Carnegie Mellon University
Software Engineering Institute

DEVSECOPS PLATFORM-INDEPENDENT MODEL: OPERATIONAL AND PERSONEL

Timothy A. Chick, Brent Frye, Bill Nichols, Chris Miller, Mary Popeck, Aaron Reffett, Geoffrey Sanders, Natasha Shevchenko, Carol Woody, Joseph Yankel

December 2021

Introduction

Refer to DevSecOps Platform-Independent Model: Requirements and Capabilities, CMU/SEI-2021-TR-010, for an Executive Summary and Introduction to the DevSecOps Platform-Independent Model. The referenced technical report covers the Dictionary, System Requirements and Strategy elements of the platform-independent model (PIM). The purpose of this paper is to cover the remaining PIM elements, which are Operational and Personnel. A summary of the PIM elements is shown in Figure 1 below.

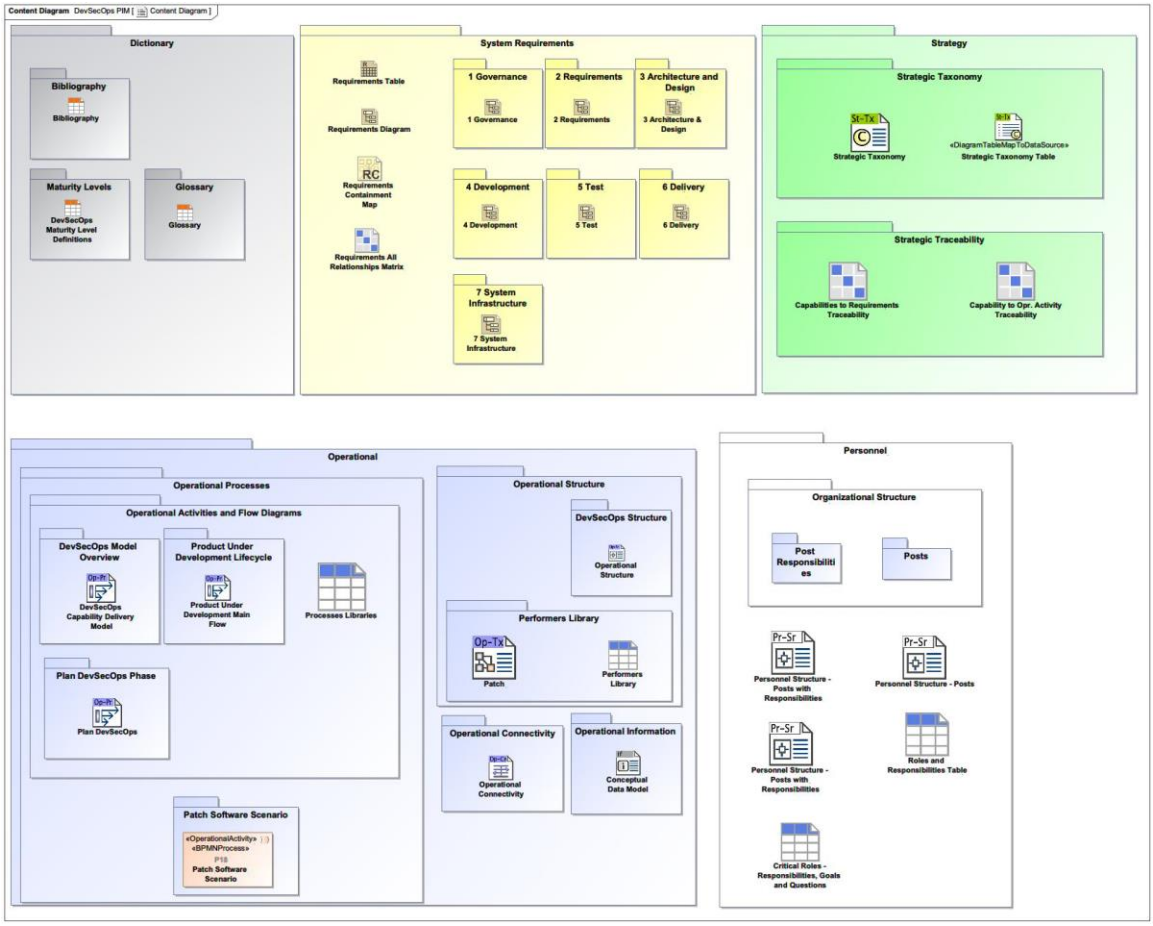

Figure 1:Platform-Independent Model Content Diagram

Operational

Operational Structure

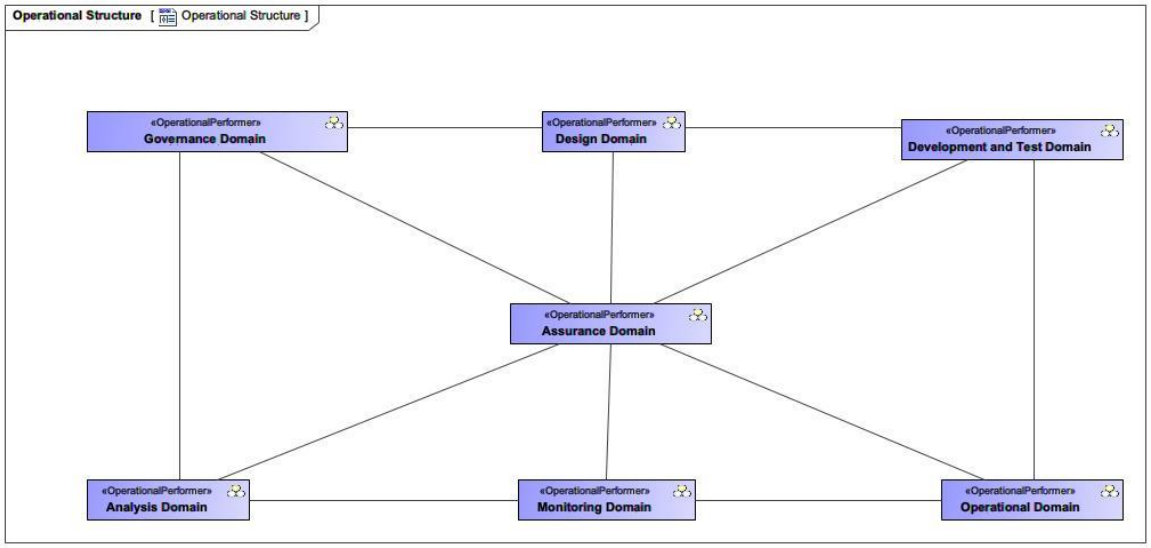

Figure 2: Operational Structure

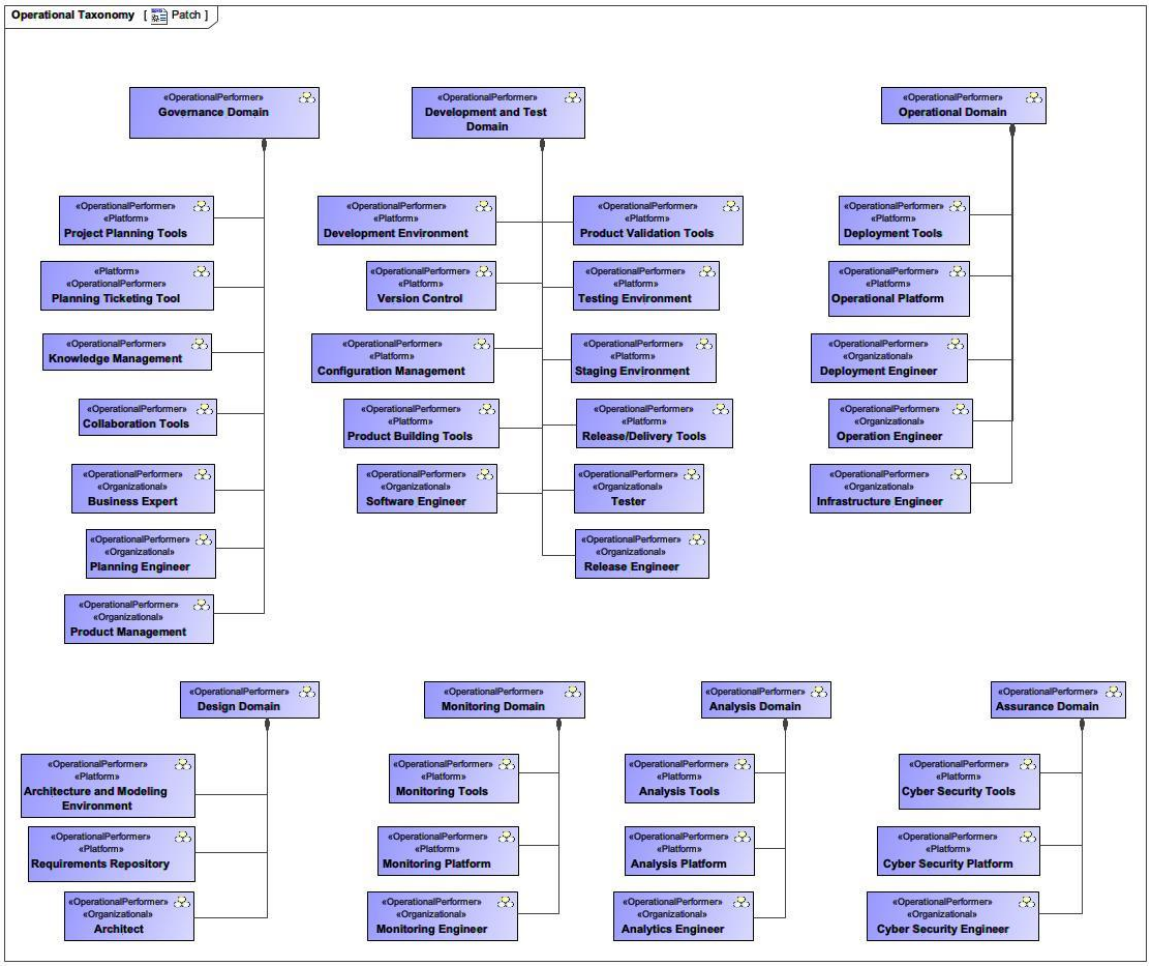

Figure 3: Operational Taxonomy

Table 1: Performers Library

#	Name	Attribute
1	Analysis Domain	analysis Tools : Analysis Tools analysis Platform : Analysis Platform analytics Engineers : Analytics Engineer monitoring Environment : Monitoring Domain cyber Security Environment : Assurance Domain governance Domain : Governance Domain
2	Analysis Platform	analysis Environment : Analysis Domain
3	Analysis Tools	analysis Environment : Analysis Domain
4	Analytics Engineer	analysis Environment : Analysis Domain
5	Architect	plan and Design Environment : Design Domain
6	Architecture and Modeling Environment	plan and Design Environment : Design Domain
7	Assurance Domain	cyber Security Tools : Cyber Security Tools cyber Security Platform : Cyber Security Platform

		cyber Security Engineers : Cyber Security Engineer development Environment : Development and Test Domain operational Environment : Operational Domain monitoring Environment : Monitoring Domain analysis Environment : Analysis Domain design Domain : Design Domain governance Domain : Governance Domain
8	Business Expert	governance Domain : Governance Domain
9	Collaboration Tools	governance Domain : Governance Domain
10	Configuration Management	development Environment : Development and Test Domain
11	Cyber Security Engineer	cyber Security Environment : Assurance Domain
12	Cyber Security Platform	cyber Security Environment : Assurance Domain
13	Cyber Security Tools	cyber Security Environment : Assurance Domain
14	Deployment Engineer	operational Environment : Operational Domain
15	Deployment Tools	operational Environment : Operational Domain
16	Design Domain	development Domain : Development and Test Domain assurance Domain : Assurance Domain architects : Architect requirements Repository : Requirements Repository architecture and Modeling Environment : Architecture and Modeling Environment governance Domain : Governance Domain
17	Development and Test Domain	configuration Management : Configuration Management version Control : Version Control developers : Software Engineer building Tools : Product Building Tools cyber Security Environment : Assurance Domain design Domain : Design Domain testing Environment : Testing Environment unit Testing Tools : Unit Testing Tools policy Validation Tools : Policy Validation Tools automated Testing Tools : Product Validation Tools staging Environment : Staging Environment release/Delivery Tools : Release/Delivery Tools testers : Tester release Engineers : Release Engineer operational Environment : Operational Domain development Environment : Development Environment
18	Development Environment	version Control : Version Control development and Test Domain : Development and Test Domain
19	Governance Domain	project Planning Tools : Project Planning Tools planning Ticketing Tool : Planning Ticketing Tool

		knowledge Management : Knowledge Management collaboration Tools : Collaboration Tools business Experts : Business Expert design Domain : Design Domain assurance Domain : Assurance Domain analysis Domain : Analysis Domain planning Team : Planning Engineer product Management : Product Management
20	Infrastructure Engineer	operational Domain : Operational Domain
21	Knowledge Management	governance Domain : Governance Domain
22	Monitoring Domain	monitoring Tools : Monitoring Tools monitoring Platform : Monitoring Platform monitoring Engineers : Monitoring Engineer operational Environment : Operational Domain analysis Environment : Analysis Domain cyber Security Environment : Assurance Domain
23	Monitoring Engineer	monitoring Environment : Monitoring Domain
24	Monitoring Platform	monitoring Environment : Monitoring Domain
25	Monitoring Tools	monitoring Environment : Monitoring Domain
26	Operation Engineer	operational Environment : Operational Domain
27	Operational Domain	deployment Tools : Deployment Tools operational Platform : Operational Platform deployment Engineers : Deployment Engineer operation Engineers : Operation Engineer testing and Integration Environment : Development and Test Domain monitoring Environment : Monitoring Domain cyber Security Environment : Assurance Domain infrastructure Team : Infrastructure Engineer
28	Operational Platform	operational Environment : Operational Domain
29	Planning Engineer	governance Domain : Governance Domain
30	Planning Ticketing Tool	governance Domain : Governance Domain
31	Policy Validation Tools	development and Testing Domain : Development and Test Domain
32	Product Building Tools	development Environment : Development and Test Domain
33	Product Management	governance Domain : Governance Domain
34	Product Validation Tools	testing and Integration Environment : Development and Test Domain
35	Project Planning Tools	governance Domain : Governance Domain
36	Release Engineer	testing and Integration Environment : Development and Test Domain
37	Release/Delivery Tools	testing and Integration Environment : Development and Test Domain

38	Requirements Repository	plan and Design Environment : Design Domain
39	Software Engineer	development Environment : Development and Test Domain
40	Staging Environment	testing and Integration Environment : Development and Test Domain
41	Tester	testing and Integration Environment : Development and Test Domain
42	Testing Environment	testing and Integration Environment : Development and Test Domain
43	Unit Testing Tools	testing Domain : Development and Test Domain
44	Version Control	development Environment : Development and Test Domain IDE : Development Environment

Table 2: Operational Connectivity

#	Exchange ID	Operational Exchange Item	Sending Operational Performer	Receiving Operational Performer
1	OE1	IE3 Code	Development Environment	Version Control
2	OE6	IE8 Strategic Vision	Business Expert	Business Expert
3	OE7	IE9 Strategic Objectives	Business Expert	Business Expert
4	OE8	IE10 Business Requirements	Business Expert	Business Expert
5	OE12	IE11 Product Concept	Product Management	Architect
6	OE14	IE11 Product Concept	Product Management	Business Expert
7	OE15	IE12 Business Plan	Business Expert	Planning Engineer
8	OE16	IE12 Business Plan	Business Expert	Business Expert
9	OE18	IE5 Business Needs	Business Expert	Business Expert
10	OE19	IE13 Business Risks	Business Expert	Business Expert
11	OE23	IE21 Program Plans	Planning Engineer	Planning Engineer
12	OE24	IE19 MVP Specification	Product Management	Business Expert
13	OE26	IE20 DevSecOps Architecture	Architect	Product Management
14	OE27	IE17 Product Architecture	Architect	Product Management

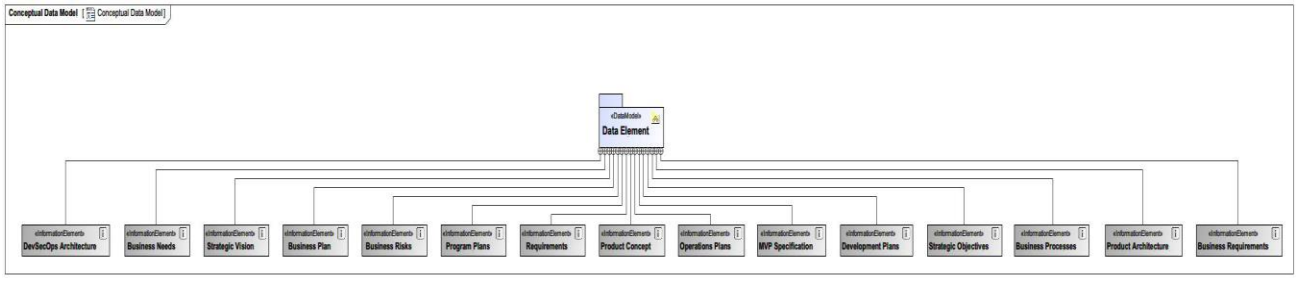

Figure 4: Operational Information – Conceptual Data Model

Operational Processes

Operational Activities and Flow Diagrams

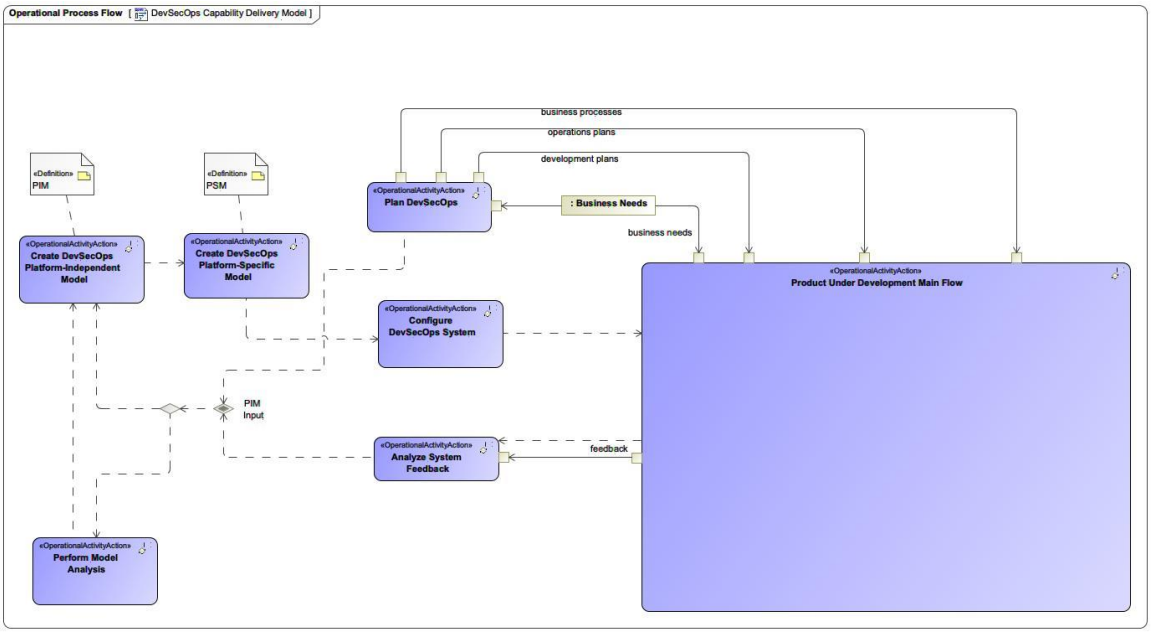

Figure 5: DevSecOps Capability Deliver Model

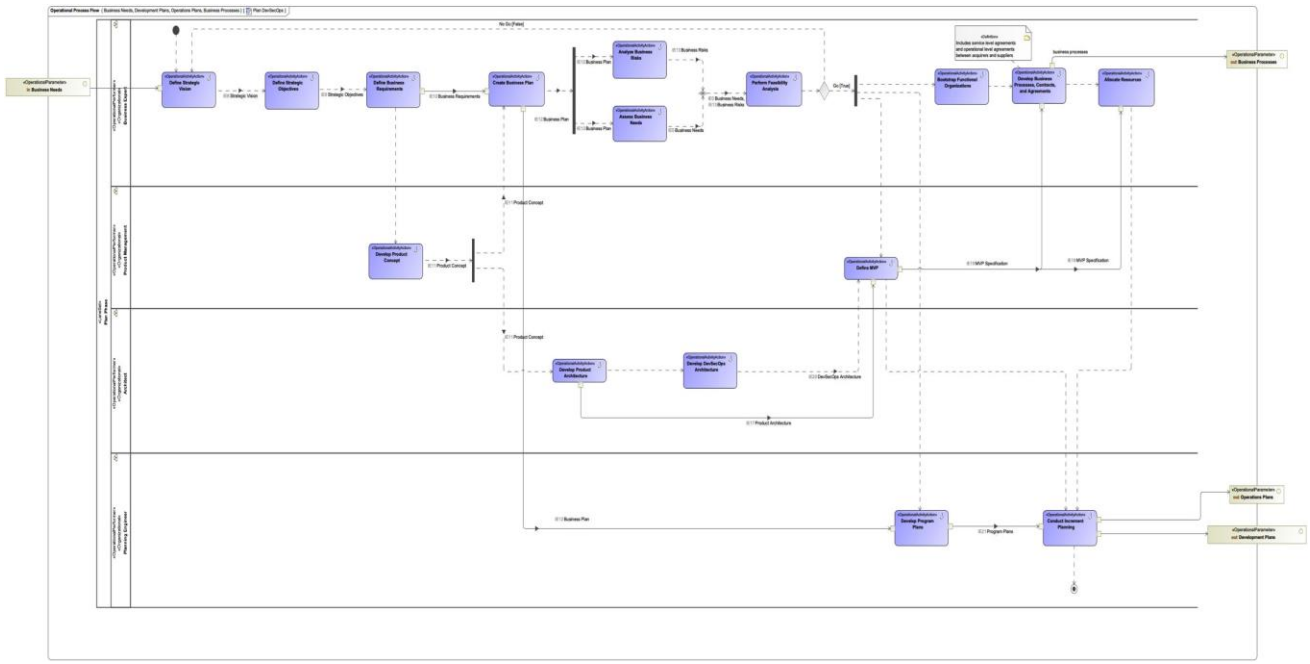

Figure 6: Plan DevSecOps

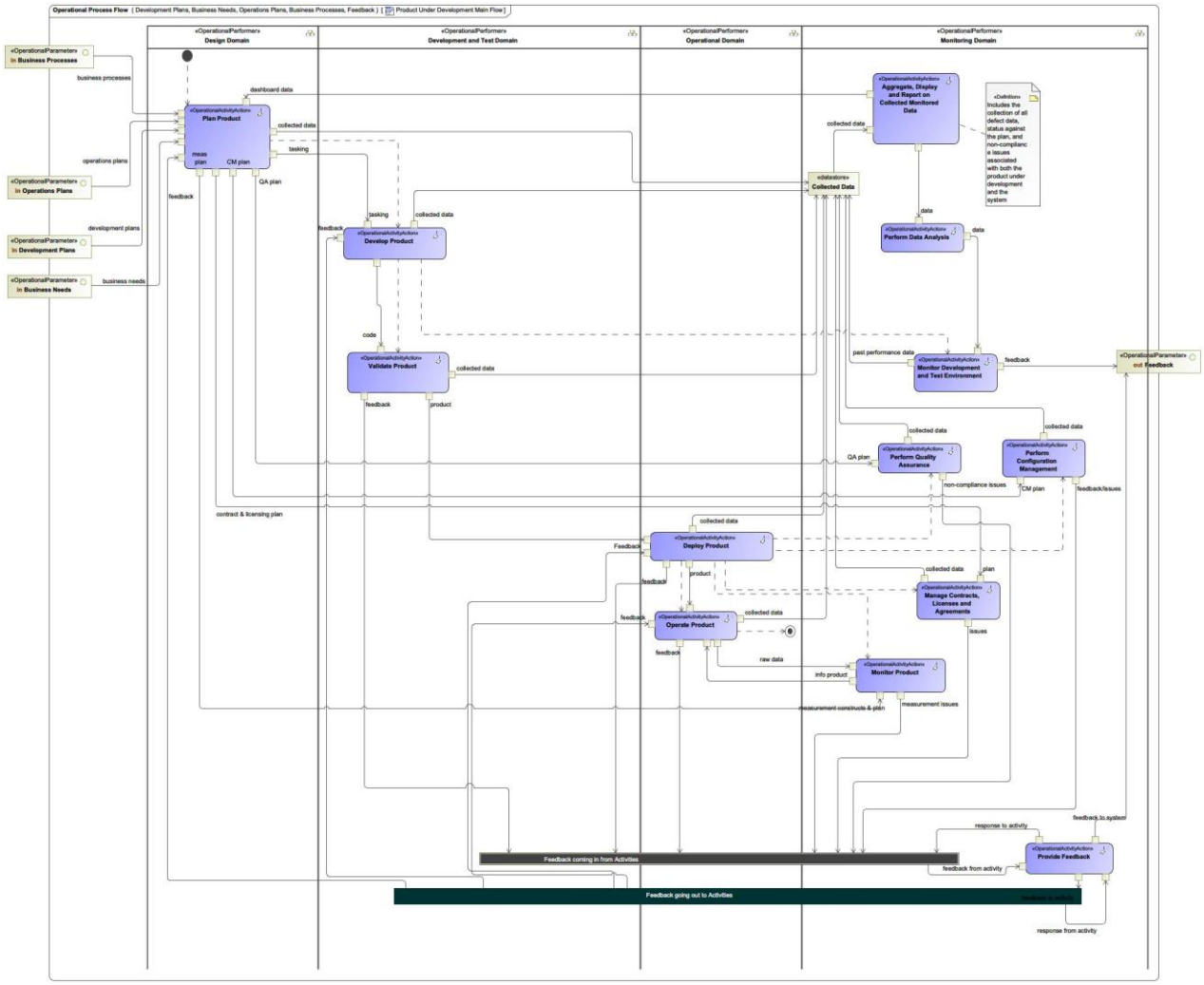

Figure 7: Product Under Development Main Flow

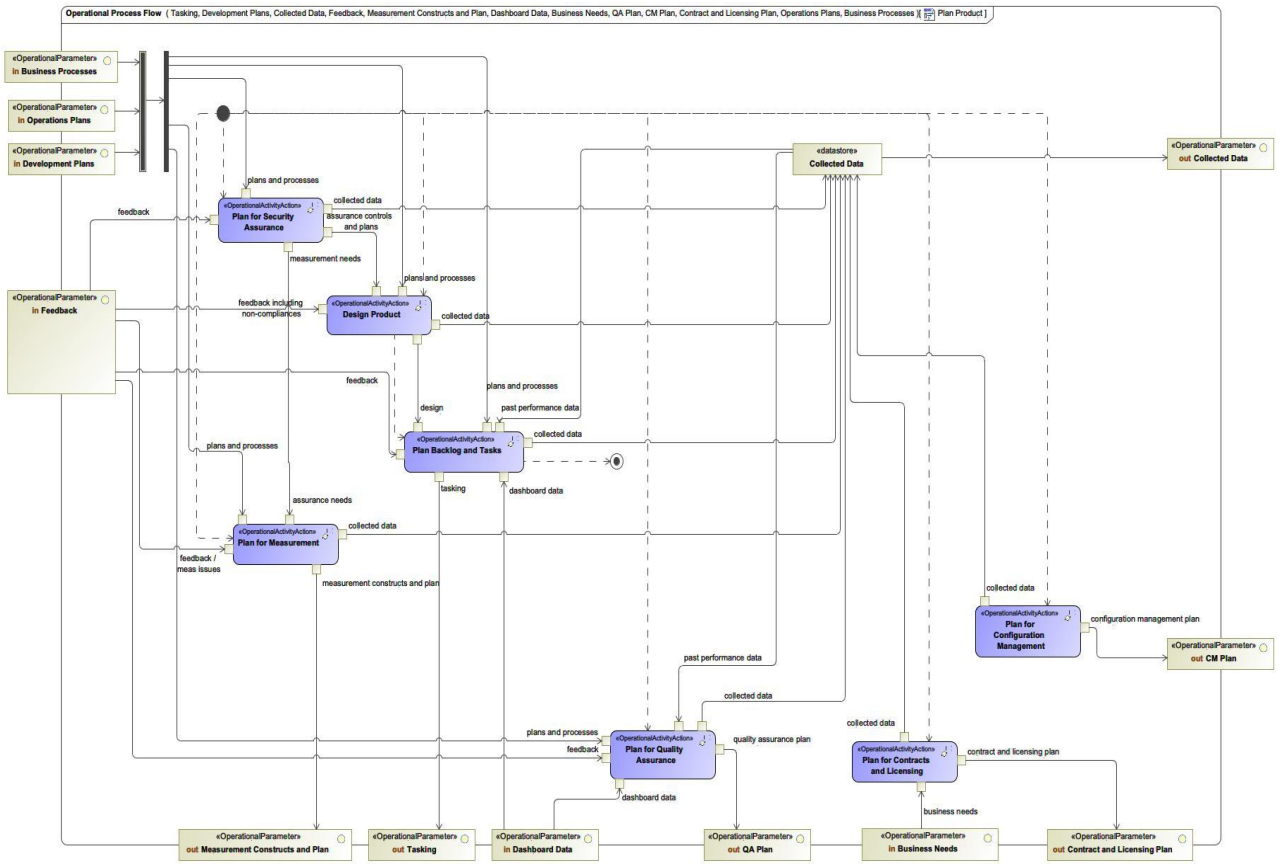

Figure 8: Plan Product

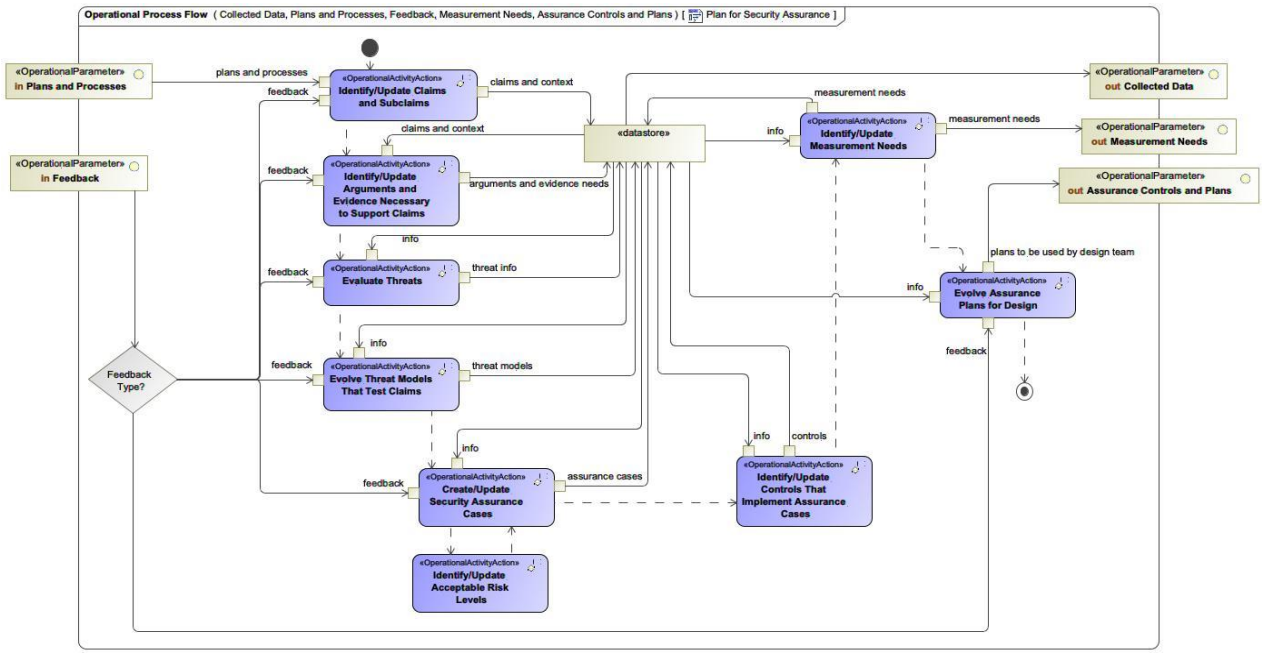

Figure 3: Plan for Security Assurance

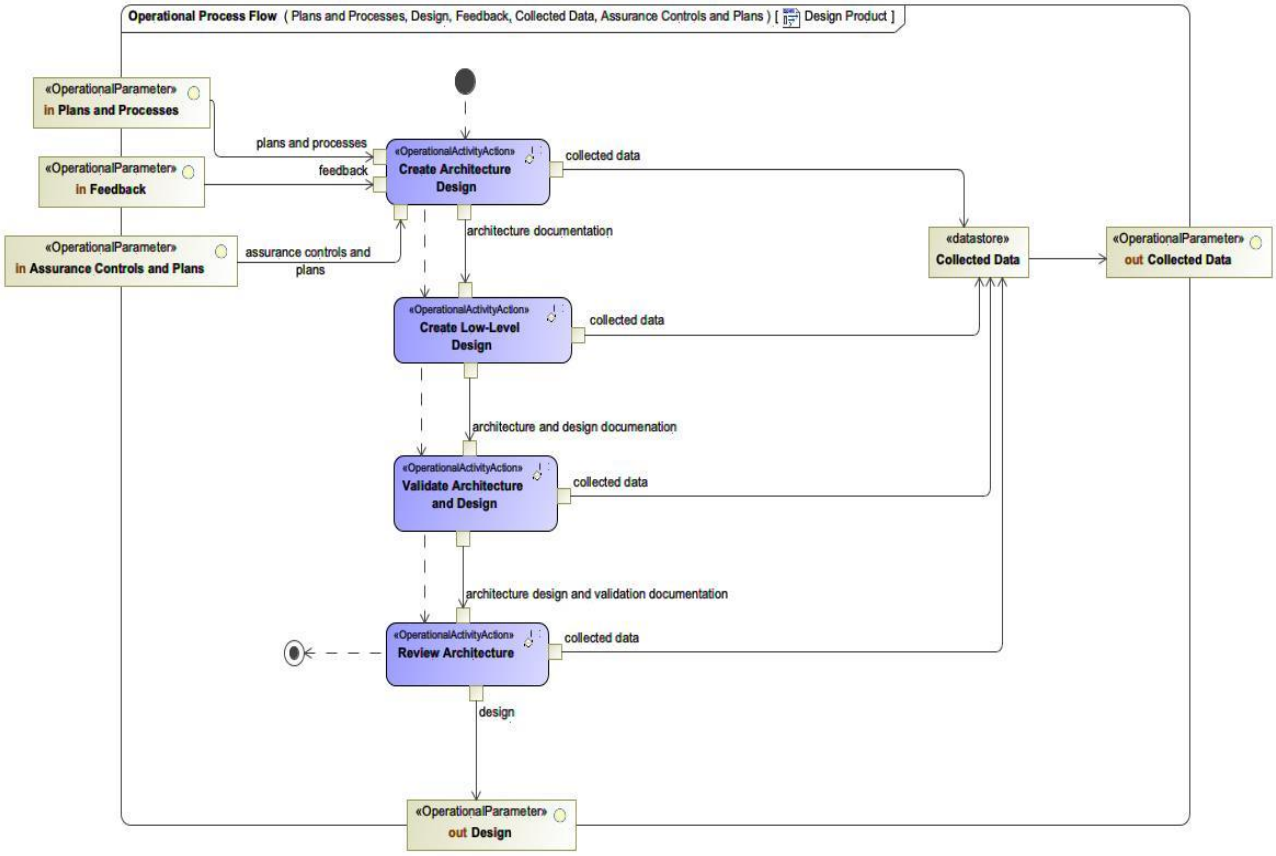

Figure 9: Design Product

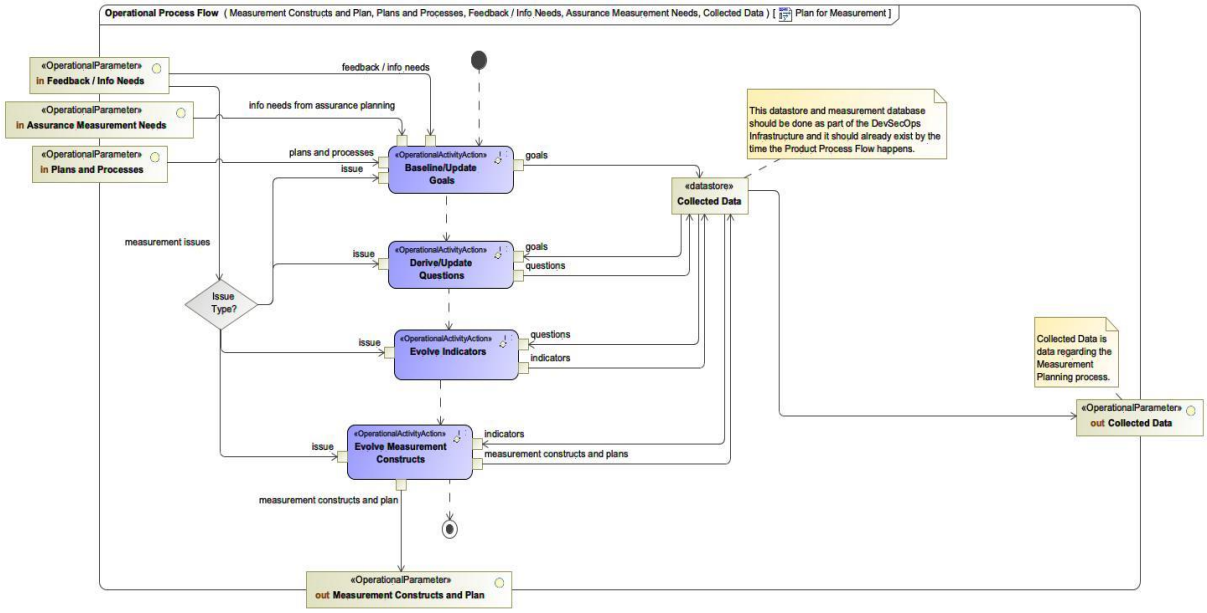

Figure 10: Plan for Measurement

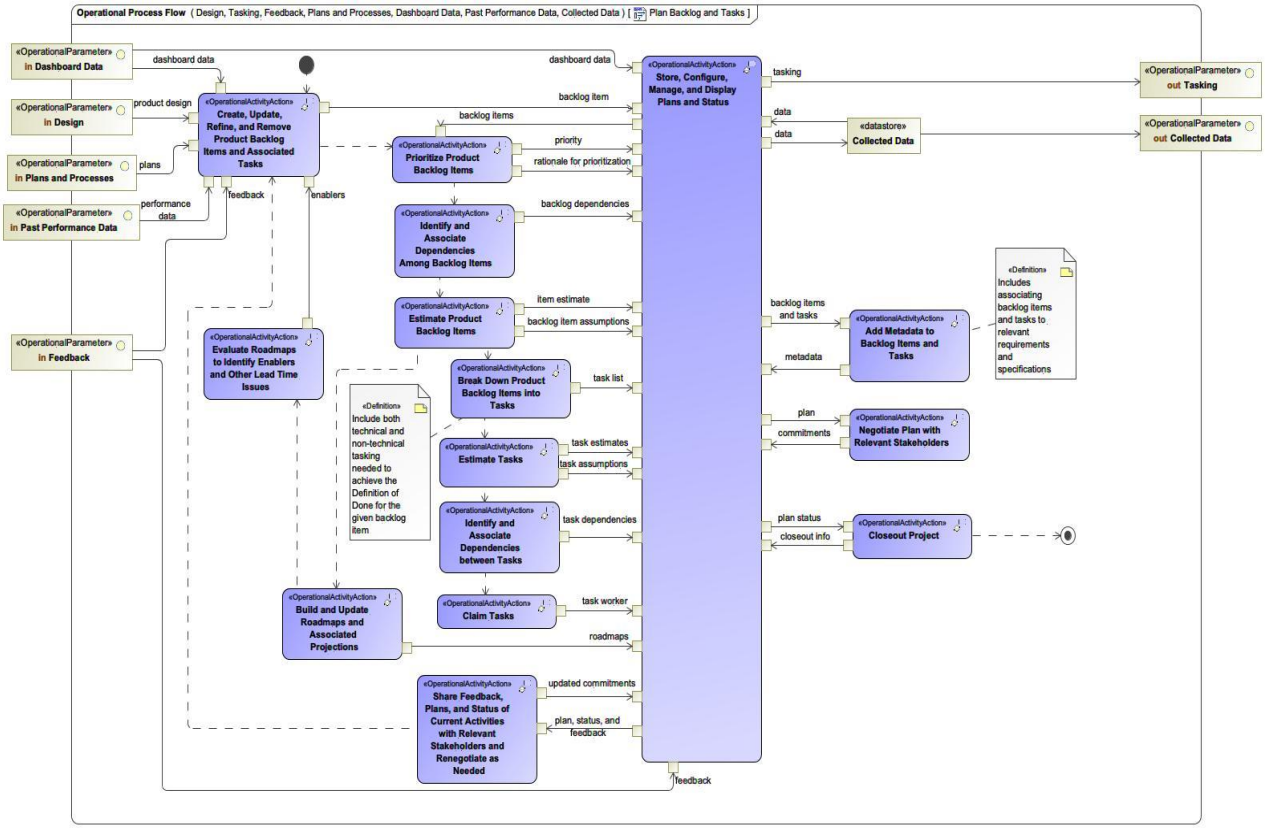

Figure 11: Plan Backlog and Tasks

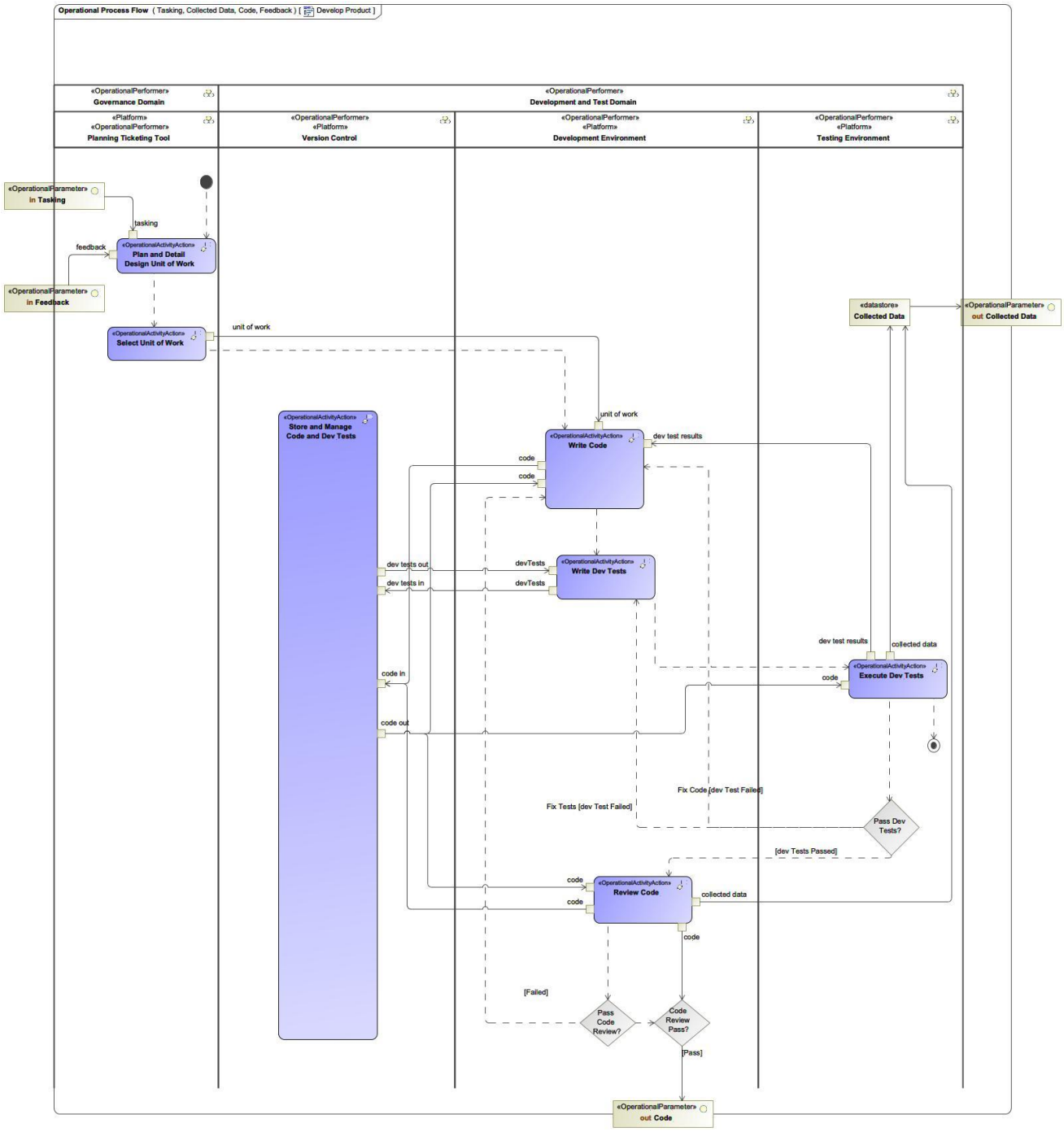

Figure 12: Develop Product

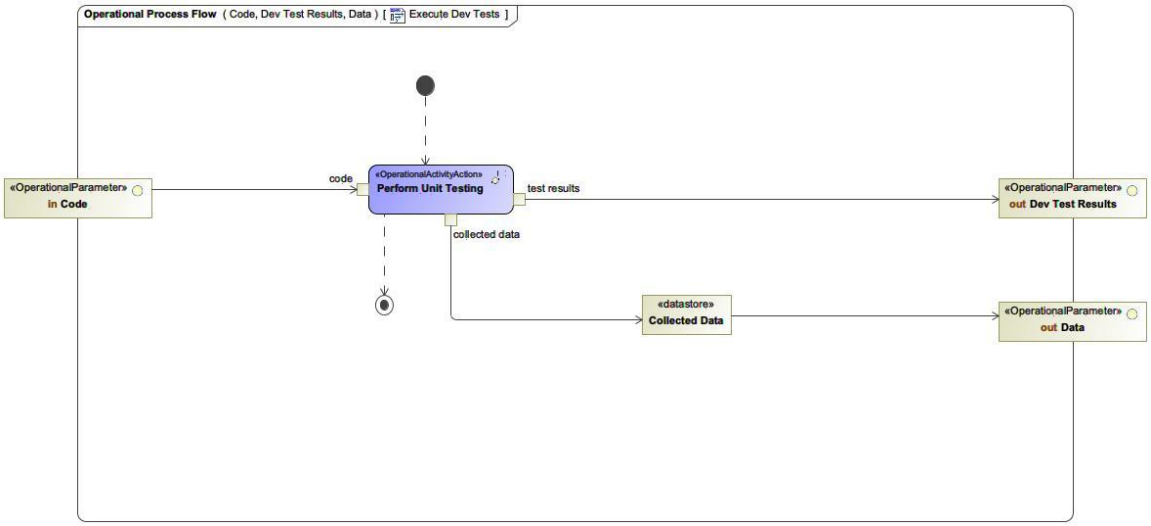

Figure 13: Execute Dev Test

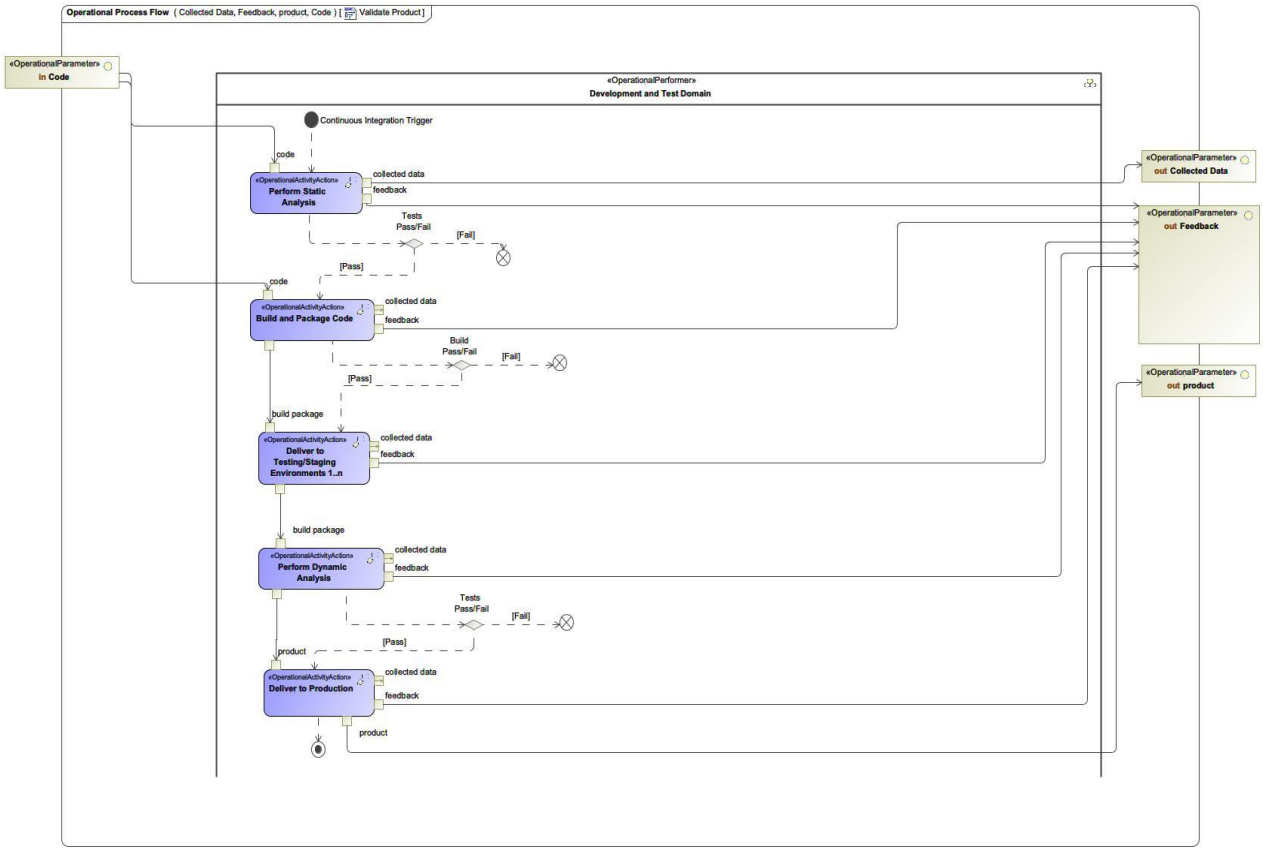

Figure 14: Validate Product

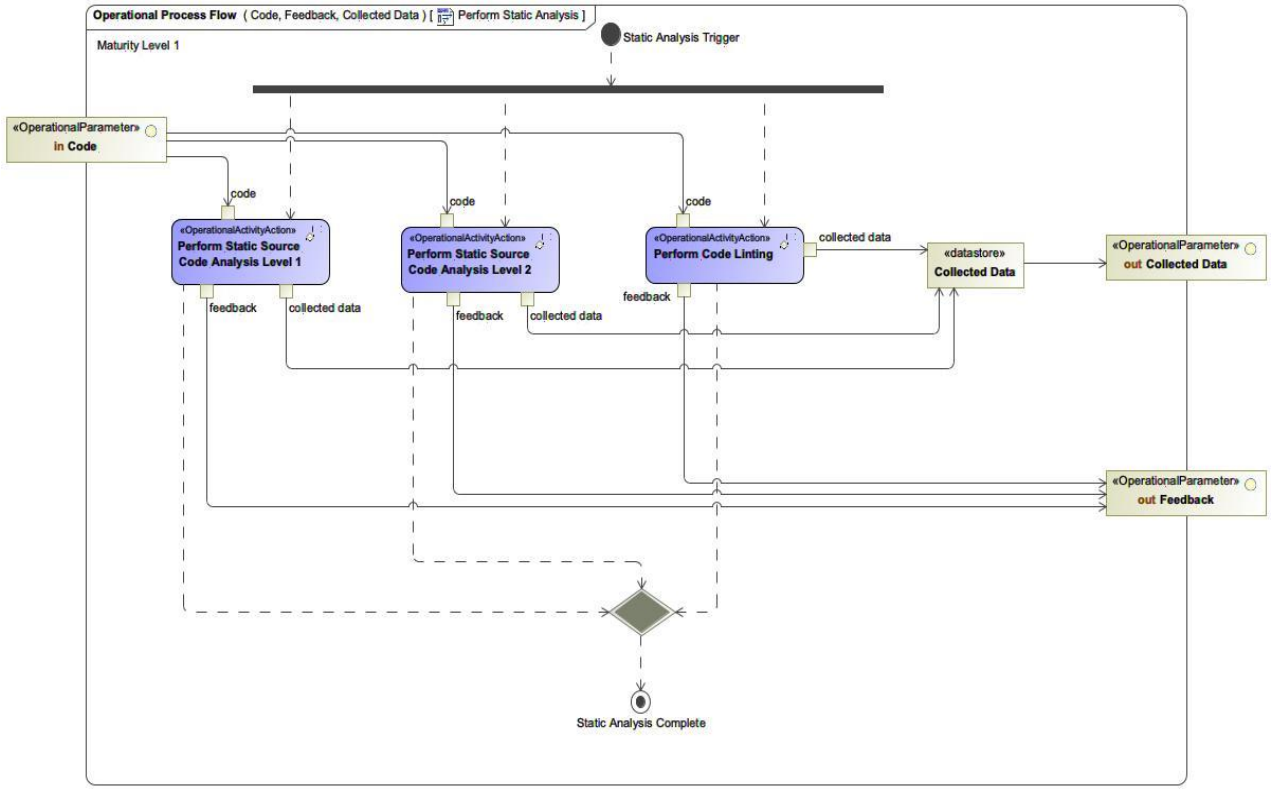

Figure 15: Perform Static Analysis

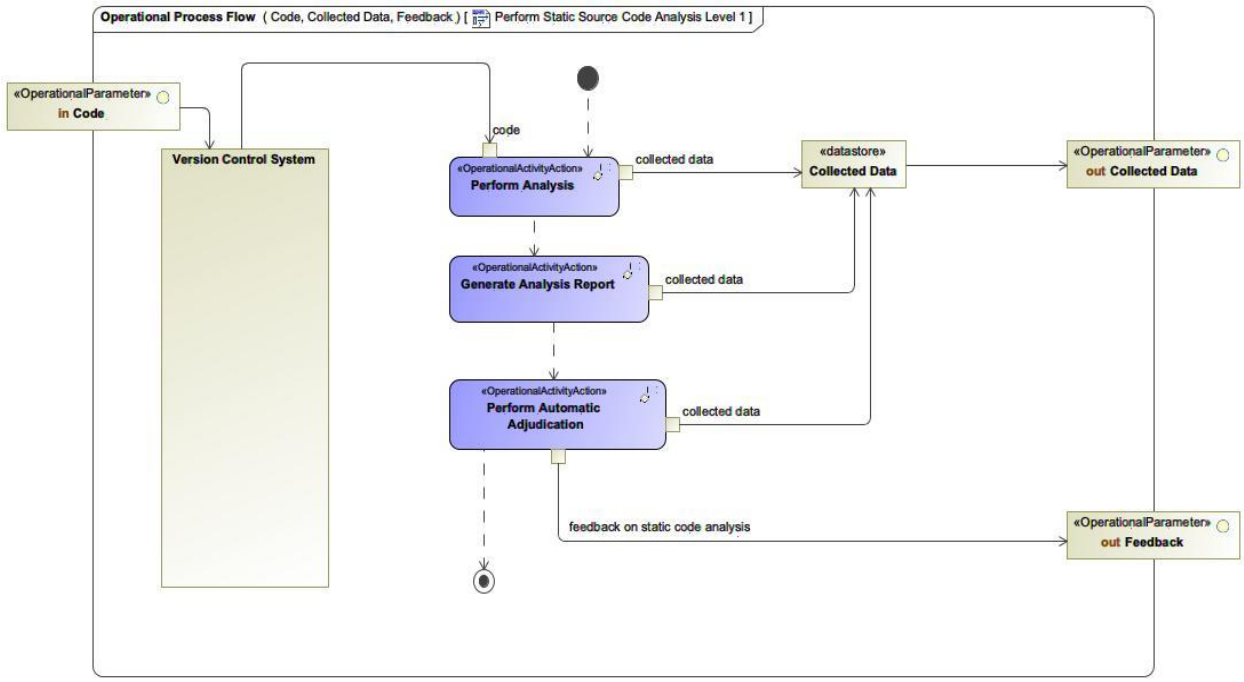

Figure 16: Perform Static Source Code Analysis Level 1

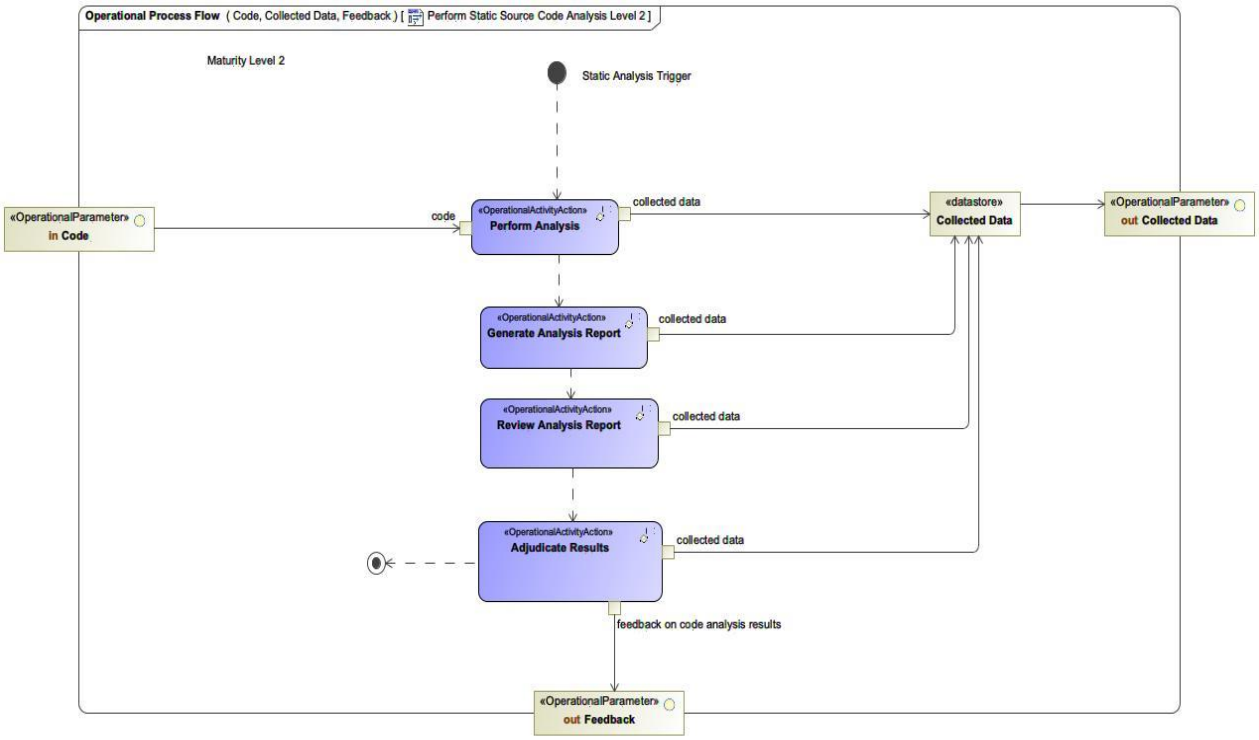

Figure 17: Perform Static Source Code Analysis Level 2

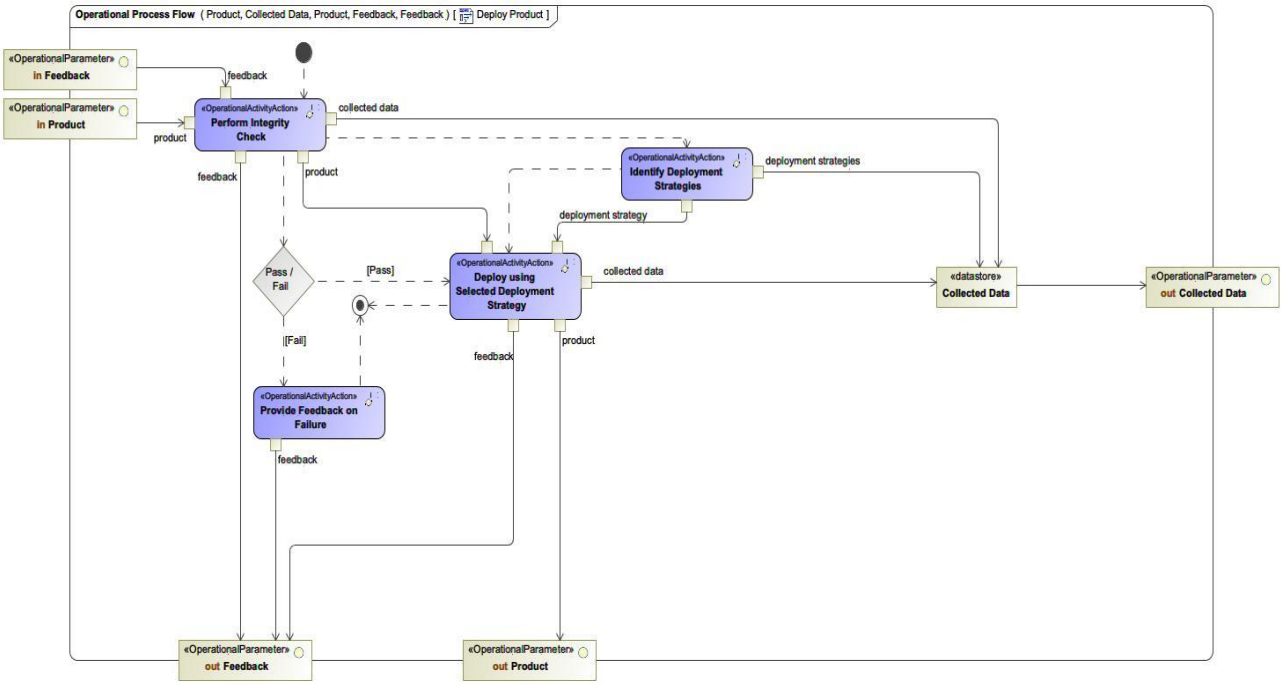

Figure 18: Deploy Product

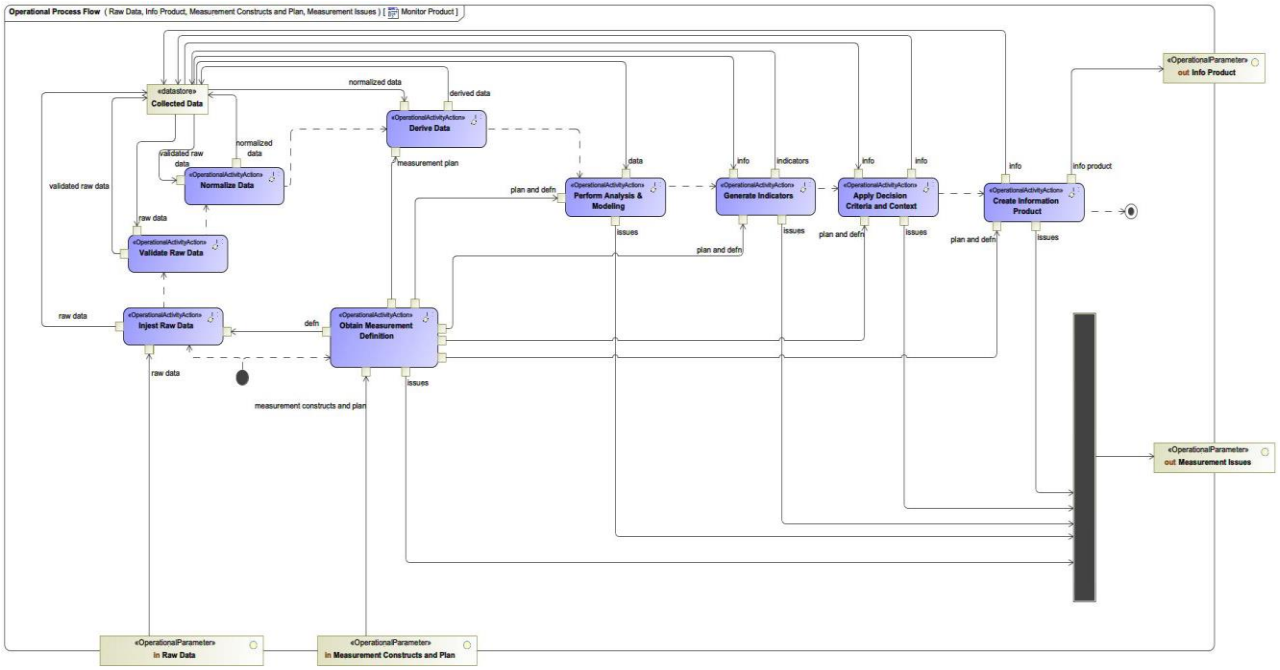

Figure 19: Monitor Product

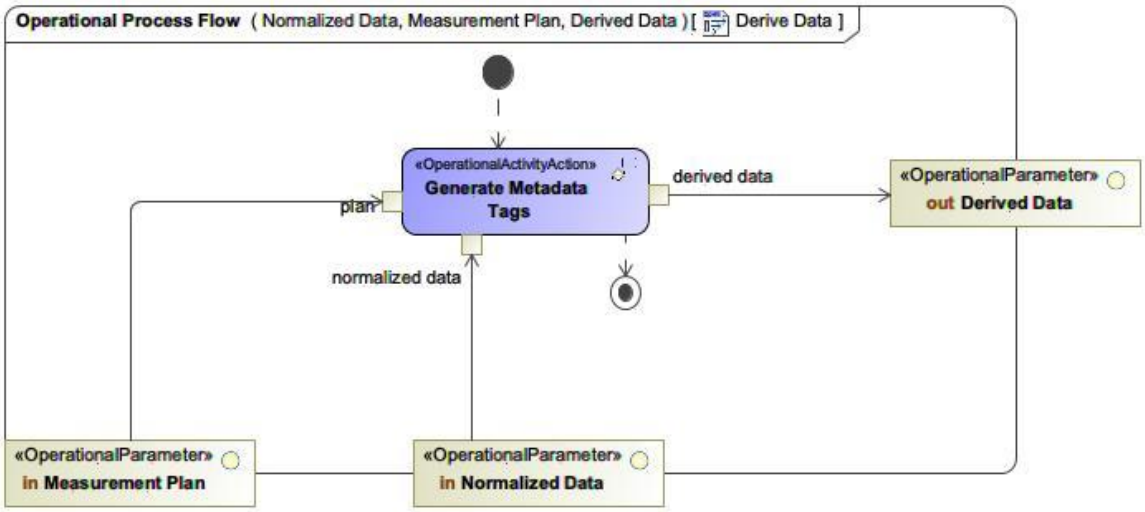

Figure 20: Derive Data

Patch Software Scenario

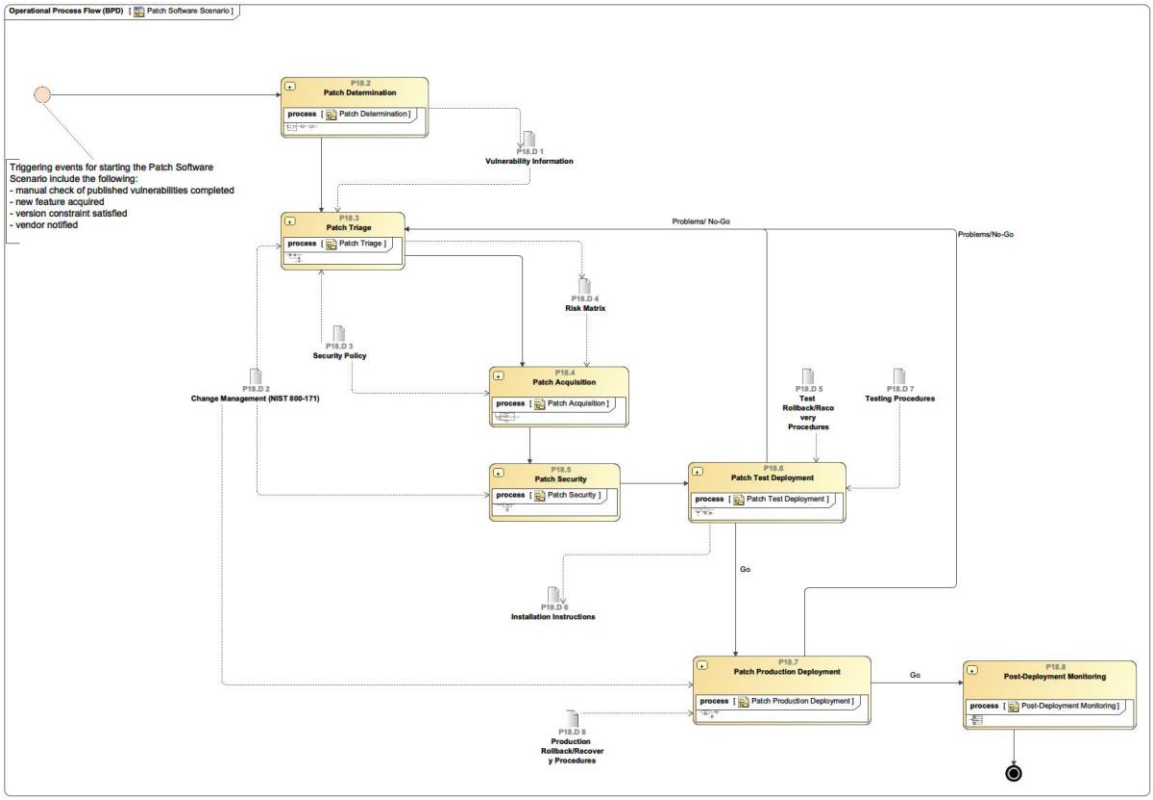

Figure 21: Patch Software Scenario

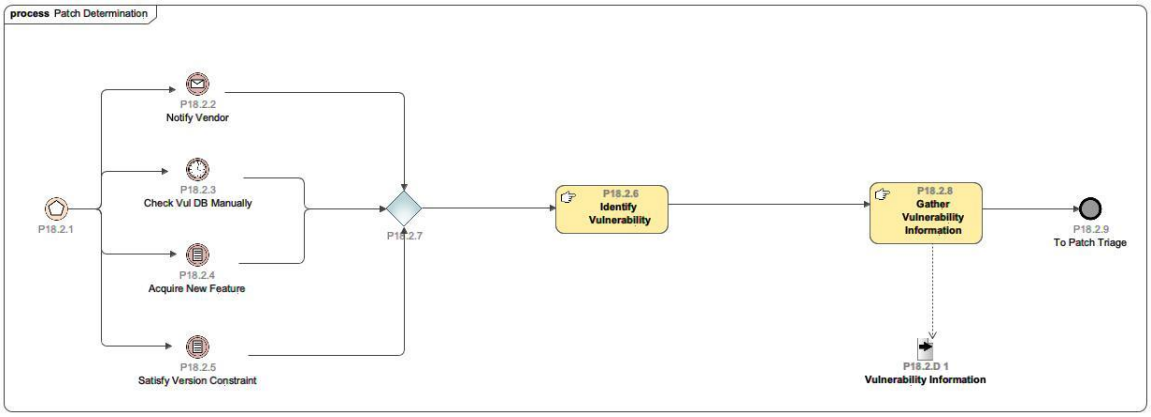

Figure 22: Patch Determination

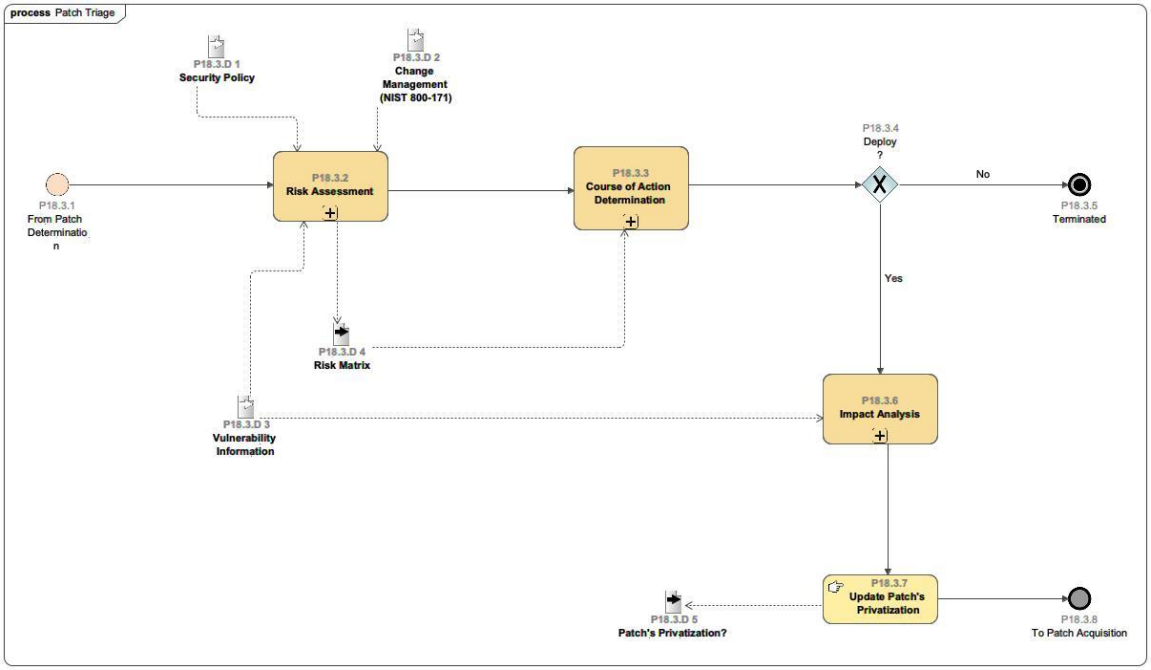

Figure 23: Patch Triage

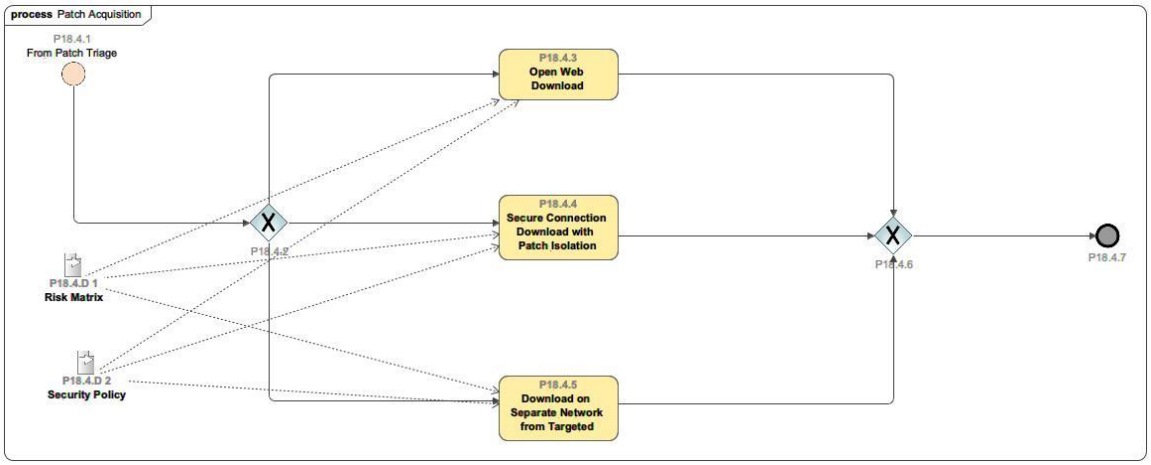

Figure 24: Patch Acquisition

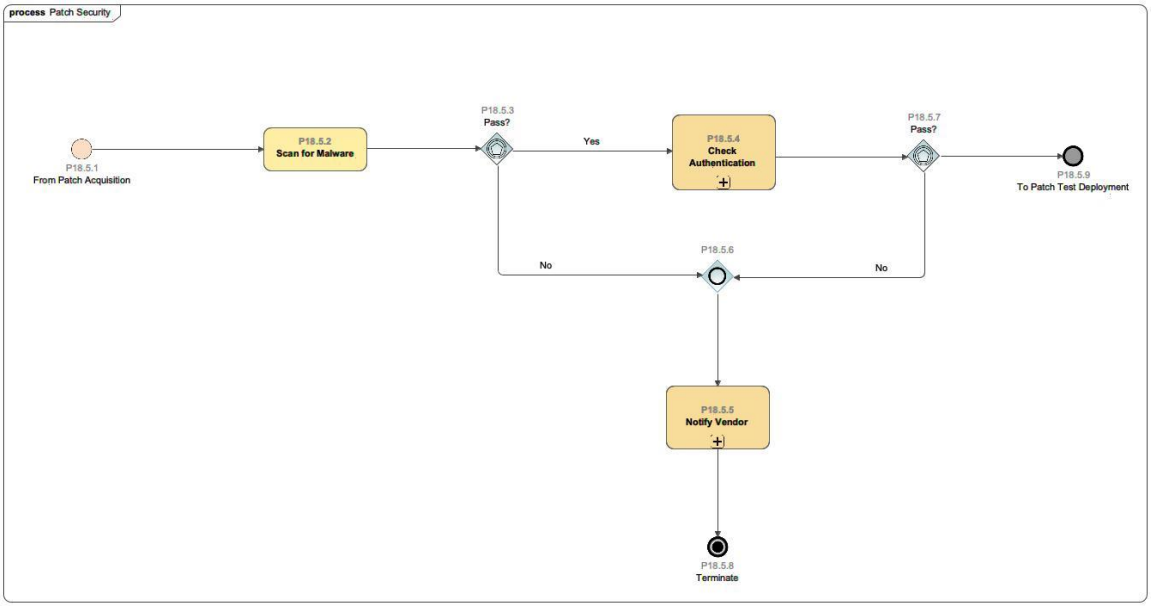

Figure 25: Patch Security

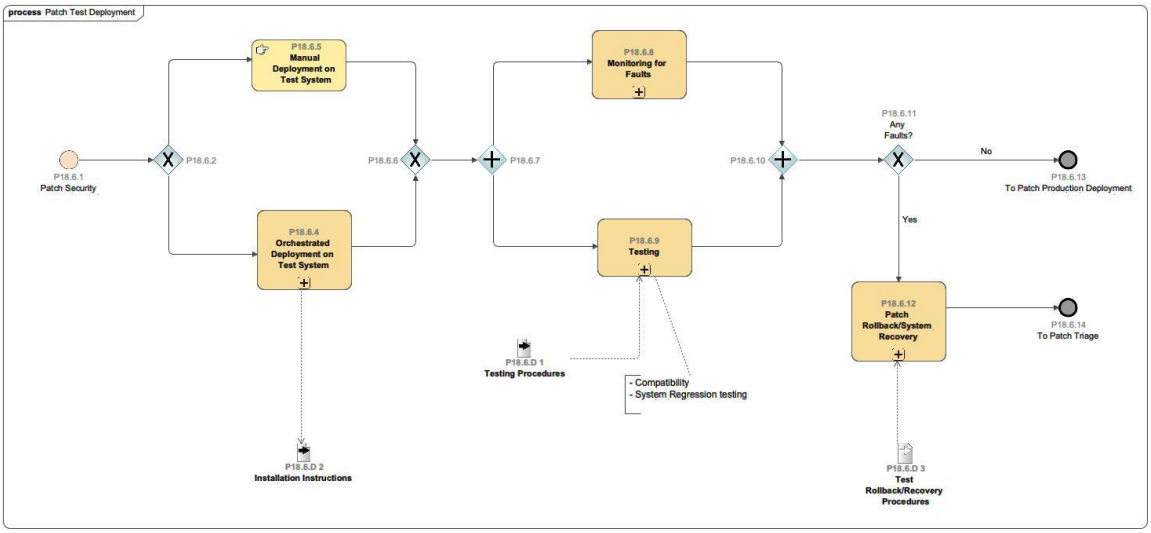

Figure 26: Patch Test Deployment

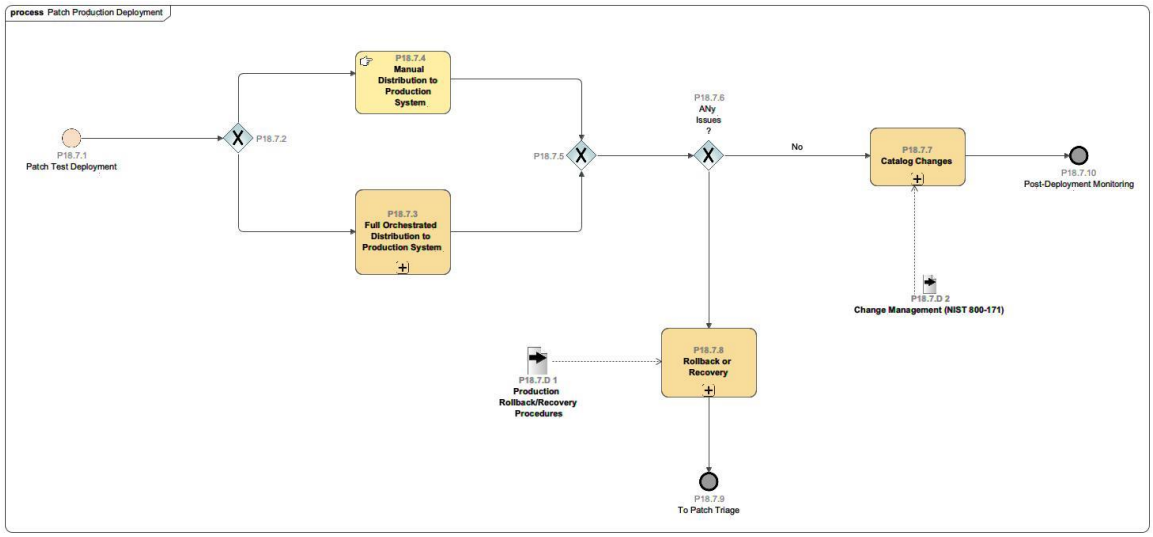

Figure 27: Patch Production Deployment

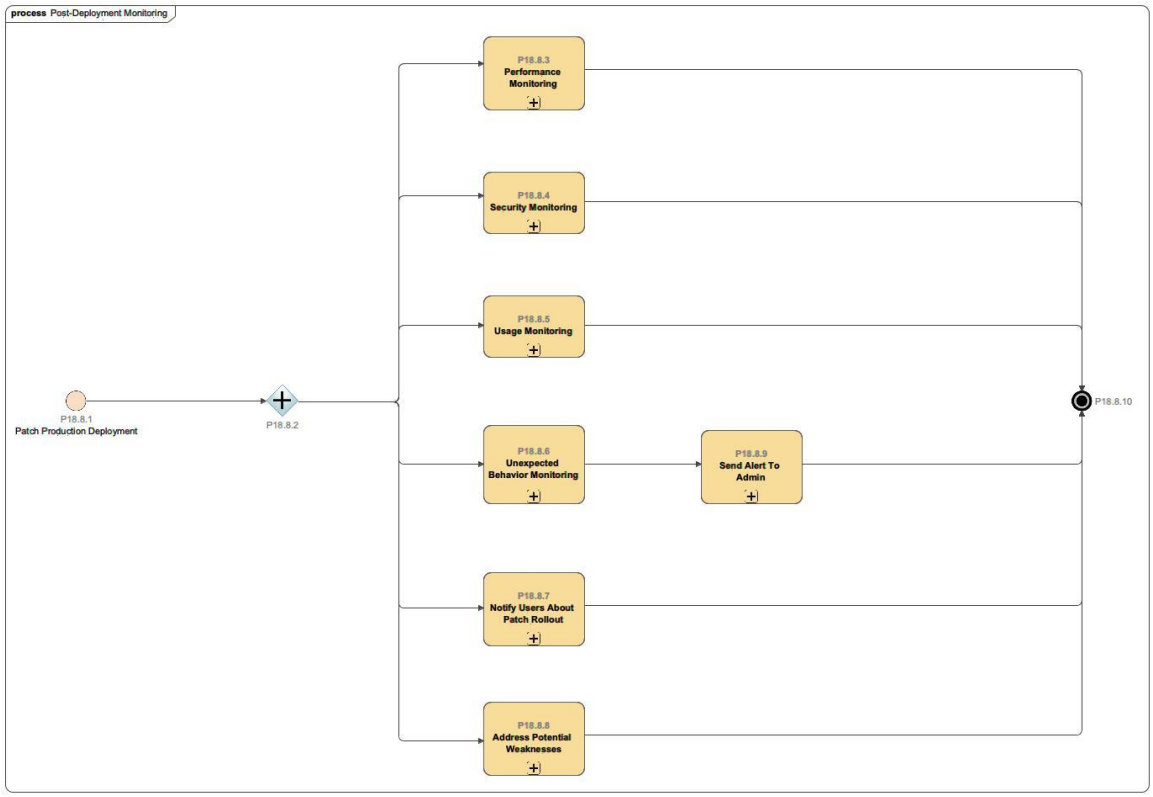

Figure 28: Post-Deployment Monitoring

Personnel

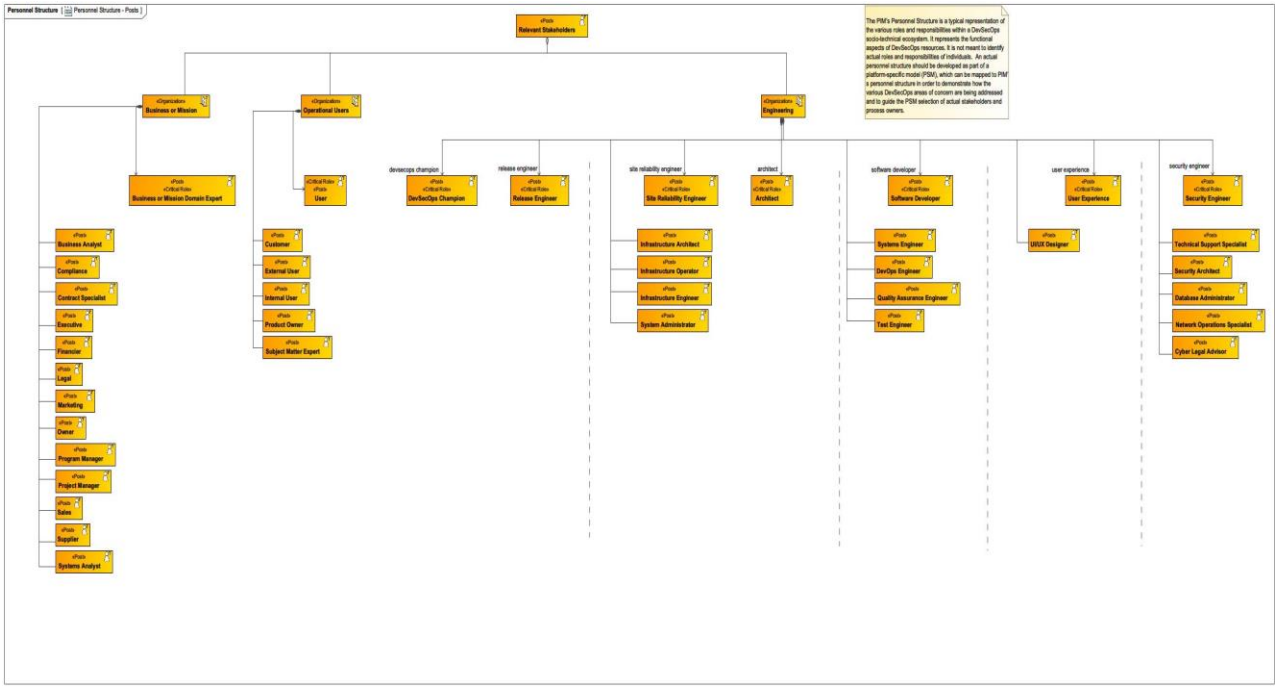

Figure 29: Personnel Structure – Posts

Table 2: Critical Roles – Responsibilities, Goals and Questions

#	Name	Attribute	Goals and Questions
1	Architect	• Analyze, design, and implement strategies for continuous deployment of production and pre-production systems, and development and test pipelines • Document how the implementation of a new system or new interface between systems impacts the current and target environment including but not limited to security posture. • Generate and implement plans for integrating new systems into existing infrastructure and employ secure configuration management processes. • Ensure that acquired or developed system(s) and architecture(s) are consistent with organization's cybersecurity architecture guidelines.	Goals: 1) Ensure continuous development and deployment including technology stack 2) Ensure test coverage and automated testing 3) Ensure pipeline infrastructure includes all required tools and processes to test system requirements (i.e., ATO) 4) Ensure that necessary records for Authority to Operate (ATO) support an audit. 5) Ensure that latencies (i.e., lead times) are not excessive for each pipeline stage 6) Ensure pipeline doesn't introduce defects 7) Ensure pipeline is secure 8) Ensure known vulnerabilities do

		• Identify and prioritize critical business functions in collaboration with organizational stakeholders. • Provide advice on project costs, design concepts, or design changes. • Analyze candidate architectures, allocate security services, and select security mechanisms. • Develop enterprise architecture or system components required to meet user needs. • Document and update as necessary all definition and architecture activities. • Integrate results regarding the identification of gaps in security architecture. • Plan implementation strategy to ensure that enterprise components can be integrated and aligned. • Translate proposed capabilities into technical requirements. • Capture and integrate essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event. • Analyze user needs and requirements to plan architecture. • Analyze system requirements • Assess effectiveness of existing systems • Recommend alternative technologies and improvements • Write detailed functional specifications that document the architecture development process. • Lead design and review of new systems • Evaluate security architectures and designs to determine the adequacy of security design and architecture proposed or provided in response to requirements contained in acquisition documents.	not escape into deployment 9) Ensure pipeline is robust and available 10) Ensure new system(s) and system interface(s) can be easily integrated into pipeline 11) Ensure all policies and guidelines are followed 12) Ensure user / business needs are adequately captured in requirements and met by the architecture and the capabilities 13) Ensure architecture is safe and secure and evolves to meet future needs Questions: 1) What are the workflow stage lead times? 2) What workflow stages are fully automated? 3) Are people adequately trained in the continuous development and deployment processes and tooling? (Staffing) 4) Are there sufficient people and resources to support the infrastructure activities? (Staffing) 5) Where are the bottlenecks or slowdowns in process/pipeline flow? 6) What is the reliability/availability of operational capabilities? 7) How long does it take to restore the system after a failure? (MTTR) 8) What steps were taken to remove vulnerabilities? 9) How many vulnerabilities were found prior to release? 10) How many vulnerabilities are discovered in each release? 11) How many vulnerabilities are estimated to escape? 12) How effective is each step in finding and removing vulnerabilities/weaknesses?

		• Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration.	13) What is the marginal cost (and lead time) to remove vulnerabilities? 14) What are the Supply Chain reported vulnerabilities (quantity, severity) 15) What is the attack surface? Is it increasing or decreasing? 16) How much margin exists for future growth? (scalability) 17) Is the system safe to operate? 18) How long does it take to conduct a full regression test? 19) How much change is there in requirements, capabilities, supply chain, tools, etc.? 20) How easy is it to integrate a new system or system interface? 21) Are all the policies and guidelines being followed? 22) Are the requirements and capabilities sufficient to meet the user / business needs?
2	Business or Mission Domain Expert	• Define the overall need in which the product under development is intended to satisfy • Analyze the need in order to describe it in sufficient details that a solution can be engineered. • Develop and track indicators of success and taking corrective actions when needed based on available information • Develop and maintain budgeting and forecasting • Perform variance analysis when actuals deviate from plan in order to determine if corrective actions are needed • Plan and monitor all activities within scope of authority • Share information between relevant stakeholders based on given stakeholder's needs • Document and maintain business processes	Goals: 1) Ensure timely Authority to Operate (ATO) 2) Ensure customer/business need is well understood. 3) Ensure a balance between customer expectations and prioritized needs and system capacity. 4) Predict market direction and future needs 5) Ensure the best promotion and marketing of the product 6) Ensure customer/business needs are translated into the development plan. Questions: 1) Is the customer/business need satisfied? 2) What is the ROI on each product release? 3) What is the ROI on improvements?

		• Lead continuous reviews of processes and develop optimization strategies • Anticipate changes to market and user needs • Coordinate the sharing of ideas and findings among relevant stakeholders • Allocate resources and maintain cost efficiency • Prioritize initiatives based on business needs and available resources • Ensure that all contractual obligations are met • Coordinate with government regulators • Ensure compliance with applicable laws and regulations • Ensure that audit discoveries and compliance violations are addressed • Plan, implement, and oversee risk identification and mitigation activities • Coordinate training needs of workforce • Develop, maintain, and enforce policies • Address relevant stakeholder concerns regarding legal compliance • Review contracts to prevent disputes and financial risks • Develop and use appropriate contract provisions and amendments which comply with legal requirements and policies • Review, negotiate, and approve contract terms and conditions • Process payments in accordance with agreed to payment structures • Report on revenue and expenditure • Create, maintain and execute sound business, program, and project plans • Manage the demands of stakeholders • Maintain financial records for all transactions and changes • Coordinate financial audits • Monitor all deposits and payments	4) What other products / innovations would satisfy the customer/business need? 5) Where is the market heading? What are the trends? 6) How good is communication among stakeholders? 7) Is the contract being met? Are all laws and regulations being met? 8) Does the contract need to be modified? 9) What is the best market strategy and how to implement it? 10) Is the system capacity adequate to deliver the product as promised? 11) Are actual personnel resources knowledgeable and trained to perform their tasks? 12) What are the significant risks and what are their mitigations? 13) Are the policies and guidelines adequate and are they being followed? 14) How can market share be increased?

		• Process invoices • Organize activities in accordance with the mission and goals of the organization • Increase brand or mission awareness • Optimize marketing strategies • Analyze the competition and prepare forecasts • Develop and manage long-term goals • Obtain funding for uninterrupted delivery of services • Evaluate user needs and promote products and services • Coordinate user needs	
3	DevSecOps Champion	• Provide leadership and vision to deliver the changes necessary in adopting DevOps practices. • Work with organizational units to remove barriers to communications and information exchange • Determine which tools or processes are best suited for solving long-term needs • Improve processes in order to better serve the customer • Determine obstacles faced by organizational units	Goals: 1) Maximize DSO Process Performance 2) Maximize Customer/User Satisfaction Questions: 1) Is the DSO process optimized? 2) Is the customer happy? 3) Is the system/product easy to use/operate? 4) What is the cost to release and maintain products? 5) What is the ROI on process improvements? 6) What is the ROI on alternative processes, tools or products? 7) How does DSO compare to the previous methodology or other possible methodologies?
4	Release Engineer	• Provide technical support of a product from development to production and maintenance. • Perform as technical liaison for Engineering and Operations on every aspect associated with final builds and control baseline issues. • Prepare, evaluate and maintain tools supporting and process automation	Goals: 1) Ensure the capability is delivered in each release; 2) Ensure capability is high quality in each release 3) Ensure capability is delivered as planned (e.g., cost, resources, people/team) per increment/release. 4) Ensure all issues during builds are addressed identified and there is a

		for software or hardware product release. • Ensure capability to compile and assemble software through source code and store tools in source control. • Design, manage and execute tools and scripts to develop different versions of products on wide-range operating systems. • Develop dashboards to quantify internal processes efficiency continuously. • Develop library of tools for automating manual workflows in development process. • Interact with release engineering and QE to debug and also resolve identified issues. • Respond constantly and aggressively to automated test and build issues. • Support integration of new technologies along with companies. • Develop and present general releases, service packs, web products and beta products. • Correct build errors working with development engineers. • Perform with others for analysis, evaluations along with design options and execute process improvements. • Perform with project teams to identify apt build schedule and initiate packaging and build process.	path to closure (including security issues). 5) Prioritize work for competing stakeholders (managing the backlog including new requirements, defect fixes, tech refresh changes, etc.) 6) Ensure efficient and effective ATO and testing is complete and documented 7) Ensure all requirements have been satisfied, presented to authorizing agents and captured in a repository. Questions: 1) What percent of the requirements/capabilities are planned, modified, and delivered for each release? 2) What percent of current release is allocated to new requirements, backlog, rework, security certifications? (Release Content) 3) How many defects (by severity) and related rework items were injected/created in each release? (Defects/Escapes) 4) Are defects being worked off at a sufficient rate? (defect backlog/inventory – shrinking/growing) 5) Does the delivered baseline meet the functional and performance requirements of the end user? 6) Is the user satisfied which each release? 7) Do features/capabilities work as expected? 8) How many teams/people were required to deliver each release compared to the plan (i.e., balancing the planned work load with the team's ability to produce it; backlog management)? (Staffing by Release)

5	Security Engineer	• Understand security sensitivities and how they affect the design, implementation, configuration, and delivery aspects of the software development lifecycle, and ensure that security considerations affect every phase of this lifecycle. • Define and prioritize essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event • Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration. • Develop/integrate cybersecurity designs for systems and networks with multilevel security requirements or requirements for the processing of multiple classification levels of data primarily applicable to government organizations (e.g., UNCLASSIFIED, SECRET, and TOP SECRET). • Document and address organization's information security, cybersecurity architecture, and systems security engineering requirements throughout the acquisition life cycle. • Employ secure configuration management processes. • Ensure that acquired or developed system(s) and architecture(s) are consistent with organization's cybersecurity architecture guidelines. • Identify and prioritize critical business functions in collaboration with organizational stakeholders.	Goals: 1) Ensure comprehensiveness of security requirements (based on customer need and environment) 2) Ensure knowledge of and minimization of attack surface 3) Ensure critical business functions are identified and prioritized 4) Ensure the pipeline and all its processes are secure 5) Provide restoration processes and procedures are available in the event of catastrophic failure 6) Ensure working knowledge of candidate architecture options and their associated security concerns 7) Ensure security incidents are detected and resolved 8) Ensure supply chain is secure 9) Minimize security incidents 10) Ensure certifications and accreditations are performed as required 11) Understand normal usage patterns and recognize abnormal usage patterns Questions: 1) What is the attack surface? Is the attack surface increasing or decreasing? 2) What can be done to reduce the attack surface? 3) What are the likely threats? 4) What types of attacks have been successful? By whom? 5) Are patches delivered as committed? 6) Are there any new CVEs that could be in the product? 7) What is the rate of security related incidents and which types occur most often? 8) How much of the backlog is security related? 9) Are preparations for ATO on

		• Perform security reviews, identify gaps in security architecture, and develop a security risk management plan. • Provide advice on project costs, design concepts, or design changes. • Provide input on security requirements to be included in statements of work and other appropriate procurement documents. • Provide input to the Risk Management Framework process activities and related documentation (e.g., system lifecycle support plans, concept of operations, operational procedures, and maintenance training materials). • Define and document how the implementation of a new system or new interfaces between systems impacts the security posture of the current environment. • Analyze candidate architectures, allocate security services, and select security mechanisms. • Develop a system security context, a preliminary system security Concept of Operations (CONOPS), and define baseline system security requirements in accordance with applicable cybersecurity requirements. • Evaluate security architectures and designs to determine the adequacy of security design and architecture proposed or provided in response to requirements contained in acquisition documents. • Write detailed functional specifications that document the architecture development process. • Analyze user needs and software requirements to determine feasibility of design within time and cost constraints.	plan/complete? Is the evidence available? 10) What is needed for continuous ATO? 11) Are there any abnormal usage patterns? How quickly are abnormal patterns detected? 12) What are the findings from code scans? 13) How long does it take to successfully complete penetration testing? 14) What is the lead time to complete security testing? 15) Are incidents being detected and resolved? 16) Is the supply chain secure? 17) How long does it take to securely recover from an attack?

		• Develop enterprise architecture or system components required to meet user needs. • Document and update as necessary all definition and architecture activities. • Determine the protection needs (i.e., security controls) for the information system(s) and network(s) and document appropriately. • Translate proposed capabilities into technical requirements. • Assess and design security management functions as related to cyberspace.	
6	Site Reliability Engineer	• Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration. • Capture and integrate essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event. • Recommend alternative technologies and improvements • Define best practices to ensure software releases are consistent and repeatable • Plan and coordinate incident responses • Respond to incidents that impact relevant stakeholders • Monitor system infrastructure to ensure that all service and operational level agreements are met • Develop and maintain response playbooks to address incidences	Goals: 1) Ensure latencies are minimized for each pipeline stage 2) Ensure the pipeline doesn't introduce defects 3) Ensure the pipeline is safe and secure 4) Ensure the pipeline is robust and available 5) Ensure all critical resources and functions are identified and prioritized 6) Ensure releases are deployed on time and with high quality 7) Provide restoration processes and procedures in the event of catastrophic failure 8) Ensure security incidents are detected and resolved 9) Ensure the supply chain is secure 10) Minimize security incidents 11) Ensure changes don't break previous functionality 12) Ensure service level agreements are satisfied Questions: 1) Are there sufficient people and resources available to support the business needs? (Staffing) 2) Where are the bottlenecks or

		• Conduct post-incident reviews, document findings, and take action based on what was learned in order to bolster the reliability of the service. • Develop software to make system and product under development infrastructure more resilient, automated, and self-healing over time • Increase reliability and performance of the system and product under development • Plan and implement disaster mitigation, prevention, and response activities • Improve the system and product under development in order to maximize effectiveness based on defined service level objectives • Reduce the time required to resolve incidents and restore service (i.e. reduce mean time to repair) • Continuously test the operational readiness and efficiency of the infrastructure • Conduct chaos experiments to see how the system and product under development infrastructure will behave under emergent behavior during runtime • Meet the system and product under development computing needs • Collect and monitor infrastructure performance data • Coordinate software and hardward installations, testing, and transitions • Introduce alternative technologies to improve or enhance infrastructure in support of system or product under development requirements • Manage the underlining security of all infrastructure components and inter-component information transfers	slow downs in process/pipeline flow? 3) What are the critical resources and functions? 4) What is the reliability/availability of the pipeline and each stage in it? 5) How long does it take to restore the system after a failure? (MTTR) 6) How much margin exists for future growth? (scalability) 7) Is the system safe to operate? 8) What is the rate of security related incidents and which types occur most often? 9) Are incidents being detected and resolved in a timely manner? 10) Is the supply chain secure? 11) How long does it take to securely recover from an attack? 12) How often do changes break existing functionality? 13) What parts of the system fail the most often? 14) Are disaster mitigation plans adequate and available? 15) How long does it take to install new components or technologies? 16) What can be done to prevent defect injection in the pipeline? 17) What are the root causes of defects in the product? 18) Are all QA activities being performed as planned? 19) Are all security assurance activities being performed as planned? 20) Has all the code been tested and gone through code analysis/peer reviews as planned?

7	Software Developer	• Turn requirements into code, build unit tests, consider initial deployment issues, develop application monitoring strategies, and review application monitoring events. • Perform secure programming and identify potential flaws in codes to mitigate vulnerabilities. • Analyze user needs and software requirements to determine feasibility of design within time and cost constraints. • Capture security controls used during the requirements phase to integrate security within the process, to identify key security objectives, and to maximize software security while minimizing disruption to plans and schedules. • Compile and write documentation of program development and subsequent revisions, inserting comments in the coded instructions so others can understand the program. • Apply cybersecurity functions (e.g., encryption, access control, and identity management) to reduce exploitation opportunities. • Identify and leverage the enterprise-wide security services while designing and developing secure applications (e.g., Enterprise PKI, Federated Identity server, Enterprise Antivirus solution) when appropriate. • Determine and document software patches or the extent of releases that would leave software vulnerable. • Translate security requirements into application design elements including documenting the elements of the software attack surfaces, conducting threat modeling, and defining any specific security criteria.	Goals: 1) Ensure the stories (i.e., requirements) are properly implemented (i.e., develop secure code) 2) Ensure users are satisfied 3) Ensure coding standards and organizational process guidelines are followed 4) Ensure interfaces are properly defined and implemented 5) Ensure software knowledge and skills are sufficient for role and purpose 6) Ensure weaknesses / vulnerabilities are not injected into the software 7) Ensure that testing is comprehensive and effective 8) Ensure changes don't break previous functionality Questions: 1) How many stories were planned versus completed (velocity)? 2) How many integration events that were planned were completed? 3) How much rework is for security related issues? 4) What was the code coverage in testing? (Code Coverage) 5) How much of the testing is automated? 6) How much rework is generated by test? (rework) 7) How many tests were performed for learning vs. for validation/verification? (test metrics) 8) What is the size of the test cases compared to the new code? (test metrics) 9) What is the provenance of libraries and are they free of vulnerabilities? How many vulnerabilities and weaknesses are inherited from third party software? (Vulnerabilities)

		• Identify and leverage the enterprise-wide version control system while designing and developing secure applications. • Identify security issues around steady state operation and management of software and incorporate security measures that must be taken when a product reaches its end of life. • Confer with systems analysts, engineers, programmers, and others to design application and to obtain information on project limitations and capabilities, performance requirements, and interfaces. • Apply coding and testing standards, apply security testing tools including "'fuzzing" static-analysis code scanning tools, and conduct code reviews. • Design countermeasures and mitigations against potential exploitations of programming language weaknesses and vulnerabilities in system and elements. • Store, retrieve, and manipulate data for analysis of system capabilities and requirements. • Conduct trial runs of programs and software applications to ensure that the desired information is produced and instructions and security levels are correct. • Correct errors by making appropriate changes and rechecking the program to ensure that desired results are produced. • Perform risk analysis [e.g., threat, vulnerability, and probability of occurrence] whenever an application or system undergoes a major change. • Direct software programming and development of documentation. • Evaluate factors such as reporting formats required, cost constraints, and	10) What is the lead time to implement a change? (Especially a High Urgency change) (Lead Time) 11) Do changes break existing functionality? (Change Failure/ Rollback) 12) Do implemented stories pass test cases? (test metrics) 13) How much rework is due to badly implemented stories or misunderstood stories? (rework) 14) What parts of the code (architecture) are likely to break? (including feature/subsystem maturity) 15) What knowledge or skills are needed to perform better? (staffing) 16) What is the acceptance rate from user representatives (e.g., stories accepted, refined, and/or rejected). How many changes are requested? 17) How many external dependencies are associated with each story? 18) What are application usage patterns? (Customer Satisfaction) 19) What is the application uptime? 20) What features/capabilities have unresolved defects/vulnerabilities? 21) What coding errors occur most often and how can they be prevented? 22) Are coding standards being followed and updated with lessons learned?

		need for security restrictions to determine hardware configuration. • Develop strategies for application monitoring and event analysis. • Identify basic common coding flaws at a high level. • Consult with customers about software system design and maintenance. • Consult with engineering staff to evaluate interface between hardware and software. • Develop software system testing and validation procedures, programming, and documentation. • Apply secure code documentation. • Identify security implications and apply methodologies within centralized and decentralized environments across the enterprise's computer systems in software development. • Address security implications in the software acceptance phase including completion criteria, risk acceptance and documentation, common criteria, and methods of independent testing. • Supervise and assign work to programmers, designers, technologists and technicians, and other engineering and scientific personnel. • Modify and maintain existing software to correct errors, to adapt it to new hardware, or to upgrade interfaces and improve performance. • Perform integrated quality assurance testing for security functionality and resiliency attack. • Develop secure code and error handling. • Analyze information to determine, recommend, and plan the development of a new application or modification of an existing application. • Design, develop, and modify software systems, using scientific analysis and	

		mathematical models to predict and measure outcome and consequences of design.	
8	User	• Use the product under development on a constant or regular basis • Provide feedback on how well the product under development functions • Determine the amount of value the product under development provides • Identify and report defects not recognized by the engineering process • Identify and request enhancements to the product under development that will increase value	Goals: 1) Ensure their needs are met with the best product possible 2) Ensure their desired changes and future needs will be met as soon as possible Questions: 1) What products will best satisfy my needs? 2) What are my future needs and what products will best satisfy them? 3) Am I happy with the product and product support? 4) Are my changes and recommendations being addressed in a timely fashion? 5) What is the cost of delaying a change or need? 6) What is the value proposition for a change or need?
9	User Experience	• User experience design, user feedback acceptance, and user experience validation of the development product outputs. • Create innovative solutions for a wide variety of product design challenges, including mobile, desktop, hardware interfaces, physical environments, and person-to-person interactions. • Lead the design for new experiences and improvements of existing experiences. • Plan and define strategy for the direction of future iterations. • Quickly iterate on multiple interactive design solutions and work through details. • Advocate for design solutions, highlighting inputs that influenced the decisions including business and user	Goals: 1) Ensure User Satisfaction 2) Ensure the features / capabilities that the user most desires are delivered as quickly as possible 3) Ensure future user needs are identified and can be satisfied Questions: 1) Is the user satisfied with delivered products & services? 2) What features/capabilities are most used and valued by the user? 3) Does the system provide the desired functionality when needed? 4) How long does it take to incorporate a user desired feature/capability? 5) Is user feedback sufficient and how could it be improved?

		goals, demographic and usage data, and research findings • Assess and optimize the performance of new and existing features by actively participating in user research and assessing performance metrics • Contribute to the group's shared knowledge of user-centered design and research methodologies • Deliver work that's not only user-friendly, aesthetically engaging, but which also produces results • Develop and maintain detailed user-interface specifications • Develop high level, detailed story-boards, mock-ups, and prototypes to effectively communicate interaction and design ideas • Present design work to multiple teams and senior leadership for review and feedback • Work alongside engineers and product managers throughout all stages of the production cycle	6) What are the future user needs and how can they be met? In what timeline? 7) How easy is the system/product to operate by the user?

Table 3: Roles and Responsibilities Table

#	Name	Uses Resource
1	Architect	• Analyze, design, and implement strategies for continuous deployment of production and pre-production systems, and development and test pipelines Document how the implementation of a new system or new interface between systems impacts the current and target environment including but not limited to security posture. • Generate and implement plans for integrating new systems into existing infrastructure and employ secure configuration management processes. • Ensure that acquired or developed system(s) and architecture(s) are consistent with organization's cybersecurity architecture guidelines. • Identify and prioritize critical business functions in collaboration with organizational stakeholders. • Provide advice on project costs, design concepts, or design changes. • Analyze candidate architectures, allocate security services, and select security mechanisms.

		• Develop enterprise architecture or system components required to meet user needs. • Document and update as necessary all definition and architecture activities. • Integrate results regarding the identification of gaps in security architecture. • Plan implementation strategy to ensure that enterprise components can be integrated and aligned. • Translate proposed capabilities into technical requirements. • Capture and integrate essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event. • Analyze user needs and requirements to plan architecture. • Analyze system requirements • Assess effectiveness of existing systems • Recommend alternative technologies and improvements • Write detailed functional specifications that document the architecture development process. • Lead design and review of new systems • Evaluate security architectures and designs to determine the adequacy of security design and architecture proposed or provided in response to requirements contained in acquisition documents. • Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration.
2	Business Analyst	Business or Mission
3	Compliance	Business or Mission
4	Contract Specialist	Business or Mission
5	Executive	Business or Mission
6	Financier	Business or Mission
7	Legal	Business or Mission
8	Marketing	Business or Mission
9	Owner	Business or Mission
10	Program Manager	Business or Mission
11	Project Manager	Business or Mission
12	Sales	Business or Mission

13	Supplier	Business or Mission
14	Systems Analyst	Business or Mission
15	Relevant Stakeholders	Business or Mission Operational Users Engineering
16	Site Reliability Engineer	• Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration. • Capture and integrate essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event. • Recommend alternative technologies and improvements • Define best practices to ensure software releases are consistent and repeatable • Plan and coordinate incident responses • Respond to incidents that impact relevant stakeholders • Monitor system infrastructure to ensure that all service and operational level agreements are met • Develop and maintain response playbooks to address incidences • Conduct post-incident reviews, document findings, and take action based on what was learned in order to bolster the reliability of the service. • Develop software to make system and product under development infrastructure more resilient, automated, and self-healing over time • Increase reliability and performance of the system and product under development • Plan and implement disaster mitigation, prevention, and response activities • Improve the system and product under development in order to maximize effectiveness based on defined service level objectives • Reduce the time required to resolve incidents and restore service (i.e. reduce mean time to repair) • Continuously test the operational readiness and efficiency of the infrastructure • Conduct chaos experiments to see how the system and product under development infrastructure will behave under emergent behavior during runtime • Meet the system and product under development computing needs • Collect and monitor infrastructure performance data • Coordinate software and hardward installations, testing, and transitions

		• Introduce alternative technologies to improve or enhance infrastructure in support of system or product under development requirements • Manage the underlining security of all infrastructure components and inter-component information transfers
17	Business or Mission Domain Expert	• Define the overall need in which the product under development is intended to satisfy • Analyze the need in order to describe it in sufficient details that a solution can be engineered. • Develop and track indicators of success and taking corrective actions when needed based on available information • Develop and maintain budgeting and forecasting • Perform variance analysis when actuals deviate from plan in order to determine if corrective actions are needed • Plan and monitor all activities within scope of authority • Share information between relevant stakeholders based on given stakeholder's needs • Document and maintain business processes • Lead continuous reviews of processes and develop optimization strategies • Anticipate changes to market and user needs • Coordinate the sharing of ideas and findings among relevant stakeholders • Allocate resources and maintain cost efficiency

		• Prioritize initiatives based on business needs and available resources • Ensure that all contractual obligations are met • Coordinate with government regulators • Ensure compliance with applicable laws and regulations • Ensure that audit discoveries and compliance violations are addressed • Plan, implement, and oversee risk identification and mitigation activities • Coordinate training needs of workforce • Develop, maintain, and enforce policies • Address relevant stakeholder concerns regarding legal compliance • Review contracts to prevent disputes and financial risks • Develop and use appropriate contract provisions and amendments which comply with legal requirements and policies • Review, negotiate, and approve contract terms and conditions • Process payments in accordance with agreed to payment structures • Report on revenue and expenditure • Create, maintain and execute sound business, program, and project plans • Manage the demands of stakeholders • Maintain financial records for all transactions and changes • Coordinate financial audits • Monitor all deposits and payments • Process invoices • Organize activities in accordance with the mission and goals of the organization • Increase brand or mission awareness • Optimize marketing strategies • Analyze the competition and prepare forecasts • Develop and manage long-term goals • Obtain funding for uninterrupted delivery of services • Evaluate user needs and promote products and services • Coordinate user needs
18	Cyber Legal Advisor	Engineering
19	Database Administrator	Engineering
20	DevOps Engineer	Engineering
21	Infrastructure Architect	Engineering
22	Infrastructure Engineer	Engineering
23	Infrastructure Operator	Engineering

24	Network Operations Specialist	Engineering
25	Quality Assurance Engineer	Engineering
26	Security Architect	Engineering
27	System Administrator	Engineering
28	Systems Engineer	Engineering
29	Technical Support Specialist	Engineering
30	Test Engineer	Engineering
31	UI/UX Designer	Engineering
32	Customer	Operational Users
33	External User	Operational Users
34	Internal User	Operational Users
35	Product Owner	Operational Users
36	Subject Matter Expert	Operational Users
37	DevSecOps Champion	• Provide leadership and vision to deliver the changes necessary in adopting DevOps practices. • Work with organizational units to remove barriers to communications and information exchange • Determine which tools or processes are best suited for solving long-term needs • Improve processes in order to better serve the customer • Determine obstacles faced by organizational units

38	Release Engineer	• Provide technical support of a product from development to production and maintenance. • Perform as technical liaison for Engineering and Operations on every aspect associated with final builds and control baseline issues. • Prepare, evaluate and maintain tools supporting and process automation for software or hardware product release. • Ensure capability to compile and assemble software through source code and store tools in source control. • Design, manage and execute tools and scripts to develop different versions of products on wide-range operating systems. • Develop dashboards to quantify internal processes efficiency continuously. • Develop library of tools for automating manual workflows in development process. • Interact with release engineering and QE to debug and also resolve identified issues. • Respond constantly and aggressively to automated test and build issues. • Support integration of new technologies along with companies. • Develop and present general releases, service packs, web products and beta products. • Correct build errors working with development engineers. • Perform with others for analysis, evaluations along with design options and execute process improvements. • Perform with project teams to identify apt build schedule and initiate packaging and build process.

39	Software Developer	• Turn requirements into code, build unit tests, consider initial deployment issues, develop application monitoring strategies, and review application monitoring events. • Perform secure programming and identify potential flaws in codes to mitigate vulnerabilities. • Analyze user needs and software requirements to determine feasibility of design within time and cost constraints. • Capture security controls used during the requirements phase to integrate security within the process, to identify key security objectives, and to maximize software security while minimizing disruption to plans and schedules. • Compile and write documentation of program development and subsequent revisions, inserting comments in the coded instructions so others can understand the program. • Apply cybersecurity functions (e.g., encryption, access control, and identity management) to reduce exploitation opportunities. • Identify and leverage the enterprise-wide security services while designing and developing secure applications (e.g., Enterprise PKI, Federated Identity server, Enterprise Antivirus solution) when appropriate. • Determine and document software patches or the extent of releases that would leave software vulnerable. • Translate security requirements into application design elements including documenting the elements of the software attack surfaces, conducting threat modeling, and defining any specific security criteria. • Identify and leverage the enterprise-wide version control system while designing and developing secure applications. • Identify security issues around steady state operation and management of software and incorporate security measures that must be taken when a product reaches its end of life. • Confer with systems analysts, engineers, programmers, and others to design application and to obtain information on project limitations and capabilities, performance requirements, and interfaces. • Apply coding and testing standards, apply security testing tools including "'fuzzing" static-analysis code scanning tools, and conduct code reviews. • Design countermeasures and mitigations against potential exploitations of programming language weaknesses and vulnerabilities in system and elements. • Store, retrieve, and manipulate data for analysis of system capabilities and requirements. • Conduct trial runs of programs and software applications to ensure that the desired information is produced and instructions and security levels are correct. • Correct errors by making appropriate changes and rechecking the program to ensure that desired results are produced.

		• Perform risk analysis [e.g., threat, vulnerability, and probability of occurrence] whenever an application or system undergoes a major change. • Direct software programming and development of documentation. • Evaluate factors such as reporting formats required, cost constraints, and need for security restrictions to determine hardware configuration. • Develop strategies for application monitoring and event analysis. • Identify basic common coding flaws at a high level. • Consult with customers about software system design and maintenance. • Consult with engineering staff to evaluate interface between hardware and software. • Develop software system testing and validation procedures, programming, and documentation. • Apply secure code documentation. • Identify security implications and apply methodologies within centralized and decentralized environments across the enterprise's computer systems in software development. • Address security implications in the software acceptance phase including completion criteria, risk acceptance and documentation, common criteria, and methods of independent testing. • Supervise and assign work to programmers, designers, technologists and technicians, and other engineering and scientific personnel. • Modify and maintain existing software to correct errors, to adapt it to new hardware, or to upgrade interfaces and improve performance. • Perform integrated quality assurance testing for security functionality and resiliency attack. • Develop secure code and error handling. • Analyze information to determine, recommend, and plan the development of a new application or modification of an existing application. • Design, develop, and modify software systems, using scientific analysis and mathematical models to predict and measure outcome and consequences of design.

40	Security Engineer	• Understand security sensitivities and how they affect the design, implementation, configuration, and delivery aspects of the software development lifecycle, and ensure that security considerations affect every phase of this lifecycle. • Define and prioritize essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event • Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recovery/restoration. • Develop/integrate cybersecurity designs for systems and networks with multilevel security requirements or requirements for the processing of multiple classification levels of data primarily applicable to government organizations (e.g., UNCLASSIFIED, SECRET, and TOP SECRET). • Document and address organization's information security, cybersecurity architecture, and systems security engineering requirements throughout the acquisition life cycle. • Employ secure configuration management processes. • Ensure that acquired or developed system(s) and architecture(s) are consistent with organization's cybersecurity architecture guidelines. • Identify and prioritize critical business functions in collaboration with organizational stakeholders. • Perform security reviews, identify gaps in security architecture, and develop a security risk management plan. • Provide advice on project costs, design concepts, or design changes. • Provide input on security requirements to be included in statements of work and other appropriate procurement documents. • Provide input to the Risk Management Framework process activities and related documentation (e.g., system life-cycle support plans, concept of operations, operational procedures, and maintenance training materials). • Define and document how the implementation of a new system or new interfaces between systems impacts the security posture of the current environment. • Analyze candidate architectures, allocate security services, and select security mechanisms. • Develop a system security context, a preliminary system security Concept of Operations (CONOPS), and define baseline system security requirements in accordance with applicable cybersecurity requirements. • Evaluate security architectures and designs to determine the adequacy of security design and architecture proposed or provided in response to requirements contained in acquisition documents.

		• Write detailed functional specifications that document the architecture development process. • Analyze user needs and software requirements to determine feasibility of design within time and cost constraints. • Develop enterprise architecture or system components required to meet user needs. • Document and update as necessary all definition and architecture activities. • Determine the protection needs (i.e., security controls) for the information system(s) and network(s) and document appropriately. • Translate proposed capabilities into technical requirements. • Assess and design security management functions as related to cyberspace.
41	User	• Use the product under development on a constant or regular basis • Provide feedback on how well the product under development functions • Determine the amount of value the product under development provides • Identify and report defects not recognized by the engineering process • Identify and request enhancements to the product under development that will increase value

42	User Experience	• User experience design, user feedback acceptance, and user experience validation of the development product outputs. • Create innovative solutions for a wide variety of product design challenges, including mobile, desktop, hardware interfaces, physical environments, and person-to-person interactions. • Lead the design for new experiences and improvements of existing experiences. • Plan and define strategy for the direction of future iterations. • Quickly iterate on multiple interactive design solutions and work through details. • Advocate for design solutions, highlighting inputs that influenced the decisions including business and user goals, demographic and usage data, and research findings • Assess and optimize the performance of new and existing features by actively participating in user research and assessing performance metrics • Contribute to the group's shared knowledge of user-centered design and research methodologies • Deliver work that's not only user-friendly, aesthetically engaging, but which also produces results • Develop and maintain detailed user-interface specifications • Develop high level, detailed storyboards, mock-ups, and prototypes to effectively communicate interaction and design ideas • Present design work to multiple teams and senior leadership for review and feedback • Work alongside engineers and product managers throughout all stages of the production cycle

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu

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.

DM21-1109