Report Documentation Page

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 FEB 2007</td>
<td></td>
<td>00-00-2007 to 00-00-2007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering Safety- and Security-Related Requirements for Software-Intensive Systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
<th>5b. GRANT NUMBER</th>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
<th>5e. TASK NUMBER</th>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carnegie Mellon University, Software Engineering Institute (SEI), Pittsburgh, PA, 15213</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
</tr>
<tr>
<td>unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
</tr>
<tr>
<td>unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE</td>
</tr>
<tr>
<td>unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same as Report (SAR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
Tutorial Goals

Familiarize Members of:

- **Safety and Security Teams with:**
  - Foundations of *Requirements Engineering*
  - *Common Concepts and Techniques from Both Disciplines*

- **Requirements Teams with the Foundations of:**
  - **Safety Engineering**
  - **Security Engineering**

Familiarize Members of all three Disciplines with:

- **Different Types of Safety- and Security-related Requirements**
- **Common Process for Engineering these Requirements**
Contents

Challenges
Common Example
Requirements Engineering Overview
Safety and Security Engineering Overview
Types of Safety- and Security-related Requirements
Common Consistent Collaborative Process
Conclusion
Challenges:
Combining Requirements, Safety, and Security Engineering
Challenges

Requirements Engineering, Safety Engineering, and Security Engineering:

• Different Communities
• Different Disciplines with different Training, Books, Journals, and Conferences
• Different Professions with different Job Titles
• Different fundamental underlying Concepts and Terminologies
• Different Tasks, Techniques, and Tools

Safety and Security Engineering are:

• Typically treated as Specialty Engineering Disciplines
• Performed separately and largely independently of the primary Engineering Workflow (Requirements, Architecture, Design, Implementation, Integration, Testing.
Challenges

Current separate Processes for Requirements, Safety, and Security are Inefficient and Ineffective.

Separation of Requirements Engineering, Safety Engineering, and Security Engineering:

• Causes *poor* Safety- and Security-related Requirements.
  — Goals rather than Requirements
  — Vague, unverifiable, unfeasible, architectural and design constraints
• Inadequate and too late to drive architecture and testing
• Difficult to achieve Certification and Accreditation
Challenges

Poor requirements are a primary cause of more than half of all project failures (defined in terms of):

- Major Cost Overruns
- Major Schedule Overruns
- Major Functionality not delivered
- Cancelled Projects
- Delivered Systems that are never used

Poor Requirements are a major Root Cause of many (or most) Accidents involving Software-Intensive Systems.

Security ‘Requirements’ often mandated:

- Industry Best Practices
- Security Functions
Challenges

How Safe and Secure is Safe and Secure *enough*?

Situation Cries out for Process Improvement:

- Better Consistency between Safety and Security Engineering
  - More consistent Concepts and Terminology
  - Reuse of Techniques
  - Less Unnecessary Overlap and Avoidance of Redundant Work
- Better Collaboration:
  - Between Safety and Security Engineering
  - With Requirements Engineering
- Better Safety- and Security-related Requirements
Three Related Disciplines

Safety Engineering

the engineering discipline within systems engineering concerned with lowering the risk of *unintentional unauthorized* harm to valuable assets to a level that is acceptable to the system’s stakeholders by preventing, detecting, and reacting to such harm, mishaps (i.e., accidents and incidents), hazards, and safety risks

Security Engineering

the engineering discipline within systems engineering concerned with lowering the risk of *intentional unauthorized* harm to valuable assets to a level that is acceptable to the system’s stakeholders by preventing, detecting, and reacting to such harm, misuses (i.e., attacks and incidents), threats, and security risks

Requirements Engineering

the engineering discipline within systems/software engineering concerned with identifying, analyzing, reusing, specifying, managing, verifying, and validating goals and requirements (including safety- and security-related requirements)
Common Example:
An Automated People Mover System
Desired Characteristics

Common Ongoing Example throughout the Tutorial
Should Not Need Special Domain Knowledge
Example System should be:

- Safety-Critical
- Realistic
- SW-Intensive
- Understandable in terms of:
  - Requirements
  - Technology
  - Hazards
Example Overview

Very Large New Zoo
Zoo Automated Taxi System (ZATS)
Example Zoo Habitat Guideway Layout
ZATS Context Diagram

Proposed ZATS:

- Taxis
- Elevated Concrete Guideway
- Taxi Stations
Zoo Automated Taxi System (ZATS)
Example Habitat Layout
ZATS Context Diagram

- **Passengers**: rides the automated taxi system (ZATS)
  - informs and alerts the managers
- **Maintainers**: maintain and monitor the ZATS
  - notifies and alerts the operators
- **Operators**: control and monitor the ZATS
  - notifies and alerts the managers
  - views the status and reports of the ZATS
- **Managers**: views the status and reports via the internet
  - alerts the operators
  - transmits requests for ZATS status and reports to the ZATS

**Zoo Automated Taxi System (ZATS)**

- **Emergency Responders**: requests emergency services from the ZATS
  - view status of the ZATS
- **Emergency Medical Technicians**
- **Fire Fighters**
- **Police**
- **Zoo Nurse**
- **Zoo Security**

- **Bank Card Processing Gateway**: obtains bank card approval to pay for zoo taxi travel cards from the ZATS
- **Zoo Information System**: obtains employee and membership information from the ZATS
Proposed Taxi Architecture

- **Computer Subsystem**
  - Radios
  - Front Window (Emergency Exit)
  - Front Door Panel
  - Display (Information, Location)
  - Back Door Panel
  - Back Window

- **Subsystems**
  - **Electrical Subsystem**
    - Entrance to Stroller/Wheelchair Area
    - Door Motor
    - GLS (Emergency Exit)
  - **Steering Mechanism**
    - PS
    - B
  - **Steered Wheel**
  - **Electric PBS**
    - BS
    - SS
  - **Front Bench Seat**
    - (Electric Batteries)
  - **Back Bench Seat**
    - (Electric Batteries)
Automated Taxis On Elevated Guideways

- Maintenance and Emergency Walkway
- Back of Taxi
- Best View
- Wheels
- Power and Communications Cables
- Guideway
- Support Pillar
- Ground Level
- Habitat with Animals
Proposed Taxi Station Network Diagram

Security Cameras (2)  Microphone  Public Address Speakers  Door Sensors (6)  Door Locks (6)  Door Motors (6)  Taxi Sensors (6)  Entry Door Speakers (2)

Audiovisual Controller  Taxi Station Switches (2)  Door Controllers (2)  Entry Door Travel Card Readers (2)

links with local traffic light controllers  dual fiber-optic network backbone  links with local traffic light controllers  dual fiber-optic network backbone

Fire Alarms  Fire Detection and Suppression System  Travel Card Vending Machines (4)

<<confidentiality>>  <<integrity>>  <<nonrepudiation>>

Taxi Station
Example Collision Hazard
Requirements Engineering: An Overview
Requirements Engineering Topics

Definition of Requirements Engineering

Requirements Engineering:

• Tasks
• Work Products

Importance and Difficulty of Requirements Engineering

Goals vs. Scenarios vs. Requirements

Types of Requirements

Characteristics of Good Requirements
Requirements Engineering

Definition

the engineering discipline within systems/software engineering concerned with identifying, analyzing, reusing, specifying, managing, verifying, and validating goals and requirements (including safety- and security-related requirements)

the cohesive collection of all tasks that are primarily performed to produce the requirements and other related requirements work products for an endeavor

Today, these RE tasks are typically performed in an iterative, incremental, parallel, and time-boxed manner rather than according to the traditional Waterfall development cycle, whereby parallel means with the:

Primary work flow disciplines such as architecting, design, and testing

Specialty engineering disciplines such as safety and security engineering
RE Tasks and Work Products

Business Analysis (i.e., Customer, Competitor, Market, Technology, and User Analysis as well as Stakeholder Identification and Profiling)

Visioning

Requirements Identification (a.k.a., Elicitation)

Requirements Reuse

Requirements Prototyping

Requirements Analysis

Requirements Specification

Requirements Management

Requirements Validation

Scope Management (Management)

Change Control (Configuration Management)

Quality Control (Quality Engineering)
Requirements Engineering Work Products

Business Analyses
Stakeholder Profiles

Vision Statement
  • Goals

Operational Concept Document (OCD)
  • Usage Scenarios

Requirements Repository and published Specifications
  • Requirements

Requirements Prototypes
Domain Model
Glossary
Importance and Difficulty of Requirements Eng.

Poor requirements are a primary cause of more than half of all:

- Project failures (defined in terms of):
  - Major cost overruns
  - Major schedule overruns
  - Major functionality not delivered
  - Cancelled projects
  - Delivered systems that are never used
- Hazards and associated Mishaps (Accidents and Safety Incidents)
- Vulnerabilities
Difficulty of Requirements Engineering

“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.”

Goals

A **goal** is an *informally documented perceived need of a legitimate stakeholder*.

- Goals are typically documented in a vision statement.
- Goals drive the analysis and formal specification of the requirements.
- Examples:
  - The system shall support user activity X.
  - The system shall be efficient.
  - The system shall be easy to use.
  - The system shall be safe to use.
- Goals are typically not verifiable.
- Goals may not be feasible.
Example ZATS Goals

Functional Goals:

• ZATS must rapidly transport patrons between the parking lots and the zoo.
• ZATS must rapidly transport patrons between habitats within the zoo.
• ZATS must allow patrons to take leisurely tours of the habitats.

Data Goal:

• ZATS must record and report appropriate system usage statistics.

Capacity Goal:

• ZATS must include sufficient taxis so that patrons need not wait long for a free taxi.

Usability Goal:

• ZATS must be very easy and intuitive for patrons to use, including those who are not very good with technology.
Usage Scenarios

A usage scenario is a specific functionally cohesive sequence of interactions between user(s), the system, and potentially other actors that provides value to a stakeholder.

Usage scenarios:

- Are instances of use cases.
- Can be either “sunny day” or “rainy day” scenarios.
- Have preconditions, triggers, and postconditions.
- Are typically documented in an Operational Concept Document (OCD).
- Drive the analysis and formal specification of the [primarily functional] requirements.
- Often include potential design information.
- Can be written in either list or paragraph form.
Example ZATS Scenario

Ride Zoo Loop Line To Restaurants for Lunch:

After the family enters a waiting taxi, Mr. Smith looks at the zoo map on its ceiling. A light representing their taxi is glowing at the Tropical Rainforest Habitat outer taxi station. He uses the control panel to select the inner taxi station at the habitat, which is the central taxi station near the restaurants and shops as a destination. He then swipes his zoo taxi debit card, and the display shows the remaining balance of $9.00 on the card. The taxi warns them to set down and thirty seconds later, the station and taxi exit doors close. Their taxi accelerates out of the taxi station and turns to the left onto the Zoo Loop Line.

Shortly after leaving the taxi station, they see a spur the angles off to their left towards a large building containing the taxi control center and maintenance facility. They continue around the outside of the zoo, passing other the Great Cats, the Wolves and Other Dogs, and the Bears habitats. Just before they reach the outer African Savanna taxi station, the guideway makes a sweeping turn to the right and they can see the parking lot on their left. Everyone looks to see if they can see the family van, but the parking lot is too big and they can only see the parking lot taxi station near where it is parked.

Soon, they pass the zoo entrance on their left and turn right to follow the main street to where the main restaurants and shops are. Their taxi passes the inner African Savanna taxi station on their right, circles around the central area, and soon pulls off the Zoo Loop Line to enter the inner Great Apes and Monkeys taxi station. Exiting the taxi when the doors open, they head down the elevator and outside for an early lunch at one of the many restaurants.
Requirements

A (product) requirement is a *mandatory* characteristic (behavior or attribute) of a product (e.g., system, subsystem, software application, or component).

- Requirements are documented in requirements specifications.
- Requirements are driven by goals.
- Example: “At each taxi station while under normal operating conditions, ZATS shall provide a taxi to passengers within an average of 5 minutes of the passengers’ request.”
- Requirements must have certain characteristics (e.g., verifiable and feasible).
Types of Requirements

- Product Requirements
  - Functional Requirements
  - Non-Functional Requirements
  - Data Requirements
  - Interface Requirements
  - Quality Requirements
  - Constraints

- Process Requirements
  - Stakeholder (Business) Requirements
  - Software Requirements
  - System/Subsystem Requirements
  - Main Mission Requirements
  - Hardware Requirements
  - Specialty Engineering Subsystem Requirements
Types of Requirements

- **Product Requirements**
  - Functional Requirements
  - Non-Functional Requirements

- **Derived Requirements**

- **Development Method Requirements**

- **Stakeholder (Business) Requirements**

- **Primary Mission Requirements**

- **Supporting Requirements**
  - Manual Procedure Requirements
  - Derived Requirements

- **System/Subsystem Requirements**
  - Software Requirements
  - Hardware Requirements

- **Safety Function / Subsystem Requirements**
  - Safety Constraints
  - Security Constraints
  - Survivability Constraints

- **Defensibility Requirements**
  - Safety Requirements
  - Security Requirements
  - Survivability Requirements

- **Quality Requirements**

- **Data Requirements**

- **Interface Requirements**

- **Constraints**

- **Derived Requirements**

- **Stakeholder (Business) Requirements**

- **Primary Mission Requirements**

- **Supporting Requirements**
  - Manual Procedure Requirements
  - Derived Requirements

- **System/Subsystem Requirements**
  - Software Requirements
  - Hardware Requirements

- **Safety Function / Subsystem Requirements**
  - Safety Constraints
  - Security Constraints
  - Survivability Constraints

- **Defensibility Requirements**
  - Safety Requirements
  - Security Requirements
  - Survivability Requirements

- **Quality Requirements**

- **Data Requirements**

- **Interface Requirements**

- **Constraints**
## Characteristics of Good Requirements

<table>
<thead>
<tr>
<th>Mandatory</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>Consistent</td>
</tr>
<tr>
<td>Cohesive</td>
<td>Usable by Stakeholders</td>
</tr>
<tr>
<td>Feasible</td>
<td>Uniquely Identified</td>
</tr>
<tr>
<td>Relevant</td>
<td>Traced</td>
</tr>
<tr>
<td>Unique</td>
<td>Externally Observable</td>
</tr>
<tr>
<td>Unambiguous</td>
<td>Stakeholder-Centric</td>
</tr>
<tr>
<td>Validatable</td>
<td>Properly Specified</td>
</tr>
<tr>
<td>Verifiable</td>
<td>Prioritized</td>
</tr>
<tr>
<td>What or How Well, not How</td>
<td>Scheduled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Managed</th>
<th>Controlled</th>
</tr>
</thead>
</table>

http://www.jot.fm/issues/issue_2003_07/column7
Safety and Security Engineering:
An Overview
Similar Definitions

Safety Engineering

the engineering discipline within systems engineering concerned with lowering the risk of *unintentional unauthorized* harm to valuable assets to a level that is acceptable to the system’s stakeholders by preventing, detecting, and reacting to such harm, mishaps (i.e., accidents and incidents), hazards, and safety risks

Security Engineering

the engineering discipline within systems engineering concerned with lowering the risk of *intentional unauthorized* harm to valuable assets to a level that is acceptable to the system’s stakeholders by preventing, detecting, and reacting to such harm, misuses (i.e., attacks and incidents), threats, and security risks
Fundamental Concepts:
A Foundation for Understanding
Fundamental Concepts

Quality Model
Safety and Security as a Quality Factors with associated Quality Subfactors
Systems responsible for Valuable Assets
Stakeholders
Accidental and Malicious Harm to Valuable Assets
Defensibility Occurrences (Accidents, Attacks, and Incidents)
Agents (External and Internal, Malicious and Non-malicious)
Vulnerabilities (system-internal sources of dangers)
Dangers (Hazards and Threats)
Defensibility Risks (Safety and Security)
Goals, Policies, and Requirements
Defenses (Safeguards and Counter Measures)
Quality Model

defines the meaning of quality for the Quality Model

1..* 1..* 1..*

Quality Factor defines a type of the quality of the

0..* 1..*

Quality Subfactor is measured using a

Quality Measure (Measurement Scale) defines a part of a type of the quality of the

1..* 1..* 1..*

System
Quality Factors

Quality Model

is measured using a

Quality Measure (Measurement Scale)

Quality Factor

Quality Subfactor

Development-Oriented Quality Factor

Usage-Oriented Quality Factor

Capacity

Configurability

Dependability

Efficiency

Interoperability

Performance

Utility

Defensibility

Robustness

Security

Safety

Survivability

Soundness

Correctness

Predictability

Operational Availability

Reliability

Stability
Safety as a Quality Factor

Safety is the Quality Factor capturing the Degree to which:

- Accidental Harm to Valuable Assets is eliminated or mitigated
- Safety Occurrences and Events (Accidents, Safety Incidents, and Hazardous Events) are eliminated or their negative consequence mitigated
- Hazards (i.e., Hazardous Conditions) are eliminated or mitigated:
  - System Vulnerabilities
  - Non-malicious Agents (humans, systems, and the environment)
- Safety Risks are kept acceptably low
- The preceding Problems are Prevented, Detected, Reacted to, and possibly Adapted to
Security as a Quality Factor

Security is the Quality Factor capturing the Degree to which:

- *Malicious Harm* to Valuable Assets is eliminated or mitigated
- *Security Occurrences and Events (Attacks, Security Incidents, and Threatening Events)* are eliminated or their negative consequence mitigated
- *Threats* (i.e., Threatening Conditions) are eliminated or mitigated:
  - System Vulnerabilities
  - Malicious Agents (humans, systems, and malware)
- *Security Risks* are kept acceptably low
- The preceding Problems are *Prevented, Detected, Reacted to*, and possibly *Adapted to*
Defensibility Quality Subfactors

- Occurrence of Unauthorized Harm
- Occurrence of Defensibility Event
- Existence of External Agent
- Existence of Internal Vulnerability
- Existence of Danger
- Existence of Defensibility Risk

Defensibility Problem Type

- Safety
- Security

Defensibility Subfactor

- Prevention
- Detection
- Reaction
- Adaptation

Quality Subfactor

is measured using a

Quality Measure (Measurement Scale)
Valuable Assets

Stakeholders have an interest in the System, which must be defended to protect Valuable Assets. Unauthorized Harm may occur to Tangible Property (Private Property), Intangible Property (Public Property), and Commercial Property. Stakeholders also value People (Human Beings, Roles Played, Organizations) and Property (Tangible, Intangible) within their Environment and Services.
Some ZATS Valuable Assets

People:
- Passengers
- Operators
- Maintainers

Property:
- Animals
- Passenger Bank Card Information
- Taxis
- Taxi Stations

Environment:
- Habitat

Services:
- Taxi Service
Types of Harm

- **Unintentional (Accidental) Harm**
  - Authorized Harm
  - Unauthorized Harm

- **Attacker-Caused (Malicious) Harm**
  - Direct Harm
  - Indirect Harm

- **Valuable Assets** may occur to Harm

- **Harm to People**
  - Death
  - Injury
  - Illness
  - Kidnap
  - Corruption (bribery or extortion)
  - Hardship

- **Harm to Property**
  - Destruction
  - Damage
  - Corruption
  - Theft
  - Unauthorized Access
  - Unauthorized Disclosure

- **Harm to the Environment**
  - Destruction
  - Damage
  - Corruption
  - Unauthorized Usage (Theft)
  - Accidental Loss of Service

- **Harm to a Service**
  - Repudiation of Transaction
  - Denial of Service (DOS)
Stakeholders

- Person
- Person Role
- Organization
- Organization Role

Stakeholder

has legitimate interest in the values

is responsible for an

System

Asset
Some ZATS Stakeholders

People:

• Emergency Responders
• Passengers
• Operators
• Maintainers
• ZATS Developers
• Zoo Employees
• Zoo Management

Organizations:

• Bank Card Processing Gateway
• Safety and Security Certification/Accreditation Bodies
• Zoo Regulatory Bodies
Accidents and Attacks

- Agents typically cause
- Vulnerabilities may cause
- Dangers may enable the occurrence of
- Defensibility Risks can be estimated using the probability of

Defensibility Occurrences

- Stakeholders have an interest in the
- Stakeholder Needs value
- System must meet
- System must defend
- Unauthorized Harm may occur to
- Defensibility Quality Factors define types of ‘quality’ of the

Software Engineering Institute | Carnegie Mellon

Engineering Safety- & Security-Related Requirements ICCBSS Tutorial
Donald Firesmith, 27 February 2007
© 2007 Carnegie Mellon University
Types of Defense Occurrences

- Defensibility Occurrences
  - Mishaps
    - Accidents
    - Safety Incidents
    - Successful Attacks
    - Unsuccessful Attacks
    - Probes
  - Misuses
    - Military Attacks
    - Survivability Incidents
  - Survivability Incidents

- Defensibility Events

- Causes:
  - Unauthorized Harm

© 2007 Carnegie Mellon University
Example ZATS Defensibility Occurrences

Accidents:
- Natural Disasters
- Taxi Accidents
- Taxi Station Accidents

Safety Incidents:
- Inadequate Headway
- Overspeed

Attacks:
- Arson
- Cyber-attacks

Security Incidents:
- Antivirus Software Works
Agents

- System Maintainer
- User
- Non-malicious Human Agent
- Non-malicious External System
- Aspect of the Natural Environment
- Attacker
- Malware
- Non-malicious Agent (Safety)
- Malicious Agent (Security)

Defensibility
- Occurrence
- Event
- Occurrence
- Event

- System-External Condition
- System-Internal Condition
- Vulnerability
- Danger
- Conditional
- Hazard (Safety)
- Threat (Security)

Defensibility
- Accident (Safety)
- Safety Incident
- Attack (Security)
- Security Incident

System Developer
- System Operator

Disgruntled Employee
- Professional Criminal
- Industrial Spy
- Foreign Government
- Terrorist
- Cracker

Backdoor
- Spyware
- Trojan
- Worm
- Virus

May include existence of a Hazard (Safety) or a Threat (Security)

May result in or exploits

May be defined in terms of the existence of system-external conditions

May be the ultimate cause of an accident or an attack

May be created and used by a malicious agent
Example ZATS Agents

Non-Malicious Agents:

• Human Agents (e.g., Developer, Maintainer, Operator, Passenger)
• External Systems (e.g., Communications Network, Electrical Power Grid)
• Natural Environment (e.g., River or Weather)

Malicious Agents:

• Attackers (e.g., Arsonists, Crackers, Terrorists, Thieves)
• Malware (e.g., virus, Trojan horse, Worm)
Vulnerabilities

Dangers

\[ \text{Defenses} \]

- eliminate or mitigate

\[ \text{Vulnerabilities} \]

- may cause

\[ \text{Defensibility Occurrences} \]

\[ \text{Agents} \]

- typically cause

\[ \text{Nonmalicious Agents} \]

\[ \text{Malicious Agents} \]

- exploit

\[ \text{Stakeholders} \]

- have an interest in the

\[ \text{System} \]

- have

\[ \text{Stakeholder Needs} \]

- must meet

\[ \text{Valuable Assets} \]

- value

\[ \text{Quality Factors} \]

- define types of ‘quality’ of the

\[ \text{Uncontrolled Harm} \]

- may occur to
Dangers and Related Concepts

- Dangers
  - are partially defined in terms of vulnerable
  - may cause
    - Dangers
      - are partially defined in terms of the existence of system-external
      - may enable the occurrence of
        - Vulnerabilities
          - may cause
            - Defensibility Risks
              - can be estimated using the probability of
                - Dangers
                  - is the expected amount of
                    - are partially defined in terms of the existence of system-internal
                      - may enable the occurrence of
                        - Agents
                          - typically cause
                            - Nonmalicious Agents
                            - Malicious Agents
                              - exploit
                                - desire
                                  - define types of 'quality' of the
                                    - unauthorized harm
                                      - may occur to
                                        - Valuable Assets
                                          - value
                                            - Stakeholder Needs
                                              - must meet
                                                - System
                                                  - must defend
                                                    - Stakeholders
                                                      - have an interest in the
                                                        - have
                                                          - exist in the
                                                            - have
                                                              - system
                                                                - Stakeholder Needs
                                                                  - value
                                                                    - Valuable Assets
                                                                      - Software Engineering Institute

© 2007 Carnegie Mellon University
Mishaps and Misuses vs. Hazards and Threats

- Nonmalicious Agents cause Mishaps
  - Mishaps
    - Accidents
      - Harm Events
    - Safety Incidents
      - Safety Events
- Malicious Agents cause Misuses
  - Misuses
    - Security Incidents
      - Security Events
- Unauthorized Harm may occur to Valuable Assets
- Unsuccessful Attacks cause Probes
- Hazards and Threats may cause the occurrence of Dangers
- Hazards and Threats cause the occurrence of Harm
- Malicious Agents cause Successful Attacks
- Successful Attacks cause Vulnerabilities

- Defensibility Events exploit Vulnerabilities
- Successful Attacks cause Harm Events
- Attack Triggers cause Threatening Events
- Threatening Events cause Security Incidents
- Security Incidents cause Attacks
- Attacks cause Successful Attacks
Defensibility Risks

- **Dangers**
  - May result in **Danger Likelihood**
    - Hazard Likelihood
    - Threat Likelihood
  - Networks of Dangerous Events
    - May cause Unauthorized Harm
      - May occur to Valuable Assets

- **Defensibility Risks**
  - Are due to
    - Harm Likelihood
    - Can be estimated in terms of
      - Hazard Likelihood
      - Threat Likelihood
    - Harm Event Conditional Likelihood
      - Can be estimated in terms of
        - Accident Likelihood
        - Successful Attack Likelihood
    - Harm Severity
      - Is the likelihood of the occurrence of
      - Software Control
        - Is software’s control over occurrence of
        - Defensibility Risks
          - Are estimated in terms of
            - Harm Likelihood
            - Can be estimated in terms of
              - Damage
              - Likely hood

Types of Risks

- Safety Risks
- Security Risks
- Survivability Risks

Defensibility Risks

Risks of Unauthorized Harm
- Expected Amount of Specific, Specific Types of, or All Unauthorized Harm (Regardless of Asset, Accident, Attack, Hazard, or Threat)

Risks to Valuable Assets
- Expected Amount of Specific, Specific Types of, or All Unauthorized Harm to Specific or Specific Types of Valuable Assets

Risks due to Accidents or Attacks
- Expected Amount of Specific, Specific Types of, or All Unauthorized Harm due to Specific or Specific Types of Accidents or Attacks

Risks due to Hazards or Threats
- Expected Amount of Specific, Specific Types of, or All Unauthorized Harm due to Specific or Specific Types of Hazards or Threats
Safety and Security Goals and Policies

**Value Assets**

**Stakeholders** must have an interest in the value of the **Stakeholder Needs** that must be met.

**System** must be defended to ensure that the **Stakeholder Needs** are met.

**Vulnerable Assets** may cause **Vulnerabilities**

**Dangers** may enable the occurrence of **Defensibility Occurrences**

**Agents** typically cause **Defensibility Policies**

**Nonmalicious Agents** may exploit **Defensibility Goals**

**Malicious Agents** desire to achieve the state desired end results regarding the **Quality Factors**

**Defensibility** mandates minimum amounts of state rules to achieve.

**Defensibility Requirements** mandate minimum amounts of state desired end results regarding the **Quality Requirements**.

**Quality Requirements** are partially defined in terms of the existence of **Quality Factors**.

**Defensibility Goals** are partially defined in terms of system-external

**Defensibility Policies** are partially defined in terms of system-internal

**Quality** factors mandate minimum amounts of state desired end results regarding the **Survivability**

**Survivability** mandates minimum amounts of state rules to achieve

**Security** mandates minimum amounts of state rules to achieve

**Safety** mandates minimum amounts of state rules to achieve

**Defensibility** mandates minimum amounts of state rules to achieve

**Defensibility Requirements** mandate minimum amounts of state rules to achieve

**System** may cause **Unauthorized Harm**

**Defensibility Occurrences** may cause **Defensibility Policies**

**Defensibility Policies** may cause **Quality Requirements**

**Quality Requirements** may cause **Defensibility Policies**
Types of Defensibility Goals

- Stakeholder Goals
- Quality Goals
- Safety Goals
- Security Goals
- Survivability Goals
- Prevention Goals
- Detection Goals
- Reaction Goals
- Adaptation Goals
- Goals involving Harm to Valuable Assets
- Goals involving Accidents, Attacks, and Incidents
- Goals involving Agents
- Goals involving Vulnerabilities
- Goals involving Dangers (Hazards and Threats)
- Goals involving Safety and Security Risks
Safety- and Security-Related Requirements
Safety- and Security-Related Requirements
Types of Safety- and Security-Related Requirements

Too often only a Single Type of Requirements is considered.

Not just:

- Special Non-Functional Requirements (NFRs):
  - Safety and Security Requirements are Quality Requirements are NFRs
- Safety- and Security-Significant Functional, Data, and Interface Requirements
- Constraints on Functional Requirements
- Architecture and Design Constraints
- Safety and Security Functions/Subsystems
- Software Requirements

Reason for Presentation Title

Safety- and Security-Related Requirements for Software-Intensive Systems
Types of Safety- and Security-Related Requirements

- **Product Requirements**
  - Functional Requirements
  - Non-Functional Requirements
  - Quality Requirements
  - Defensibility Requirements
  - Safety Requirements
  - Security Requirements
  - Survivability Requirements

- **Derived Requirements**
  - Stakeholder (Business) Requirements

- **Development Method Requirements**

- **System/Subsystem Requirements**
  - Software Requirements
  - Hardware Requirements
  - Manual Procedure Requirements
  - Supporting Requirements
  - Safety Function/Subsystem Requirements
  - Security Function/Subsystem Requirements
  - Survivability Constraints

- **Defensibility Constraints**

- **Stakeholder (Business) Requirements**

- **Software Requirements**

- **Hardware Requirements**

- **Manual Procedure Requirements**

- **Supporting Requirements**

- **Derived Requirements**

- **Quality Requirements**

- **Data Requirements**

- **Interface Requirements**

- **Constraints**

- **System/Subsystem Requirements**

- **Defensibility Constraints**

- **Safety Function/Subsystem Requirements**

- **Security Function/Subsystem Requirements**

- **Survivability Constraints**

- **Defensibility Constraints**

- **Safety Constraints**

- **Security Constraints**

- **Survivability Constraints**
Types of Defensibility-Related Requirements

- Safety Requirements
- Security Requirements
  - Defensibility Requirements
    - Safety-Significant Requirements
    - Security-Significant Requirements
  - Defensibility Function/Subsystem Requirements
  - Safety Constraints
  - Security Constraints
    - Defensibility Constraints
  - System Requirements
    - Defensibility-Related Requirements
    - Safety-Related Requirements
    - Security-Related Requirements
Types of Safety-Related Requirements

- Safety-Intolerable Requirements (SIL = 5)
- Safety-Critical Requirements (SIL = 4)
- Safety-Major Requirements (SIL = 3)
- Safety-Moderate Requirements (SIL = 2)
- Safety-Minor Requirements (SIL = 1)

Safety Integrity Level (SIL)

- Safety-Independent Requirements (SIL = 0)
- Safety-Significant Requirements (SIL = 1 - 5)

- Functional Requirements
- Data Requirements
- Interface Requirements
- Quality Requirements

- Asset / Harm Requirements
- Safety Incident Requirements
- Hazard Requirements
- Safety Risk Requirements

- Protect Valuable Assets Requirements
- Detect Safety Incidents Requirements
- React to Safety Incidents Requirements

- Non-Safety Quality Requirements
- Safety Requirements

- Safety Constraints

- Main Mission Requirements
- Safety System Requirements
Safety and Security Requirements

Safety and Security Requirements are Quality Requirements.

Quality Requirements are Product Requirements that specify a mandatory amount of a type of product quality (i.e., quality factor or quality subfactor).

Quality Requirements should be:

- Scalar (How Well or How Much)
- Based on a Quality Model
- Specified in Requirements Specifications
- Critically Important Drivers of the Architecture
Components of a Quality Requirement

- **Quality Requirement**: Specifies a minimum level of quality of the system.
- **Condition**: Describes aspect of quality of system.
- **Mandatory System-Specific Quality Criterion**: Provides evidence of existence of quality factor.
- **Measurement Threshold**: Is measured against.
- **Quality Subfactor**: Is measured using a quality measure (measurement scale).
- **Quality Factor**: Defines the meaning of quality for the system.
- **Quality Model**: Specifies a minimum level of quality of the system.
Safety- and Security-Significant Requirements

Are identified based on Safety or Security (e.g., hazard or threat) Analysis

Subset of non-Safety and non-Security Requirements:

- Functional Requirements
- Data Requirements
- Interface Requirements
- Other Quality Requirements
- Constraints

Safety/Security Integrity Level (SIL) is not 0:

- May have minor Safety/Security Ramifications
- May be Safety- or Security-Critical
- May have intolerable Safety or Security Risk
SILs and SEALs

Safety/Security Integrity Level (SIL)

a category of required safety or security for safety- or security-significant requirements.

Safety/Security Evidence Assurance Level (SEAL)

a category of required evidence needed to assure stakeholders (e.g., safety or security certifiers) that the system is sufficiently safe or security (i.e., that it has achieved its required SIL).

SILs are for requirements

SEALs are for components that collaborate to fulfill requirements (e.g., architecture, design, coding, testing)
Safety and Security Function/Subsystem Rqmts.

**Defensibility Function/Subsystem Requirements** are requirements for functions or subsystems that exist strictly to improve defensibility (as opposed to support the primary mission requirements).

- **Safety Function or Subsystem Requirements** are requirements for safety functions or subsystems.
- **Security Function or Subsystem Requirements** are requirements for security functions or subsystems.
Safety Function/Subsystem Requirements

Functions or subsystems strictly added for safety:

- Aircraft Safety Subsystems:
  - Collision Avoidance System
  - Engine Fire Detection and Suppression
  - Ground Proximity Warning System (GPWS)
  - Minimum Safe Altitude Warning (MSAW)
  - Wind Shear Alert
- Nuclear Power Plant:
  - Emergency Core Coolant System

All requirements for such functions/subsystems are safety-related.
Example Safety Function/Subsystem Requirements

“Except when the weapons bay doors are open or have been open within the previous 30 seconds, the weapons bay cooling subsystem shall maintain the temperature of the weapons bay below X° C.”

“The Fire Detection and Suppression Subsystem (FDSS) shall detect smoke above X ppm in the weapons bay within 2 seconds at least 99.9% of the time.”

“The FDSS shall detect temperatures above X° C in the weapons bay within 2 seconds at least 99% of the time.”

“Upon detection of smoke or excess temperature, the FDSS shall begin fire suppression within 1 second at least 99.9% of the time.”
Security Function/Subsystem Requirements

Functions or subsystems strictly added for security:

- Access Control Function
- Encryption/Decryption Subsystem
- Firewalls
- Intrusion Detection System
- Virus Protection Application

All requirements for such functions/subsystems are security-related.

Look in the Common Criteria for many reusable example security function requirements.
Safety and Security Constraints

A Constraint is any Engineering Decision that has been chosen to be mandated as a Requirement. For example:

- Architecture Constraints
- Design Constraints
- Implementation Constraints (e.g., coding standards or safe language subset)
- Testing Constraints

A safety constraint is any constraint primarily intended to ensure a minimum level of safety (e.g., a mandated safeguard).

Safety and Security Standards often mandate Industry Best Practices as Constraints.
Example ZATS Safety Constraints

“When the vehicle is stopped in a station with the doors open for boarding, the horizontal gap between the station platform and the vehicle door threshold shall be no greater than 25 mm (1.0 in.) and the height of the vehicle floor shall be within plus/minus 12 mm (0.5 in.) of the platform height under all normal static load conditions…”

Automated People Mover Standards – Part 2: Vehicles, Propulsion, and Braking (ASCE 21-98)

“Oils and hydraulic fluids shall be flame retardant, except as required for normal lubrication.”

Note need to define flame retardant and normal lubrication.
Common Process:
A Basis for Effective Collaboration
Defensibility & Requirements Engineering

Defensibility-Related Requirements Engineering performs Stakeholder Analysis, Asset Analysis, Vulnerability Analysis, Event Analysis, Agent Analysis, Danger Analysis, Risk Analysis, Significance Analysis, Defense Analysis, System Analysis, and Stakeholders. Defensibility-Work Products includes Requirements Identification, Requirements Analysis, Requirements Validation, and Defensibility-Related Requirements. The Safety Team collaborates with the Security Team, and both teams are involved in the process. Stakeholders include Safety Team and Security Team, with Subject Matter Experts also participating.
Systems Analysis

Safety Team collaborates with Security Team.

Safety and Security Engineering:
- Vision Statement
- Context Diagram
- Goals
- ConOps
- Scenarios
- Use Cases
- Requirements Models
- Requirements Specifications
- Requirements
- Architecture Model
- Architecture Documentation

Requirements Engineering:

System Analysis performs Understand Requirements.

Requirements Team

Architecture Team
Asset Analysis

Safety and Security Engineering

- Stakeholders
- Subject Matter Experts
- Requirements Team
- Safety Team
- Security Team
- Project Documentation (RFP, Contract, ConOps)
- Generic / Reusable Asset Tables
- Standard / Reusable Asset Value and Harm Severity Categories
- Generic / Reusable Asset Value and Harm Severity Tables
- Standard / Reusable Asset-Harm Goals

Asset Analysis

- prepares
- Stakeholder Analysis
- Asset Use Analysis
- Value Analysis
- Harm Analysis
- Asset Identification
- Asset Table
- Stakeholder Table
- Asset Usage Table
- Asset Value and Harm Table
- Asset-Harm Goals

Requirements Engineering

- performs
- Requirements Identification
- Requirements Analysis
- Requirements Validation

Safety and Security Engineering

- Stakeholders
- Safety Team
- Security Team
- Subject Matter Experts
Defensibility Occurrence Analysis

Occurrence Table
- Asset Table
- Asset Value and Harm Table
- Generic / Reusable Attack Type Lists
- Generic / Reusable Defensibility Occurrence Table
- Standard / Reusable Occurrence Likelihood Categories
- Generic / Reusable Occurrence Goals

Safety Team
- Collaborates with Security Team
- Performs Occurrence Analysis

Security Team

Safety and Security Engineering
- Occurrence Identification
- Abuse Tree Analysis
- Abuse Case Analysis
- Goal Identification
- Defensibility Occurrence Table
- Abuse Trees
- Abuse Cases
- Defensibility Event Goals

Requirements Engineering
- Requirement Identification
- Requirements Analysis
- Requirements Validation
- Abuse Requirements

Stakeholders
- Provide input during Occurrence Analysis

Subject Matter Experts
- Provide input during Occurrence Analysis

Project Documentation (RFP, Contract, ConOps)

Software Engineering Institute | Carnegie Mellon
Example Abuse (Attack) Tree

Industrial Spy
Attacker wants to Steal ZATS Intellectual Property
Attacker performs Industrial Espionage

Nation State Spy

Current or Former Employee

Attack Goal

Legend
- Instantiation
- Inheritance (and)
- Aggregation (or)
- Association

Insider Recruitment

Dumpster Diving

Internet Access

Plant Spy

Physical Access

Social Engineering

Control Facility Access

Maintenance Facility Access

Taxi Access

Zoo Administration Office Access

Blackmail

Bribery

Extortion

Janitor Recruitment

Maintainer Recruitment

Operator Recruitment

Domain Name Determination

ZATS Firewall Exploit

Web Server Exploit

Control Server Exploit

Break-in

Impersonation

PMI Employee

Safety / Security Accradiator

Safety Inspector

Inheritance (and)

Aggregation (or)

Association
Example Abuse (Mishap and Attack) Tree
Example Abuse (Mishap and Misuse) Cases

- **Storm**
  - **Snow Storm**
  - **Ice Storm**
  - **Tornado**
  - **Hurricane**

- **Thunder Storm**
  - **Strike With Lightning**
  - **Damage Taxis**

- **Rain on Guideways**
  - **Coat Guideways With Ice**
  - **Make Guideways Unsafe**

- **Snow on Guideways**
  - **Drop Debris On Guideways**
  - **Blow Down Control Facility**
  - ** Blow Down Maintenance Facility**
  - **Blow Down Taxi Station**

- **Start Fire**
  - **Blow Down Building**
  - **Blow Down Building**
Vulnerability Analysis

- Safety Team collaborates with Security Team
- Safety and Security Engineering
- Requirements Team performs
- Vulnerability Analysis
- System Vulnerability Analysis
- Organization Vulnerability Analysis
- Defensibility Compliance Repository
- Requirements Identification
- Requirements Analysis
- Requirements Validation
- Vulnerability Requirements
- Vulnerability Constraints
- Quality Engineers, Testers, and Maintainers
- Safety Team
- Security Team
- Architects, Designers, and Implementers
- Analysts
- Asset Value and Harm Table
- Actual / Proposed System Architecture
- Actual / Proposed System Design
- Actual / Proposed System Implementation
- ISO 15026
- ISO 27001
- NIST 800-37
- NPCIC
- Cybersecurity Framework
- CMMI
- Process Improvement
- Policies
- Asset Management
Agent Analysis

Agent Analysis

Safety Team

Security Team

collaborates with

Agent Analysis

performs

Agent Identification

Potential Agent List

Agent Profiling

Agent Profiles

Agent Occurrence Analysis

Agent Occurrence Table

Agent Goal Development

Agent-Related Goals

Support Requirements Engineering

Requirements Identification

Requirements Analysis

Requirements Validation

Stakeholders

Subject Matter Experts

Safety Team

Security Team

Stakeholders

Subject Matter Experts

Safety Team

Security Team

Safety and Security Engineering

Safety and Security Engineering

Project Documentation (RFP, Contract, ConOps)

Generic / Reusable Agent Lists

Generic / Reusable Agent Profiles

Generic / Reusable Agent-Related Goals

Standard / Reusable Agent-Related Requirements

Agent-Related Requirements Engineering
Danger Analysis

- Safety Team
  - Subject Matter Experts
  - Stakeholders
- Security Team
  - System Safety and Security Documentation
  - Other System Documentation
  - Non-System Documentation
  - Generic / Reusable Danger Lists
  - Generic / Reusable Danger Profiles
  - Generic / Reusable Danger Likelihoods

Danger Analysis
- performs
- collaborates with

Safety and Security Engineering
- Danger Identification
- Danger Profiling
- Danger Cause Analysis
- Cause Analysis
- Root Cause Analysis
- Common Cause Analysis
- Danger Effects Analysis
- Danger Likelihood Analysis
- Danger (Hazard & Threat) Profiles
- Danger (Hazard & Threat) Cause and Effects Diagrams
- Defensibility Compliance Repository

Requirements Engineering
- Requirements Identification
- Requirements Analysis
- Requirements Validation
- Hazard Requirements
- Threat Requirements
- Generic / Reusable Hazard and Threat Requirements

Requirements Team
- Subject Matter Experts
- Stakeholders
- Security Team
- Safety Team

Defensibility Compliance Repository

Software Engineering Institute | Carnegie Mellon

Engineering Safety- & Security-Related Requirements ICCBSS Tutorial
Donald Firesmith, 27 February 2007
© 2007 Carnegie Mellon University
Example Fault Tree

- Passenger falls out of open door of moving taxi
  - Passenger inattentive and near taxi door
    - Door opens on moving taxi
      - Taxi door is unlocked
        - Taxi door lock sensor fails closed
        - Taxi door lock sensor fails locked
      - Door motor opens taxi door
        - Taxi computer fails
          - Taxi computer motor fails open
  - Train starts moving with open door
    - Taxi door fails to close
      - Taxi computer fails
        - Taxi motor fails on
          - Taxi computer fails

- Taxi starts to move
  - Taxi computer fails
    - Taxi computer motor fails on
      - Taxi computer fails

Example Cause and Effect Tree

Legend
- Event
- Hazardous Event
- Accident
- State
- Hazard
- Harm

Taxi(s) crash off of guideway into habitat
- Animal(s) are harmed
- Pedestrian(s) are harmed
- Patrons' vehicles are harmed

Two taxis approach point where their individual guideways merge
- Taxi 2 location sensor has failed
- Taxi 2 power fails on
- Taxi 2 brakes fail off
- Right-of-way not requested
- Lack of right-of-way ignored

Taxi (2) without right-of-way fails to yield
- Two taxis moving too fast to stop

Taxi 1 with right-of-way fails to yield
- Taxi 1 transmitter has failed
- Taxi 1 computer has failed
- Failure to yield not observed
- No warning sent
- No warning received
- Warning ignored

Guideway is damaged
- Taxi(s) are harmed
- Taxi(s) crash off guideway into parking lot
- Taxi(s) are destroyed
- Taxi(s) are damaged
Defense Analysis

Subject Matter Experts

Stakeholders

Safety and Security Requirements

Generic / Reusable Safeguard and Countermeasure Lists

Standard Defense Functionality and Constraint Requirements

Safety and Security Assurance Level (SAL) Allocations

Defense Analysis

Defends

Collaborates with

Performs

Defence Type Identification

Defence Function Identification

Defence Selection

Defence Adequacy Analysis

Requirements Identification

List of Defense Functions / Subsystems

Vendor Trade Studies

Countermeasure and Safeguard Selection Reports

Requirements Validation

Requirements Engineering

Requirements Team

Defence Functionality Requirements

Defence Constraints

Architecture Team

Collaborate in the performance of

Architecting

Requirements Team

Stakeholders

Subject Matter Experts

Safety Team

Security Team
Conclusion:

Process Improvement Recommendations
Conclusion

Engineering safety-significant requirements requires *appropriate*:

- Concepts
- Methods
- Techniques
- Tools
- Expertise

These must come from *both*:

- Requirements Engineering
- Safety Engineering
Conclusion

There are four types of Safety- and Security-related Requirements:

- Safety and Security Quality Requirements
- Safety- and Security-Significant Requirements
- Safety and Security Function/Subsystem Requirements
- Safety and Security Constraints

Different Types of Safety- and Security-related Requirements have different Structures.

These different Types of Requirements need to be identified, analyzed, and specified differently.
Conclusion

Processes for Requirements Engineering, Safety Engineering, and Security Engineering need to be:

- Properly interwoven.
- Consistent with each other.
- Performed collaboratively and in parallel (i.e., overlapping in time).
Process Improvement Recommendations


Better Integrate Safety and Security Processes:

- Concepts and Terminology
- Techniques and Work Products
- Provide Cross Training

Better Integrate Safety and Security Processes with Requirements Process:

- Early during Development Cycle
- Clearly define Team Responsibilities
- Provide Cross Training

Develop all types of Safety- and Security-related Requirements.

Ensure that these Requirements have the proper Properties.