

Modeling Requirements with MBSE

Nataliya Shevchenko

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

Copyright 2022 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.

CERT® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM22-0811

About Me



Senior Member of Technical Staff, *Security Automation Department, CERT*
20+ years of experience across Software Development Life Cycle



Software development lifecycle, software architecture principles & practices, systems engineering, MBSE, and threat modeling



BS/MS degrees in Mathematics from Donetsk State University in Ukraine
MS degrees in Software Engineering from Carnegie Mellon University



Email: san@sei.cmu.edu

Office: 412.268.9620

Publications: <https://insights.sei.cmu.edu/authors/nataliya-shevchenko/>

Agenda

- Definitions and Assumptions
 - Requirements
 - MBSE
- Requirements in SysML
- Requirements in a Model
 - Organization and Initial Analysis
 - Traceability and Relationships
 - Requirements Diagram
 - Requirements Table
- Requirements Analysis in a Model
 - Matrix
 - Metric
 - Map
 - Find Errors and Gaps
- Questions

Definitions and Assumptions



Definitions and Assumptions:

Requirements

A description of the problem(s) that future systems should solve.

Business requirements:

High-level statements of the goals, objectives, or needs of an organization.

User requirements:

Mid-level statements of the needs of a particular stakeholder or group of stakeholders.

System requirements:

Usually detailed statements of capabilities, behavior, and information that the solution will need including detailed statements of the conditions under which the solution must remain effective, qualities that the solution must have, or constraints within which it must operate. System requirements include non-functional requirements, often called [quality attributes](#) or “ilities,” such as security, usability, testability, and modifiability.

Definitions and Assumptions: ***MBSE***

Model-based systems engineering is a formalized methodology that supports the requirements, design, analysis, verification, and validation associated with the development of complex systems.

Modeling Language: SySML

Modeling Environment: Digital modeling environment

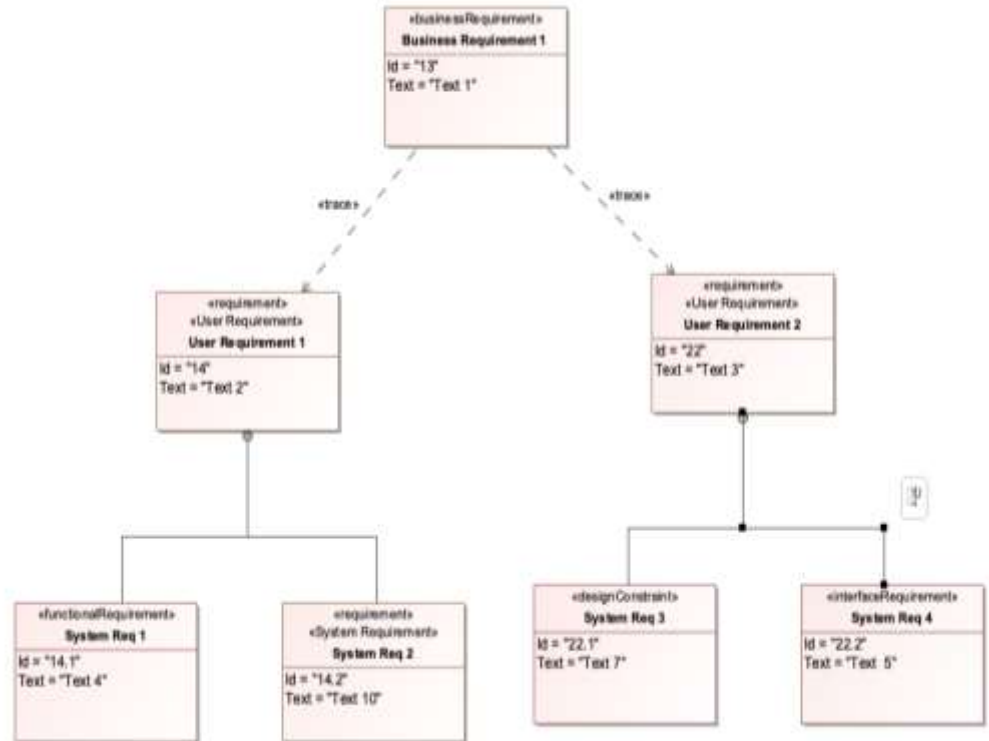
Requirements in SysML



Requirements Mapping

- Business requirement → SysML *business requirement*
- User requirement → SysML *generic requirement* with *user requirement* stereotype
- System functional requirement → SysML *generic requirement* with *system requirement* stereotype or SysML *functional requirement* subclass
- System non-functional requirements → SysML *design constraint*, *usability requirement*, *performance requirement*, *interface requirement*, *physical requirement*

Requirements Mapping: *Hierarchy*

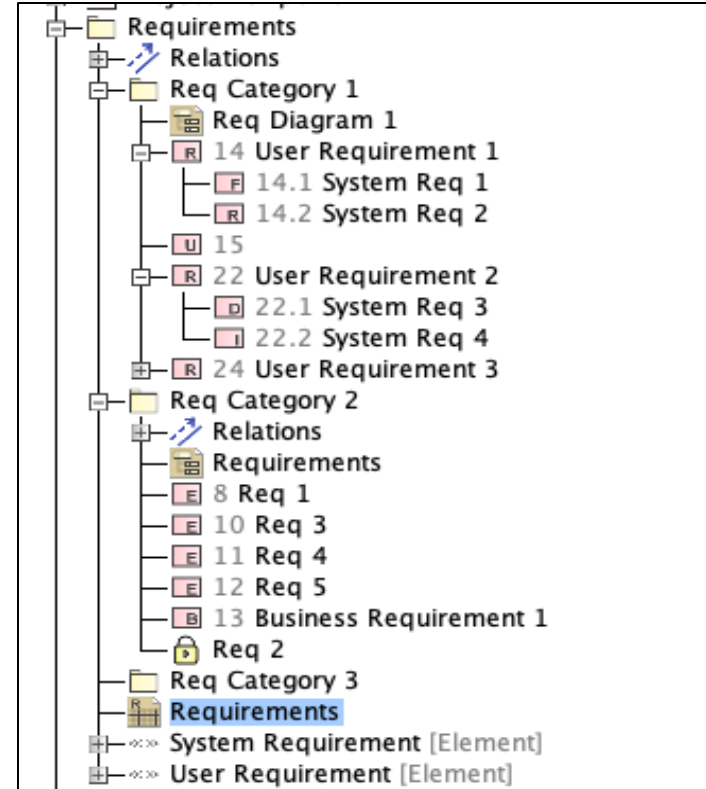


Requirements in a Model



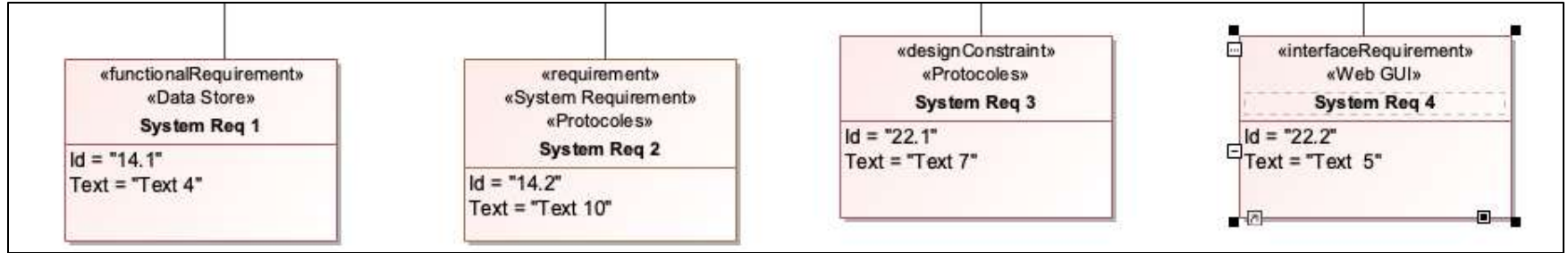
Organization and Initial Analysis: *Categorization*









- Categories based on:
 - Type of requirements
 - Functionality
 - Part of a business process
 - Subsystems
 - Components
- Categories as Packages



Organization and Initial Analysis 2

- Categories as Custom Stereotypes



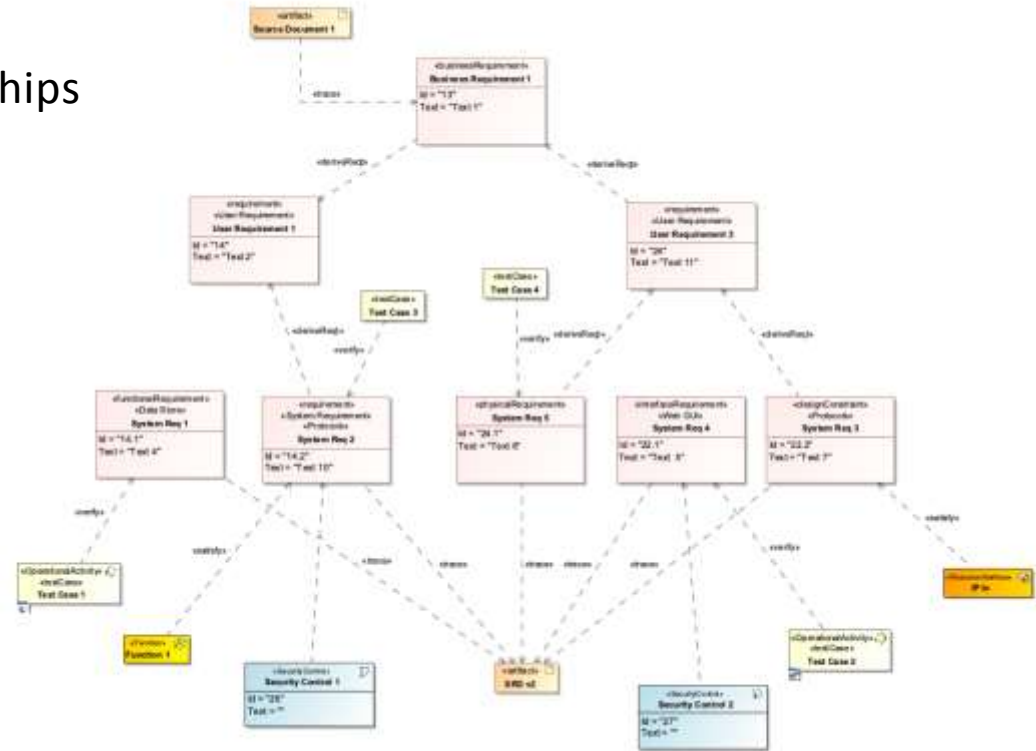
#	Name	Text	Applied Stereotype
1	 14.1 System Req 1	Text 4	 functionalRequirement [Class] «>> Data Store [Element]
2	 14.2 System Req 2	Text 10	 Requirement [Class] «>> System Requirement [Element] «>> Protocoles [Element]
3	 22.1 System Req 3	Text 7	 designConstraint [Class] «>> Protocoles [Element]
4	 22.2 System Req 4	Text 5	 interfaceRequirement [Class] «>> Web GUI [Element]

Traceability and Relationships

- Requirement was published in the System Requirements Document (SRD) – **Trace**
- Other requirements were derived from this requirement – **Derive**
- Requirement was decomposed – **Containment**
- Requirement refines any use case – **Refine**
- Requirement was verified by a test case or test process – **Verify**
- This requirement is the same as another requirement – **Copy**
- Parts of the conceptual and/or solution architecture is satisfying the requirement – **Satisfy**

Requirements Diagram

- Depict all elements and relationships directly related to a requirement
- Assist with the analysis



Requirements Table

- Traditional and convenient way to look at the requirements
- Convenient and effective way to import the requirements by bulk











Id	Name	Text	Traced To	Satisfied By	Derived From	Verified By
	Business Capabilities					
32	32 Traveler Information Management	Text 3				
32.1	32.1 Traveler Profile Management	Text 4		Traveler Profile Management System	32 Traveler Information Management	
32.2	32.2 Traveler History Management	Text 5			32 Traveler Information Management	Test Case 1
34	34 Communication Management	Text 6				
34.1	34.1 Mobile Communication Management	Text 7			34 Communication Management	
34.1.1	34.1.1 Mobile Communication Security	Text 8			34.1 Mobile Communication Management	
34.2	34.2 Optic Communication Management	Text 9			34 Communication Management	
34.2.1	34.2.1 Optic Communication Security	Text 12			34.2 Optic Communication Management	Test Case 2
35	35 Information Management	Text 13				
35.1	35.1 Information Lifecycle Management	Text 14			35 Information Management	
35.2	35.2 Information Organization	Text 15			35 Information Management	
36	36 Route Management	Text 16				
36.1	36.1 Route Analysis	Text 17			36 Route Management	
36.1.1	36.1.1 Statistical Analysis	Text 18	34 User Requirement 1		36.1 Route Analysis	
36.1.2	36.1.2 Route Visualization	Text 19	35 User Requirement 2		36.1 Route Analysis	
36.2	36.2 Route Planning	Text 20			36 Route Management	
37	37 Traffic Controls Information Management	Text 21				
38	38 Travel Condition Information Management	Text 22				
39	39 Traffic Controls Strategies Management and Optimization	Text 23				

- Requirements Analysis in a Model



Coverage Metric

- To evaluate a current state of the model
- Calculates statistics of how many requirements are covered with relationship
- Allows to monitor changes of a specific aspect of the model in time

#	 Date	 Scope	 Requirements	 Covered By Design Percentage	 Covered By Design	 Covered By Test Cases Percentage	 Covered By Test Cases
1	2020.12.04 18.49	 Requirements	15	26.6667	4	13.3333	2
2	2020.12.04 18.50	 Requirements	15	33.3333	5	13.3333	2
3	2020.12.04 18.53	 Requirements	15	26.6667	4	20	3

Dependency Map












- Put a specific requirement at the top of the tree
- Shows all or selected types of relationships for the requirement
- Shows multiple levels of the relationships
- Shows how the requirement relates to other domain



Find Errors: *Requirements Table*

- Requirements can be derived only from higher level of requirements
- In the table, there is two errors

Can you find the errors?

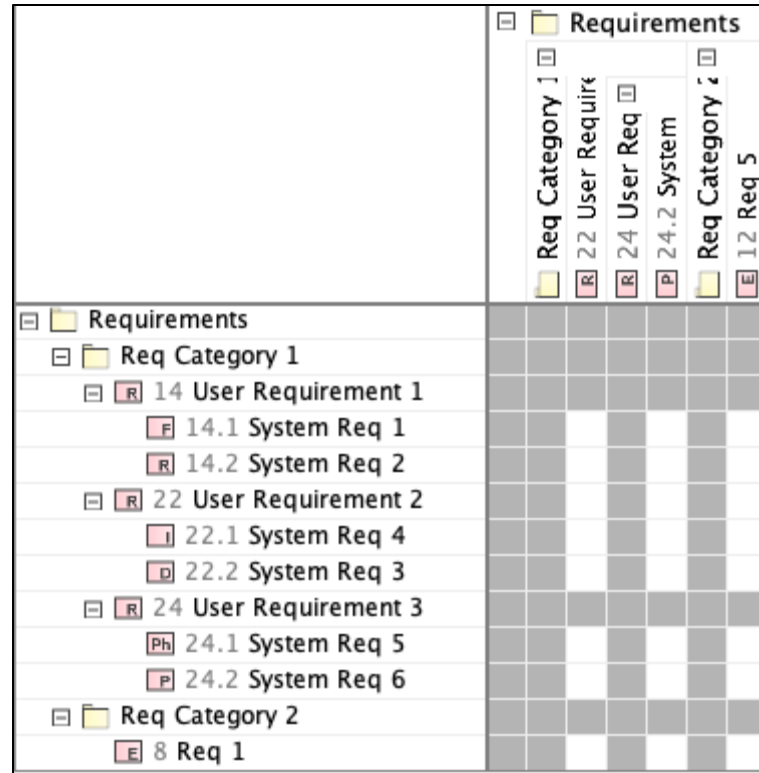
#	△ Name	Text	Derived	Derived From
1	 13 Business Requirement 1	Text 1	 24 User Requirement 3	 14 User Requirement 1
2	 14.1 System Req 1	Text 4	 14 User Requirement 1	
3	 14.2 System Req 2	Text 10		 14 User Requirement 1
4	 22.1 System Req 4	Text 5		 22 User Requirement 2
5	 22.2 System Req 3	Text 7		 24 User Requirement 3

Business requirement cannot be derived from User requirement

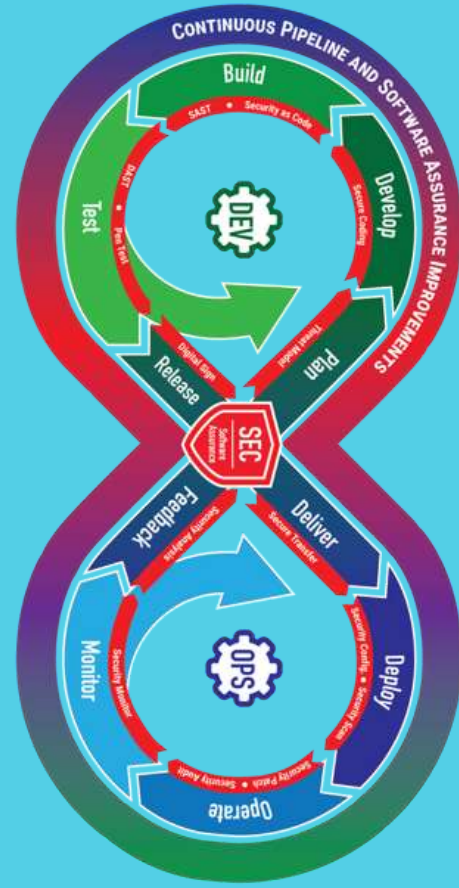
User requirement cannot be derived from System requirement

Find Gaps: “Negative” Dependency Matrix

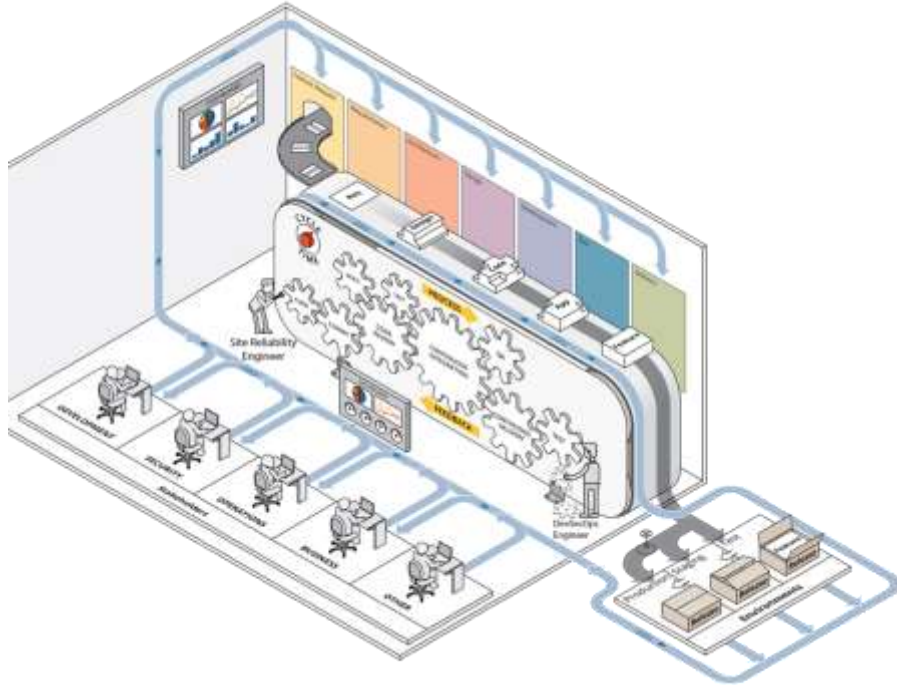
- To file a “negative” space – elements without the relationships



- Requirements in Real Model



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 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

Legend	
	Trace

System Requirements		
	DevSecOps Pipeline [Strategic Taxonom]	
	Configuration Management	28
	Deployment	10
	Hosting Services	37
	Integration	6
	Monitor & Control	50
	Planning & Tracking	34
	Quality Assurance	17
	Software Assurance	65
	Solution Development	41
	Verification & Validation	25

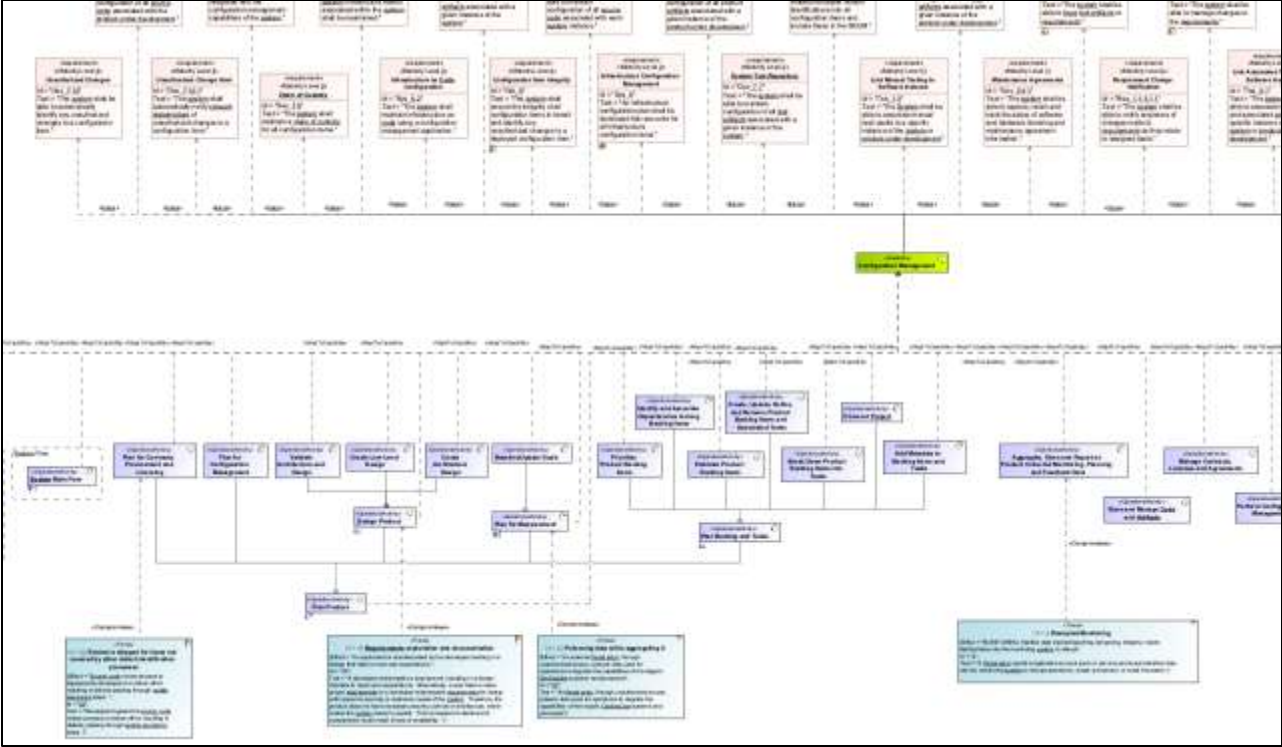
Legend	
	Trace

System Requirements	
1 Governance	
2 Knowledge Management	
3 Requirements	
4 Development	
5 Configuration Management	
6 Test	
7 Delivery	
8 System Infrastructure	

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

DevSecOps is a complex interconnected system

A Capability traced to Requirements



DevSecOps Requirements are Satisfied by Op. Activities

Legend		System Requirements	
Satisfies		1 Governance	
<ul style="list-style-type: none"> Gov_1 Track Changes Associated to Gov_2 Track Progress with Sc Gov_2.1 Provide Filtered Views Gov_2.2 Road Mapping Gov_2.3 Capture Work Gov_2.3.1 Remove Work Gov_2.3.2 Add Work Gov_2.4 Team and Organiz Gov_2.4.1 Subordinate Plans Gov_2.4.2 Plan Transparency Gov_3 Task Creation Gov_3.1 Task Metadata Gov_3.1.1 Requirements Gov_3.1.1.1 Mapping Req Gov_3.1.2 Task Assignment Gov_3.2 Dependency Tracking Gov_4.1 Development Progress Gov_4.2 Requirements Metrics Gov_4.3 Code Coverage Metrics Gov_4.4 Continuous Data Monit Gov_4.5 Security Metrics Gov_4.6 Stakeholder Metrics Gov_5.1 Planning and Trac Gov_5.1.1 Commitment Gov_5.1.1.1 External Con Gov_5.1.1.2 Changes to C Gov_5.1.2 Assumptions and Gov_5.1.3 Stakeholder F Gov_5.1.3.1 Requirement Gov_5.1.3.2 Test Active Gov_5.1.3.3 Quality Assur Gov_5.1.3.4 Configurator Gov_5.1.3.5 Agreement L Gov_5.1.3.6 Risk Activit Gov_5.1.4 Change Managem Gov_5.2 Documented Policies ar Gov_5.5 Software Lifecycle Gov_5.4 Service and Opera Gov_5.4.1 Maintenance Age Gov_5.4.2 Agreement Requir Gov_5.5 Roles and Responsibility Gov_5.6 Decision Points Gov_5.6.1 Automated D Gov_5.6.1.1 Automated C Gov_5.6.2 Decision Point Not Gov_5.6.3 Decision Point Dat Gov_5.6.4 Decision Logging Gov_5.7 Measurement Strategy Gov_5.8 Software Certification Gov_6.1 System Monitoring Info Gov_6.2 System Noncompl Gov_6.2.1 Embedded System Gov_6.3 Infrastructure as Code Gov_6.4 Security Risk 			
Plan DevSecOps Phase			
P4 Create Business Strategy and Tactics	9	2	6
Product Under Development Lifecycle	3	2	6
P2 Product Under Development Main Phases	14	17	8
P2-1 Plan Product	4	5	2
P2-2 Develop Product	5	6	2
P2-2-1 Select Unit of Work	13	13	6
P2-2-2 Plan and Detail Design	6	6	3
P2-2-4 Write Code	3	3	3
P2-2-5 Write Dev Tests	3	3	3
P2-2-6 Review Code and Artifacts	14	14	8
P2-2-7 Execute Dev Tests	2	2	2
P2-2-8 Store and Manage Collected	14	14	4
P2-2-9 Release Code for Validation	2	2	2
P2-4 Validate Product	40	17	2
P2-4-1 Perform Static Analysis	2	2	2
P2-4-2 Build and Package Code	2	2	2
P2-4-3 Deliver to Testing/Staging	7	7	7
P2-4-4 Perform Dynamic Analysis	1	1	1
P2-4-5 Deliver to Production	2	2	2
P2-5 Deploy Product	2	2	2
P2-6 Operate Product	4	4	4
P2-7 Monitor Product	1	1	1
P2-8 Manage Contracts, Licenses and	1	1	1
P2-9 Provide Feedback	1	1	1
P2-10 Perform Quality Assurance	1	1	1
P2-11 Perform Data Analysis	1	1	1

Questions

