Identifying Threats Using a DevSecOps Platform Independent Model

Timothy A. Chick
CERT Systems Technical Manager, CMU-Software Engineering Institute
Adjunct Faculty Member, CMU-Software and Societal Systems Department (S3D)

Carnegie Mellon University
Software Engineering Institute

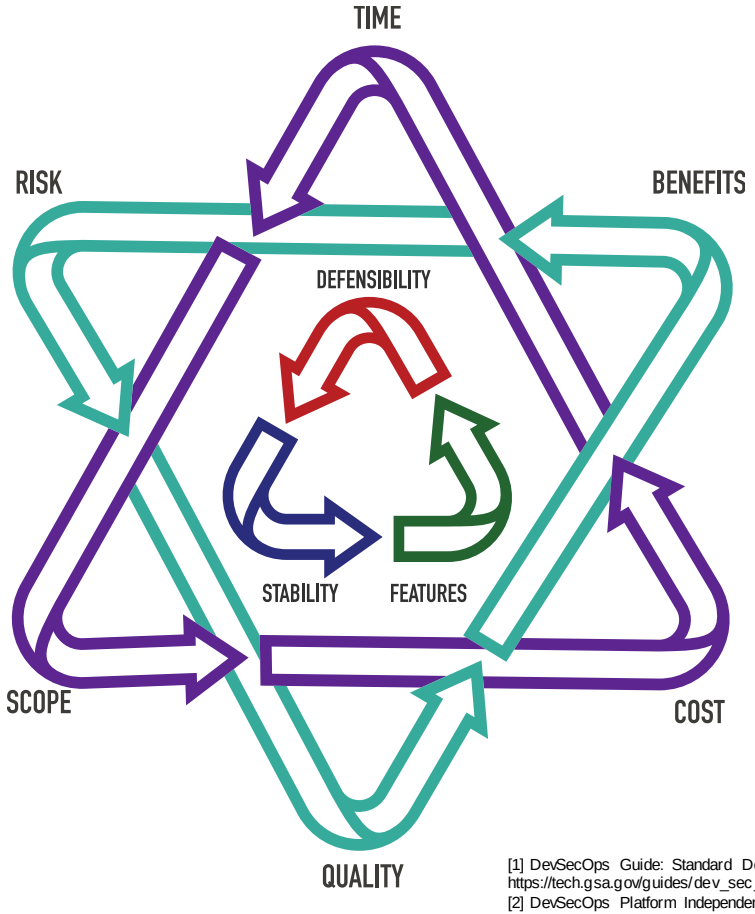
CyLab Carnegie Mellon University Security and Privacy Institute

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.

DM22-0944

DevSecOps: Modern Software Engineering Practices and Tools that Encompass the Full Software Lifecycle



DevSecOps is 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].

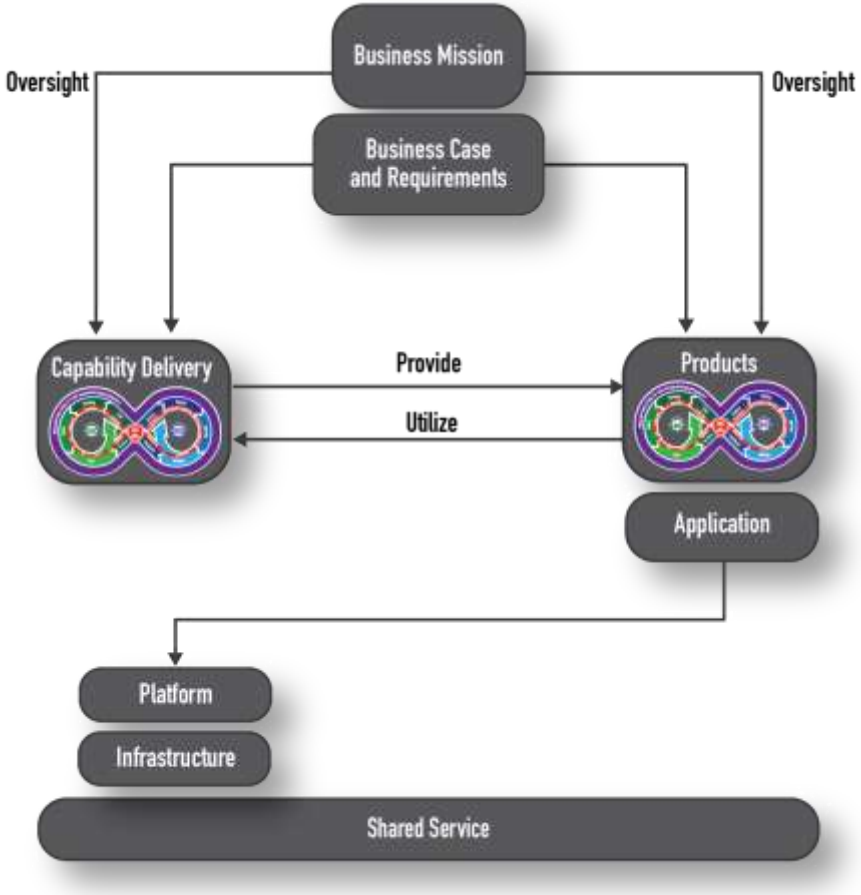
A **DevSecOps Pipeline** attempts to seamlessly integrate "three traditional factions that sometimes have opposing interests:

- **development**; which values features;
- **security**, which values defensibility; and
- **operations**, which values stability [2]."

Not only does one need to balance the factions. They must do so in a way that balances **risk**, **quality** and **benefits** within their **time**, **scope**, and **cost** constraints.

[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
[2] DevSecOps Platform Independent Model, https://cmu-sei.github.io/DevSecOps-Model/

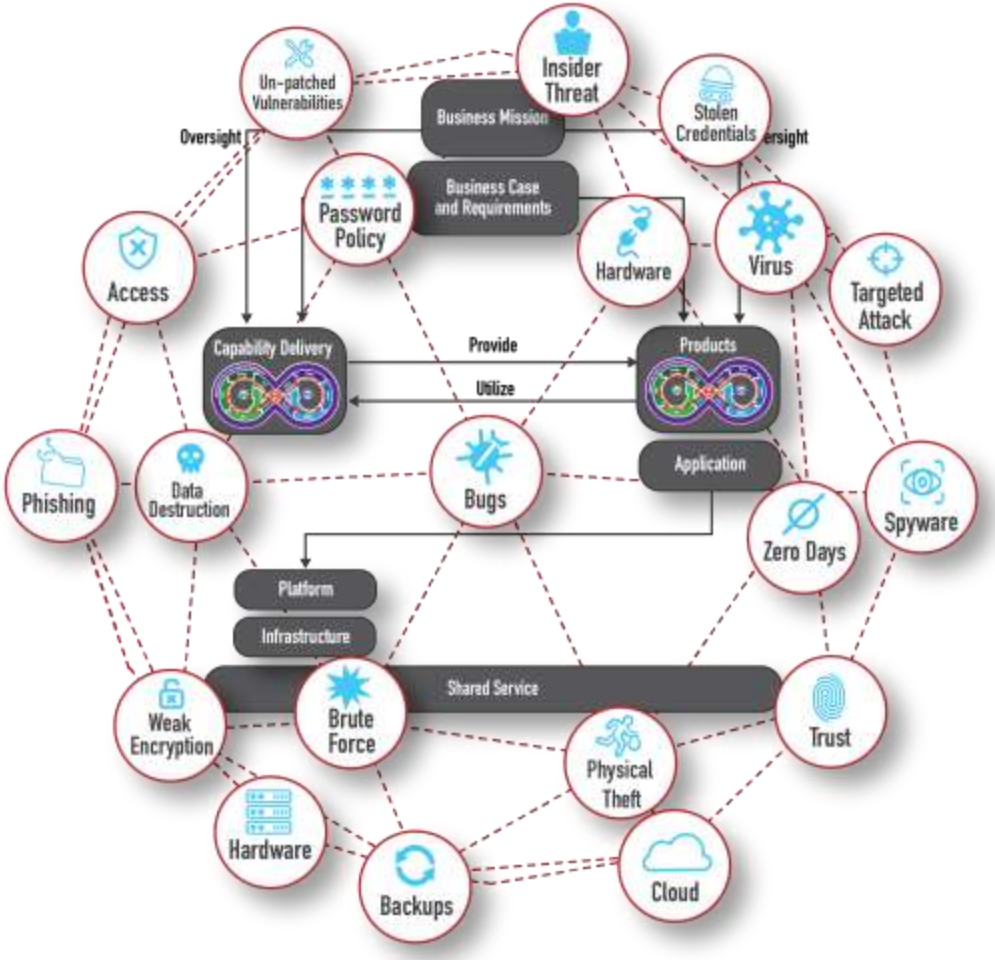
An Enterprise View



All DevSecOps-oriented enterprises are driven by three concerns:

- **Business Mission** – captures stakeholder needs and channels the whole enterprise in meeting those needs. It answer the questions *Why* and *For Whom* the enterprise exists
- **Capability to Deliver Value** – covers the people, processes, and technology necessary to build, deploy, and operate the enterprise's products
- **Products** – the units of value delivered by the organization. Products utilize the capabilities delivered by the software factory and operational environments.

Challenge: Cybersecurity of Pipeline and Product

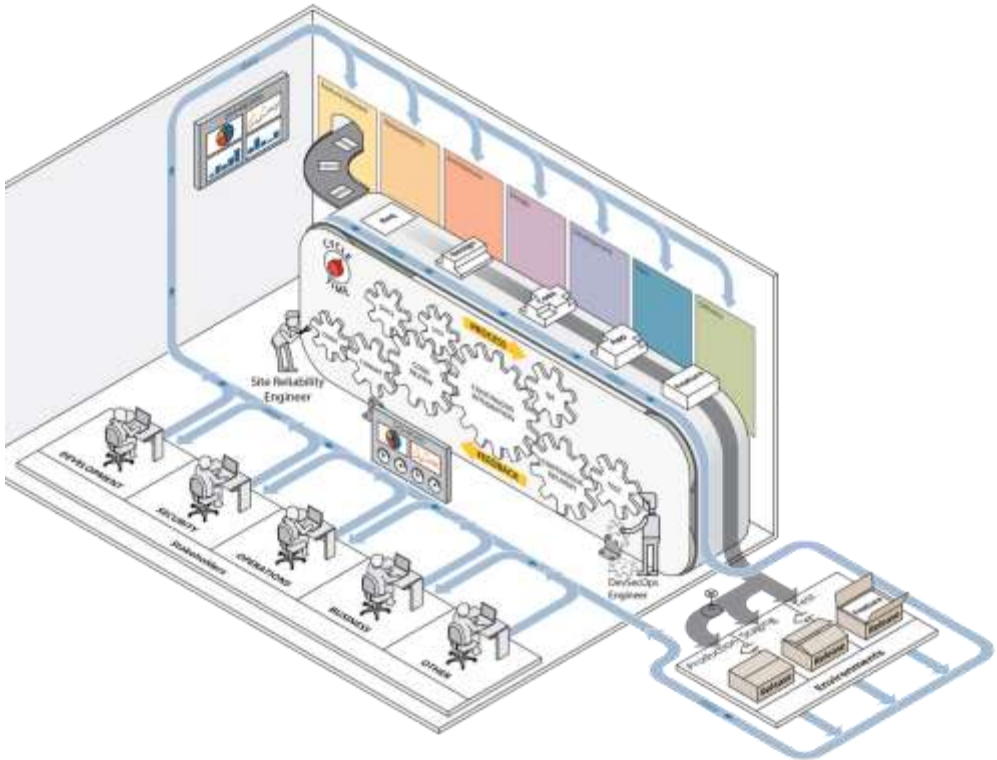


The tight integration of Business Mission, Capability Delivery, and Products, using integrated processes, tools, and people, increases the attack surface of the product under development.

Managing and monitoring all the various parts to ensure the product is built with sufficient cybersecurity and the pipeline is maintained to operate with sufficient cybersecurity is complex.

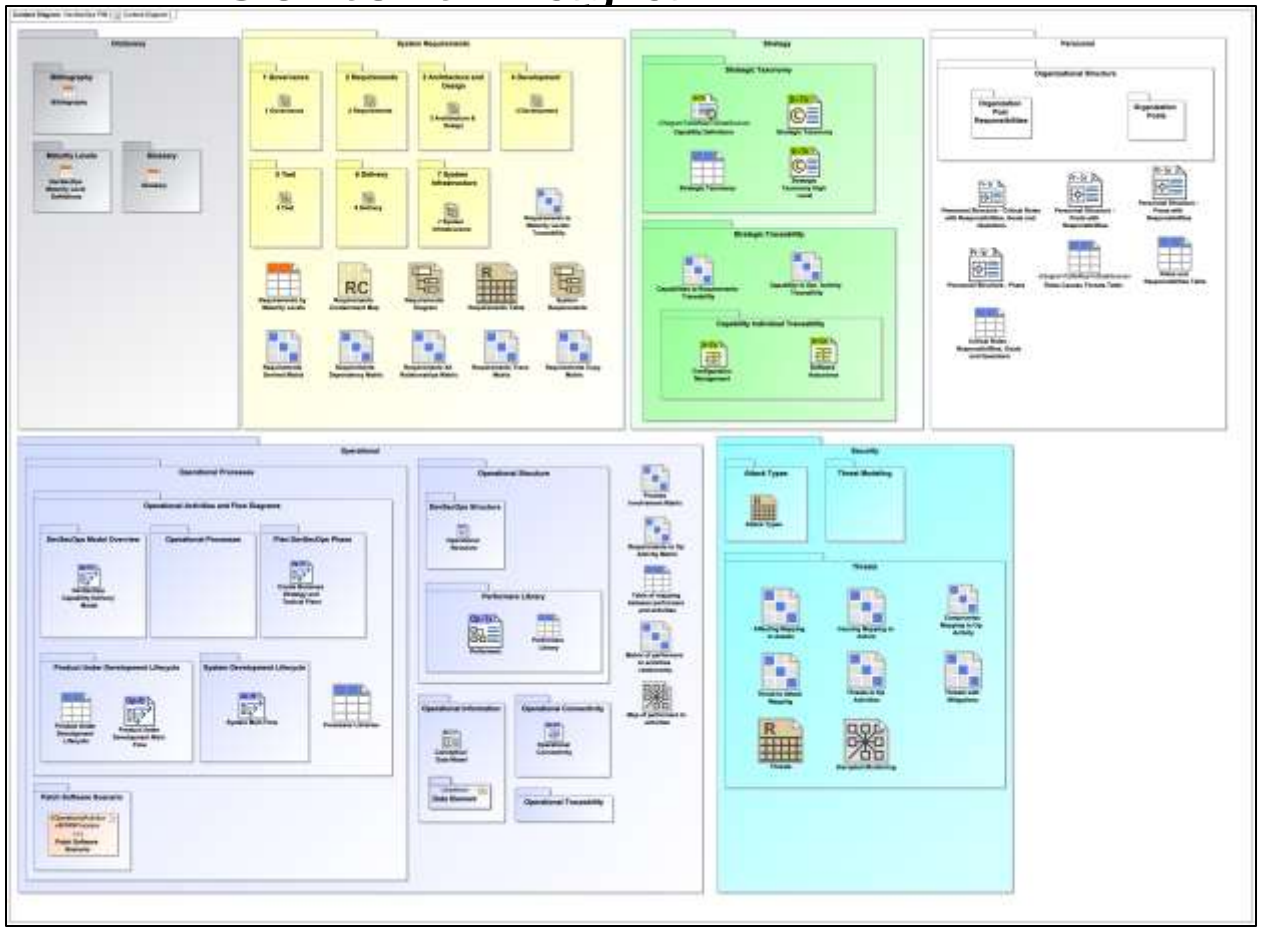
How do you focus attention to areas of greatest concern for security risks and identify the attack opportunities that could require additional mitigations?

DevSecOps Platform Independent Model (PIM)



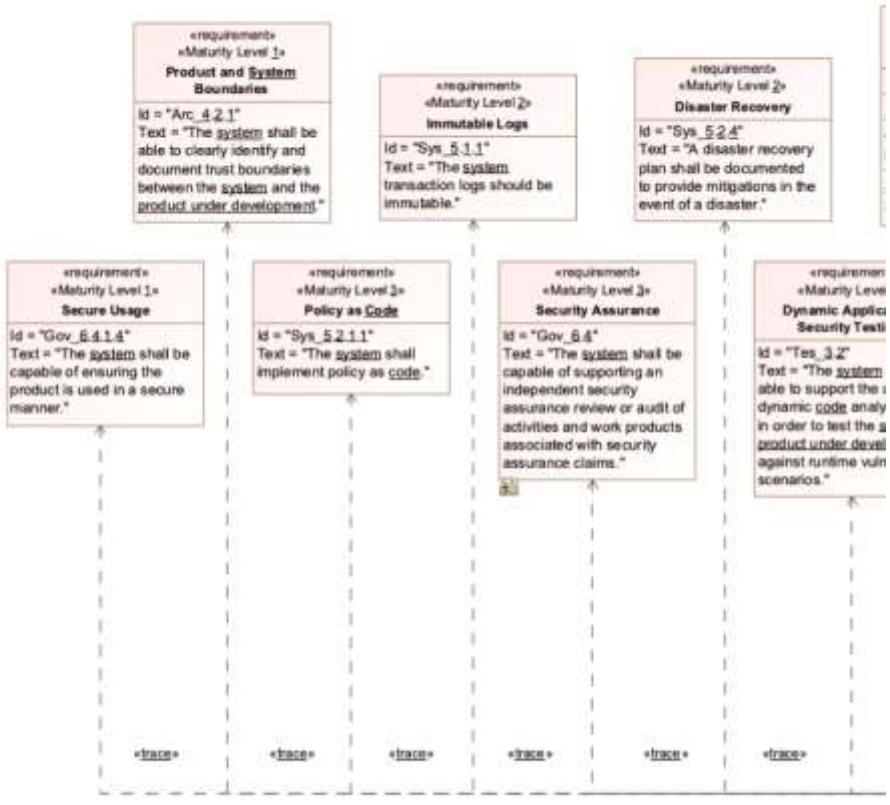
- is an authoritative reference to fully design and execute an integrated Agile and DevSecOps strategy in which all stakeholder needs are addressed
- enables organizations to implement DevSecOps in a secure, safe, and sustainable way in order to fully reap the benefits of flexibility and speed available from implementing DevSecOps principles, practices, and tools
- was developed to outline the activities necessary to consciously and predictably evolve the pipeline, while providing a formal approach and methodology to building a secure pipeline tailored to an organization's specific requirements

DevSecOps PIM - Content Diagram



https://cmu-sei.github.io/DevSecOps-Model/

DevSecOps Requirements



Example of Requirements Representation in Diagrams from PIM

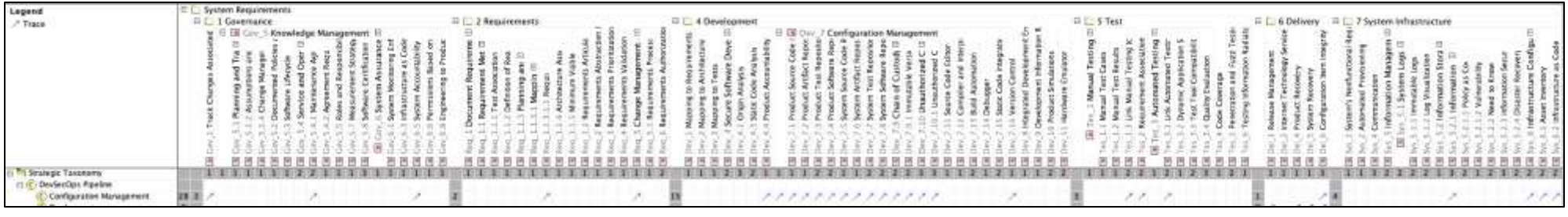
All requirements are organized into categories based on logical and functional groupings:

- Governance
- Requirements
- Architecture and Design
- Development
- Test
- Delivery
- System Infrastructure

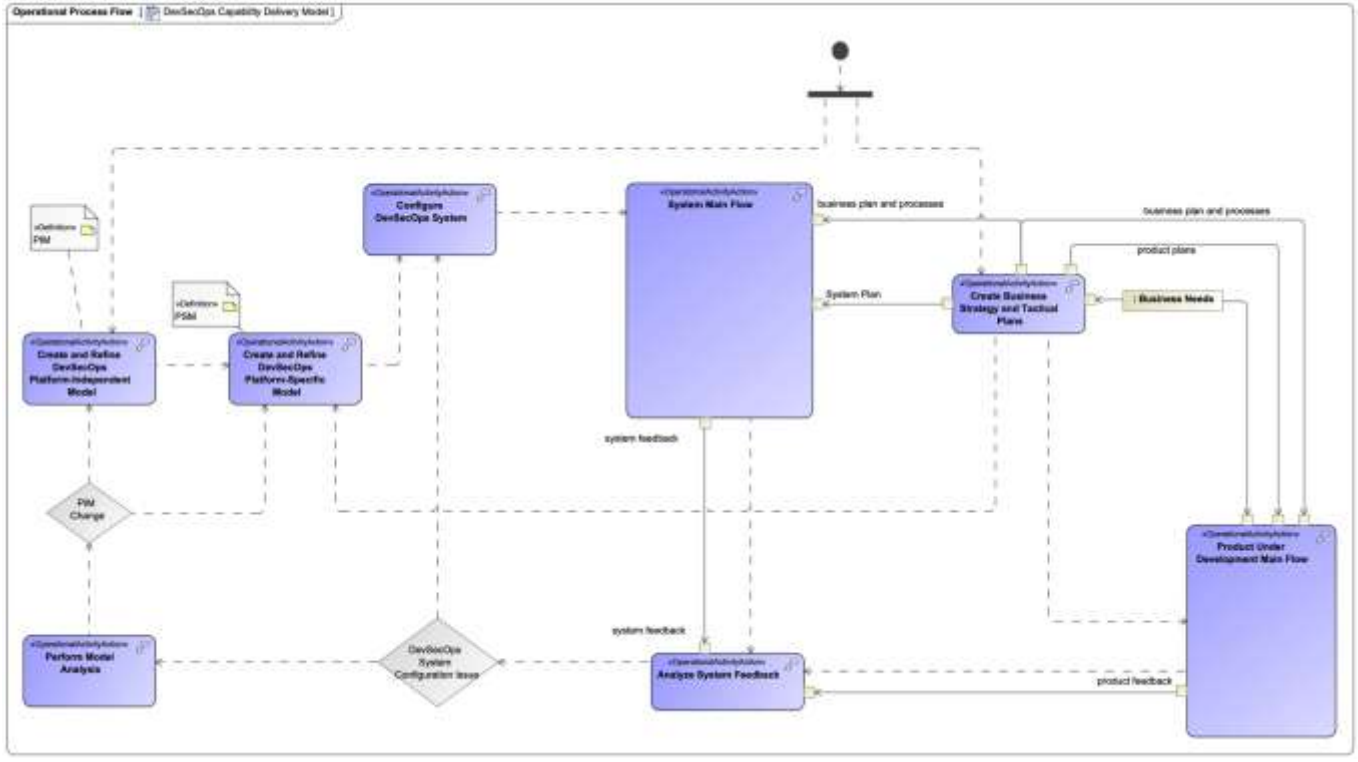
DevSecOps Capability/Strategic Viewpoint

A capability is a high-level concept that describes the ability of a system to achieve or perform a task or a mission.

All requirements in the DevSecOps PIM were allocated to corresponding capabilities.



DevSecOps Operational Viewpoints

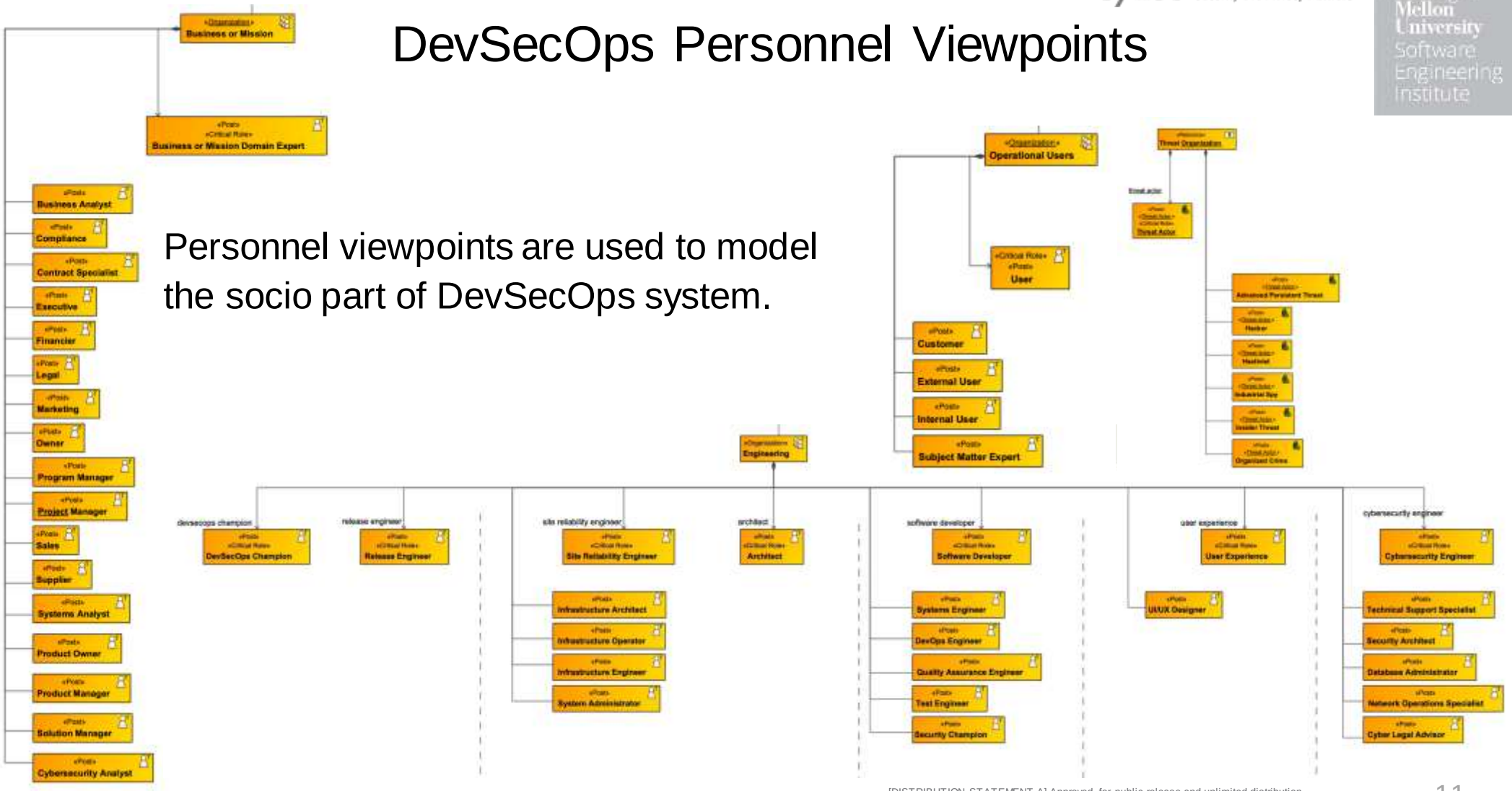


An operational model for a system describes behavior of the system to conduct enterprise operations.

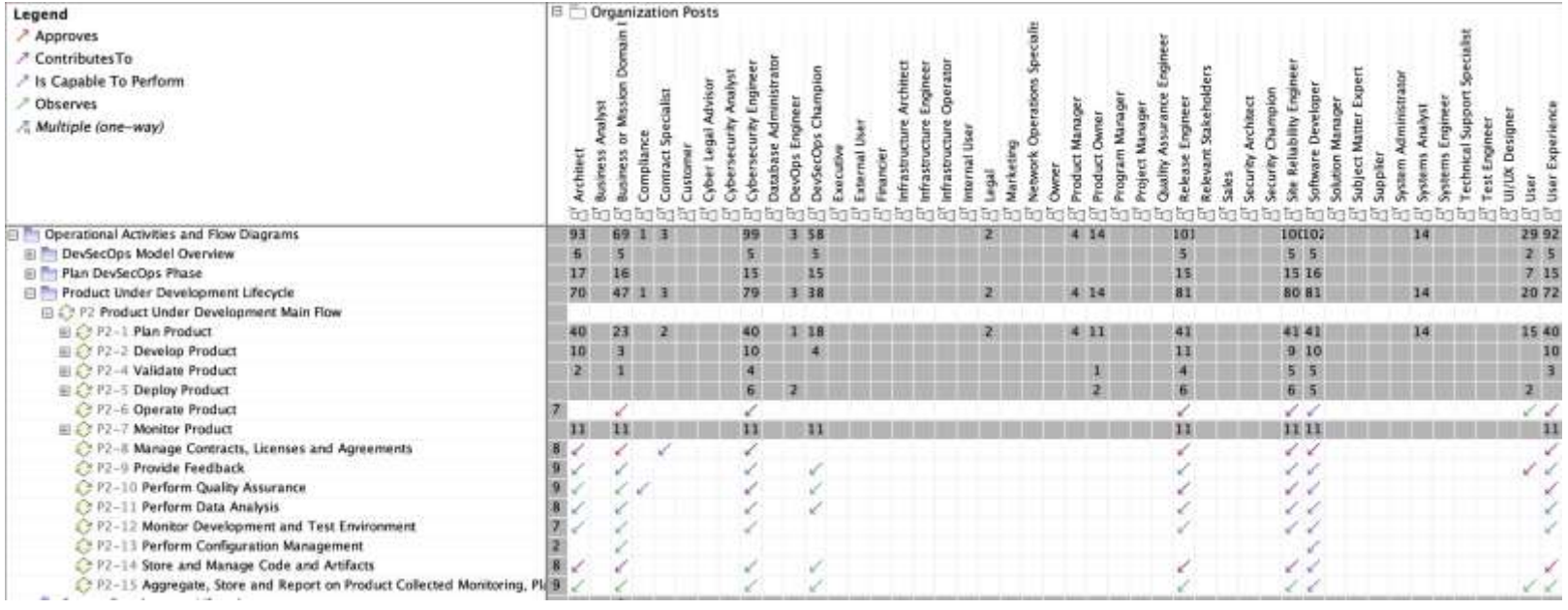
The main operational processes for DevSecOps includes development process for the product, as well as the DevSecOps process itself.

DevSecOps Personnel Viewpoints

Personnel viewpoints are used to model the socio part of DevSecOps system.



Everyone Plays a Role in DevSecOps



Critical Roles are mapped to Operational Activities.

Threat Scenario Generation Workshop

Purpose	Identify threat scenarios for a given system	
Entry Criteria:	The following Unified Architecture Framework (UAF) defined views have been created for the system under evaluation: • Requirements Diagrams • Operational Process Flows • Relationships between Operational Activities and System Requirements • Operational resource structure, Posts (i.e. roles) and corresponding responsibilities including the Involvement relationships.	
General	• As the system architecture and associated system instantiation evolves, so will the threats and corresponding mitigations. While this process defines an approach to systematically define applicable threat scenarios for the given system, threats should be identified, evaluated, and captured continuously outside this process. • During the structured and unstructured brainstorming activities, there are no right or wrong ideas. The goal is to identify any reasonable action that can be taken to exploit the various activities within the system to ultimately impact the final product. The ideas will be evaluated later in the process.	
Step	**Activities**	**Description**
1	Planning	• Identify relevant stakeholders. Participants must contain a mix of engineering, operational, user, business, and cyber security experience. • Schedule a date and time, or series of events, in which all relevant stakeholders can actively participate.
2	Kick-off Event	• Review the workshop process and introduce participants • Discuss the goals and objectives of the workshop • Introduce participants to the concept of system threats and review a few example threat scenarios that follow the format of the Threat Scenario Template.

3	System and Architectural Overview	• Outline system purpose and constraints
		• Review system's architectural views and relationships
		  ○ Requirements
		  ○ Strategy
		  ○ Personnel
		  ○ Operational

4	Operational Process Flow Focus Area	• Select an operational process flow to focus the threat scenario generation • Review the selected operational process flow to gain understanding of the process, data flow between operational activities, and performers involved. This may include reviewing associated requirements to understand the scope and context of the various operational activities.
5	Unstructured Brainstorming	• Select an operational activity within the operational process flow • Either working individually or in pairs, brainstorm threats for the selected operational activity and write them down. Threats can bridge multiple operational activities. The brainstormed ideas should be captured in the individual's natural language. • Using an affinity diagram, organize the threats identified by the whole group and remove duplicates. • Create a list of potential threats to the system.
6	Structured Brainstorming	• Use the same operational activity as in step 5. • Break into groups of 2-3 people. • In small groups, identify ways that the operational activity may be exploited to interrupt the confidentiality, integrity, and/or availability of the system. Utilize the Process Specific STRIDES Threat Modeling Taxonomy to reduce individual bias and to holistically identify threats to the given activity. • Using an affinity diagram, organize the threats identified by the whole group and remove duplicates. • Add new threats to the list of potential threats to the system created in step 5.
7	Define Threat Scenarios	• If this is the first time any of the participates have written threat scenarios, select a threat from the list and complete the Threat Scenario Template as a group. Repeat until everyone understands how to complete the Threat Scenario Template. • Break into small groups of 3-4 people. • Divide the list of potential threats to the system between the small groups. Alternatively, create a pull system in which the small groups claim a potential threat from a centralized list as needed. • In small groups, complete the Threat Scenario Template for each assigned, or pulled, potential threat. • Review and update all completed threat scenarios as a whole group, removing or consolidating duplicates.
8	Operational Activity Threat Identification	• Select next operational activity within the selected operational process flow. • Repeat steps 5-7. • Repeat step 8 until threats have been identified for all operational activates within the selected operational process flow.
9	Identify Operational Process Flow Threats	• Repeat steps 4-8 until threats have been identified for all operational process flows for the given system.
10	Consolidate and Review	• Consolidate all threat scenarios into a central list. • Review and accept the threat scenarios
Exit Criteria		A list of structured threat scenarios that cover the operational activities in the given system.

Threat Scenarios

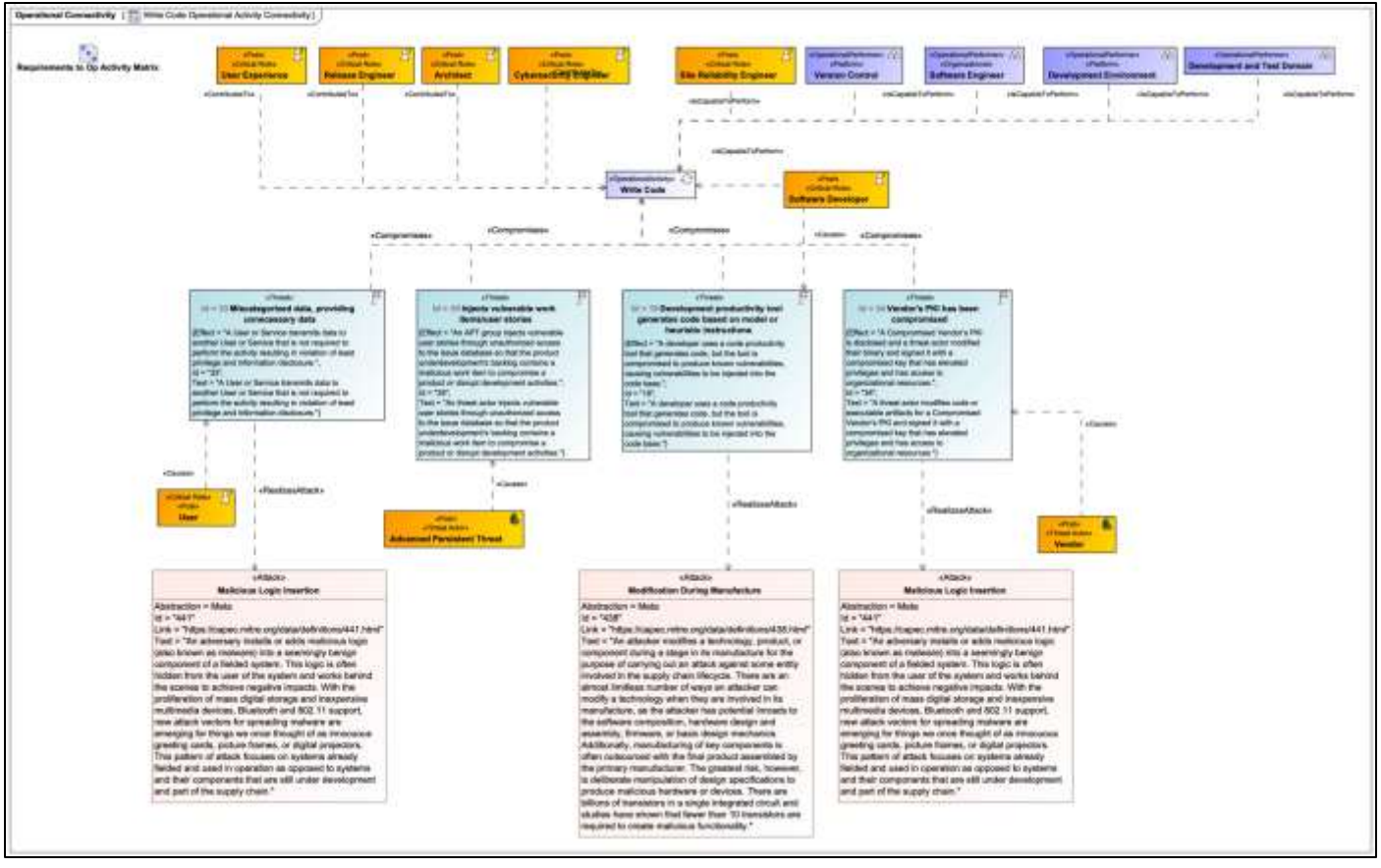
Template:

Part	Description
Activity	The activity diagrammed in the PIM or PSM. There can be more than one activity applied to the Threat Scenario.
Actor	The person, or group, that is behind the threat scenario. Threat actors can be malicious or unintentional. Developing a standard set of actors is beneficial for this step. Persona non grata could be useful in determining malicious actors. Threat actor may be a person, or group, internal to an organization structure.
Action	A potential occurrence of an event that might damage an asset, a mission, or goal of a strategic vision.
Attack	An action taken that utilizes one of more vulnerabilities to realize a threat to compromise or damage an asset, a mission, or goal of a strategic vision.
Asset	A resource, person, or process that has value.
Effect	The desired or undesired consequence resulting from the attack.
Objective	The threat actor's motivation or objective for conducting the attack
Statement	Structured prose summarizing the 6-part security scenario

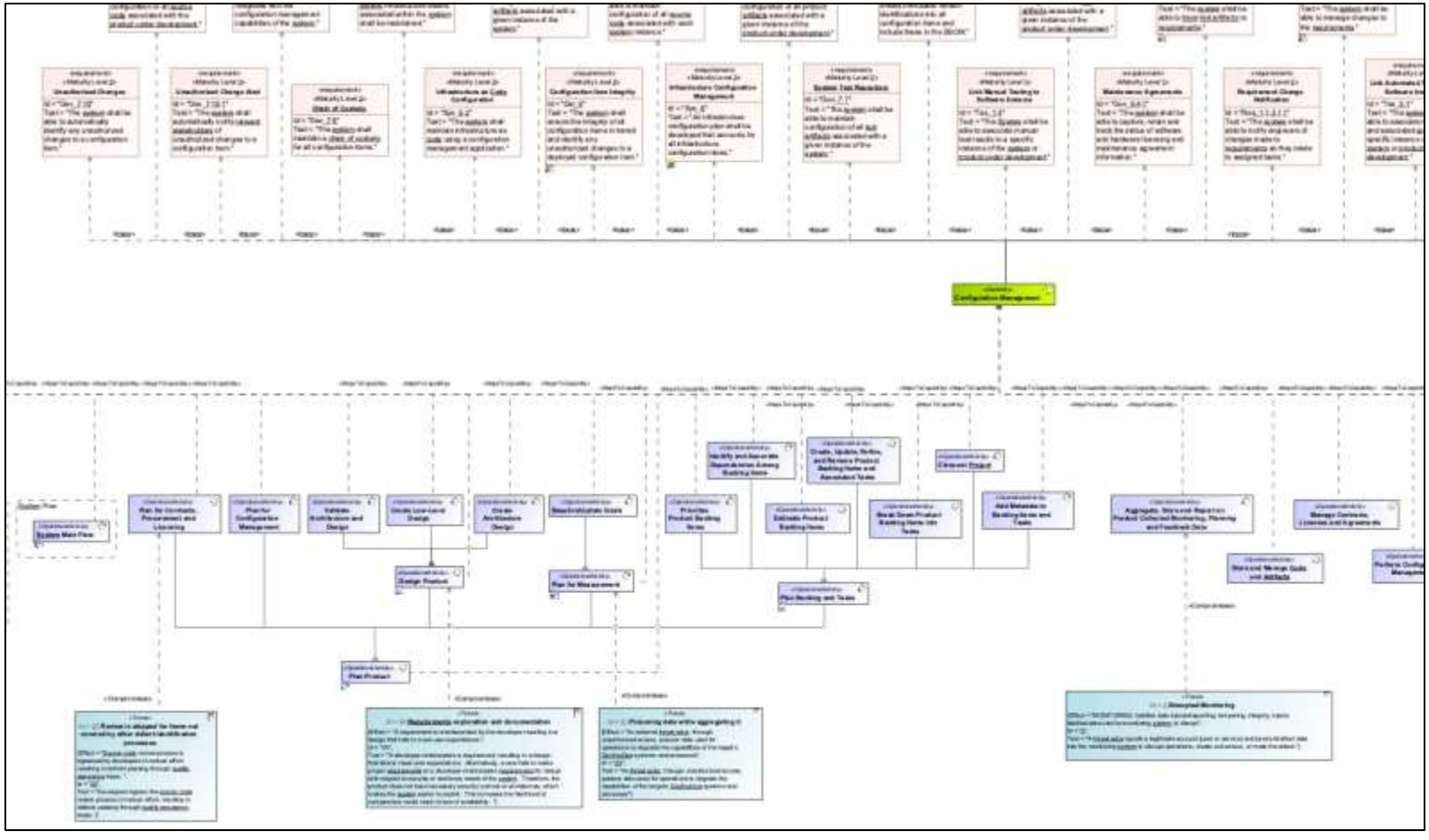
Example:

Part	Description
Activity	Develop Product, Static and Dynamic Analysis
Actor	Insider Threat
Action	Results from analysis are disclosed for effect
Attack	Information Disclosure
Asset	Analysis Results
Effect	Damage organization, vulnerabilities are publicly enumerated for a product under development
Objective	Develop a targeted exploit for the product under development, financial attack
Statement	An insider threat publicly releases the results of static and dynamic analysis to the public to damage the organization's reputation.

Example Threat Modeling Diagram for Write Code Operational Activity



Capturing the Complexity of the DevSecOps System



Example of Threats Traced to Capabilities via Operational Activities

https://cmu-sei.github.io/DevSecOps-Model/

Summary

The use of model based systems engineering in the design, implementation, and sustainment of your DevSecOps socio-technical system will assist you in building a system that is:

- Trustworthy – No exploitable vulnerabilities exist, either maliciously or unintentionally inserted.
- Predictable – When executed, software functions as intended and only as intended.
- Timely – Features are delivered as the speed of relevance.