

USAF High Assurance and DevSecOps Challenges

Timothy A. Chick
Carol Woody, Ph.D.

December 2020

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

Copyright 2020 Carnegie Mellon University.

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM20-1104

Several Major Air Force Program are Facing Similar Issues

- Hardware-based solution => Software-intensive system
- Waterfall methodology => Agile DevSecOps approach
- Program owned infrastructure => AF shared infrastructure (Platform One, Ski Camp)
- ATO for 3 years => Continuous ATO with integrated monitoring
- Static certified and unchanged within the Nuclear boundary => Software-based periodically refreshed Nuclear Surety environment

Today: Program Offices Whac-A-Mole



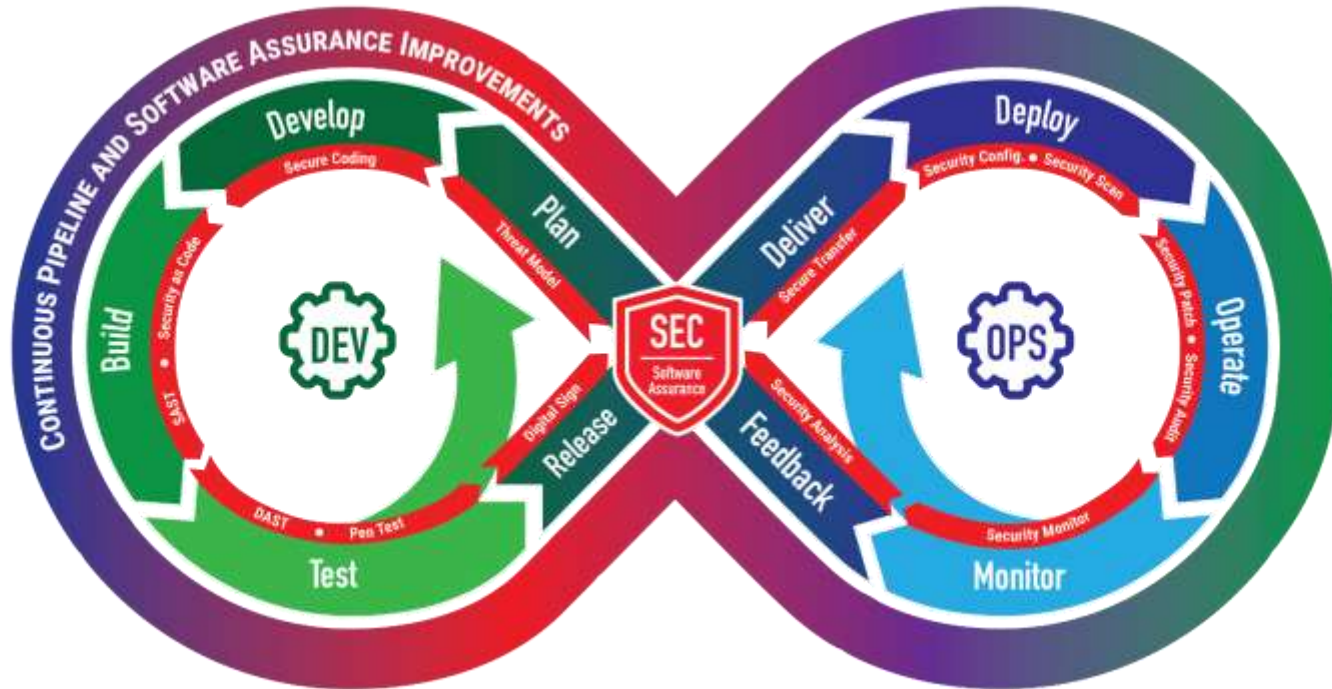
Winning in Features and Warfighter Effectiveness, but Losing in Defensibility and Stability

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 to address insufficient security and operations practices due to the narrow focus on speed of delivery.

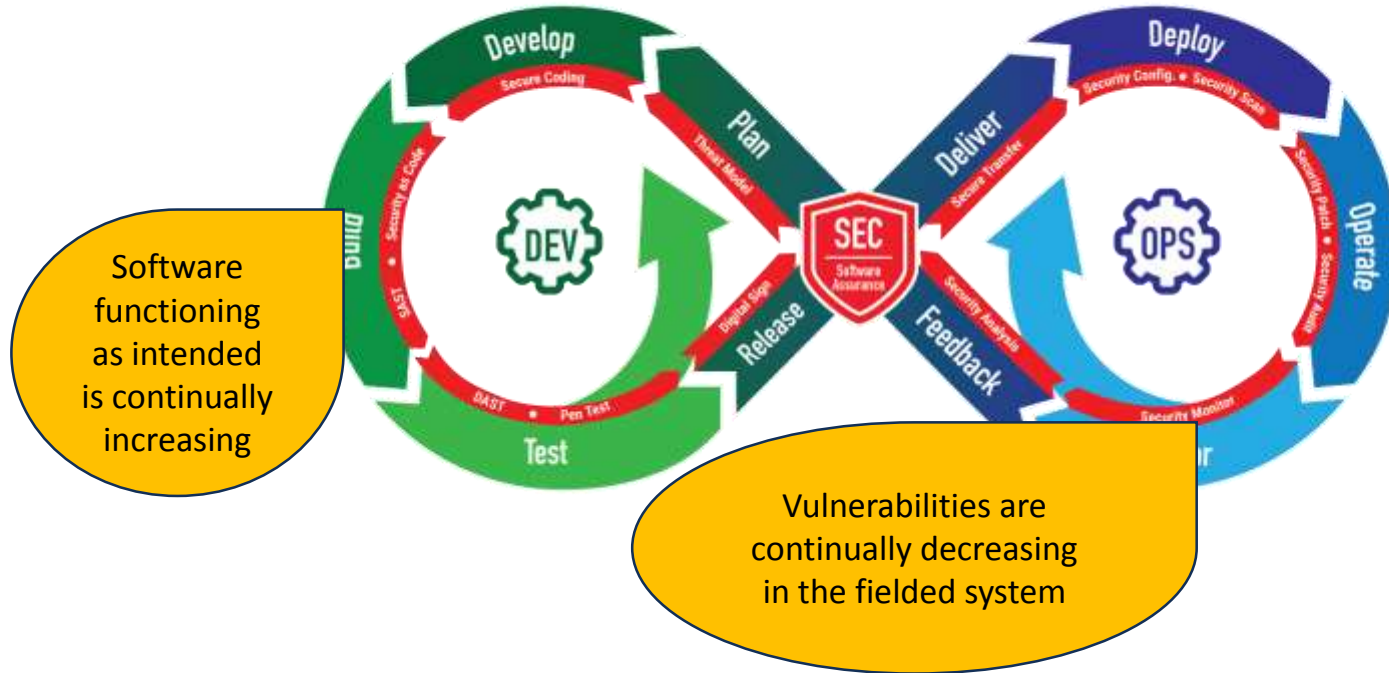
These things occur, even in small relatively successful programs, when technical debt and insufficient security and operational practices are in place due to lack of knowledge, experience, and reference material to fully design and execute an integrated DSO strategy in which all stakeholder needs, including cybersecurity, are addressed.

Without the ability to perform formal analysis of a system’s numerous parameters, program offices are forced to play Whac-A-Mole and hope for the best.

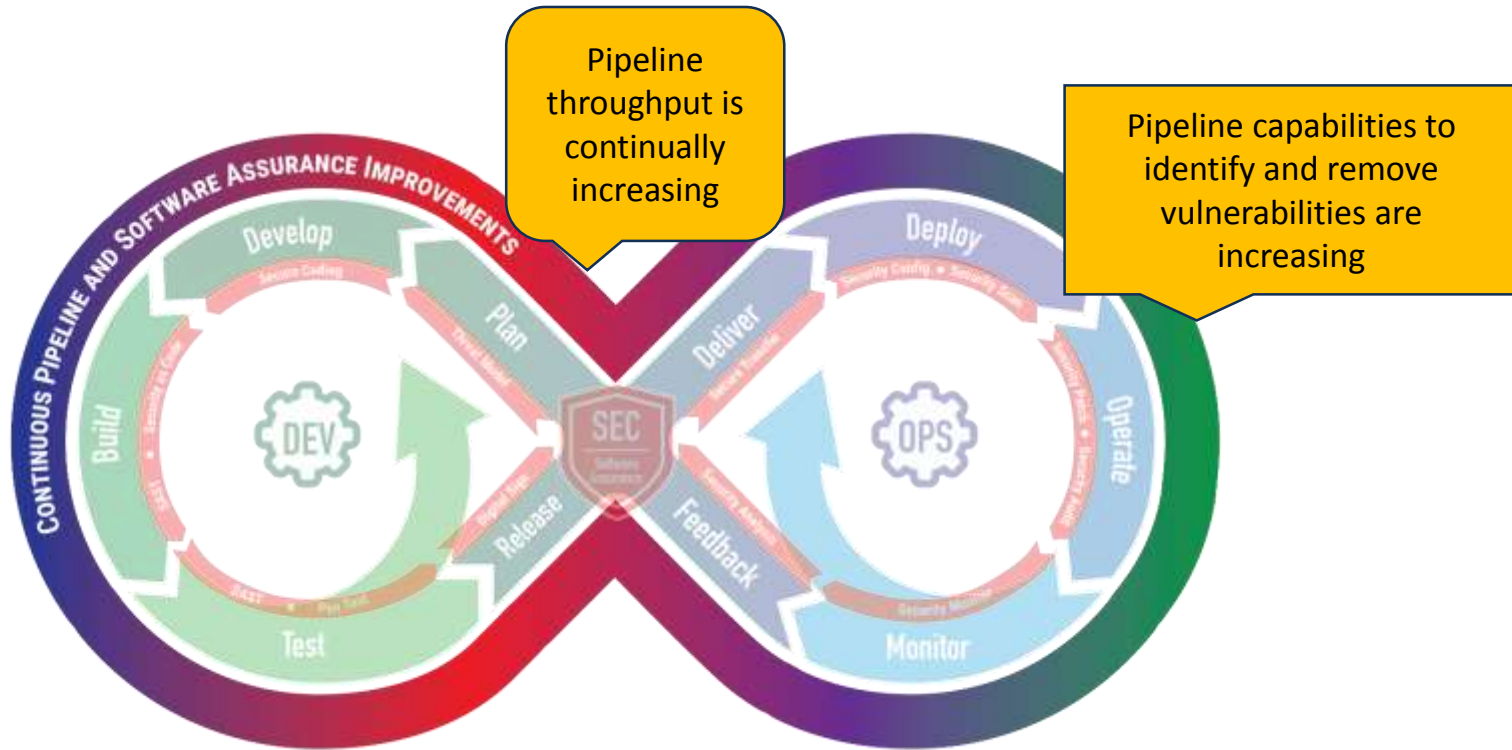
DevSecOps Pipeline Can Provide Integrated Assurance



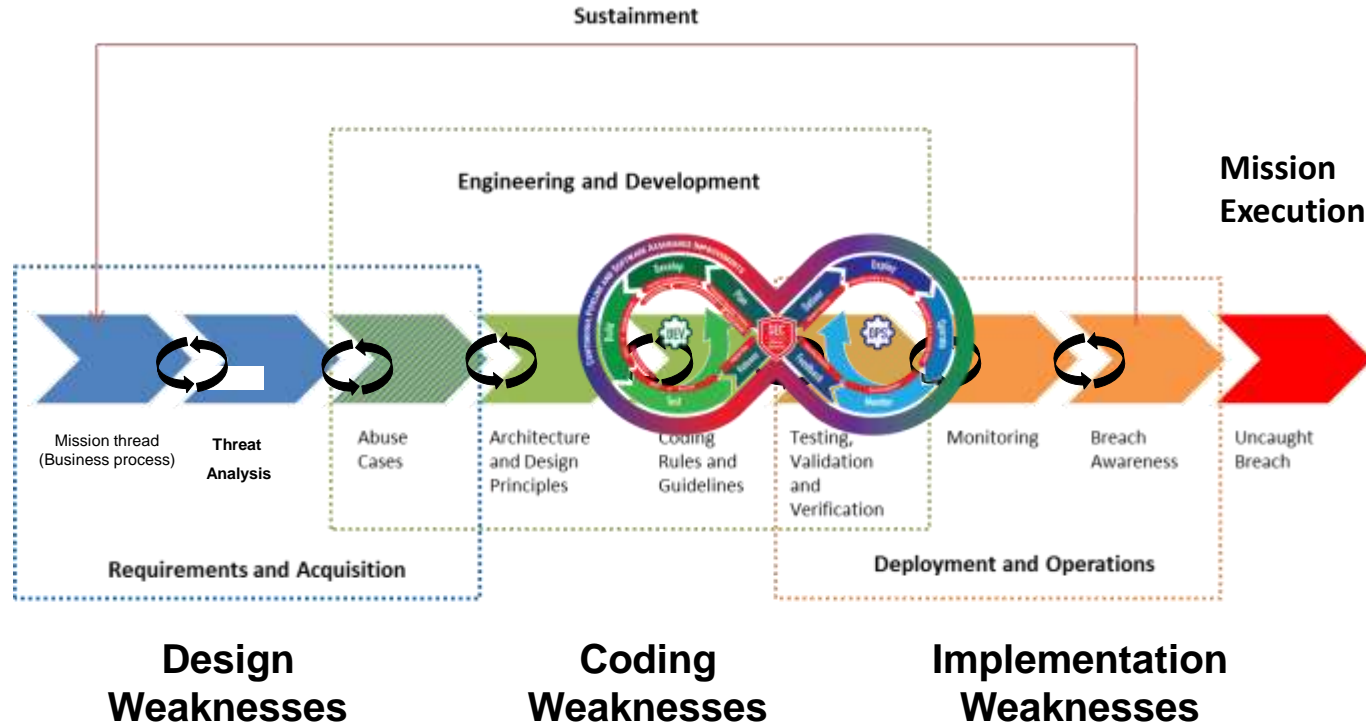
Development Factory Can Deliver Increased Product Assurance



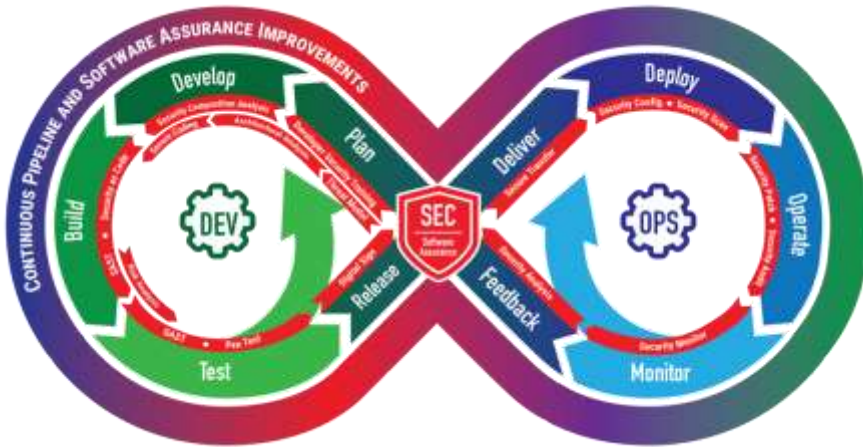
DevSecOps Pipeline Can Promote Increased Assurance



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



DevSecOps: a Complex Socio-Technical Information System

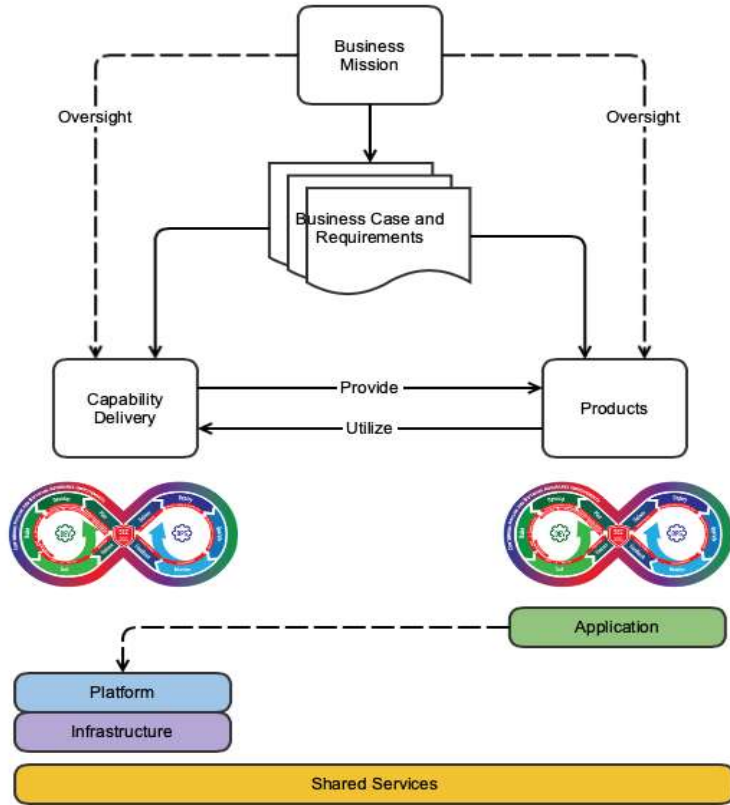


DSO is an approach that integrates development (Dev), security (Sec), and delivery/operations (Ops) of software systems to reduce the time required to move from need to capability and provide CI/CD with high software quality [13].

The DSO CI/CD pipeline is a **socio-technical system made up of both a collection of software tools and processes** [14].

It 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.

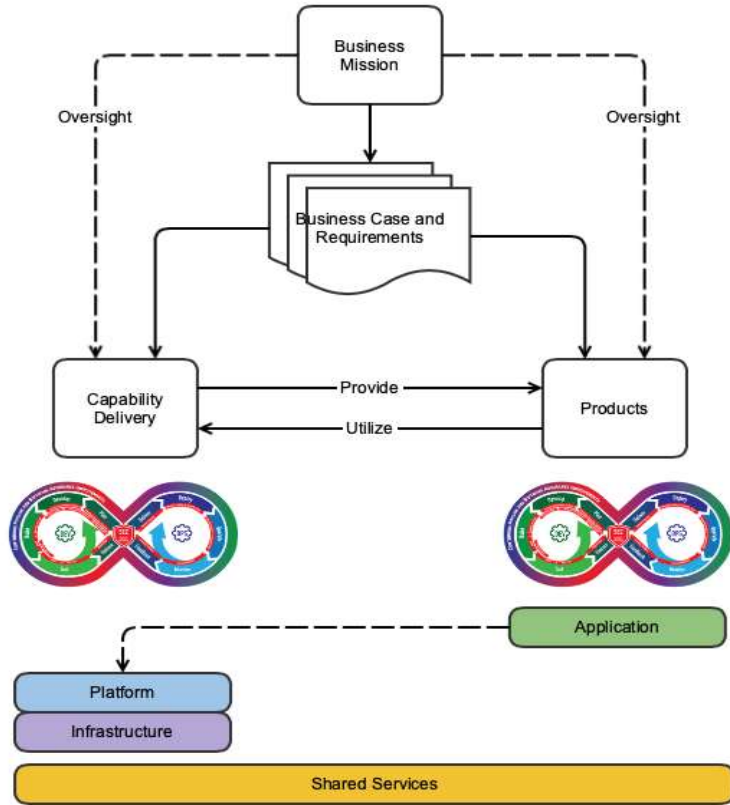
Challenge 1 for DSO: connecting process, practice, & tools



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



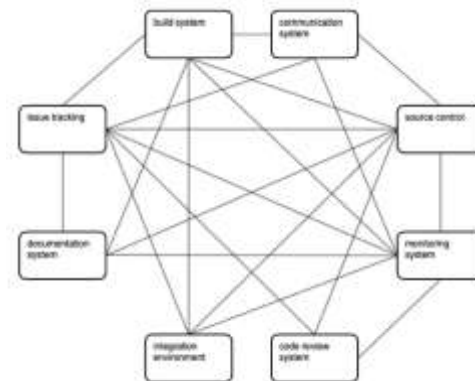
Managing and monitoring all of the various parts to ensure the product is built with sufficient cybersecurity and the pipeline is maintained to operate with sufficient cybersecurity is complex. Cybersecurity demands effective governance to address:

- 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.

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.

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

Example of Complexity: Tool Management -3

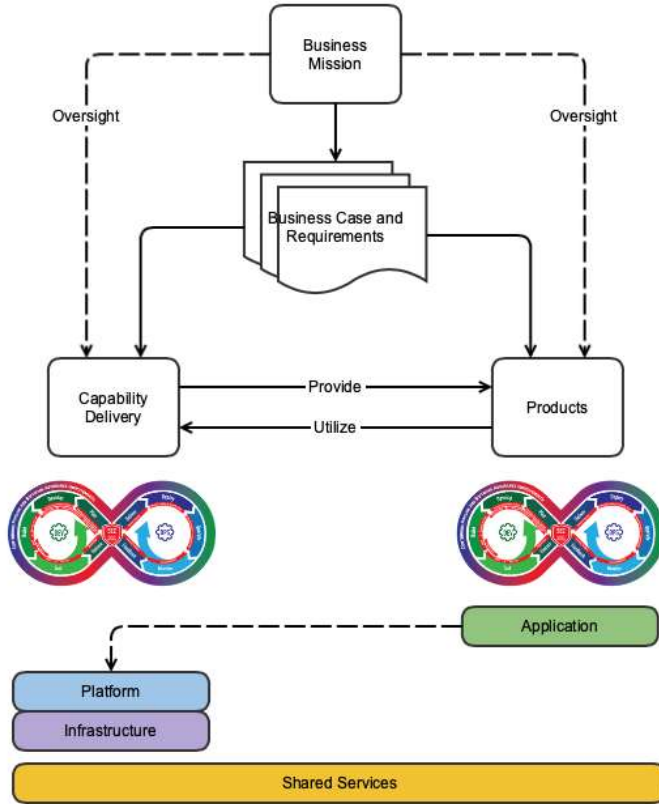
A range of processes can be allocated to various pipeline administrative roles.

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...?

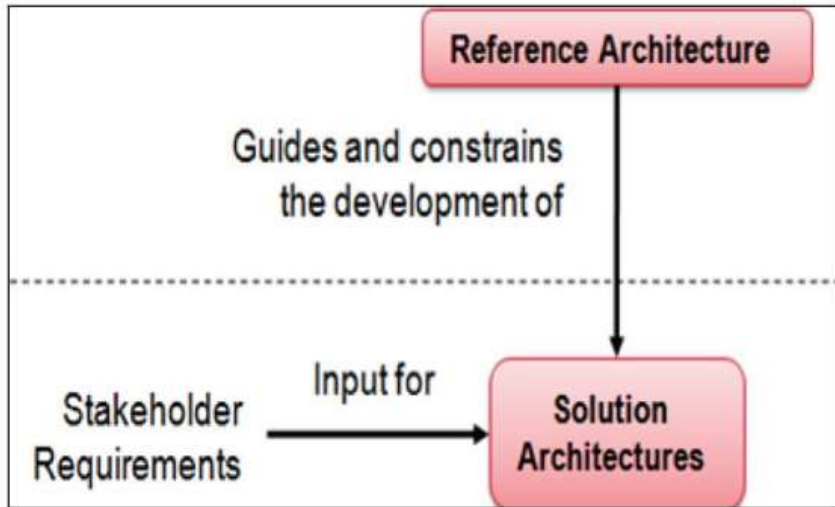


Create a Platform Independent Model (PIM) of a DevSecOps (DSO) System in order to be able to:

- Specify the DSO requirements to the lead system integrators who need to develop a platform-specific weapon solution that includes the weapon system and CI/CD pipeline
- Assess and analyze alternative pipeline functionality and feature changes as the weapon system evolves
- Apply DSO methods to complex weapon systems that do not follow well-established software architectural patterns commonly used in industry
- Provide a basis for threat and attack surface analysis to build a cyber assurance case

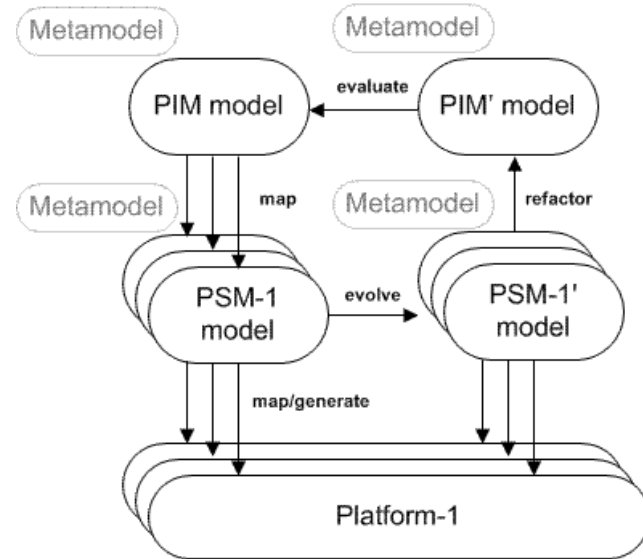
Reference Architecture/Platform Independent Model (PIM)

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 [18].



NOTE: PSM = Platform Specific Model

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.



Enterprise Architecture (EA) Approach

Why EA?: To provide the knowledge and tools that will enable program offices to adopt DSO in support of the new software acquisition pathway [9].

An EA approach will expand the focus of DSO to the whole enterprise.

- What capabilities does the enterprise need to be successful?
- How must existing functional roles (e.g., program managers, finance, contracting, logistics, security, engineers) adapt to succeed in DSO?
- What new roles (e.g., product development, software engineering, information security, IT operations) are required?
- What critical processes must be implemented for DSO and what information do they need?
- What existing policies, guidelines, and best practices provide requirements and constraints for the enterprise to achieve its DSO goals?

Goal: What is the *Minimum Viable Enterprise* that is capable of defining, developing, and operating a *Minimum Viable Product*?

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.

Examples of applying model-based engineering to complex socio-technical systems include the following:

- A System Integrator used model-based engineering for several border security projects [4]
- The DoD using model-based engineering for designing forward base camp [5]
- An evaluation of Norwegian pension system [2]

Future: Program Office Topple



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

Combining the DSO PSM with the system's architecture to build the single architecture, enables program offices to become organizations driven by smart automation, where delivery of a secure and resilient application quickly is the objective.

Through proper balance, programs will be able “to maintain a constant pace (i.e., play Topple) indefinitely.” [11]

Contact Information



Carol Woody, Ph.D.

cwoody@cert.org

Timothy A. Chick

tchick@sei.cmu.edu

Web Resources

<https://sei.cmu.edu/>

References

- [1] Haskins, C. 2008a. "Using patterns to transition systems engineering from a technological to social context," Systems Engineering, 11: 147–155.
- [2] Palmer, E. 2016. "Investigating structural gender inequality in the Norwegian pension system: An example of using MBSE in the evaluation of social systems," 26th Annual INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18-21, 2016
- [3] Oosthuizen, R., Venter, J.P., Serfontian, C. 2018 "Model Based Systems Engineering Process for Complex Command and Control Systems," 23rd International Command and Control Research and Technology Symposium (ICCRTS 2018), Pensacola, USA, 6-9 November 2018
- [4] Asan, E., Albrecht, O., Bilgen, S. 2013 "Handling Complexity in System of Systems Projects – Lessons Learned from MBSE Efforts in Border Security Projects," 4th International Conference on Complex System Design & Management, pp. 281-299.
- [5] Miller, Lori Ann, "Modeling forward base camps as complex adaptive sociotechnical systems" (2012). Masters Theses. 6943.
- [6] SEBoK, [https://www.sebokwiki.org/wiki/Sociotechnical_System_\(glossary\)](https://www.sebokwiki.org/wiki/Sociotechnical_System_(glossary))
- [7] Information system, https://en.wikipedia.org/wiki/Information_system
- [8] DoD Enterprise DevSecOps Reference Design, https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583
- [9] DoD Software Acquisition Pathway Interim Policy and Procedures, <https://aaf.dau.edu/aaf/software/>
- [11] Principles behind the Agile Manifesto, <https://agilemanifesto.org/principles.html>
- [13] Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments, CMU/SEI-2020-TR-002
- [14] Len Bass, Ingo Weber, and Liming Zhu. 2015. DevOps: A Software Architect's Perspective (1st ed.). Addison-Wesley Professional.
- [15] system, <https://www.dictionary.com/browse/system>
- [16] Industry Expertise, <https://www.nomagic.com/industry-expertise>
- [17] OMG Standards for BPM, <https://www.omg.org/technology/readingroom/BPM.htm>
- [18] DoD Reference Architecture Description, https://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf