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

- **Passengers**: rides the Zoo Automated Taxi System (ZATS) and informs and alerts the Operators.
- **Maintainers**: maintain and monitor the ZATS and notifies and alerts the Managers.
- **Operators**: view ZATS status and reports via the Internet, notifies and alerts the Managers, control and monitor the ZATS, and views the status and reports of the ZATS.
- **Managers**: views ZATS status and reports and alerts the Operators.
- **Internet**: transmits requests for ZATS status and reports to the Operators.

Zoo Nurse requests emergency services from the Emergency Responders.

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

Zoo Nurse obtains bank card approval to pay for zoo taxi travel cards from the Bank Card Processing Gateway and obtains employee and membership information from the Zoo Information System.

- **Bank Card Processing Gateway**: obtains bank card approval to pay for zoo taxi travel cards.
- **Zoo Information System**: obtains employee and membership information.

This diagram illustrates the interactions between different roles involved in the operation and management of the Zoo Automated Taxi System (ZATS).
Proposed Taxi Architecture

- Computer Subsystem
- Radios
- Front Window (Emergency Exit)
- Front Door Panel
- Display (Information, Location)
- Back Door Panel
- Back Window
- Front Bench Seat (Electric Batteries)
- Place for strollers or wheelchair
- Back Bench Seat (Electric Batteries)
- Steering Mechanism
- Electrical Subsystem
- Door Motor
- Steered Wheel
- Electric PBS
- Drive Wheel
- Back Door Panel
- B
Automated Taxis On Elevated Guideways

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

Taxi Station

- 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

Door Controllers (2)

Taxi Station Switches (2)

Travel Card Vending Machines (4)

Fire Detection and Suppression System

Smoke Detectors

Fire Alarms

links with local traffic light controllers

dual fiber-optic network backbone

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

- **Hardware Requirements**

- **Main Mission Requirements**

- **Specialty Engineering Subsystem Requirements**
Types of Requirements

- Product Requirements
  - Functional Requirements
  - Non-Functional Requirements
    - Quality Requirements
      - Safety Requirements
      - Security Requirements
      - Survivability Requirements
    - Data Requirements
    - Interface Requirements
  - Derived Requirements
    - Stakeholder (Business) Requirements
    - Development Method Requirements

- Primary Mission Requirements
  - Defensibility Requirements
    - Safety Constraints
    - Security Constraints
    - Survivability Constraints
  - System/Subsystem Requirements
    - Software Requirements
    - Hardware Requirements
    - Manual Procedure Requirements

- Supporting Requirements
  - Derived Requirements
  - Stakeholder (Business) Requirements
  - Development Method Requirements

- System/Subsystem Requirements
  - Safety Function/Subsystem Requirements
  - Security Function/Subsystem Requirements

- Stakeholder (Business) Requirements
  - Derived Requirements
  - Development Method Requirements

- Defensibility Requirements
  - Safety Constraints
  - Security Constraints
  - Survivability Constraints

- Derived Requirements
  - Stakeholder (Business) Requirements
  - Development Method Requirements

- Development Method Requirements
  - Derived Requirements
  - Stakeholder (Business) Requirements

- Stakeholder (Business) Requirements
  - Derived Requirements
  - Development Method Requirements
### 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>
<tr>
<td></td>
<td>Managed</td>
</tr>
<tr>
<td></td>
<td>Controlled</td>
</tr>
</tbody>
</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 \textit{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 \textit{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

- Quality Model
  - Quality Factor
    - Quality Subfactor
      - Quality Measure
        (Measurement Scale)
  - System
    - Quality Factor
      - defines a type of the quality of the System
    - Quality Subfactor
      - defines a part of a type of the quality of the System
    - Quality Measure
      - measured using a Quality Scale

The Quality Model defines the meaning of quality for the System.
Quality Factors

- Quality Model
  - Quality Factor
    - Development-Oriented Quality Factor
    - Usage-Oriented Quality Factor
      - Performance
      - Utility
      - Efficiency
      - Interoperability
      - Availability
      - Predictability
      - Correctness
      - Soundness
      - Defensibility
      - Robustness
      - Security
      - Safety
      - Survivability
      - 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
- Prevention
- Detection
- Reaction
- Adaptation

Quality Factor → Quality Subfactor → Defensibility Subfactor → Defensibility Solution Type → Quality Measure

Quality Measure is measured using a Quality Model

Defensibility Problem Type

Safety Security
Valuable Assets

Stakeholders → value → System → Valuable Assets

Stakeholders have an interest in the System, which must be defended.

UnAuthorized Harm may occur to Valuable Assets.

Valuable Assets include:

- People
  - Human Beings
  - Roles Played
  - Organizations

- Property
  - Tangible Property
  - Intangible Property

- Environment

- Services
  - Private Property
  - Public Property
  - Commercial Property
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

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

- **Security**
  - Attacker-Caused (Malicious) Harm
  - Unauthorized Usage (Theft)
  - Unauthorized Access
  - Unauthorized Disclosure

- **Survivability**
  - Unauthorized Di sclosure

- **Harm**
  - Direct Harm
  - Indirect Harm
  - Harm to People
    - Death
    - Injury
    - Illness
    - Kidnap
    - Corruption (bribery or extortion)
    - Hardship
  - Harm to Property
    - Destruction
    - Damage
    - Corruption
    - Theft
  - Harm to the Environment
    - Destruction
    - Damage
    - Loss of Use
  - Harm to a Service
    - Corruption
    - Unauthorized Usage (Theft)
    - Accidental Loss of Service
    - Denial of Service (DOS)
    - Repudiation of Transaction

- **Valuable Assets**
  - e.g., caused to enemy forces by weapons systems

- **Survivability**
  - Accidental Loss of Service
  - Repudiation of Transaction

- **Harm**
  - Direct Harm
  - Indirect Harm
Stakeholders

Person -- Person Role -- Organization -- Organization Role

Stakeholder

- has legitimate interest in the values
- is responsible for an Asset
- is responsible for an System
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, which may cause Dangers. Dangers may enable the occurrence of Defensibility Risks, which can be estimated using the probability of Defensibility Occurrences.

Stakeholders have an interest in the System, which must meet Stakeholder Needs and must be defended to protect Valuable Assets. Stakeholders may cause Unauthorized Harm, which may occur to the System, define types of 'quality' of the Valuable Assets.
Types of Defense Occurrences

- **Defensibility Occurrences**
  - **Mishaps**
    - Accidents
      - Successful Attacks
      - Unsuccessful Attacks
    - Safety Incidents
  - Misuses
    - Civilian Attacks
    - Security Incidents
    - Probes
  - Survivability Occurrences
    - Military Attacks
    - Survivability Incidents
- **Defensibility Events**

- **Unintended Harm**

- **Cause**
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
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

- Defenses
  - eliminate or mitigate
- Stakeholders
  - have
  - have an interest in the
  - must meet
  - must defend
- System
  - exist in the
- Stakeholder Needs
  - value
- Valuable Assets
  - may cause
- Dangers
  - are partially defined in terms of the existence of system-internal
  - may cause
- Defensibility Occurrences
  - may cause
- Unauthorized Harm
  - may occur to
- Quality Factors
  - define types of ‘quality’ of the
  - exploit
  - desire
  - typically cause
- Agents
  - Nonmalicious Agents
  - Malicious Agents
  - typically cause
  - exploit
- Defensibility
  - define types of ‘quality’ of the
  - exploit
- Agents
  - Nonmalicious Agents
  - Malicious Agents
  - typically cause
  - exploit
- Dangers
  - are partially defined in terms of the existence of system-internal
  - may cause
- Valuable Assets
  - may cause
Dangers and Related Concepts

Dangers

Defensibility Risks

are partially defined in terms of the existence of system-external

is the expected amount of

can be estimated using the probability of

are partially defined in terms of the existence of system-internal

may enable the occurrence of

Vulnerabilities

may cause

Defensibility Occurrences

Agents

typically cause

Nonmalicious Agents

Malicious Agents

exploit

desire

define types of 'quality' of the

UnAUTHORIZED HARM

may occur to

System

must defend

Stakeholder Needs

must meet

have

Dangers

have an interest in the

exist in the

Stakeholders

have

Vulnerable

may cause

Valuable Assets

value

Stakeholder Needs

must meet

Software Engineering Institute | Carnegie Mellon

Engineering Safety- & Security-Related Requirements ICCBSS Tutorial
Donald Firesmith, 27 February 2007
© 2007 Carnegie Mellon University
Mishaps and Misuses vs. Hazards and Threats

- Mishaps
  - Nonmalicious Agents
  - Accidents
  - Safety Incidents
  - Harm Events
  - Accident Triggers
  - Hazardous Events
  - Safety Incidents
  - Valuable Assets
  - Harm Events
  - Accidents
  - Safety Incidents
  - Valuable Assets
  - Unauthorized Harm
  - Unsuccessful Attacks
  - Security Incidents
  - Threatening Events
  - Security Incidents
  - Threatening Events
  - Security Incidents
  - Security Events

- Misuses
  - Malicious Agents
  - Security Incidents
  - Attacks
  - Successful Attacks
  - Security Incidents
  - Attack Triggers
  - Harm Events
  - Security Events
  - Vulnerabilities
  - Defensibility Events
  - Events
  - Vunerabilities
  - Exploit
  - Unsuccessful Attacks
  - Probes
  - Dangers
  - Hazards
  - Hazards
  - harm Events
  - enable the occurrence of
  - cause the occurrence of
  - cause the occurrence of
  - cause the occurrence of
  - cause the occurrence of
  - cause the occurrence of
  - cause the occurrence of
  - cause the occurrence of
Defensibility Risks

Defensibility Risks can be estimated in terms of Harm Likelihood, Software Control, or Harm Severity.

Harm Likelihood is the likelihood of the occurrence of a danger event.

Danger Likelihood is the conditional likelihood given danger of occurrence of an event.

Hazard Likelihood and Threat Likelihood may result in Networks of Dangerous Events.

Networks of Dangerous Events may cause Unauthorized Harm.

Unauthorized Harm may occur to Valuable Assets.

Is software's control over occurrence of events can be estimated in terms of Successful Attack Likelihood, Accident Likelihood, and Harm Event Conditional Likelihood.
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

- **Safe**
- **Security**
- **Survivability**

- **Defensibility Requirements**
- **Defensibility Policies**

**Stakeholders** have an interest in the system, which must meet the stakeholder needs, and must defend the valuable assets. VULNERABILITIES may cause DEFENSIBILITY OCCURRENCES, which may enable the occurrence of AGENTS. AGENTS may typically cause DAMAGES, and MALICIOUS AGENTS may exploit to define types of 'quality' of the system. NONMALICIOUS AGENTS may cause HARM, unauthorized harm, and DANGERS may cause DEFENSIBILITY GOALS, which are partially defined in terms of system-external state desired end results regarding state, and DEFENSIBILITY REQUIREMENTS mandate minimum amounts of state rules to achieve DEFENSIBILITY. DANGERS, AGENTS, and VULNERABILITIES are partially defined in terms of system-internal state rules to achieve DEFENSIBILITY. Quality Requirements mandate minimum amounts of DEFENSIBILITY.
Types of Defensibility 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
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
Types of Defensibility-Related Requirements

- Safety Requirements
  - Safety Significance
    - Safety Function/Subsystem Requirements
  - Safety Constraints
- Security Requirements
  - Security Significance
    - Security Function/Subsystem Requirements
  - Security Constraints
- Defensibility Requirements
  - Defensibility Significance
    - Defensibility Function/Subsystem Requirements
  - 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)

- **Main Mission Requirements**
  - **System Requirements**
    - **Functional Requirements**
    - **Data Requirements**
    - **Interface Requirements**
    - **Quality Requirements**
    - **System Requirements**
    - **Safety System Requirements**

- **Safety Incident Requirements**
  - **Detect Safety Incidents Requirements**
  - **React to Safety Incidents Requirements**

- **Hazard Requirements**
  - **Safety Risk Requirements**
  - **Safety Constraints**

- **Protect Valuable Assets Requirements**
  - **Non-Safety Quality Requirements**
  - **Safety Requirements**

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

- **Safety Integrity Level (SIL)**
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.
  - Must meet or exceed the measurement threshold.

- **Condition**
  - Describes aspect of quality of the system.
  - Restricts applicability of the system-specific quality criterion.

- **System-Specific Quality Criterion**
  - Provides evidence of existence of quality factor.

- **Quality Factor**
  - Quality subfactor provides evidence of existence of the quality factor.

- **Quality Subfactor**
  - Is measured using a quality measure.

- **Quality Measure**
  - (Measurement scale)

- **Measurement Threshold**
  - Is measured against.

- **Quality Model**
  - Defines the meaning of quality for the system.

- **System**
  - 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)

“For 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

Defensibility Analysis

System Analysis

Stakeholder Analysis

Asset Analysis

Vulnerability Analysis

Event Analysis

Agent Analysis

Danger Analysis

Risk Analysis

Significance Analysis

Defense Analysis

Defensibility-Work Products

Safety and Security Engineering

Requirements Engineering

Requirements Team

Requirements Identification

Requirements Analysis

Requirements Validation

Defensibility-Related Requirements

Stakeholders

Subject Matter Experts

Safety Team

Security Team
Systems Analysis

Safety Team

Security Team

collaborates
with

performs

System Analysis

Safety and Security Engineering

Requirements Engineering

Vision Statement

Context Diagram

Goals

ConOps

Scenarios

Use Cases

Requirements Models

Requirements Specifications

Requirements

Architecture Model

Architecture Documentation

Requirements Team

Safety and Security Team

Understand
Requirements

Understand
Architecture

Software Engineering Institute

Carnegie Mellon

Engineering Safety- & Security-Related Requirements ICCBSS Tutorial
Donald Firesmith, 27 February 2007
© 2007 Carnegie Mellon University
Asset Analysis

Sanity Team collaborates with Security Team

Safety and Security Engineering

Asset Analysis

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

Preparation

- Asset Identification
- Stakeholder Analysis
- Asset Use Analysis
- Value Analysis
- Harm Analysis

Asset Table

- Asset Stakeholder Table
- Asset Usage Table
- Asset Value and Harm Table
- Asset-Harm Goals

Requirements Team

- performs
- Requirements Identification
- Requirements Analysis
- Requirements Validation

Safety Team, Security Team, Stakeholders, Subject Matter Experts

Requirements Engineering

Engineering Safety- & Security-Related Requirements ICCBSS Tutorial
Donald Firesmith, 27 February 2007
© 2007 Carnegie Mellon University
Defensibility Occurrence Analysis

- Subject Matter Experts
- Stakeholders
- Project Documentation (RFP, Contract, ConOps)
- 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

Occurrence Analysis

- Occurrence Identification
- Abuse Tree Analysis
- Abuse Case Analysis
- Goal Identification

Safety and Security Engineering

- Defensibility Occurrence Table
- Abuse Trees
- Abuse Cases
- Event Goals

Requirements Engineering

- Requirements Identification
- Requirements Analysis
- Requirements Validation

Requirements Team performs

Stakeholders
- Subject Matter Experts
- Safety Team
- Security Team

Defensibility Event Goals
Example Abuse (Attack) Tree

Industrial Spy

Attacker

wants to

performs

Steal ZATS Intellectual Property

Industrial Espionage

Attack Goal

Attacks

Legend

Instantiation

Inheritance (and)

Aggregation (or)

Association

Nation State Spy

Current or Former Employee

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

Inheritance (and)

Aggregation (or)

Association
Example Abuse (Mishap and Attack) Tree

Defensibility Occurrence

ZATS Control Facility Attack or Mishap causes Damage to or Destruction of ZATS Control Facility

Attack (A)
Mishap (M)

Fire
Explosion

Building Collapse
Data Center
Explosion
Fire
probably causes

Flooding
Roof

Roof

Fire Suppression (M)
River Flooding (M)
Sprinkler System (M)
Storm Surge (M)
Water Pipe Leak (M)

Power Loss (M)
Malware (A)
Sabotage (A)
Telecommunications Failure (M)
Vandalism (A)

Explosive (A)
Propane (M)

Earthquake (M)
Sinkhole (M)
Tornado (M)

Cooling Loss (M)
Hardware Failure (M)

Arson (A)
Electrical Fire (M)
Furnace Fire (M)
Lightning Fire (M)
Wild Fire (M)

Snow Load (M)
Volcanic Ash (M)
Wind Damage (M)

Hurricane (M)
Thunderstorm (M)
Tornado (M)
Example Abuse (Mishap and Misuse) Cases

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

- **Tornado**
  - **Drop Debris On Guideways**
  - **Blow Down Building**

- **Snow Storm**
  - **Rain on Guideways**
  - **Coat Guideways With Ice**

- **Ice Storm**
  - **Snow on Guideways**

- **Thunder Storm**
  - **Strike With Lightning**

- **Hurricane**
  - **Blow Down Building**
  - **Blow Down Control Facility**
  - **Blow Down Maintenance Facility**
  - **Blow Down Taxi Station**

- **Damage Taxis**
  - **Start Fire**
  - **Make Guideways Unsafe**

- **Taxis**
  - **Make Guideways Unsafe**
Vulnerability Analysis

Vulnerability Analysis

Safety Team collaborates with Security Team

Vulnerability Analysis performs

- Vulnerability Identification
- System Vulnerability Analysis
- Organization Vulnerability Analysis

Vulnerability Table

Defensibility Compliance Repository

Requirements Team performs

- Requirements Identification
- Requirements Analysis
- Requirements Validation

Vulnerability Requirements

Vulnerability Constraints

Architects, Designers, and Implementers

Quality Engineers, Testers, and Maintainers

Safety Team

Security Team

Actual / Proposed System Architecture

Actual / Proposed System Design

Actual / Proposed System Implementation

Asset Value and Harm Table

Safety and Security Engineering

Requirements Engineering

Software Engineering Institute | Carnegie Mellon

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

87
Agent Analysis

Agent Analysis

- Subject Matter Experts
- Stakeholders
- Project Documentation (RFP, Contract, ConOps)
- Generic / Reusable Agent Lists
- Generic / Reusable Agent Profiles
- Generic / Reusable Agent-Related Goals
- Safety Team
- Security Team

Collaborates with

performs

Agent Analysis

Stakeholders provide input during

Requirements Engineering

Safety and Security Engineering

Requirements Team

performs

Agent-Related Requirements

Requirements Identification

Agent-Related Goals

Agent Profiles

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 Analysis

Requirements Validation

Subject Matter Experts

Safety Team

Security Team

Stakeholders
Danger Analysis
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 fails unlocked
  - Taxi computer fails
- Taxi door opens taxi door
  - Taxi door motor fails open
  - Taxi computer fails

Train starts moving with open door

- Warning is ineffective
  - Taxi computer fails
- Taxi door fails to close
  - Taxi door lock sensor fails closed
  - Taxi door lock sensor fails locked
- Taxi starts to move
  - Taxi motor fails on
  - Taxi computer fails
Example Event Tree

Passenger near open door on moving taxi

- Passenger protected
- Passenger may fall out of stopped taxi
- Passenger likely to fall out of stopped taxi
- Passenger may fall out of moving taxi
- Passenger likely to fall out of moving taxi
- Passenger likely to fall out of moving taxi
Example Cause and Effect Tree
Defense Analysis

Diagram showing the process of defense analysis, including interactions between stakeholders, defense teams, and security teams. The process involves:

- Defense Analysis
- Safety and Security Engineering
- Requirements Engineering

Key activities include:
- Stakeholder input
- Defense Type Identification
- Defense Functionality Identification
- Market Research
- Defense Selection
- Defense Adequacy Analysis
- Countermeasure and Safeguard Selection Reports
- Vendor Trade Studies
- Countermeasure and Safeguard Type Lists
- List of Defense Functions / Subsystems

Collaborators and teams include:
- Safety and Security Requirements
- Generic / Reusable Safeguard and Countermeasure Lists
- Standard Defense Functionality and Constraint Requirements
- Safety and Security Assurance Level (SAL) Allocations
- Architecture Team
- Requirements Team
- Stakeholders
- Subject Matter Experts
- Safety Team
- Security Team

The diagram illustrates the flow of information and the collaborative effort in defense analysis and engineering.
Defense Certification and Accreditation

- Safety Team
- Security Team
- Defensibility Certification and Accreditation
  - Performs with Safety Certification Repository
  - Performs with Security Certification Repository
- Safety and Security Engineering
  - Safety Certification
    - Safety Case(s)
    - Safety Certificate
  - Security Certification
    - Security Case(s)
    - Security Certificate
  - Safety Accreditation
    - Safety Authorization
  - Security Accreditation
    - Security Authorization
- Requirements Engineering
  - Requirements Specification
  - Requirements Validation
  - Requirements Team
    - Performs
- Project Management
  - Management Team
  - Performs
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.
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.