



# SE Use Cases

## SysML Assessment and Roadmap

### Working Group

John Watson

Lockheed Martin

OMG Technical Conference

Boston MA

6/17/2014



- Assessment method
- Summary of what was done
- Key Findings (list below is for every finding)
  - Area of improvement
  - What evaluation criteria are most impacted
  - What category of action would address this area of improvement (spec change, tool improvement, process change, other)
- Priority (per discussion)
- Future assessment effort (per discussion)



- Intent – Capture a Repository of Systems Engineering (SE) modeling practices
- Capture in a SysML Model
  - Define the Context
    - System Engineering Development Environment
  - Identify the SE Process Use Cases within the defined context
    - First Pass – Review each Use Case’s goal and Primary Actor
      - Available in a document
    - Selected 2-3 that will provide the most benefit
    - For each selected Use Case;
      - Derive and explore behavior
      - Iterate with domain experts until sufficient
      - Ultimately Identify/Derive new SysML Requirements
      - Determine how we are, or need to, support in SysML

# Approach - What do SE Process Use Cases produce?

- Definitions:
  - Process defines what tasks need to be done
  - Method defines how to do the tasks
- The SE Process Use Cases will represent the Process not the Methods
- Therefore, the SE Process Use Cases will identify:
  - What behavior is required in the SE Development Environment for **Systems Engineers** to do their work
  - Evaluate how **SysML** is used to support each behavioral entity
- Process Behavior Examples;
  - Conduct a Review, Capture System behavior/structure/requirements, measure change impact, produce a deliverable, select a multi-domain view, create a baseline, assess change impact, etc.
- Derive requirements from the resulting behavior



- Used the SEBoK to identify
  - Description of Knowledge Areas
  - Organized based on 15288 Life Cycle Stages
- ISO/IEC 15288
  - Definition of Life Cycle Stages
  - Content of Life Cycle Stages



- Architect a Development System
  - Decompose System into a set of Logical Components with Interfaces
    - Example Components – Modeling Tools, Analysis Tools, CM tools. Etc.
    - This effort will be as domain independent as possible
- Provide input, clarity and vision to:
  - Standards Groups –
    - Identifying enhancements to existing standards
    - Demonstrate the need of **integration threads across standards**
    - Identify new standards
  - Tool Vendors –
    - A full view of the need
    - A specification of what needs to be built Tool Vendors
    - They provide the Methods



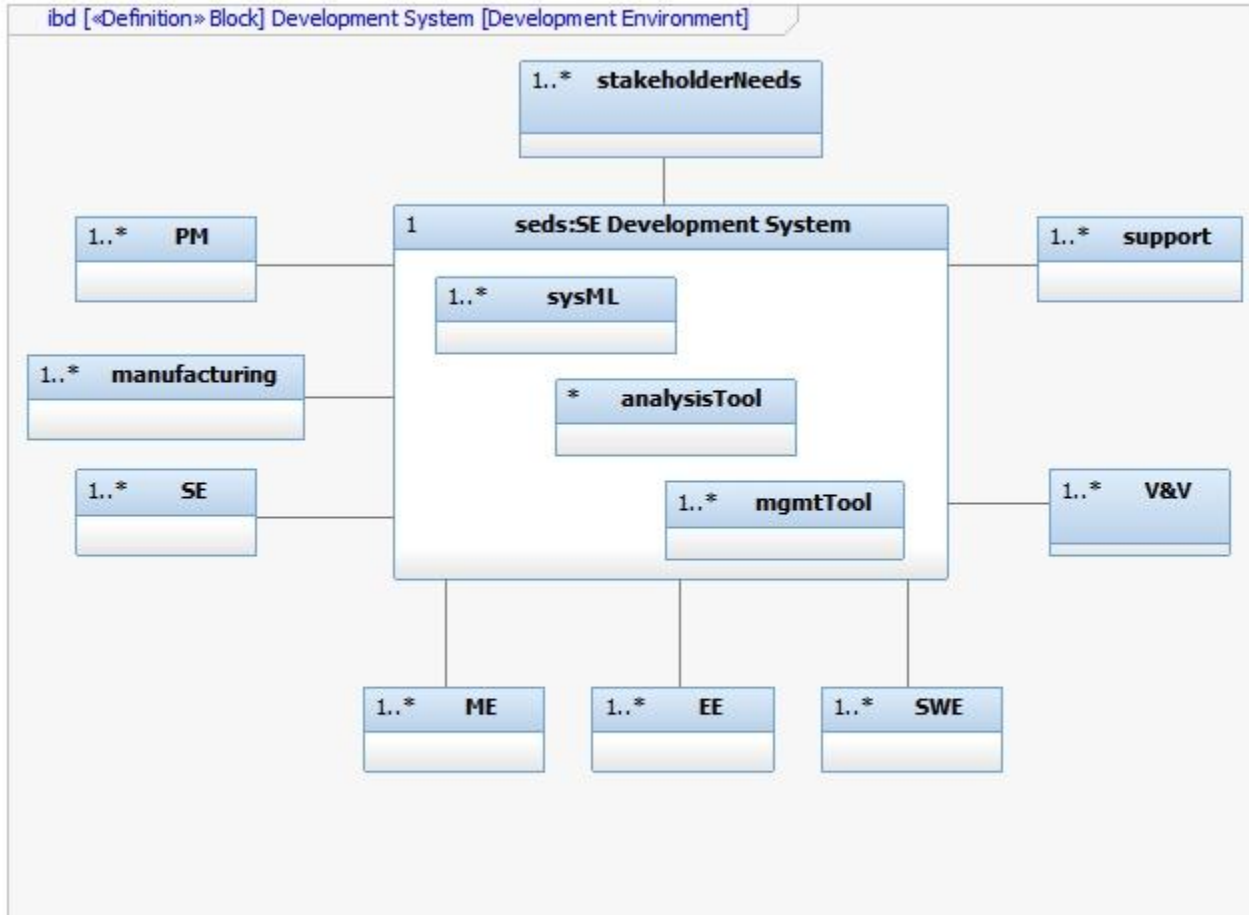
- Schedule – WAG
  - Context Diagram - May 6<sup>th</sup> - Done
  - Review Use Case survey, update and select focused Use Cases –May 27<sup>th</sup> Done
  - Evaluate Selected Use Cases, refine as necessary – Complete June 17<sup>th</sup> - Done
  - Summarize SysML Impact – Complete July 15<sup>th</sup> ???



# SUMMARY OF WHAT WAS DONE



# SE Development Environment Context

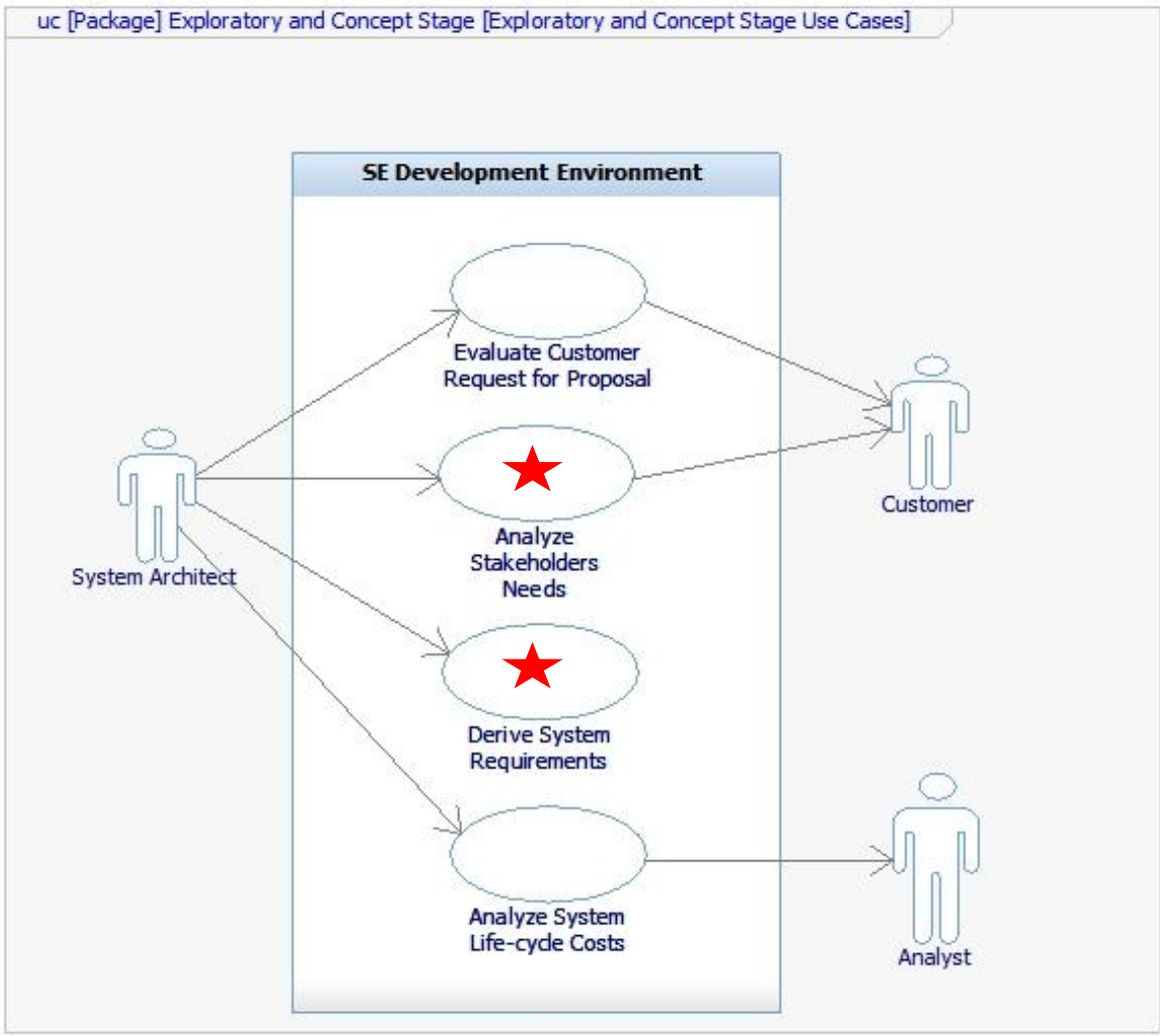


To measure SysML effectiveness we need to understand the context and how it is used

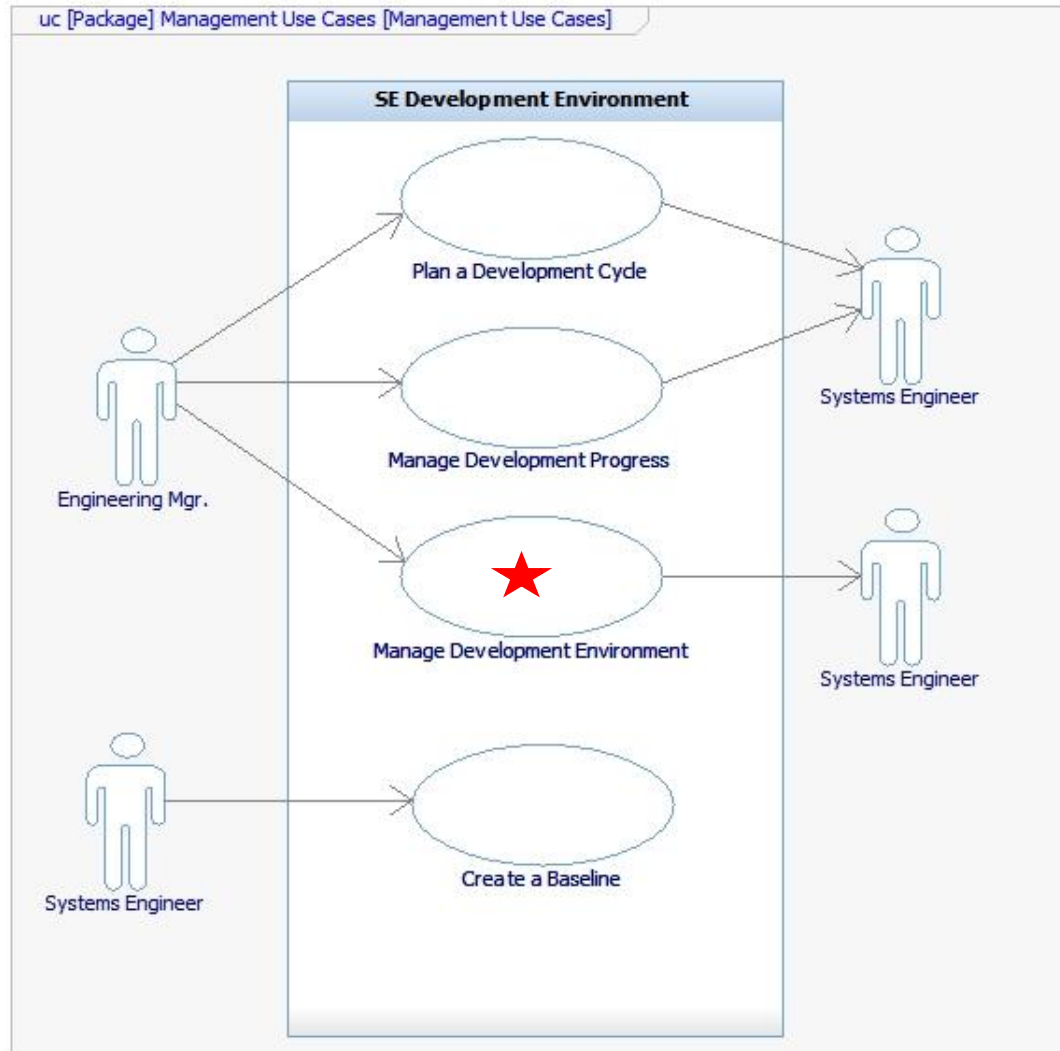


- **Exploratory/Concept Stage**
- **System Development Stage**
  - Management Use Cases
  - SE Domain Use Cases
  - Validation and Verification Use Cases
- **Production Stage Use Cases**
- **Product and Service Life Management**

# Exploratory/Concept Stage

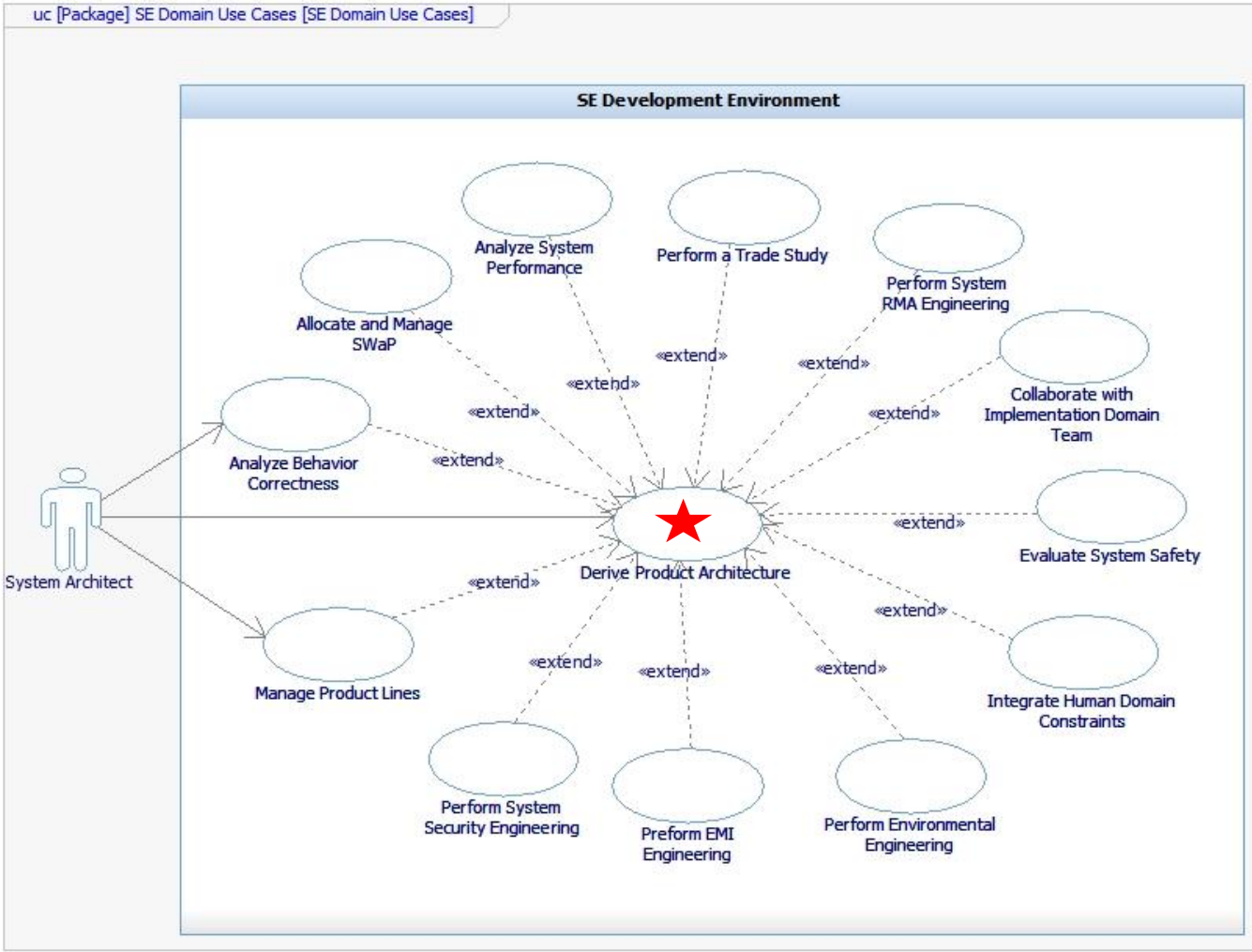


# System Development Stage – Management Use Cases



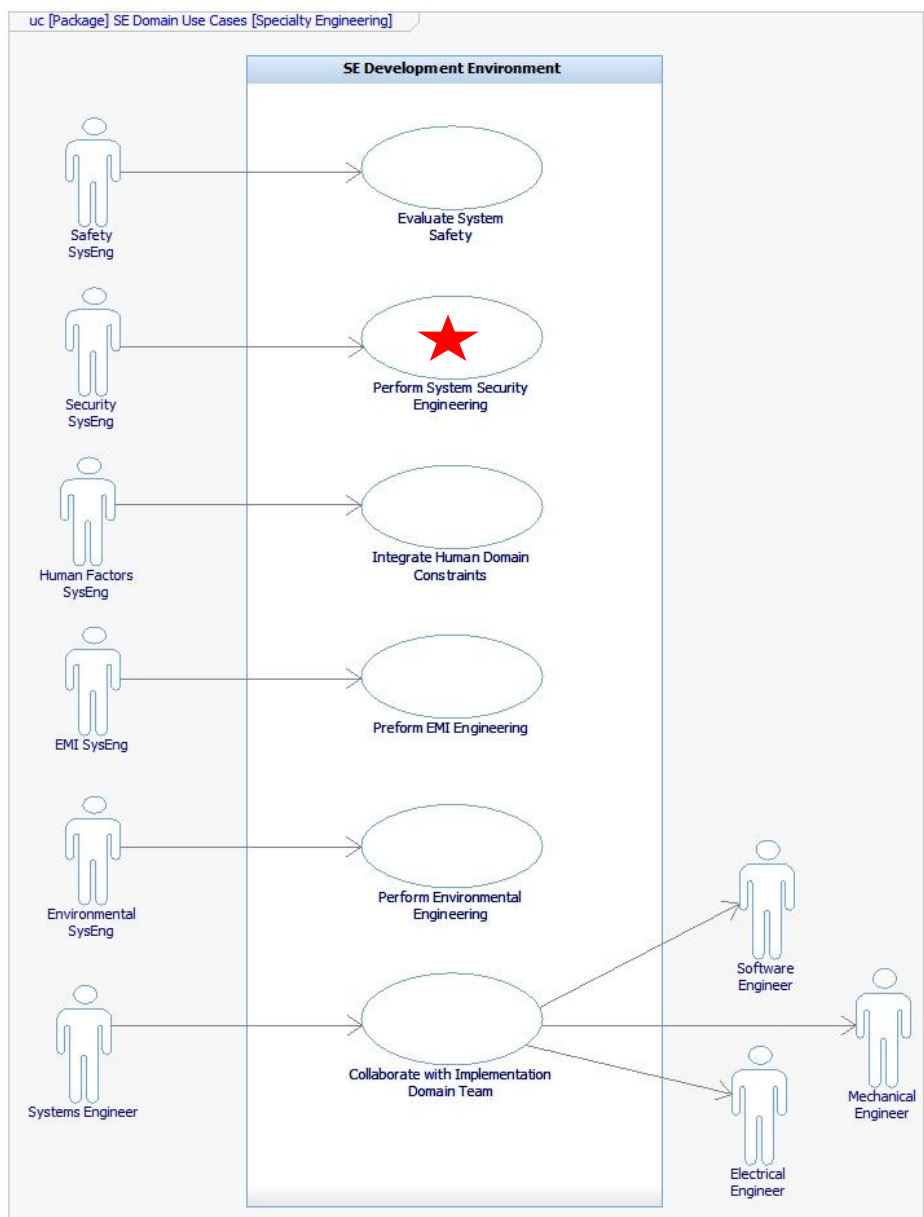


# System Development Stage – System Development



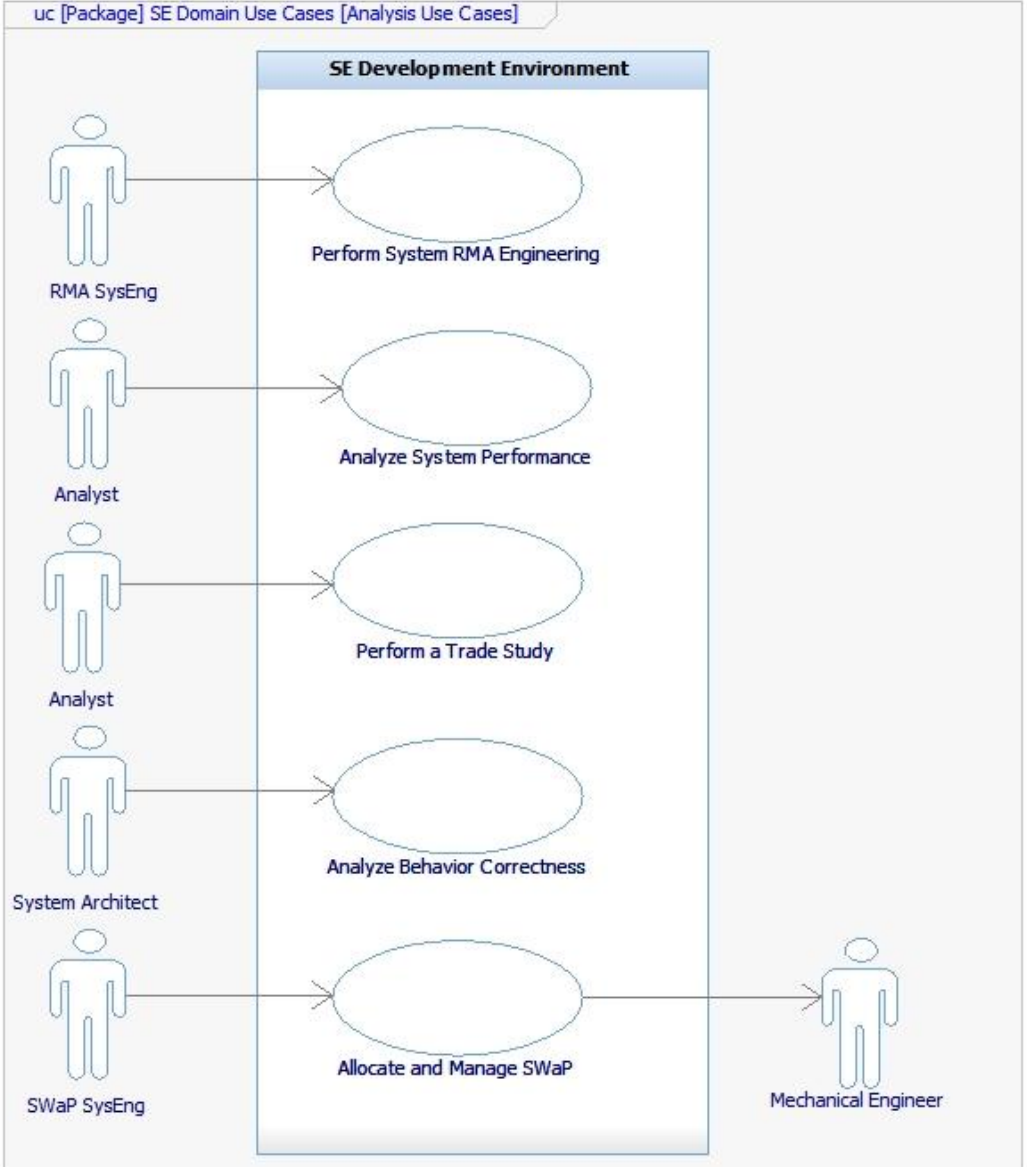


# System Development Stage – System Development - Specialty

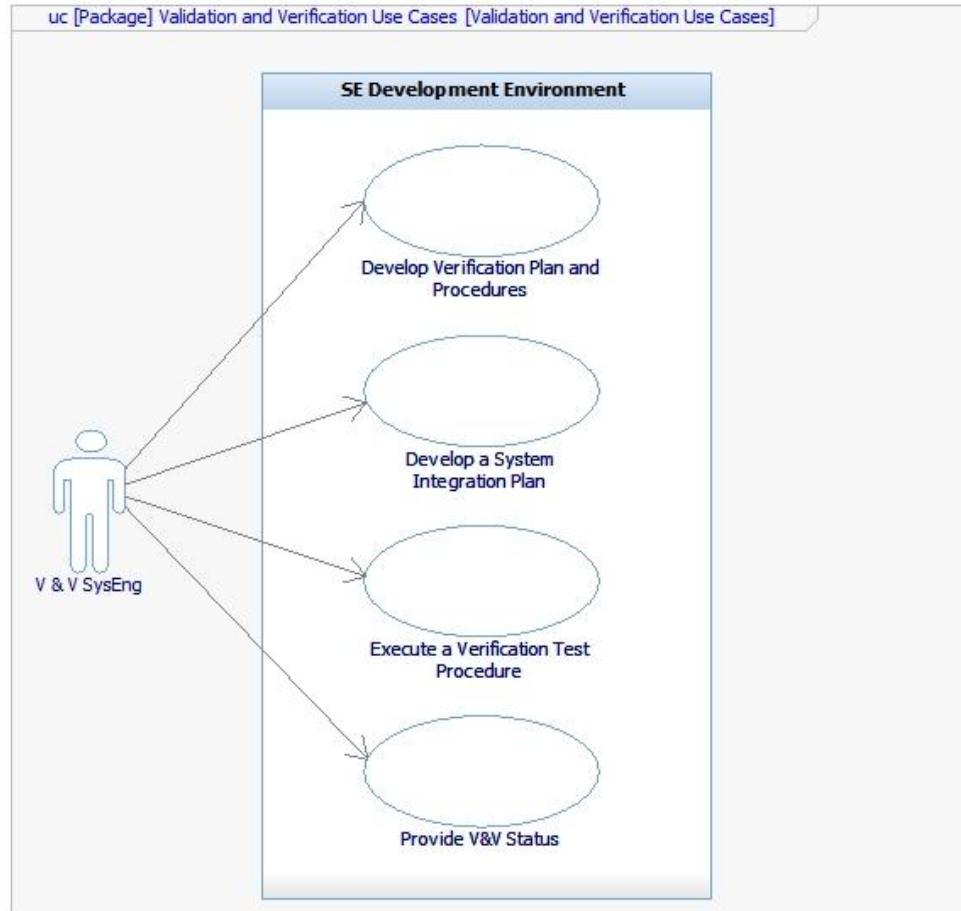




# System Development Stage – System Development Analysis

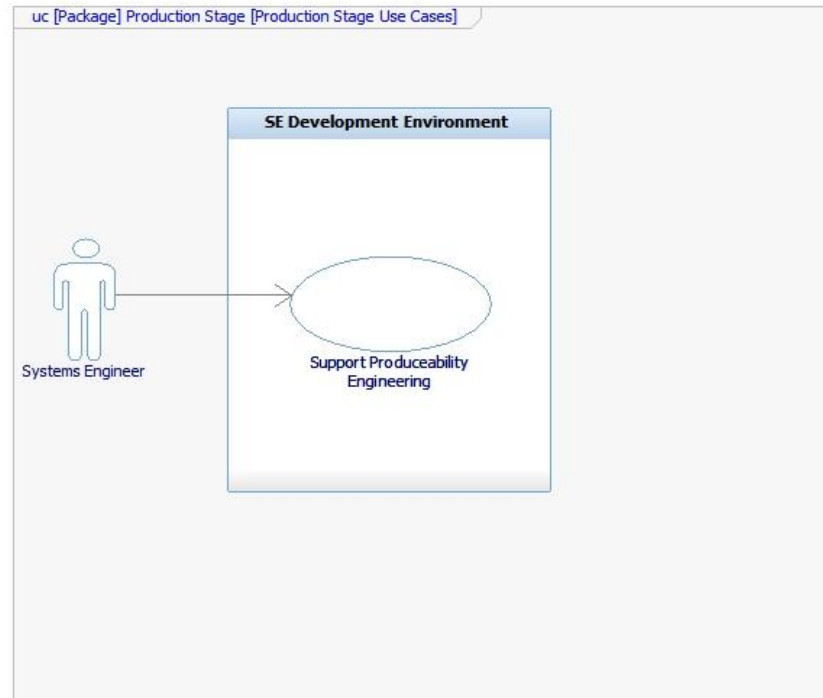


# System Development Stage – Validation and Verification

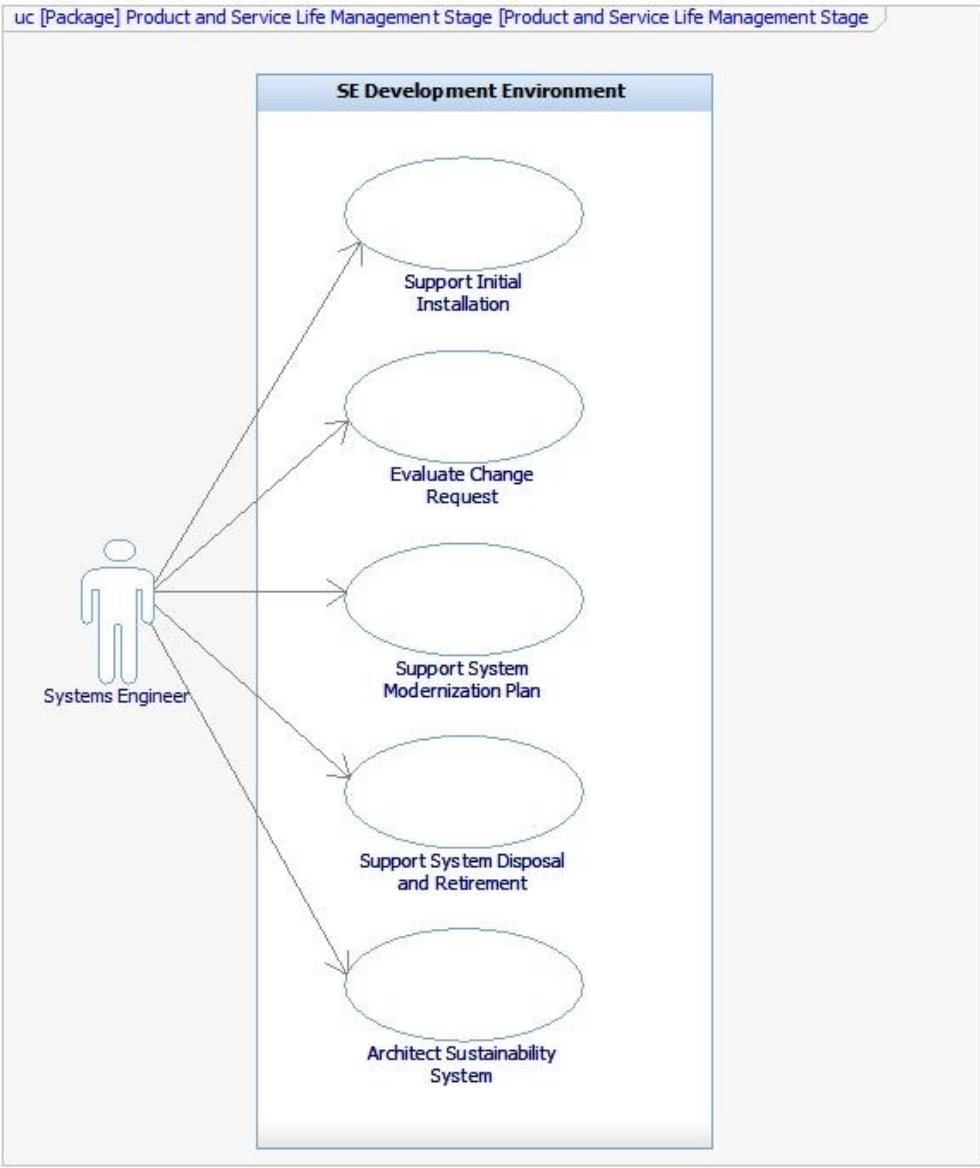




# Production Stage Use Cases



# Product and Service Life Management Use Cases



# Example Use Case to Evaluate SysML Usage

## Perform System Security Engineering



- **Goal** – The goal of this use case is to incorporate in the system of interest the necessary security design features to meet the needs of the customer.
- **Primary Actor** – SE Security Specialist
- **Secondary Actors** –
- **Preconditions** –
  1. A list of known potential threats are available
  2. A list of applicable policy documentation is available

# Activity - Perform System Security Engineering



This use case begins early in the development cycle and continues to iterate through the remaining development cycles as the product matures.

1. Obtain and/or define the customer's security protection goals for the following security domains including:
  1. Information security governance and risk management
  2. Access control
  3. Cryptography
  4. Physical (environmental) security
  5. Security architecture and design
  6. Business continuity and disaster recovery planning
  7. Telecommunications and network security
  8. Application development security
  9. Operations security
  10. Legal, regulations, investigations, and compliance
2. Capture the system vulnerabilities by analyzing the known or perceived threats and their behavior.



# Activity - Perform System Security Engineering

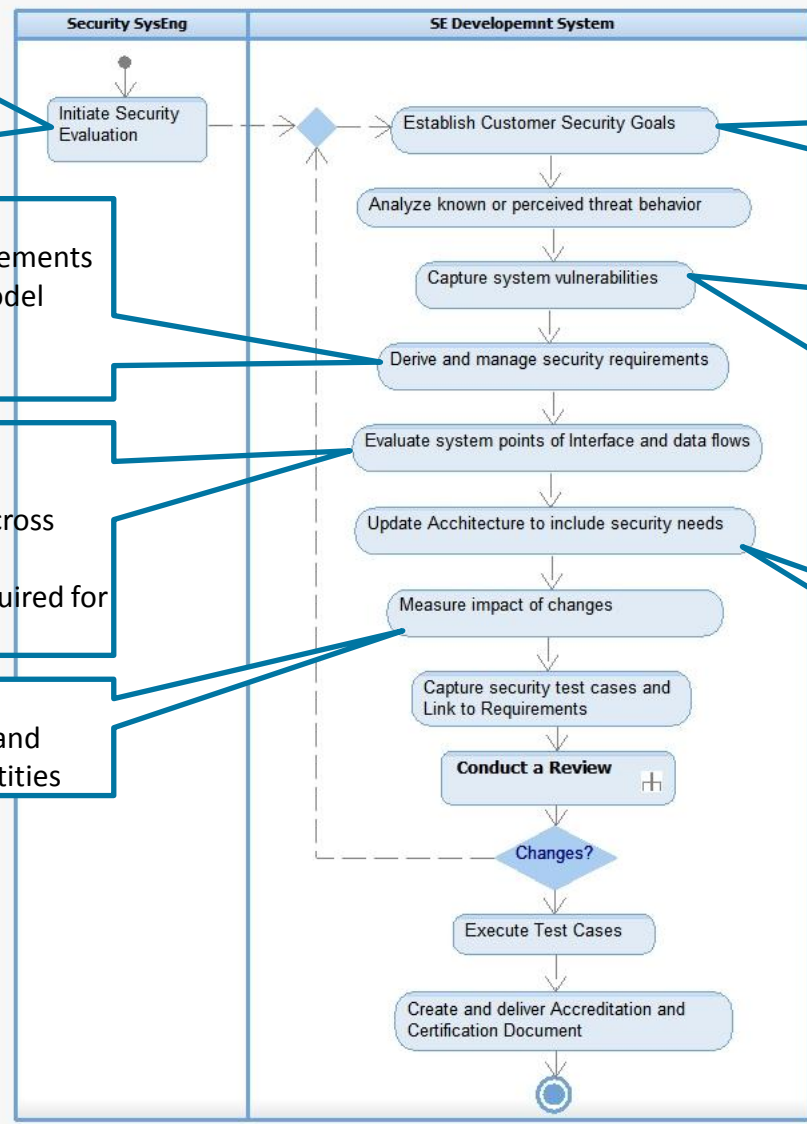
3. Derive a set of security requirements that address the vulnerabilities and other applicable security policy documents.
4. Evaluate points of Interface;
  1. Identify all external interface points
  2. Identify internal interface points of major subsystems such as server farms, sensors, security management, business network, etc.
  3. Identifying the points of interface may have been completed earlier in a use case such as “Derive Product Architecture”.
  4. Determine and capture the level of security required for the information exchanged at the points of interface.
5. Capture the security architecture design that satisfy these requirements and minimize or contain the vulnerabilities.
6. Measure the change impact to other domains and mitigate issues
7. Conduct appropriate reviews within engineering and with the customer
8. Capture test cases that validate the security requirements have been reached.
9. If the proposed design does not meet the System goals, refine the design.
10. Prepare the necessary documentation for system accreditation and certification.

**Post Conditions** – Accreditation Certificate is submitted



# Perform System Security Engineering UC

act [Use Case] Perform System Security Engineering [Security Engineering Task]



1. SEDS shall be compatible with domain specific profiles that include domain specific Ionic representations

1. SEDS shall allow users to;

- derive and capture Requirements
- Produce traceability to model elements
- Manage Requirements

1. SEDS shall capture Interfaces specifications:

- Identify data that flows across interfaces
- Capture security level required for each Data item

1. SEDS shall assess Change Impact, showing two domains at a time, and highlight one or more domain entities

1. SEDS shall create and share validated reference libraries

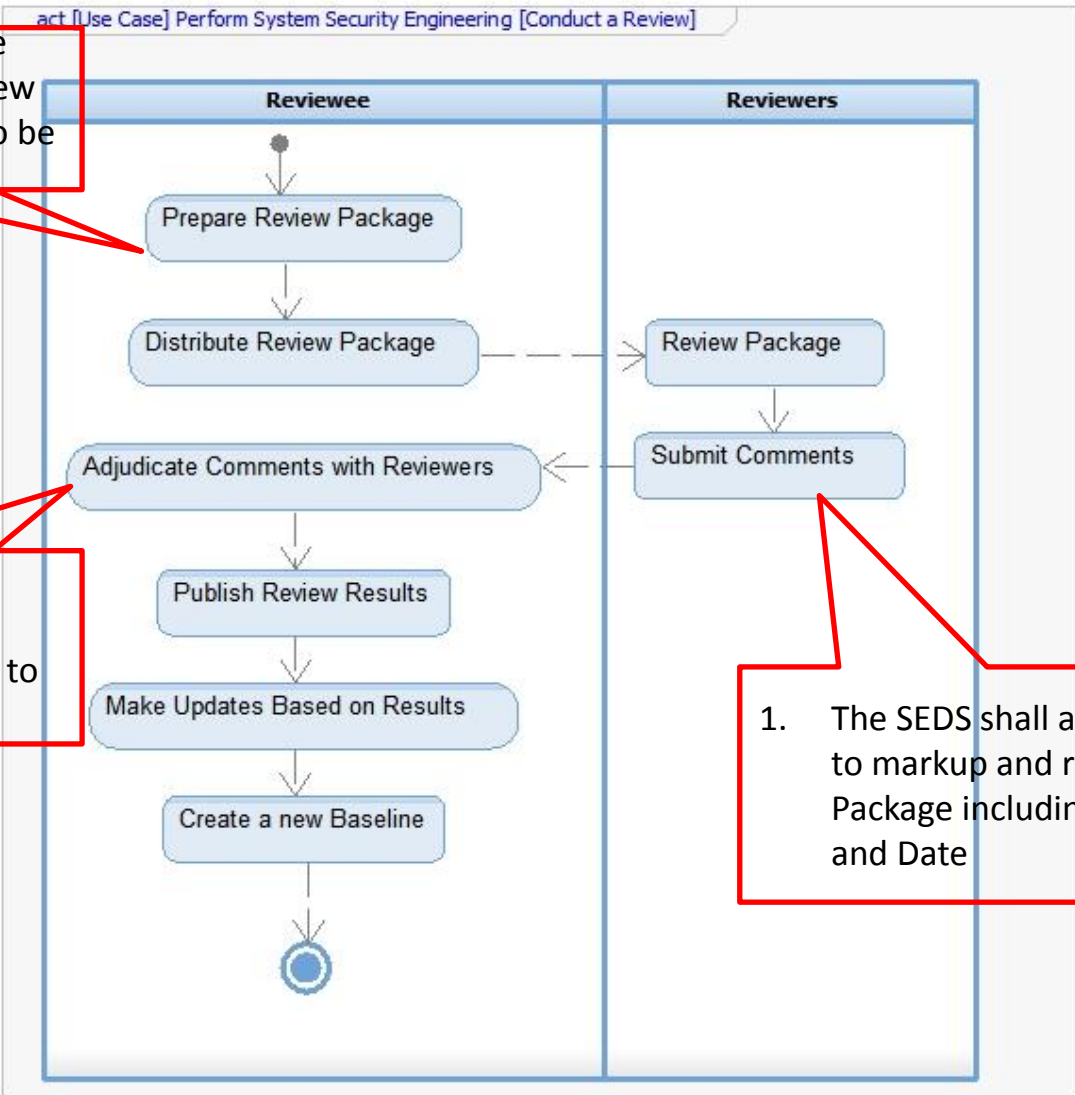
1. SEDS shall:

- Define and organize UCs
- Create Activities to expose vulnerabilities
- Integration with threat analysis tools

1. SEDS shall view one or more Domain entities and suppress or gray out other entities



# Artifact Review Pattern



1. The SEDS shall allow the Reviewee to create a view to isolate information to be reviewed.

1. The SEDS shall allow a reviewee to evaluate comments and respond to the reviewer.

1. The SEDS shall allow a reviewer to markup and return a Review Package including the Reviewer and Date



# Use Case Maturity Level

Category	Use Case	Maturity
Exploratory and Concept Stage	Analyze Stakeholders Needs	Evolving Description
Exploratory and Concept Stage	Derive System Requirements	Evolving Description
Exploratory and Concept Stage	Analyze System Life-cycle Costs	Early Text Description
Exploratory and Concept Stage	Evaluate Customer Request for Proposal	Evolving Description
Management Use Cases	Manage Development Progress	Early Text Description
Management Use Cases	Manage Development Environment	Evolving Description
Management Use Cases	Create a Baseline	Early Text Description
Management Use Cases	Plan a Development Cycle	Evolving Description
Product and Service Life Management Stage	Support System Disposal and Retirement	Goal Only
Product and Service Life Management Stage	Support System Modernization Plan	Goal Only
Product and Service Life Management Stage	Evaluate Change Request	Evolving Description
Product and Service Life Management Stage	Support Initial Installation	Goal Only
Product and Service Life Management Stage	Architect Sustainability System	Early Text Description
Production Stage	Support Produceability Engineering	Early Text Description
SE Domain Use Cases	Perform a Trade Study	Early Text Description
SE Domain Use Cases	Analyze System Performance	Early Text Description

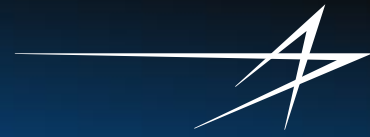




# Use Case Maturity

Category	Use Case	Maturity
SE Domain Use Cases	Manage Product Lines	Goal Only
SE Domain Use Cases	Integrate Human Domain Constraints	Early Text Description
SE Domain Use Cases	Perform Environmental Engineering	Goal Only
SE Domain Use Cases	Collaborate with Implementation Domain Team	Early Text Description
SE Domain Use Cases	Preform EMI Engineering	Goal Only
SE Domain Use Cases	Derive Product Architecture	Evolving Description
SE Domain Use Cases	Analyze Behavior Correctness	Goal Only
SE Domain Use Cases	Evaluate System Safety	Evolving Description
SE Domain Use Cases	Perform System RMA Engineering	Evolving Description
SE Domain Use Cases	Allocate and Manage SWaP	Evolving Description
SE Domain Use Cases	Perform System Security Engineering	Early Activity
Validation and Verification Use Cases	Execute a Verification Test Procedure	Goal Only
Validation and Verification Use Cases	Develop Verification Plan and Procedures	Early Text Description
Validation and Verification Use Cases	Develop a System Integration Plan	Early Text Description
Validation and Verification Use Cases	Provide V&V Status	Goal Only

# Summary of Use Cases Elaborated



Category	Use Case	Selection
Exploratory and Concept Stage	Analyze Stakeholders Needs	Selected
Exploratory and Concept Stage	Derive System Requirements	Selected
Management Use Cases	Manage Development Environment	Selected
SE Domain Use Cases	Derive Product Architecture	Selected
SE Domain Use Cases	Perform System Security Engineering	Selected & Pattern



# Use Case Pattern

**Activity** -This use case begins early in the development cycle and continues to iterate through the remaining development cycles as the product matures.

Early in the development process there may not be sufficient design detail to determine if the product needs can be met with 100% certainty. Therefore assumptions and experience are required to fill those gaps until the design matures. These unknowns, risks and assumptions need to be captured and managed.

1. Establish the customer needs that are impacted by this domain including Measures of Effectiveness
2. Capture information design and assumptions
3. Measure domain design effectiveness, typically by use of analysis
4. Measure impact in other domain areas
5. Refine design
6. If the proposed design does not meet the System goals execute the following steps.
7. Iterate back to the appropriate previous step until design seems sound
8. Conduct appropriate peer/engineering/customer level reviews
9. Capture results of task including defined metrics

**Post Conditions -**



# Use Cases Based on Pattern

Category	Use Case	Selection
SE Domain Use Cases	Analyze System Performance	Pattern
SE Domain Use Cases	Manage Product Lines	Pattern
SE Domain Use Cases	Integrate Human Domain Constraints	Pattern
SE Domain Use Cases	Perform Environmental Engineering	Pattern
SE Domain Use Cases	Preform EMI Engineering	Pattern
SE Domain Use Cases	Analyze Behavior Correctness	Pattern
SE Domain Use Cases	Evaluate System Safety	Pattern
SE Domain Use Cases	Perform System RMA Engineering	Pattern
SE Domain Use Cases	Allocate and Manage SWaP	Pattern
SE Domain Use Cases	Perform System Security Engineering	Selected & Pattern



- Key Findings
  - There are Process Patterns that may be helpful to mine
  - The use cases provide validation of their derived requirements
  - The use cases have to be vetted



- Pyster, A. and D.H. Olwell (eds). 2013. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 1.2. Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed DATE. [www.sebokwiki.org/](http://www.sebokwiki.org/)
- International Standard – ISO/IEC 15288 and IEEE 15288 – 2008, Second Edition 2008-02-01, Systems and software engineering - System life cycle processes
- Pramanik, Sarah. “Security Architecture Approaches.” 2013. Crosstalk November/December



# BACKUP SLIDES



# SE Domain Integration and Information Layers

- Each SE Domain;
  - Contributes domain specific information
  - Has Responsibility for their information content
  - Information content can reference content from other domains
  - Iterates solution with other Domains
  - Has one or more views to information content
  - Defines and manages Requirements
  - Measures impact of their changes in their domain and across other domains
  - Conducts Reviews
  - Produces Deliverables

