Tutorial T3

Engineering Safety-Related Requirements for Software-Intensive Systems

Donald Firesmith, Software Engineering Institute, USA
Topics

❖ Importance of Safety-Related Requirements
❖ Automatic People Mover Example Overview
❖ Basic Safety Concepts
❖ Safety-Related Requirements:
  ● Safety Requirements
  ● Safety-Significant Requirements
  ● Safety System Requirements
  ● Safety Constraints
❖ A Process for Producing Safety-Related Requirements
Importance of Requirements

- Poor requirements cause more than half of all project failures:
  - Major cost overruns
  - Major schedule overruns
  - Major functionality not delivered
  - Cancelled projects
  - Delivered systems that are never used
Difficulty of Requirements

“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.”

Importance of Accidents

- Accidents can have expensive and potentially fatal repercussions:
  - Mars Climate Orbiter ($125 million)
  - Therac–25
  - Bhopal (3–10K deaths, 500K injured)
Poor Requirements Cause Accidents

- Most accidents are caused by poor requirements:
  - “For the 34 (safety) incidents analyzed, 44% had inadequate specification as their primary cause.”

  Health and Safety Executive (HSE), *Out of Control: Why Control Systems Go Wrong and How to Prevent Failure* (2nd Edition), 1995

- “Almost all accidents related to software components in the past 20 years can be traced to flaws in the requirements specifications, such as unhandled cases.”

Poor Requirements

- **Ambiguous Requirements:**
  - Developers misinterpret Subject Matter Experts intentions.
  - The system shall be safe.”
  - How safe? Safe in what way?

- **Incomplete Requirements:**
  - Developers must guess SME intentions.
  - The system shall do X.”
  - In what state? When triggered by what event? How often? How fast? For whom?

- **Missing Requirements:**
  - What shall the system do if it can’t do X?
  - Unusual combinations of conditions result in accidents.
  - What shall the system do if event X occurs when the system is simultaneously in states Y and Z?
More Problems and Challenges

❖ Inappropriate architecture and design constraints unnecessarily specified as requirements
  ● Use ID and password for identification and authentication.

❖ Separation of requirements engineering and safety engineering:
  ● Different disciplines with different training, books, journals, and conferences.
  ● Different professions with different job titles.
  ● Different fundamental underlying concepts and terminologies
Safety Engineering

- **Safety engineering** is the *engineering discipline* within *systems engineering* that lowers the *risk of accidental harm to valuable assets* to an *acceptable level* to *legitimate stakeholders*.

Note:
- Engineering Discipline
- Systems Engineering (not just software)
- Risk
- *Accidental* Harm
- Harm to Valuable Assets
- *Acceptable Level* of Risk
- *Legitimate* Stakeholders
Tutorial Example: Characteristics

- Common Ongoing Example throughout Tutorial
- Safety-Critical SW-Intensive System
- Realistic Example System
- No Special Domain Knowledge Needed

- Understandable:
  - Requirements
  - Technology
  - Hazards
Tutorial Example: Overview

- Very Large New Zoo
- Zoo Automated Taxi System (ZATS)
- Typical Habitat
- Typical Automated Taxi Station
- ZATS Domain Model
- Taxi Object Model
Tutorial Example: Very Large New Zoo
Zoo Automated Taxi System (ZATS)
Typical Habitat
Typical Automated Taxi Station
ZATS Domain Model

- Daily Schedule
- Dispatcher
  - Monitors and controls
  - Dispatches and monitors taxis via
- Virtual Person
  - Requests trips and pay
- Taxi Drivers
  - Drive and monitor
  - Travels along
- Passengers
  - Ride in
  - Enter and exit taxis at
- Taxis
  - Stop at
  - Connect
- Guideways
  - Are in
- Taxi Stations
- Regions
  - Habitats
  - Parking Lots
  - Maintenance Facility
Taxi Object Model
Basic Safety Concepts

- Safety as a Quality Factor of a Quality Model
- Safety Quality Subfactors
- Valuable Assets
- Accidental Harm to Valuable Assets
- Safety Incidents (Accidents & Near Misses)
- Hazards
- Safety Risks
- Goals, Policies, and Requirements
- Safeguards (Safety Mechanisms)
- Vulnerabilities (system-internal sources of dangers)
Quality Model

- **Quality Model** – a hierarchical model (i.e., a collection of related abstractions or simplifications) for formalizing the concept of the quality of a system in terms of its quality factors, quality subfactors, quality criteria, and quality measures.
Quality Factors

Quality Model

Quality Factor

Development-Oriented Quality Factor

Usage-Oriented Quality Factor

Capacity

Configurability

Dependability

Efficiency

Interoperability

Defensibility

Soundness

Safety

Survivability

Correctness

Predictability

Robustness

Security

Continuity

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 prevented, detected, reacted, and adapted
- Accidents (and near misses) are eliminated or their negative consequence mitigated
- Hazards are eliminated or mitigated
- Safety risks are acceptably low
Defensibility Subfactors

- Harm
  - Incident
  - Danger
  - Risk

- Prevention
  - Detection
  - Reaction
  - Adaptation

- Defensibility Problem Type

- Defensibility Solution Type

- Safety

- Defensibility

- Quality Factor

- Quality Subfactor

- System-Specific Quality Criterion

- Quality Measure

- System

- Quality of the System

Describes the quality of the System

Is measured using Quality Measure

Provides evidence for existence of System-Specific Quality Criterion

Provides evidence for existence of Quality Subfactor

Provides evidence for existence of Quality Factor

Defensibility Subfactor

Defensibility

Quality Subfactor

Quality Factor

Safety

Defensibility

Harm

Incident

Danger

Risk

Prevention

Detection

Reaction

Adaptation

Tutorial T3

Engineering Safety-Related Requirements for Software-Intensive Systems
Safety Subfactors

Safety Subfactor:
- Accidental Harm
- Safety Incident
- Hazard
- Safety Risk

System-Specific Safety Criterion:
- Prevention
- Detection
- Reaction
- Adaptation

Safety Problem Type:
- Safety Incident
- Hazard
- Safety Risk

Safety Solution Type:
- Prevention
- Detection
- Reaction
- Adaptation

Safety Measure:
- provides evidence for existence of
- measures
- describes the safety of the System

System-Specific Safety Criterion:
- is measured using
Valuable Assets

- A valuable asset is anything of significant value to a legitimate stakeholder that should be protected from accidental (or malicious) harm by the system.
ZATS Valuable Assets

- What are the Valuable Assets for which ZATS is responsible for protecting against accidental harm?
- How valuable are these assets to the Zoo (and society)?
Accidental Harm

- **Harm** is any significant negative consequence to a valuable asset

- **Accidental harm** is any harm due to an accident
ZATS Harm to Valuable Assets

○ What kinds of accidental harm can occur to the Valuable Assets for which ZATS is responsible?

○ How should these kinds of harm be categorized in terms of harm severity, and how should the categories be defined?
  ● Catastrophic
  ● Critical
  ● Major
  ● Minor
  ● Negligible
Safety Incidents

- An **incident** is an unplanned (but not necessarily unexpected) series of one or more related **events** that either did cause or could have caused (accidental or malicious) harm to one or more valuable assets
  - A **safety incident** is an incident involving actual or potential accidental harm
Incidents and their Relationships

- Near Miss (Close Call)
- Accident
- Successful Attack
- Unsuccessful Attack
- Probe or Scan
- Attack
- DoS
- Man-in-the-Middle
- Virus
- Security Incident
- Attacker
- Unauthorized Harm
- Valuable Asset
ZATS Safety Incidents

- What kinds of safety incidents can occur if not prevented?
  - Accidents
  - Near misses

- What kind of harm can these accidents cause to what valuable assets?

- How likely can these safety incidents be allowed to be?
Safety Hazards

- **Danger** (Defensibility) is one or more conditions, situations, or states of a system that in conjunction with condition(s) in the environment of the system can cause or contribute to the occurrence of an incident:
  - **Hazard** (Safety) is a danger that can cause or contribute to the occurrence of a safety incident.
  - **Threat** (Security and Survivability) is a danger that can cause or contribute to the occurrence of a security or security incident (i.e., a vulnerability combined with an attacker with means, motive, and opportunity).
Dangers and their Relationships

- Safety
- Security
- Survivability

Hazard

Threat

Attacker

involves the existence and profile of

Danger

may result in

Incident

may cause

Unauthorized Harm

may occur to

Valuable Asset

is responsible for protecting or not harming any

relevant

State

Environment

System
ZATS Hazards

lical hazards (hazardous conditions) might exist?

What kinds of safety incidents can these hazards cause?

What kinds of events can cause these safety hazards to exist?

Can the existence of these hazards be detected?
Safety Risks

- **Risk** is the combination of the severity of harm to a valuable asset with the likelihood that the harm will occur.

- **Harm severity** is usually set *conservatively* to the maximum credible category of harm.

- The likelihood of harm is the likelihood of danger multiplied by the likelihood that the danger results in a harm-causing incident (e.g., accident or attack).
Safety Risk Matrix

Safety Risks can be categorized (for example) as:
- Intolerable
- Undesirable
- As Low As Reasonably Practical (ALARP)
- Acceptable

<table>
<thead>
<tr>
<th>Safety Risks / Safety Integrity Levels (SILs)</th>
<th>Frequency of Accident / Hazard Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harm Severity</td>
<td>Frequent</td>
</tr>
<tr>
<td>Catastrophic</td>
<td>Intolerable</td>
</tr>
<tr>
<td>Critical</td>
<td>Intolerable</td>
</tr>
<tr>
<td>Major</td>
<td>Undesirable</td>
</tr>
<tr>
<td>Minor</td>
<td>Undesirable</td>
</tr>
<tr>
<td>Negligible</td>
<td>ALARP</td>
</tr>
</tbody>
</table>
ZATS Safety Risks

- How would you develop a safety matrix for ZATS?
  - How would you categorize and define harm severity?
  - How would you categorize and define likelihood?

- How would you categorize, define, and assign safety risks to the safety risk matrix cells?

- What would be some of the ramifications of your choices?
Safety Goals

- **Safety Goals** are high-level stakeholder desires regarding safety:
  - “The system must be safe.”
  - “There can be no serious accidents.”
  - “The system will never kill or injure its users.”

- Goals are typically ambiguous or unrealistic (i.e. impossible to guarantee).

- Goals are *not* requirements.

- A *major* problem is safety goals that are specified as if they were verifiable requirements.
ZATS Safety Goals

- What do you think some of the safety goals for the ZATS should be?
- Are they realistic and verifiable?
- Do different stakeholders have different safety goals?
Safety Policies

- **Policy** – a strategic decision that establishes a desired goal.

- **Safety policy** – a policy that establishes a safety goal:
  - “The overall responsibility for safety must be identified and communicated to all stakeholders.”
  - “A hazard analysis shall be performed during early in the project.”
  - “All users will have safety training.”

- Tend to be process rather than product oriented.

- Safety policies are collected into safety policy documents.

- In practice, safety policies are confused with requirements and policy documents may sometimes include requirements. Why is this a problem?
Requirements

A requirement is a statement that formally specifies a necessary capability or characteristic of a business enterprise, application (system or SW), component, or application domain.

Good requirements must be:
- Mandatory (i.e., required)
- Cohesive
- Consistent
- Correct
- Feasible
- Relevant
- Unambiguous
- Uniquely Identifiable
- Verifiable and Validatable
- What, not how (architecture, design, or implementation)
Safeguards (Safety Mechanisms)

- **A safeguard** is a kind of defense that fulfills a safety-related requirement and thereby eliminates or reduces the impact of a safety vulnerability.

- A safeguard is a part of the system (e.g., component, procedure, training).

- Only relevant to requirements if specified as safety constraints.
Safety Vulnerabilities

- A **safety vulnerability** is a weakness in the architecture, design, implementation, integration, or deployment of a system that enables a hazard to exist or an accident to occur.

- Only relevant to requirements if a requirement needs to be specified to prevent the vulnerability or mitigate its negative consequences

- For example, if taxi doors did not have locks or lock sensors.
Putting the Safety Concepts Together

- **Requirement**
  - Functional Requirement
  - Non-Functional Requirement
  - Quality Criterion
  - Quality Measure

- **Safety Goal**
  - states the importance of achieving a target level of
  - mandates a desired criterion of

- **Safety Policy**
  - specifies a
  - mandates a minimum amount of

- **Safety Risk**
  - fulfills a
  - mandates elimination or reduction of a
  - includes relevant states of the

- **Safety Requirement**
  - includes relevant states of the

- **Safeguard**
  - exists because of actual or potential
  - eliminates or reduces

- **Vulnerability**
  - exploits

- **Safety Incident**
  - may cause

- **Accidental Harm**
  - may occur to an

- **Asset**
  - is responsible for an

- **System**
  - has a legitimate interest in the

- **Stakeholder**
  - is valuable to a

- **Accident**
  - Near Miss

- **Hazard**
  - is due to

- **Environment**
  - includes relevant states of the

- **Quality Factor**
  - describes a quality attribute of a

- **Money**
  - Facilities
  - Services
  - Software
  - Hardware
  - Data

- **People**

- **Property**

- **Environment**

- **Services**

- **Societal**

- **Safety**

- **Accidental**
  - Harm

- **Harm**
  - may cause

- **Incident**
  - may result in

- **Vulnerability**
  - has

- **Safeguard**
  - fulfills a

- **Requirement**
  - Quality Criterion
  - Quality Measure

- **Safety Goal**
  - states the importance of achieving a target level of
  - mandates a desired criterion of
Safety-Related Requirements

- Safety Requirements
- Safety-Significant Requirements
- Safety System Requirements
- Safety Constraints
Types of Requirements

- Process Requirements
- Product Requirements
  - Functional Requirements
  - Non-Functional Requirements
    - Data Requirements
    - Interface Requirements
    - Quality Requirements
    - Constraints
      - Defensibility Requirements
      - Safety Constraints
      - Safety Requirements
      - Security Requirements
      - Survivability Requirements

- System Requirements
- Software Requirements
- Hardware Requirements
- Security System Requirements
Safety-Related Requirements

- **Safety-Related Requirements** are any system requirements having *significant safety ramifications*:
  - **Safety Requirements** are requirements that specify mandatory amounts of pairs of subfactors of the safety quality factor.
  - **Safety-Significant Requirements** are non-safety primary mission requirements with significant safety ramifications.
  - **Safety System Requirements** are requirements for safety systems or subsystems (as opposed to primary mission requirements).
  - **Safety Constraints** are constraints intended to ensure a minimum level of safety.
Safety-Related Requirements

Safety Integrity Level (SIL)

- 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-Significant Requirements
- SIL = 1

Safety-Independent Requirements
- SIL = 0

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

Non-Safety Quality Requirements

Safety Constraints
- Protection of Valuable Assets Requirements
- Detection of Safety Incidents Requirements
- Reaction to Safety Incidents Requirements
- Adaption to Safety Incidents Requirements

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

Constraints
- Main Mission Requirements
- Safety System Requirements
[Pure] Safety Requirements

- A safety requirement is a kind of defensibility requirement because safety is a type of defensibility. (Safety requirements are like security requirements.)

- Safety requirements specify minimum required amounts of:
  - Safety
  - A quality subfactor of safety:
    - Defensibility Problem Type: Accidental Harm, Safety Incident, Hazard, Safety Risk
    - Defensibility Solution Type: Prevention, Detection, Reaction, Adaptation

- A safety requirement is a combination of a safety criterion and a minimum threshold on a safety measure.
Quality Requirements

- Quality Requirements are based on a quality model:

```
Quality Model

Quality Factor
  \hspace{1cm} \text{provides evidence for existence of}

Quality Subfactor
  \hspace{1cm} \text{provides evidence for existence of}

Quality Measure with Threshold
  \hspace{1cm} \text{measures}

System-Specific Quality Criterion
  \hspace{1cm} \text{describes quality of}

System

Quality Requirement
```
Safety Requirements

- Safety Requirements are a kind of quality requirement.

Diagram:

- Quality Model
  - Safety
  - Quality Factor
    - Quality Subfactor
      - System-Specific Quality Criterion
        - System
  - Quality Measure with Threshold
    - Safety Requirement
      - Quality Requirement

- Provides evidence for existence of measures
- Requires minimum amount of
- Describes quality of
Safety Requirements (Simplified)

- Previous figure with supertypes removed for simplicity:
Based on Safety Subfactors

Accidental Harm
- Safety Incident
  - Hazard
- Safety Risk

Safety Incident

Hazard

Safety Risk

Prevention
- Detection
- Reaction
- Adaptation

Safety Problem Type

Safety Solution Type

is measured using

Safety Measure

provides evidence for existence of

provides evidence for existence of

measures

describes the safety of the

System-Specific Safety Criterion

System

Safety

Safety Subfactor
Safety Requirements

Based on the previous figure, there are twelve types of safety requirements:

- Accidental harm prevention, detection, and reaction
- Safety incident prevention, detection, and reaction
- Hazard prevention, detection, and reaction
- Safety risk prevention, detection, and reaction
Example Safety Requirements

- “The system shall not cause more than X amount of accidental harm per year.”
- “The system shall not cause more than X safety incidents (accidents, near misses) per passenger mile traveled.”
- “The system shall not cause hazard X to exist more than Y percent of the time.”
- “The system shall not allow a safety risk level of X to exist.”
- “The system shall detect accidents of type X Y percent of the time.”
- “The system shall react to accidents of type X by performing Y.”
ZATS Safety Requirements

- What are some reasonable ZATS safety requirements related to preventing:
  - Accidental harm to valuable assets?
  - Safety incidents from occurring?
  - Hazards from existing?
  - Safety risks from being too high?

- How about:
  - *Detecting* accidental harm, safety incidents, hazards, and risks?
  - *Reacting* to the detection of harm, incidents, hazards, and risks?
  - *Adapting* ZATS to better handle future harm, safety incidents, hazards, and risks?
Safety-Significant Requirements

- Are identified based on safety (hazard) analysis
- Subset of non-safety requirements:
  - Functional Requirements
  - Data Requirements
  - Interface Requirements
  - Non-safety Quality Requirements
  - Constraints
- Safety Integrity Level (SIL) is not 0:
  - May have minor safety ramifications
  - May be safety-critical
  - May have intolerable safety risk
SILs and SEALs

- **Safety Integrity Level** – a category of required safety for safety-significant requirements.

- **Safety Evidence Assurance Level** – a category of required evidence needed to assure stakeholders (e.g., safety certifiers) that the system is sufficiently safe (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-Significant Requirements (cont)

- Require enhanced Safety Evidence Assurance Levels (SEALs) including more rigorous development process (including better requirements engineering):
  - Formal specification of requirements
  - Fagan inspections of requirements

- Too often SEALs only apply to design, coding, and testing:
  - Safe subset of programming language
  - Design inspections
  - Extra testing
Example Safety-Significant Requirements

- Requirements for controlling elevator doors:
  - Keep doors closed when moving
  - Not crush passengers

- Requirements for firing missiles from military aircraft:
  - When to arm missile
  - Controlling doors providing stealth capabilities
  - Detecting weight-on-wheels

- Requirements for chemical plant:
  - Mixing and heating chemicals
  - Detecting temperature and pressure
ZATS Safety-Significant Requirements

- What are some reasonable ZATS functional requirements with safety ramifications?
- What is a reasonable data requirement with safety ramifications?
- What is a reasonable interface requirement with safety ramifications?
- What SIL level (e.g., intolerable, undesirable, tolerable, insignificant) should be assigned to these safety-significant requirements?
Safety System Requirements

- Systems or components strictly added for safety:
  - Emergency core coolant system for nuclear power plant
  - Fire detection and suppression system for aircraft
  - Emergency full pump cut off for gas station
  - Emergency stop for escalators

- All requirements for such systems are safety-related
Example Safety System Requirements

- “Except when the weapons bay doors are open or have been open within the previous 30 seconds, the weapons bay cooling system shall maintain the temperature of the weapons bay below $X^\circ$ C.”

- “The Fire Detection and Suppression System (FDSS) shall detect smoke above $X$ ppm in the weapons bay within 5 seconds.”

- “The FDSS shall detect temperatures above $X^\circ$ C in the weapons bay within 2 seconds.”

- “Upon detection of smoke or excess temperature, the FDSS shall alert the pilot within 1 second and begin fire suppression.”
ZATS Safety System Requirements

- Would the ZATS system need a safety subsystem?
- If so, what would that subsystem be and what would a few of its requirements be?
Safety Constraints

- A constraint is any engineering decision that has been chosen to be mandated as a requirement. For example:
  - Architecture constraints
  - Design constraints
  - Implementation (e.g., coding) constraints
  - Testing constraints

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

- Safety standards often mandate best practices safety constraints.
Example 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.”
ZATS Safety Constraints

osi What would be reasonable safety constraints to specify on the ZATS system?
Safety & Requirements Engineering

- Set Safety Goals
  - Safety Team
  - Safety Program Planning
  - Safety Significance Analysis
  - Safety Control Analysis

Safety Goals

Application Visioning

Application Vision Statement (ConOps)

System Requirements Specification

Safety-Related Requirements

System Requirements

Safety-Related Requirements Elicitation

Requirements Elicitation

Safety Analysis

Safety Requirements

Safety System Requirements

Safety Constraints

System Requirements

Requirements Specification
Safety Program Planning

Safety Team

Set Safety Policy

Set Safety Goals

Determine Safety Categories

Develop Safety Program

Safety Policy

Safety Goals

Asset Value Categories

Harm Severity Categories

Hazard Likelihood Categories

Safety Incident Likelihood Categories

Safety Risk Matrix

Safety Integrity Levels (SIL)

Safety Evidence Assurance Levels

Safety Program Plan

Subject Matter Experts

Stakeholders

Project Documentation (RFP, Contract, ConOps)

Legacy Documentation

Generic / Reusable Safety Categories

Standard / Reusable Safety Evidence Assurance Levels

Safety Team performs

Safety Program Planning

Set Safety Policy

Set Safety Goals

Determine Safety Categories

Develop Safety Program

Subject Matter Experts

Stakeholders

Project Documentation (RFP, Contract, ConOps)

Legacy Documentation

Generic / Reusable Safety Categories

Standard / Reusable Safety Evidence Assurance Levels
Safety Analysis

Safety Team

Architecture Team

Prelim. Hazard Analysis

System Hazard Analysis

Safety Analysis

Asset Analysis

Safety Incident Analysis

Hazard Analysis

Safety Risk Analysis

Safety Significance Analysis

Safety Control Analysis

Asset Safety Requirements

Accident Safety Requirements

Hazard Safety Requirements

Safety Risk Safety Requirements

Safety Requirements

Safety-Significant Requirements

Safety System Requirements

Safety Constraints

Safety-Related Requirements

Supports

Helps perform

Identifies
Asset Analysis

Subject Matter Experts

Safety Team

Performs

Stakeholders

Project Documentation (RFP, Contract, ConOps)

Generic / Reusable Asset Lists

Generic / Reusable Asset / Harm Tables

Standard / Reusable Harm Severity Categories

Standard / Reusable Asset / Harm Requirements

Asset Analysis

Value Analysis

Harm Analysis

Asset Identification

Asset / Harm Requirements Production

Asset List

Asset Value and Harm Table

Asset / Harm Requirements

Helps perform

Requirements Team

Safety Team
Safety Incident Analysis

Subject Matter Experts

Safety Team

- Stakeholders

- Project Documentation (RFP, Contract, ConOps)
- Generic / Reusable Safety Incident Type Lists
- Asset Value and Harm Table
- Harm Severity Categories
- Generic / Reusable Safety Incident / Harm Tables
- Standard / Reusable Safety Incident Likelihood Categories
- Safety Incident Likelihood Categories
- Standard / Reusable Safety Incident Requirements

Safety Incident Analysis

- Safety Incident Type Identification
- Safety Incident Harm Analysis
- Safety Incident Likelihood Analysis
- Safety Incident Requirements Production

Stakeholders

Safety Team

helps perform

Requirements Team
Hazard Analysis

- Hazard Identification
- Hazard Categorization
- Network of Causes Analysis
- Root Cause Analysis
- Common Cause Analysis
- Hazard Likelihood Analysis
- Hazard Effects Analysis
- Hazard Cause & Effect Diagrams and Tables
- HAZOP/FEMA
- Hazard Categories
- Fault/Event Trees
- Hazard Cause & Effect Diagrams and Tables
- Hazard Likelihood Table
- Hazard Reports
- Hazard Safety Requirements
- Hazard List

Subject Matter Experts

Stakeholders

Safety Team performs

Project Documentation (System Architecture)

Generic / Reusable Hazard Lists

Standard / Reusable Hazard Categories

Standard / Reusable Hazard Likelihoods

Generic / Reusable Hazard Safety Requirements

Requirements Team helps perform

Hazard Analysis

Requirements Team

Tutorial T3 Engineering Safety-Related Requirements for Software-Intensive Systems
Safety Risk Analysis

Subject Matter Experts

Stakeholders

Performs

Safety Team

Harm Severity Categories

Safety Risk Table

Generic / Reusable Safety Risk Matrices

Standard / Reusable Safety Risk Categories

Accident / Hazard Likelihood Categories

Standard / Reusable Safety Integrity Levels

Standard / Reusable Safety Evidence Assurance Levels (SEALs)

Safety Risk Categories

Generic / Reusable Safety Risk Requirements

Safety Risk Determination

Safety Risk Estimation

Safety Risk Requirements Production

Safety Risks

Accident Type Safety Risk Table

Hazard Safety Risk Table

Safety Risk Requirements

Helps perform

Requirements Team

Tutorial T3

Engineering Safety-Related Requirements for Software-Intensive Systems

74
Safety-Significance Analysis

Subject Matter Experts

Stakeholders

Safety Team performs

Requirements Team helps perform

Safety-Significant Requirements Identification

Identify Safety-Significant Functional Requirements

Identify Safety-Significant Data Requirements

Identify Safety-Significant Interface Requirements

Identify Safety-Significant Non-Quality Requirements

Categorization of Safety-Significant Requirements

Safety Integrity Level (SIL) Allocation

Safety Evidence Assurance Level (SEAL) Allocation

Functional Requirements

Data Requirements

Interface Requirements

Non-Safety Quality Requirements

Safety Risk Tables

Safety Integrity Levels

Engineering Safety-Related Requirements for Software-Intensive Systems
Safety Control Analysis

Safety Team

Architecture Team

performs

supports

Subject Matter Experts

Stakeholders

Safety-Significant Requirements

Safety Analyses

System Architecture

Safety Control Analysis

Safety Controls Identification

Updated System Architecture

Safety System Identification

Safety System Requirements Allocation

Safety Constraints Determination

Safety Constraints

Requirements Team

Updated System Architecture
Conclusion

- Engineering safety-significant requirements requires concepts, methods, techniques, and expertise from both requirements engineering and safety engineering.

- There are multiple types of safety-related requirements that have different forms and that need to be analyzed and specified differently.

- Look for my upcoming book by the same title.

- For more information, check out my repository of over 1,100 free open source reusable process components including many on safety at www.opfro.org.