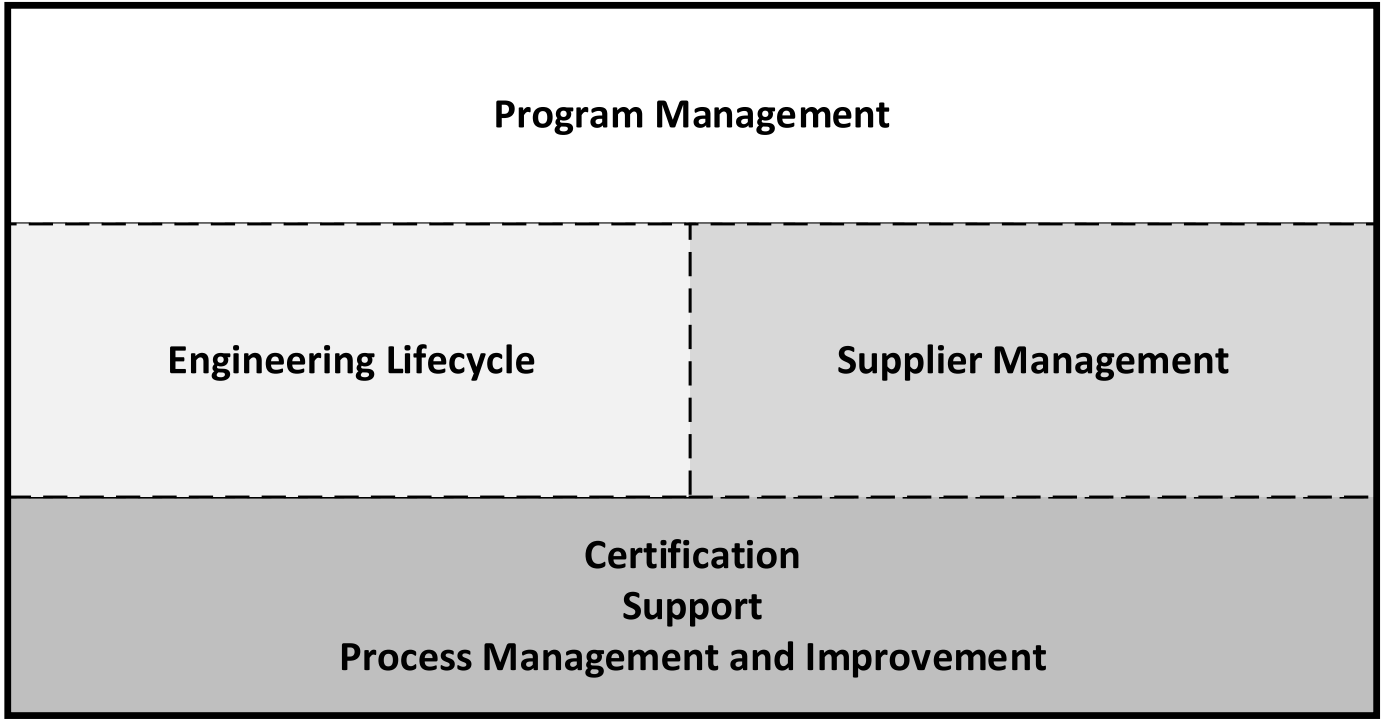
Cybersecurity risks must be managed continuously during operations to ensure that evolving security and resilience requirements are met, effectively and efficiently.

- 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.



Supply Chain Risk Management and Security Must Align Across Six Key Lifecycle Areas



A Cybersecurity Engineering Strategy for DevSecOps

Cybersecurity Strategy is Key to Success

Effective DevSecOps Cybersecurity Requires a Good Strategy

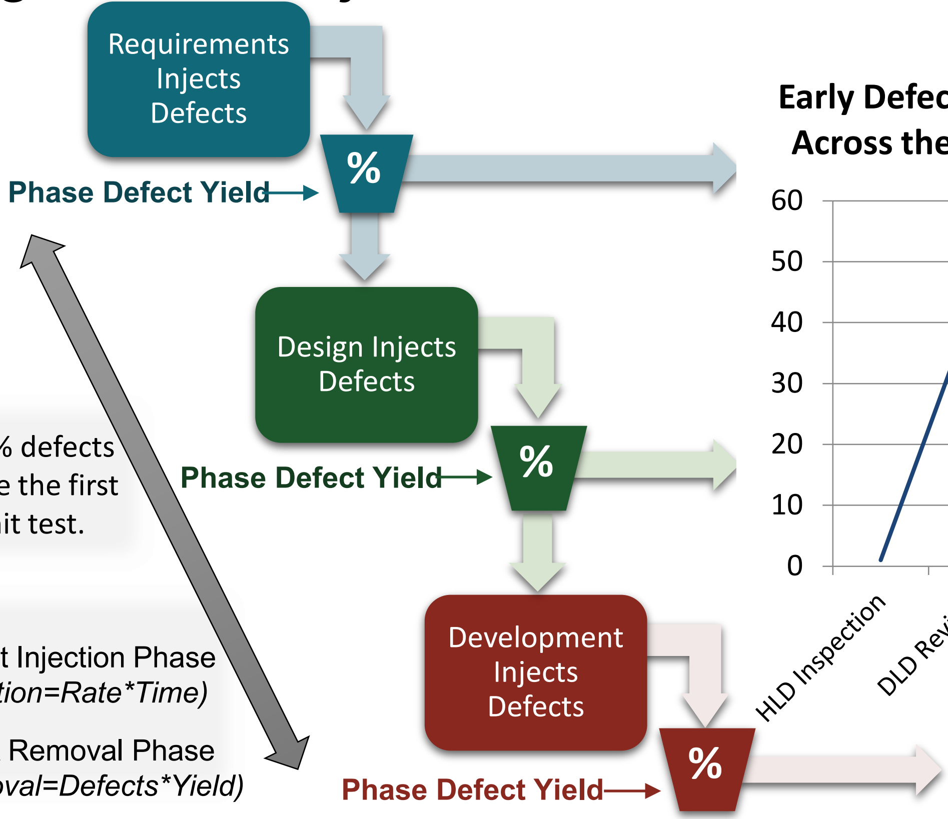
How will risk be identified, prioritized, and addressed in the DevSecOps pipeline?

- What cybersecurity requirements will be built into the pipeline?
- What tools will be integrated into the pipeline for vulnerability tracking and removal?
- What measurements will be implemented in the pipeline to monitor the processes and the product?
- How will the monitoring feed pipeline and product maturity?

How will the supply chain (3rd party code and components) be acquired implemented, and maintained

- How will trusted dependencies be implemented and monitored?
- How will coordination of supply chain participants be managed

Manage Defect Injection and Removal for Early Detection

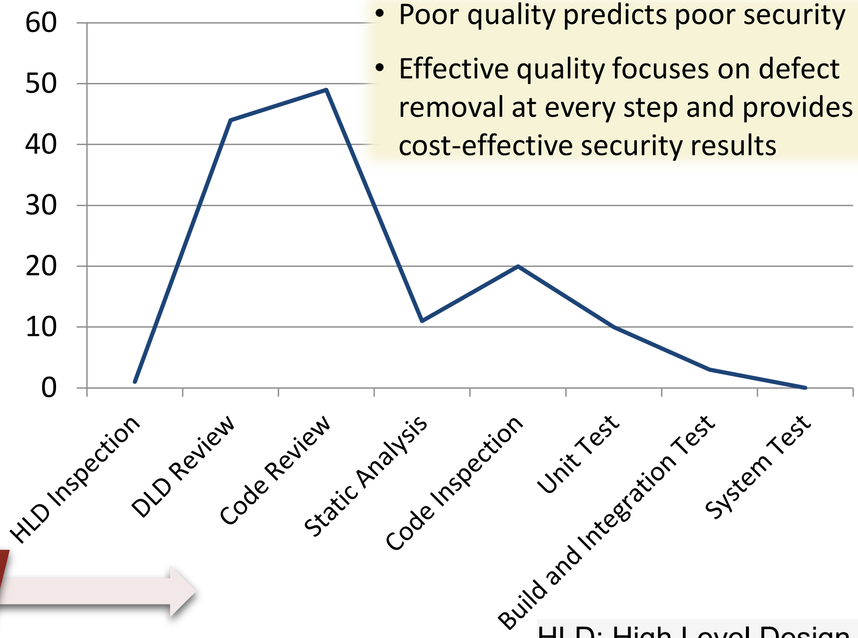


Process yield: % defects removed before the first compile and unit test.

Defect Injection Phase
*(Injection=Rate*Time)*

Defect Removal Phase
*(Removal=Defects*Yield)*

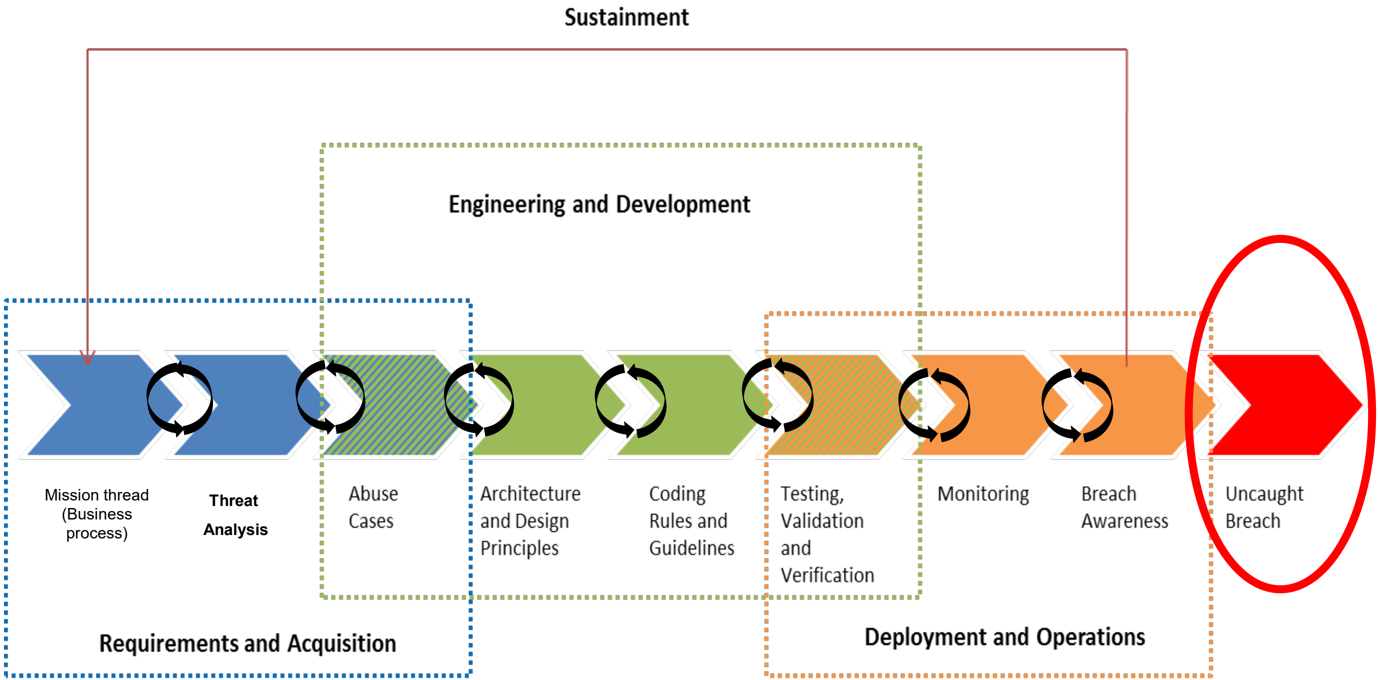
Early Defect Removal Across the Lifecycle



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

HLD: High Level Design
DLD: Detailed Level Design

Continuous Focus on Cybersecurity Risk Across the Lifecycle is Critical to Operational Mission Success



A Cybersecurity Engineering Strategy for DevSecOps

Final Thoughts

Build and Implement a Cybersecurity Strategy

Establish a plan for sufficient system and software cybersecurity engineering to ensure the operational mission(s) continue, even under cyber attack.

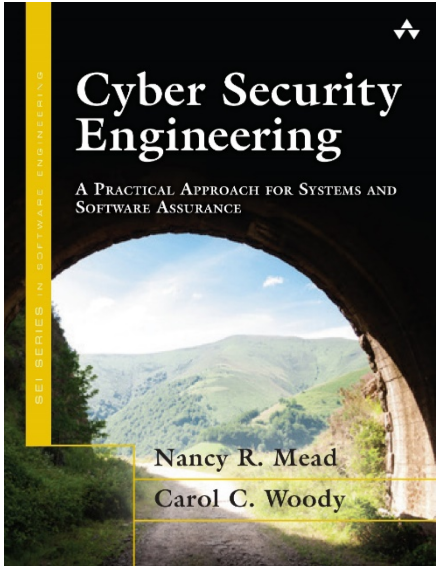
Elements in the strategy include:

- Establish security requirements to ensure confidentiality, integrity, availability (CIA)
- Monitor the pipeline and product for CIA in operational systems and software
- Monitor to recognize, resist, and recover from attacks
- Implement appropriate lifecycle processes and practices to reduce operational vulnerabilities
- Establish coordination and communication capabilities among the many participants to ensure timely and effective response

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/