DevSecOps Pipeline for Complex Software-Intensive Systems: Addressing Cybersecurity Challenges

Carol Woody, Ph.D.

January 2021

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

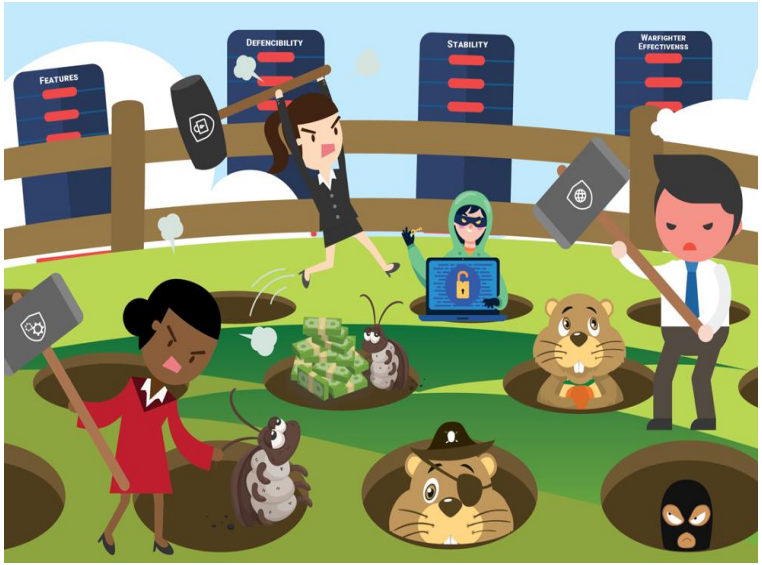
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-1104

Major DoD Program are Shifting to Commercially Successful Approaches to Improve Delivery Speed and Cost

- Hardware-based solution => Software-intensive system
- Waterfall methodology => Agile DevSecOps approach
- Program owned infrastructure => Shared infrastructures (e.g. Cloud)

Today: Program Offices Whac-A-Mole



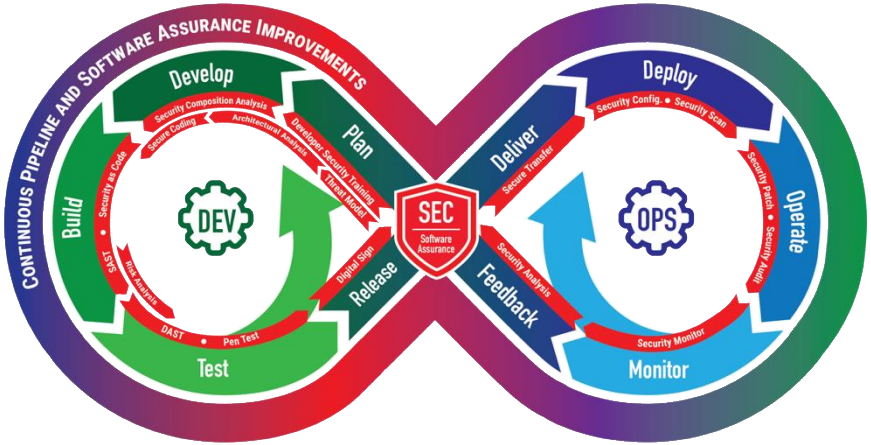
Winning in Features and Warfighter Effectiveness, but Losing in Defensibility and Stability

Just applying tools and automation does not ensure improvement.

In June of 2020 a generally successful DoD program completed an 8 week “Hardening the Software Factory” effort in order to address accumulated technical debt and insufficient security and operations practices due to the narrow focus on speed of delivery.

Missing is the ability to perform formal analysis of a system’s numerous parameters so instead program offices are forced to play Whac-A-Mole and hope for the best.

DevSecOps: a Complex Socio-Technical Information System



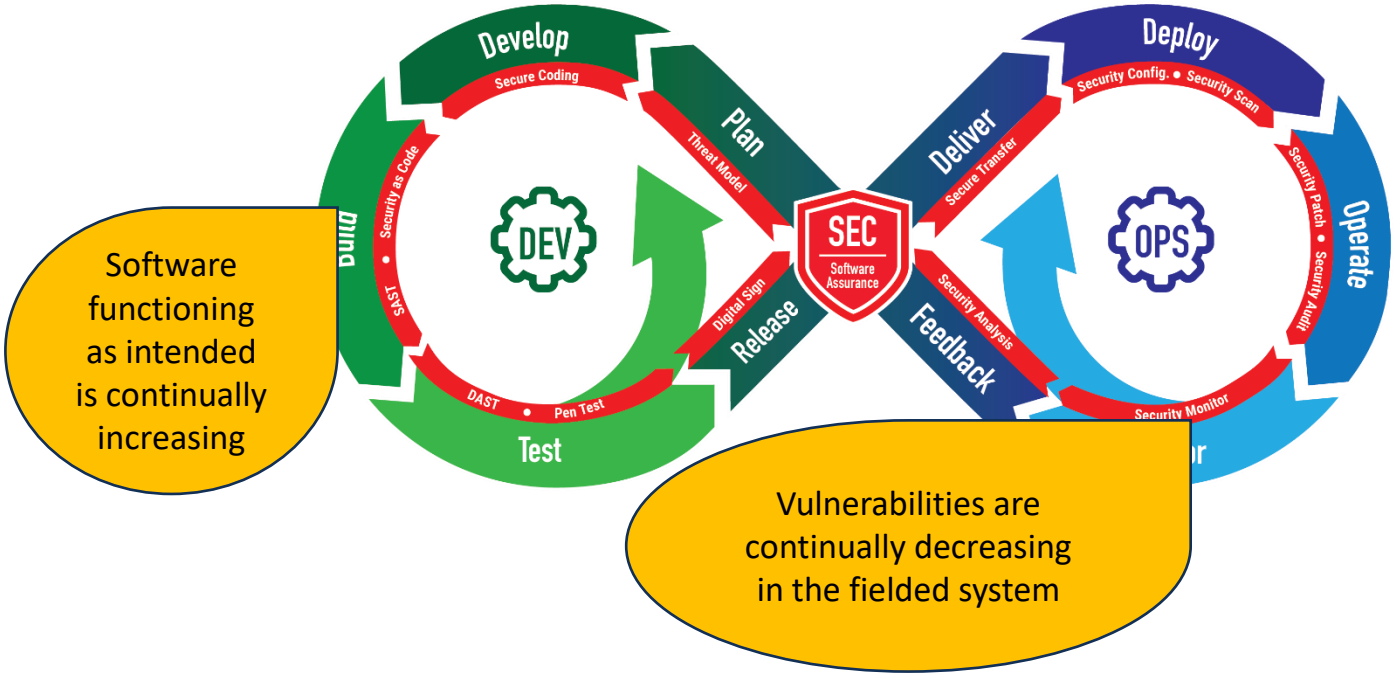
DSO is an approach that integrates development (Dev), security (Sec), and delivery/operations (Ops) of software systems to reduce the time required to move from need to capability and provide CI/CD with high software quality [1].

The DSO Continuous integration (CI) and Continuous Development (CD) pipeline is a **socio-technical system made up of both a collection of software tools and processes** [2].

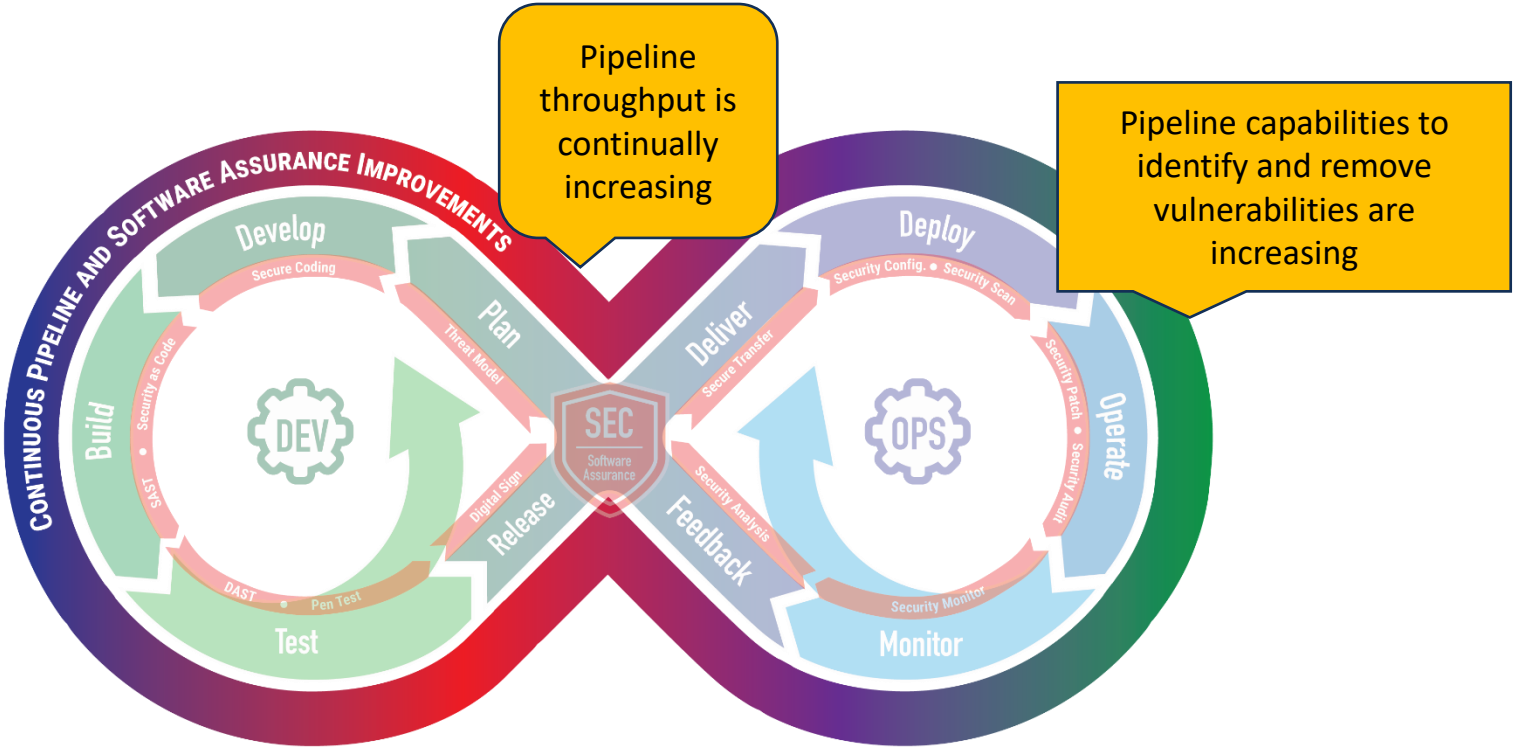
DSO CI/CD is **not a system to be built or acquired**, it is a personal and organizational **mindset** defining processes for the rapid development, fielding, and operations of software and software-based systems **utilizing automation where feasible** in order to achieve the desired throughput of new features and capabilities.

[1] Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments, CMU/SEI-2020-TR-002
[2] Len Bass, Ingo Weber, and Liming Zhu. 2015. DevOps: A Software Architect's Perspective (1st ed.). Addison-Wesley Professional.

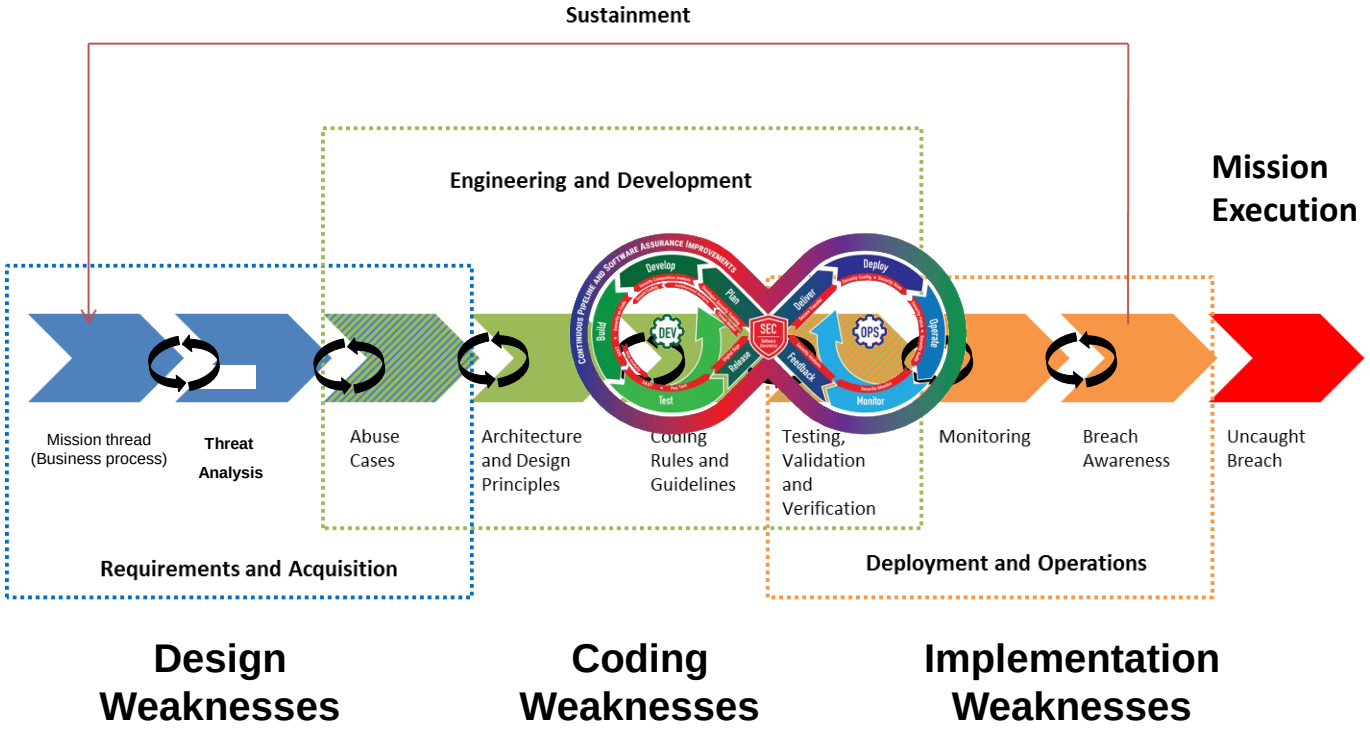
Development Factory Can Deliver Increased Product Assurance



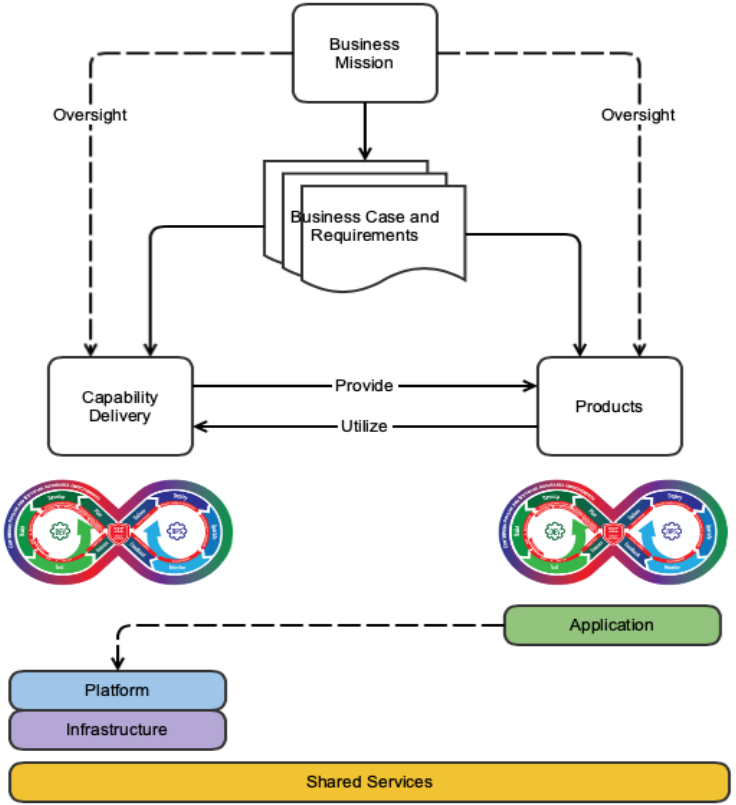
DevSecOps Pipeline Infrastructure Can Promote Increased Assurance



Cybersecurity Is an Acquisition Lifecycle Challenge and the DSO Pipeline is only a Piece of the Puzzle



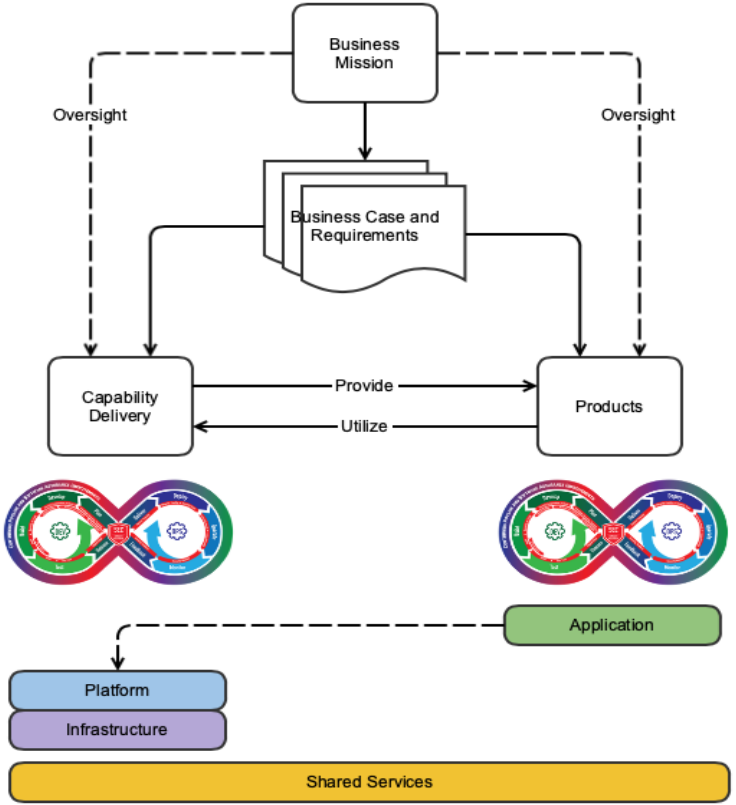
Challenge 1 for DSO: connecting process, practice, & tools



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.

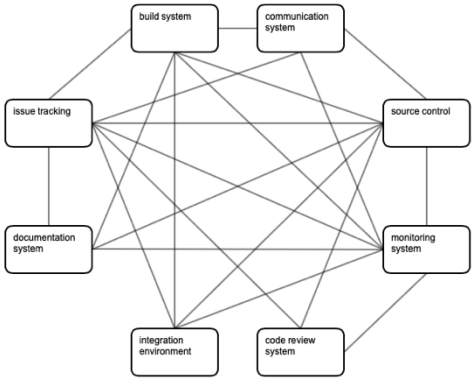
Cybersecurity demands effective governance to address:

- 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?

Example of Complexity: Tool Management -1

Tool groups connect roles to capabilities in a pipeline integrating Dev, Sec, and Ops capabilities with interactions that did not exist before

Administrative resources now control what each participant can see and do, way beyond the typical responsibility of authentication and authorization.



Tool Group	#Coupling	Interface
Issue Tracking System	7	create, modify, delete, read issues where an issue has some schema definition
Code Review System	2	create review, start review, add source files to review, add comments to review, create issue from review item, resolve issue from review item, close review
Monitoring System	7	write message; write metric; display metric; create, modify, delete, read alarm threshold on metric; notify on alarm; show dashboard; process message; extract metric
Integration and Test Environment	3	deploy system, tear down system, execute tests, collect test results
Documentation System	3	create, modify, delete document where a document has some schema definition
Build System	6	execute build; create, modify, delete, read build definition where a build is a collection of steps executed to create artifacts that can be executed
Source Control System	6	create, modify, delete, read repository; write source files to repository; modify source lines in repository; read repository
Communication System	4	create, modify, delete, read channel; read and write comment to channel where a channel is an interactive conversation of text between human users with machine users making contributions

Example of Complexity: Tool Management -2

Each tool type requires specific technical skills that must be drawn from the integrated capabilities (Dev, Sec & Ops) and work together in the process flow.

- Product build should move through the security activities as part of the pipeline flow.
- Security considerations can be in the control gates for the pipeline flow.

Process Type	Process	Security Activities
Dev	Plan	Threat Model
	Code	Secure Coding
	Build	SAST, Security as Code
	Test	DAST, Pen Test
	Release	Digital Sign
Ops	Deliver	Secure Transfer
	Deploy	Security Configuration and Scan
	Operate	Security Patch and Audit
	Monitor	Security Monitor
	Feedback	Security Analysis

Pipeline flow does not address security for the pipeline's capabilities

- Pipeline security must be integrated into the roles and responsibilities of those that administer and support these capabilities.
- Pipeline administrators should perform the similar processes and use similar tools, but they are applied to different content.

Example of Complexity: Tool Management -3

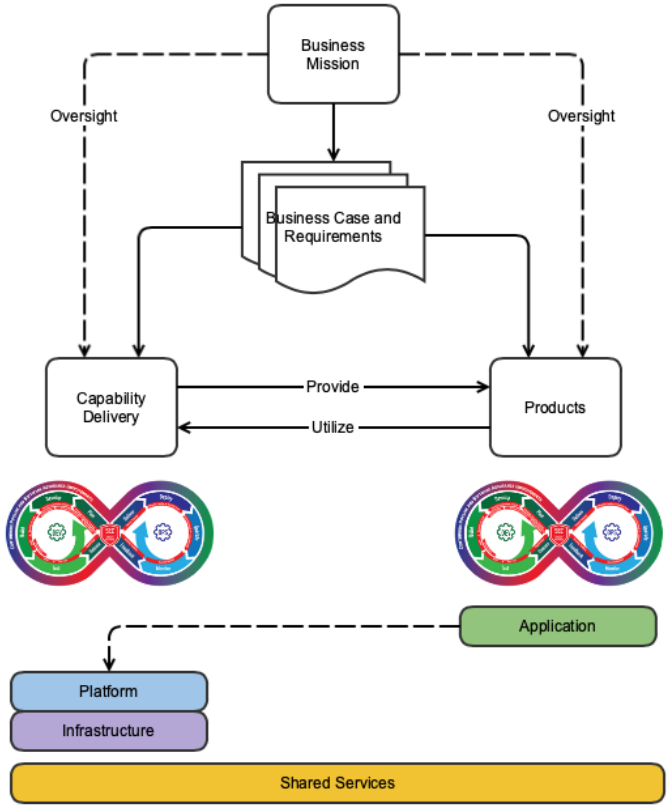
A range of processes can be allocated to various pipeline administrative roles.

Each process focuses on a different component of the pipeline, but all processes are needed to keep the pipeline functioning effectively.

Due to this complexity and repurposing of existing administrative resources without integrated oversight, infrastructure services and development tool types are increasingly the target of attacks.

Operational Process	Component	Role
Add Hardware	Host System	infra
Code Software	Source Control System Issue Tracking System IdAM Communication System Code Review System	dev
Configure Infrastructure	Host System	infra
Decommission Hardware	Host System	infra
Deploy Application	Any	ops
Disaster Recovery	Any	all
Install Software	Any	admin
Manage Incidents	Monitoring System	admin
Manage Users	IdAM System	admin
Monitor Infrastructure	Monitoring System	infra
Operate Solutions	Any	ops
Patch Infrastructure	Host System	infra
Patch Software	Any	admin
Perform Backup	Any	admin
Review Logs	Monitoring System	ops
Test Applications	Any	dev

What Are We Trying to Do…?

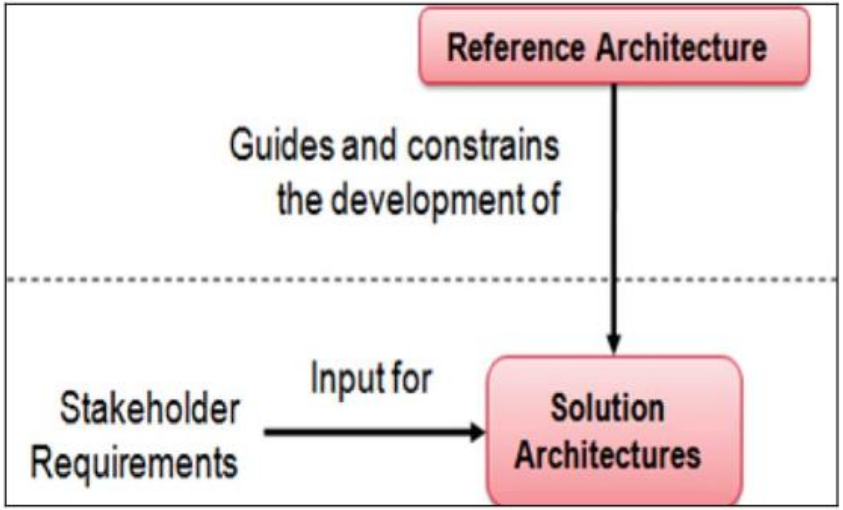


Create a Platform Independent Model (PIM) of a DevSecOps (DSO) System that codifies and integrates critical capabilities in order to be able to:

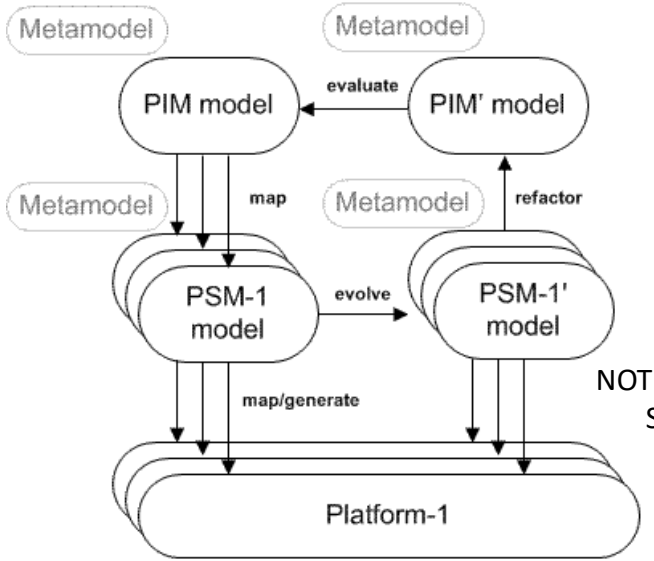
- Apply DSO methods to complex systems that do not follow well-established software architectural patterns commonly used in industry
- Specify the DSO requirements to the lead system integrators who need to develop a platform-specific solution
- Provide a basis for threat and attack surface analysis to build a cyber assurance case

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 [3].



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

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

What Is New In Our Approach and Why Do We Think It Will Be Successful?

Approach

- Build a DSO PIM using model-based engineering methodology. The model will encode the complex socio-technical system of tools, processes, and human interactions within a DSO CI/CD pipeline.

Reason for Success

- A DSO PIM will bridge the gap between high-level system concept/views and actual instantiations.
- A DSO PIM will provide a single source of truth in which multiple perspectives and views can be analyzed.

Additional Details are Available in our Publication:

The Journal on Systemics, Cybernetics and Informatics: JSCI, Volume 18 - Number 5 - Year 2020, pp. 31-36, ISSN: 1690-4524 (Online)
http://www.iiisci.org/journal/sci/issue.asp?is=ISS2005

Future: Program Office Will Continuously Balance Features with Stability and Security

PIM will explicitly identify points (e.g. requirements, constraints, and conditions) that should be addressed or mitigated as well as mechanisms to manage coverage of the points.

PSMs will present solutions based on the PIM.

Using provided modeling mechanisms will allow for the comparison of PSMs, analyzing of trade-offs and balancing the system dynamically.

Contact Information

Carol Woody, Ph.D.

cwoody@cert.org

Primary Co-Author: Timothy A. Chick

tchick@sei.cmu.edu

Web Resources
https://sei.cmu.edu/