Engineering Safety- and Security-Related Requirements for Software-Intensive Systems

6th International Workshop on Software Engineering for Secure Systems (SESS'10) Workshop at the 32nd ICSE Conference, Cape Town, South Africa

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Donald Firesmith
2 May 2010

Contents

Three Disciplines:

Requirements, Safety, and Security Engineering

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, vulnerabilities, 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, vulnerabilities, 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)

Challenges:

Combining Requirements, Safety, and Security Engineering

Challenges$_1$

Requirements engineering, safety engineering, and security engineering have different:

- *Communities*
- *Disciplines* with different training, books, journals, and conferences
- *Professions* with different *job titles*
- Fundamental underlying *concepts* and *terminologies*
- *Tasks, techniques,* and *tools*

Safety and security engineering are:

- Typically treated as *secondary specialty engineering* disciplines
- Performed separately from, largely Independently of, and lagging behind the primary engineering workflow:
  (requirements, architecture, design, implementation, integration, testing, deployment, sustainment)

Challenges$_2$

Current separate methods for performing requirements, safety, and security engineering are inefficient and ineffective.

Separation of requirements engineering, safety engineering, and security engineering:

- Causes *poor* safety- and security-related requirements that are often:
  - Vague, unverifiable, unfeasible, architectural and design constraints
  - Capabilities or goals rather than requirements
  - Inadequate and too late to drive architecture and testing
- Makes it unnecessarily harder to achieve certification and accreditation

Challenges$_3$

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
- Large number of defects delivered
- Delivered systems that are never used

Poor requirements are one major root cause of many (or most) accidents involving software-intensive systems.

Most mandated security “requirements” are actually constraints such as :

- Security functions or subsystems
- Industry “best practices”

Challenges$_4$

How safe and secure is safe and secure *enough*?

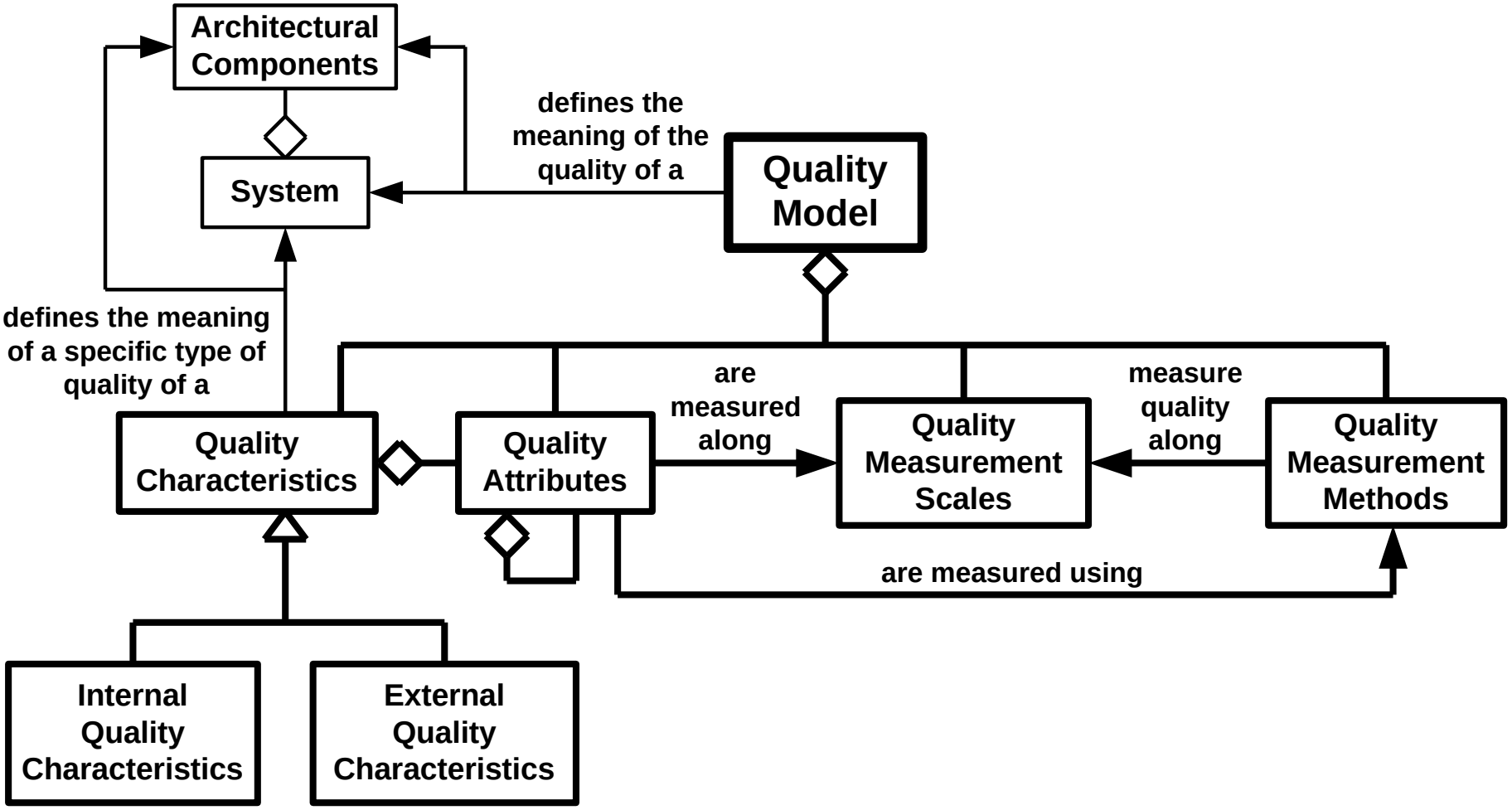
Situation cries out for process improvement:

- Better consistency between safety and security engineering
  - More consistent concepts and terminology
  - Reuse of techniques across disciplines
  - Less unnecessary overlap and avoidance of redundant work
- Better collaboration:
  - Between safety and security engineering
  - With requirements engineering
- Better safety- and security-related requirements

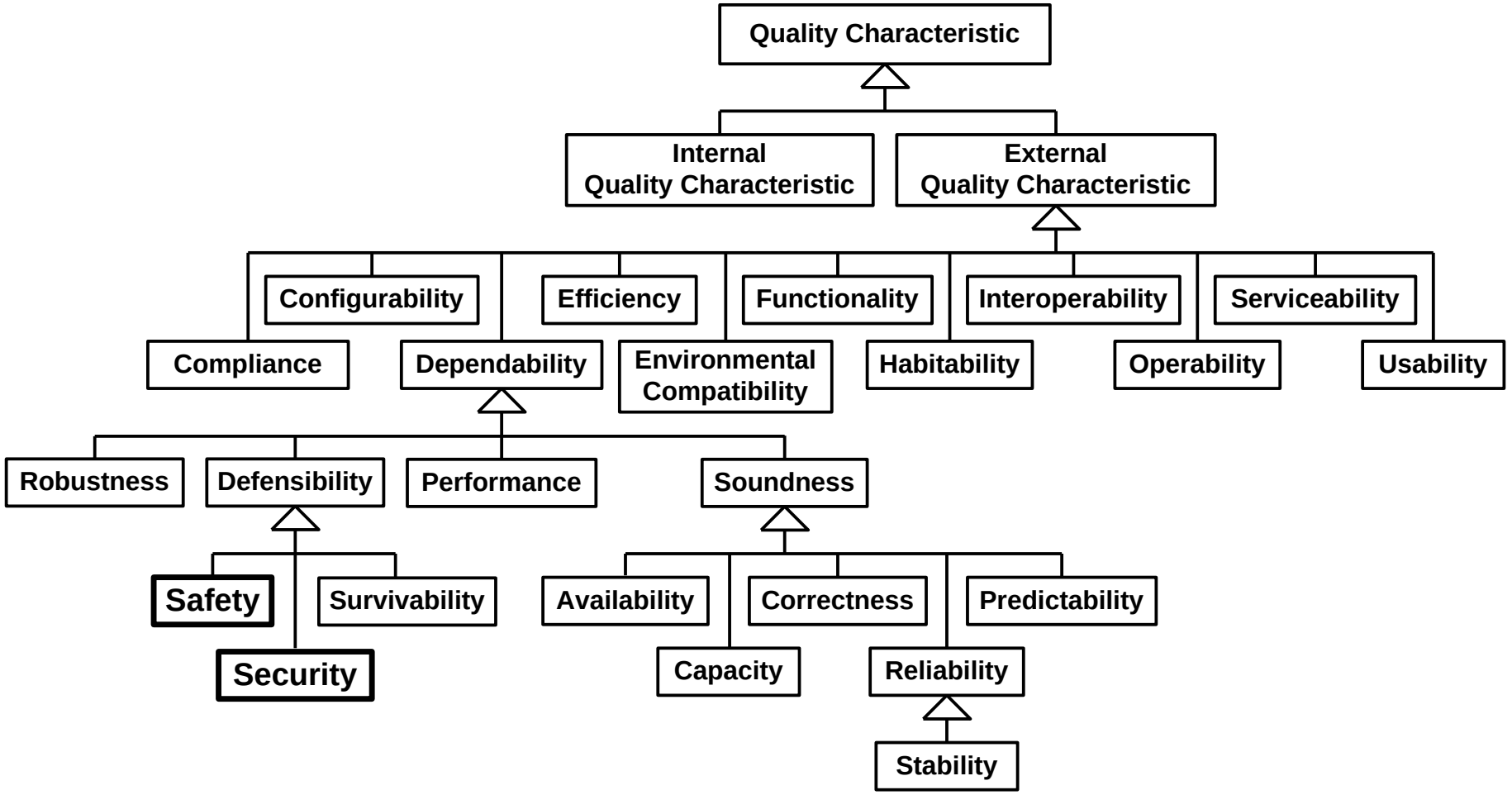
Fundamental Concepts:

A Foundation for Understanding

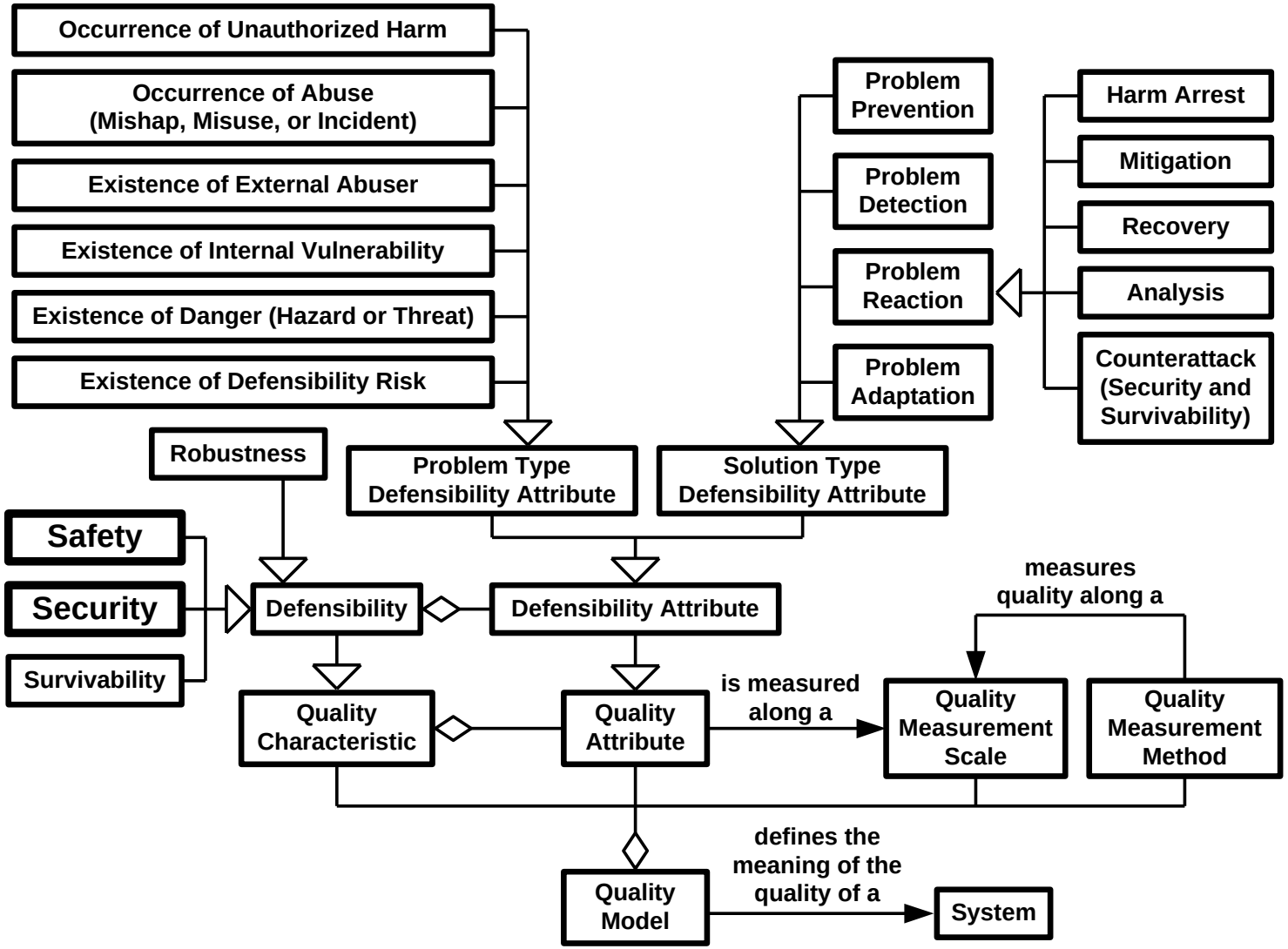
Quality Model



Quality Characteristics (External)



Defensibility Quality Attributes



Defensibility

Defensibility

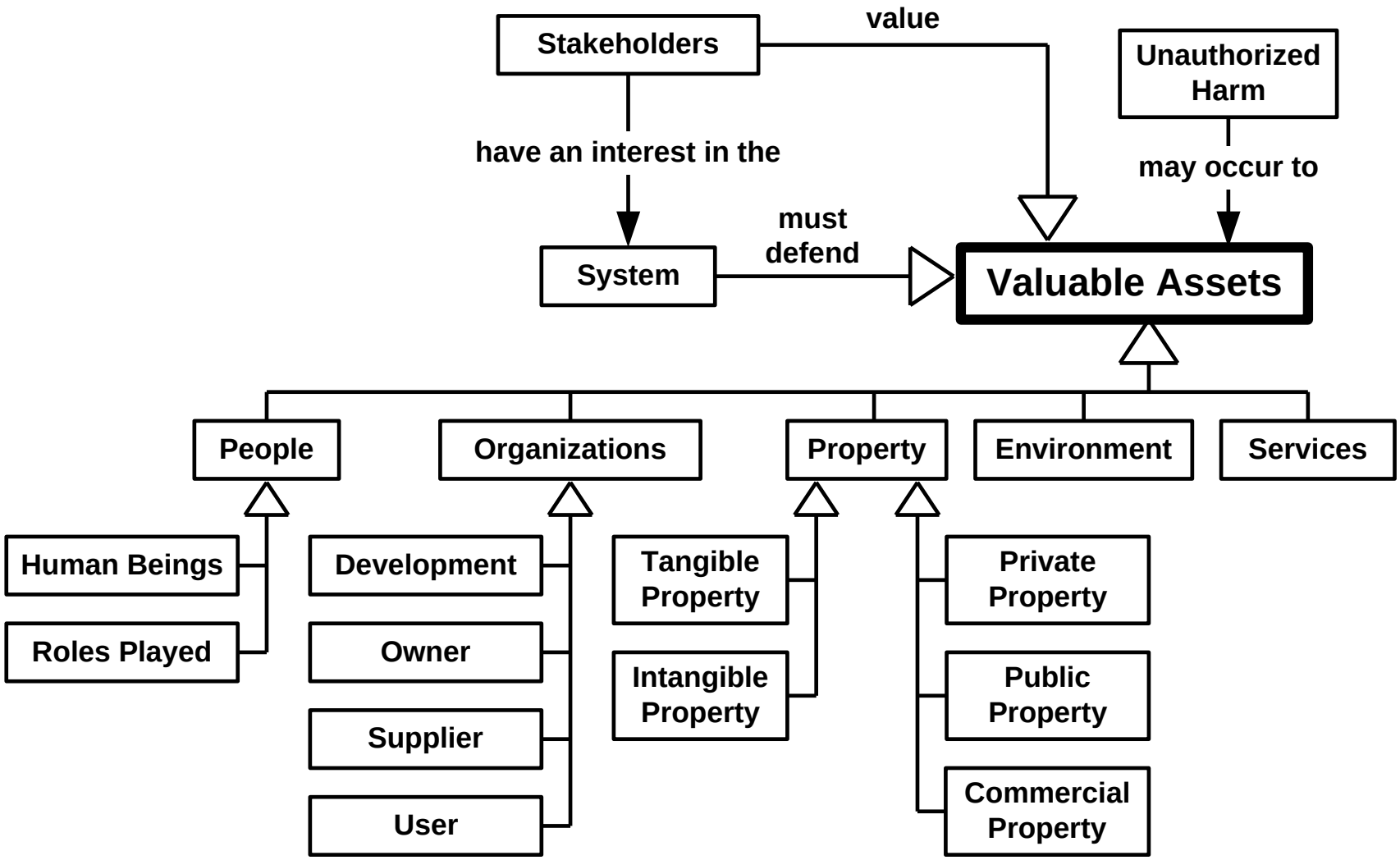
the quality characteristic capturing the degree to which the system:

- Properly prevents, detects, reacts to, and adapts to:
  - Unintended and unauthorized *harm* to *valuable assets* due to the occurrence of
  - *Abuses* enabled by the existence of
  - *Dangers*
- Has *defensibility risks* that are acceptably low to its *stakeholders*

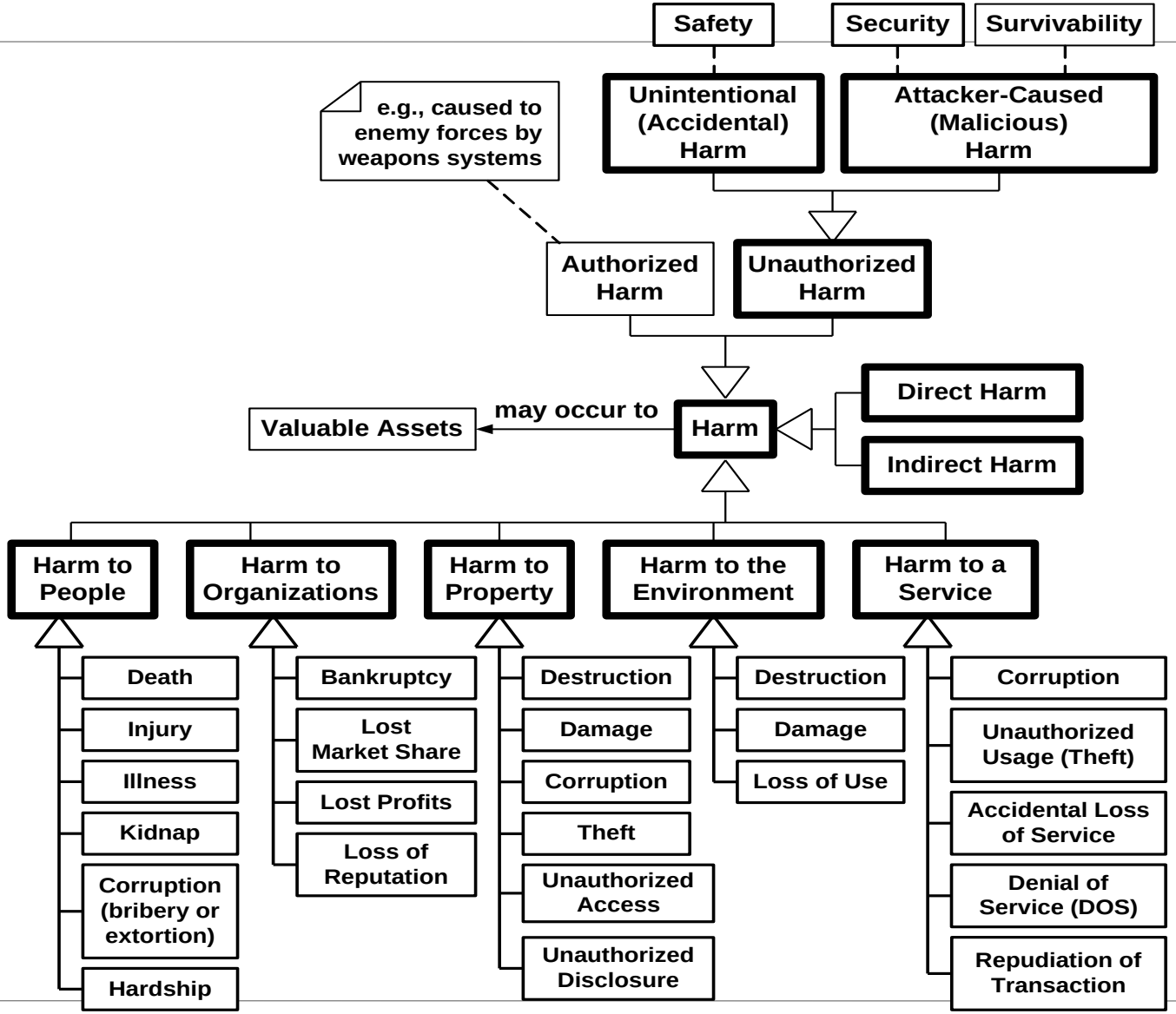
Safety and security are defined in a similar manner by replacing:

- Abuse with either mishap (safety) or misuse (security)
- Danger with either hazard (safety) or threat (security)
- Defensibility risks with safety risks and security risks

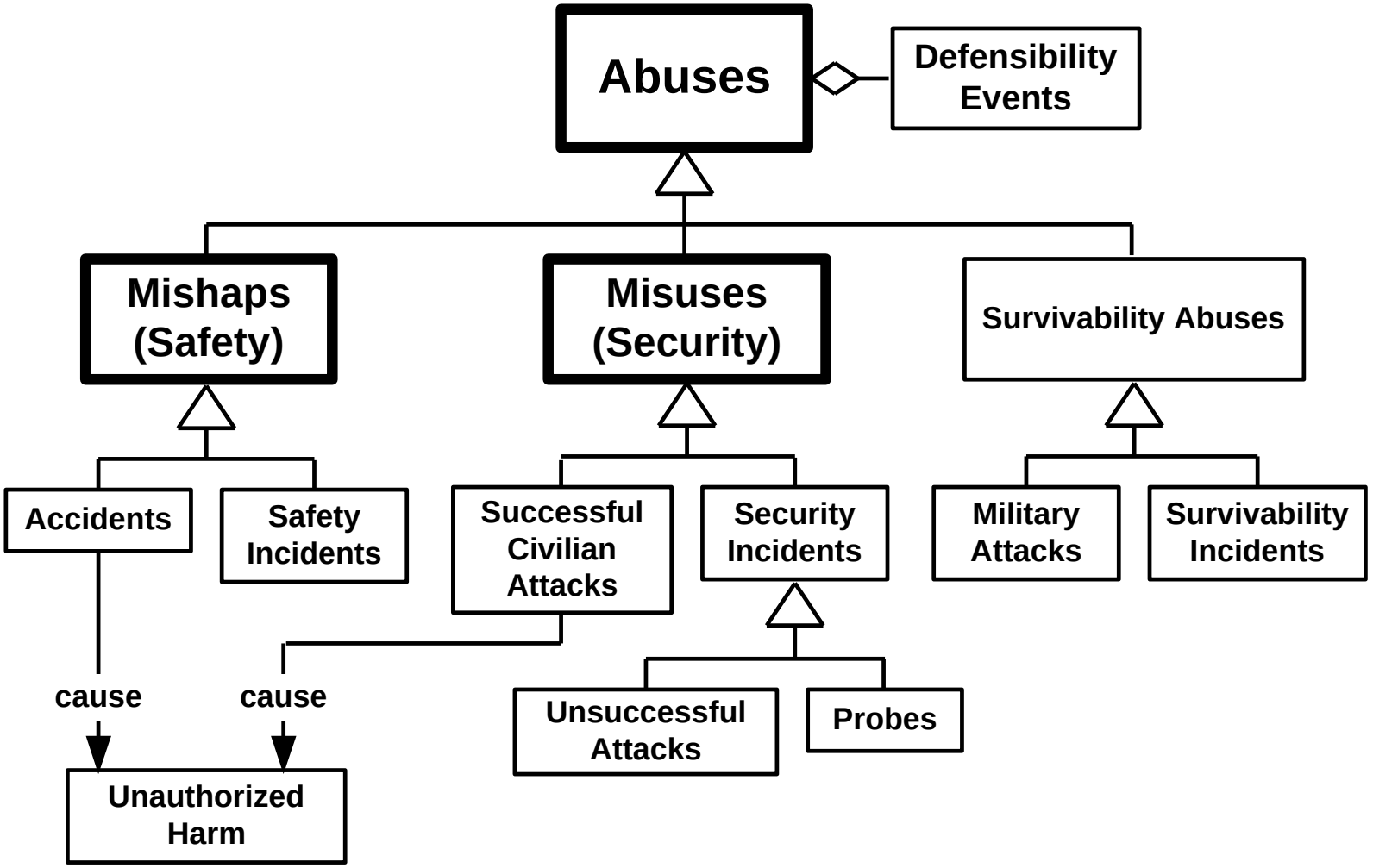
Unauthorized Harm to Valuable Assets



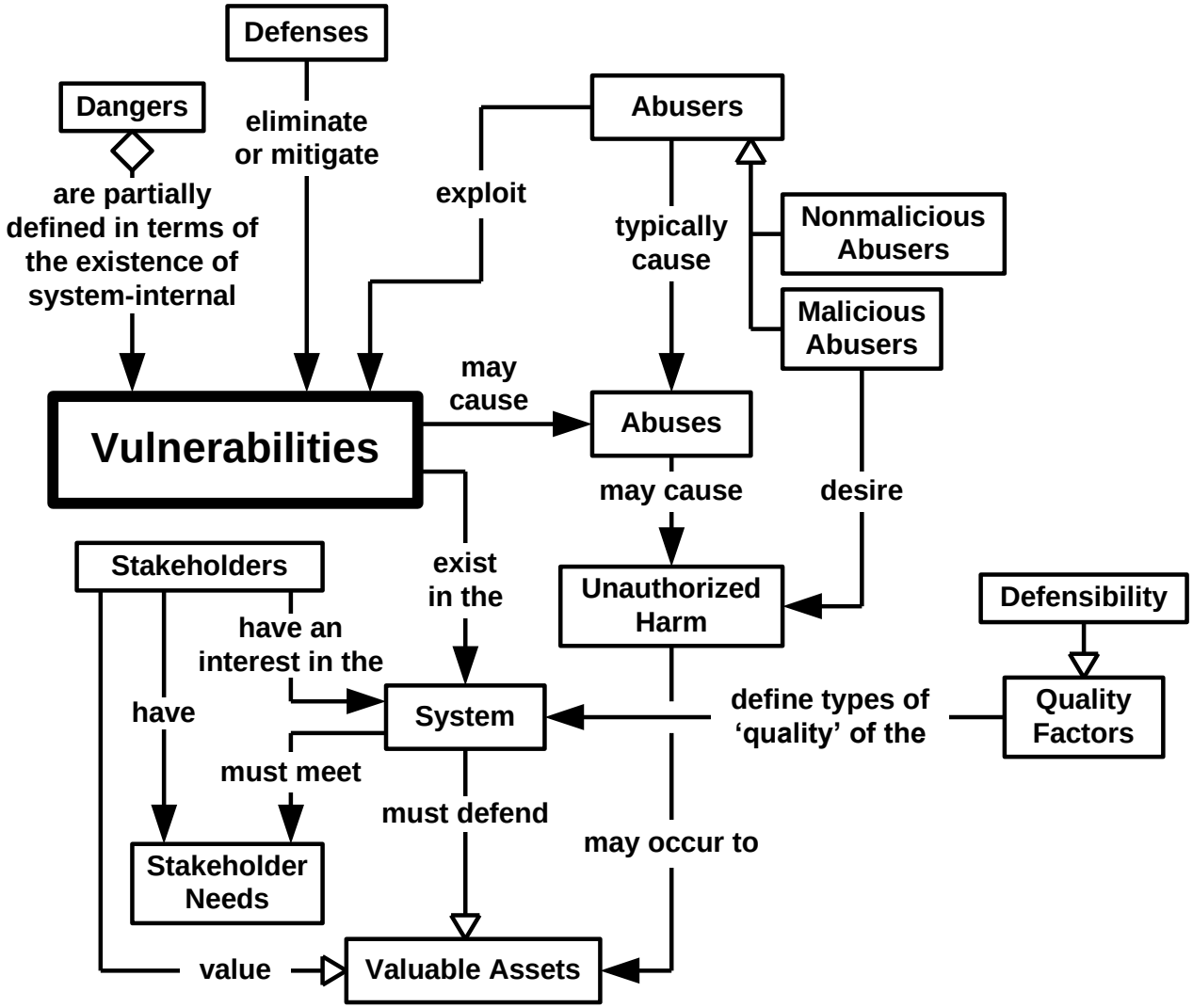
Types of Harm



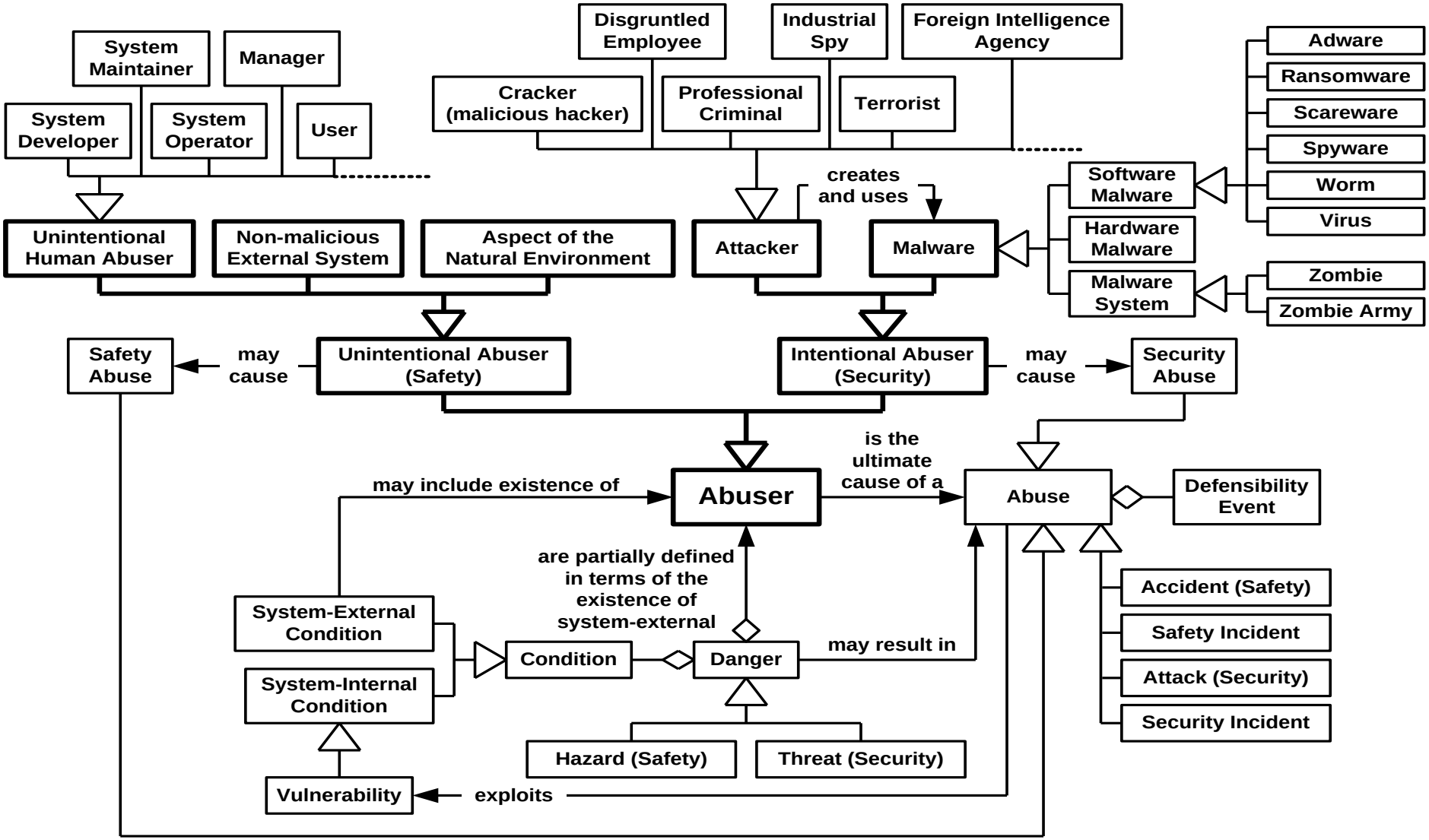
Types of Abuses



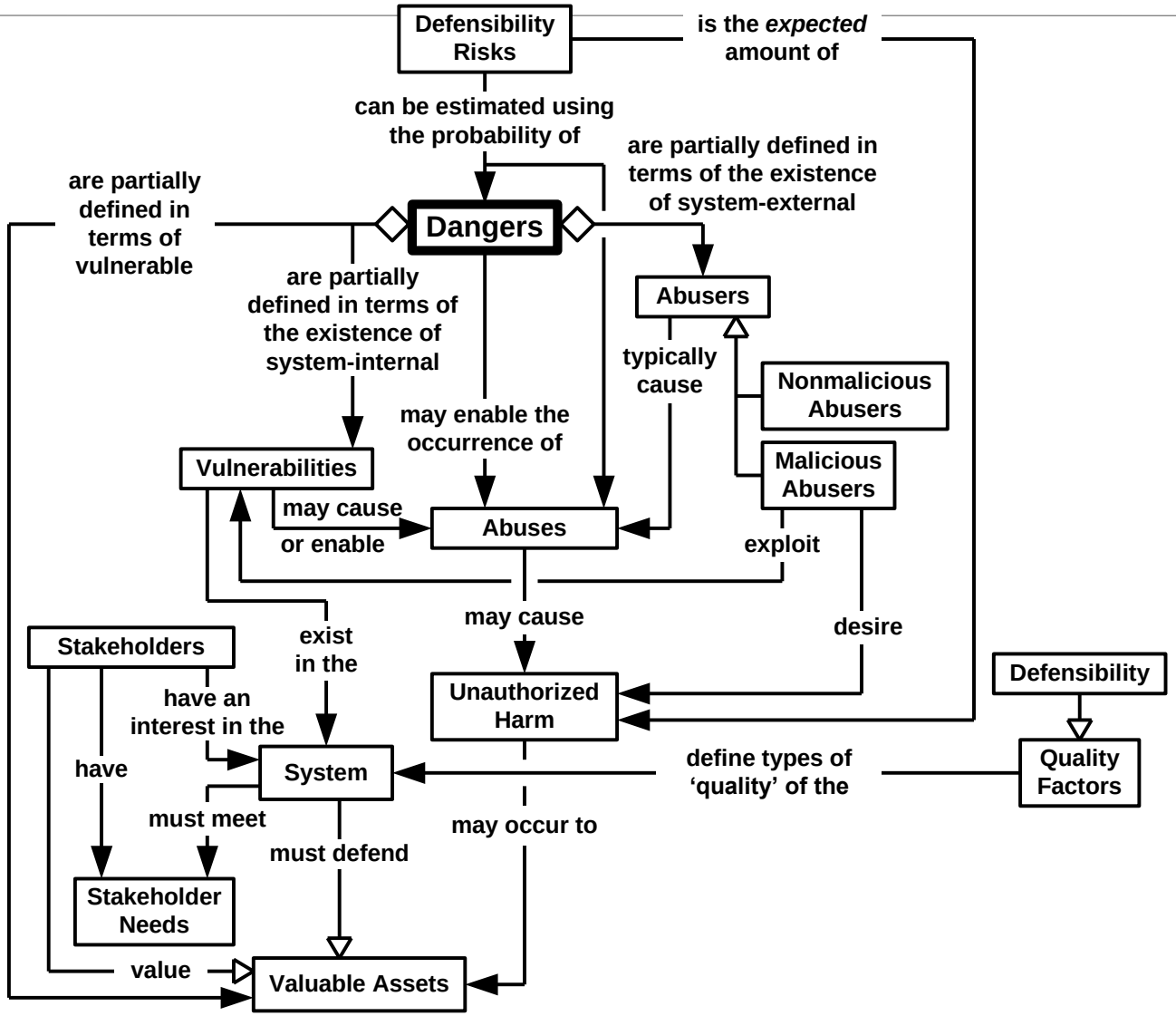
Vulnerabilities



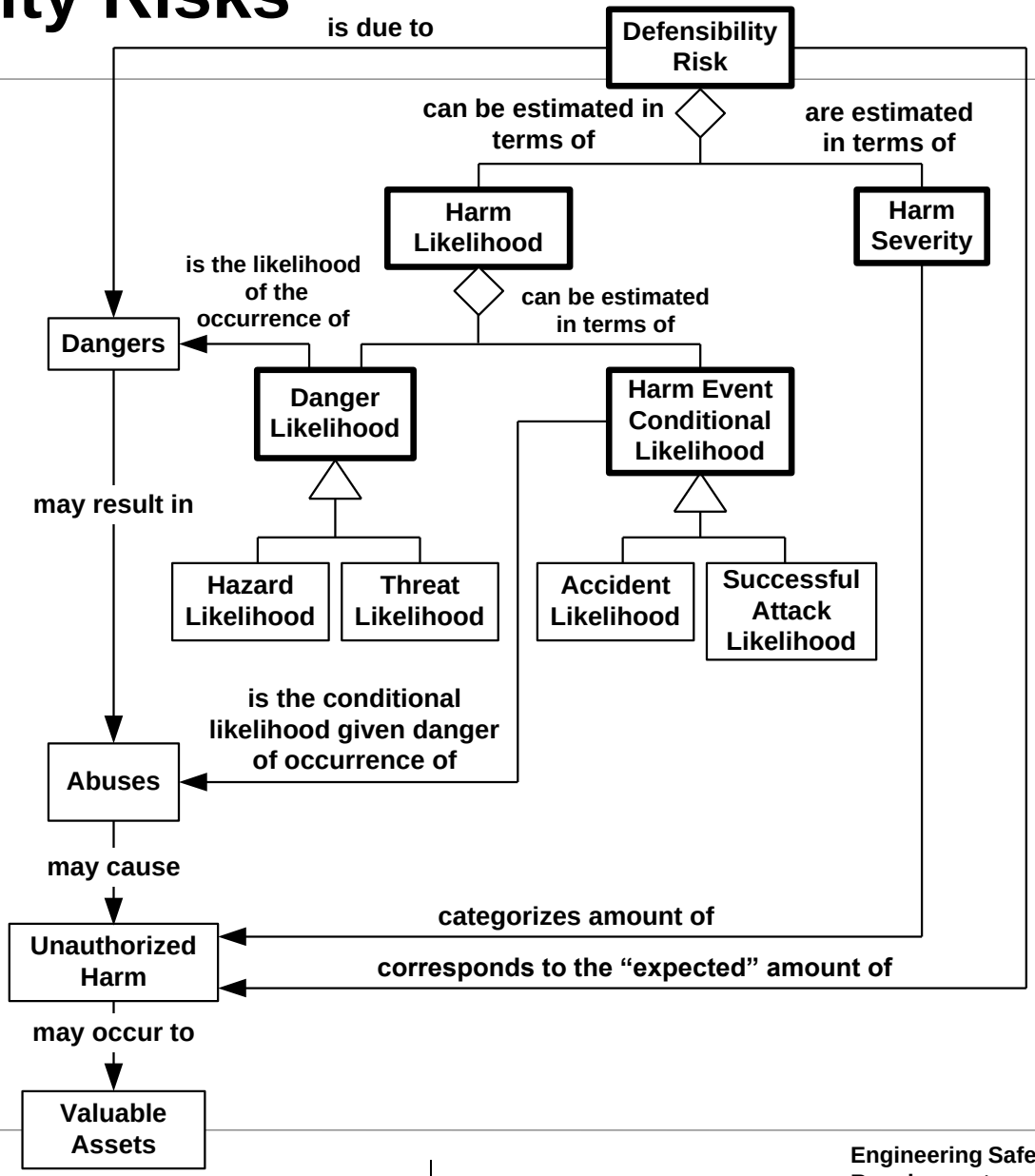
Types of Abusers



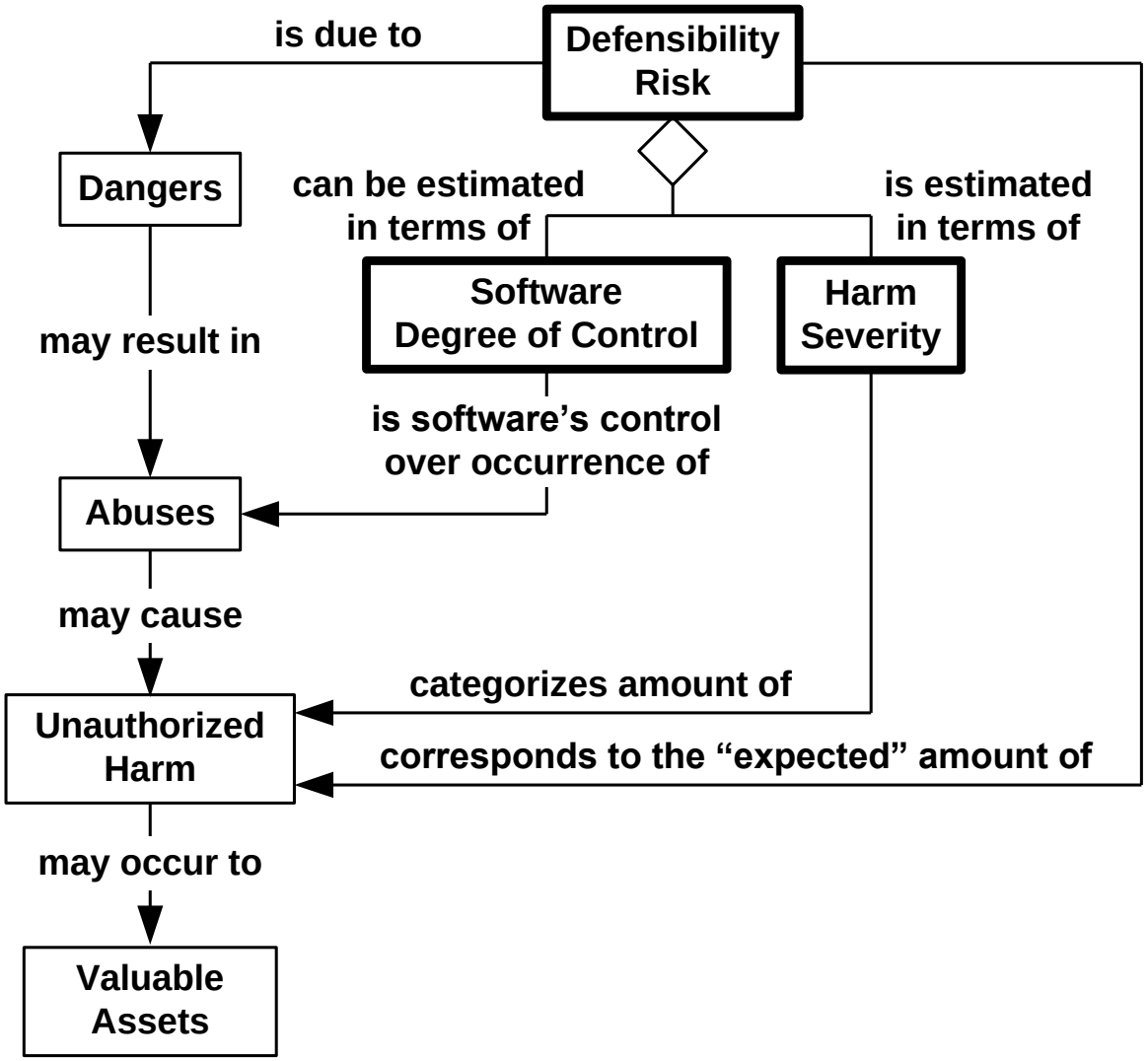
Dangers



Defensibility Risks



Risk in terms of Software Degree of Control



Safety- and Security-Related Requirements

Types of Safety- and Security-Related Requirements

Too often only a Single Type of Requirements is considered.

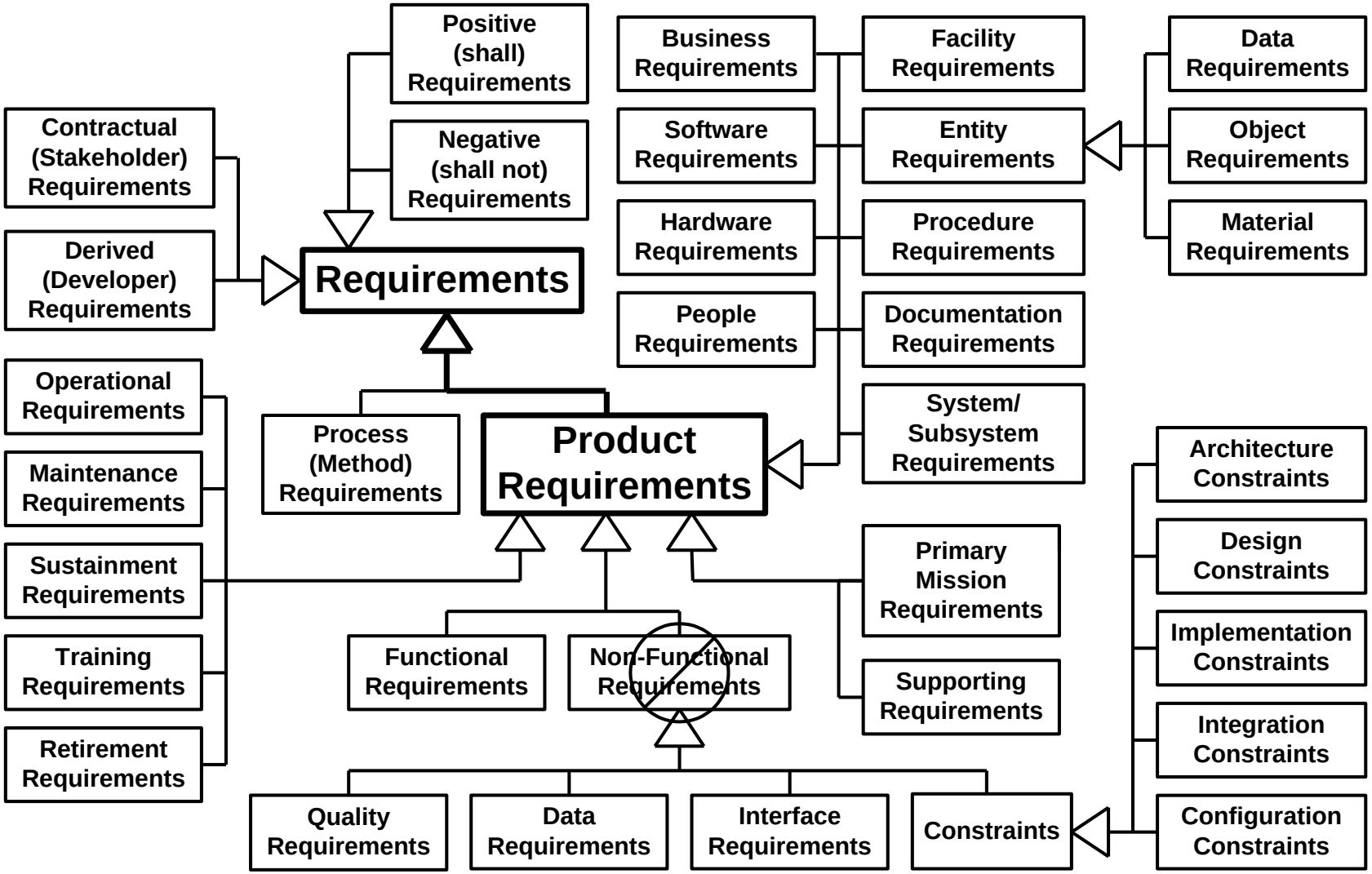
Not just:

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

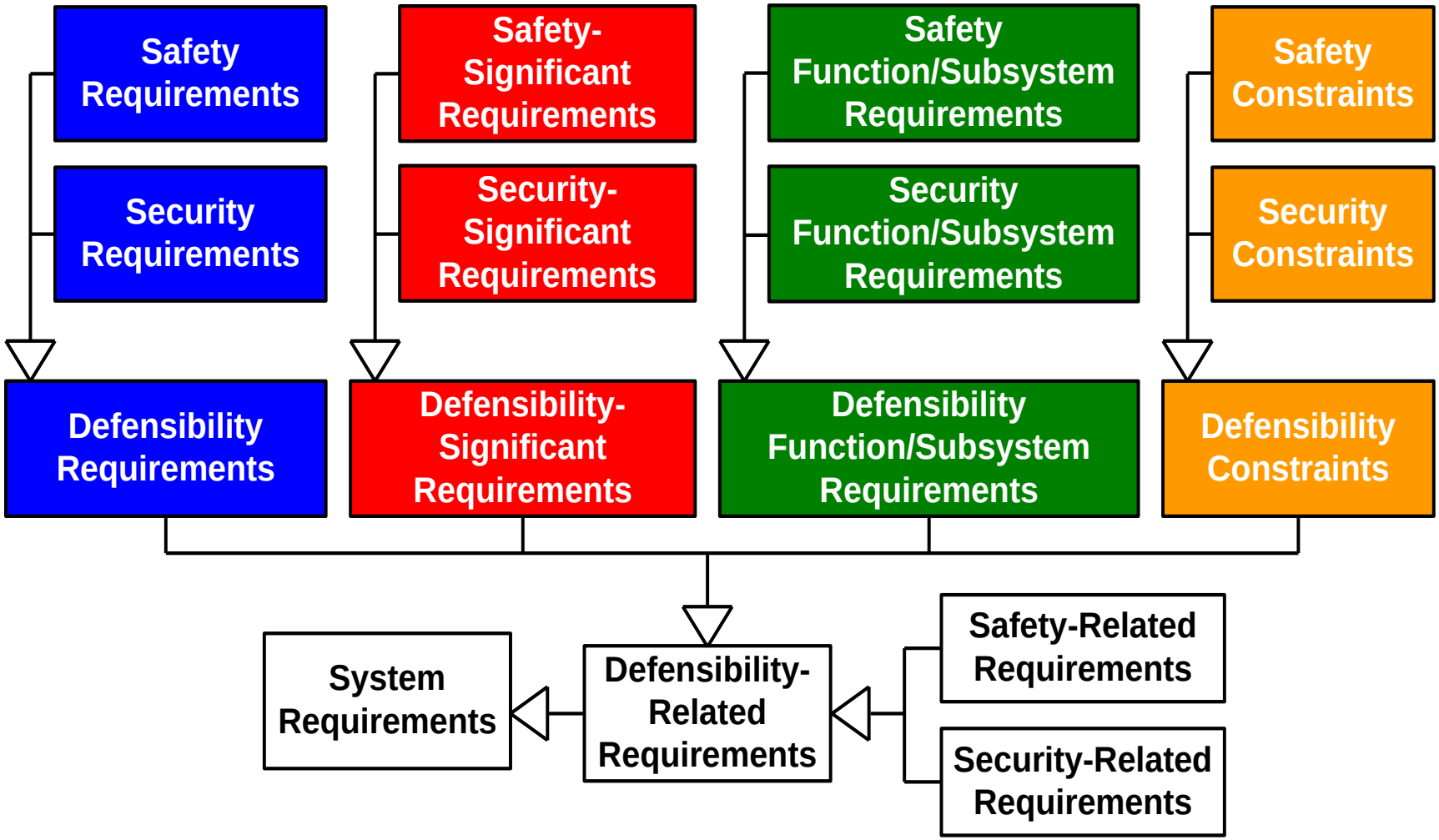
Reason for Presentation Title

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

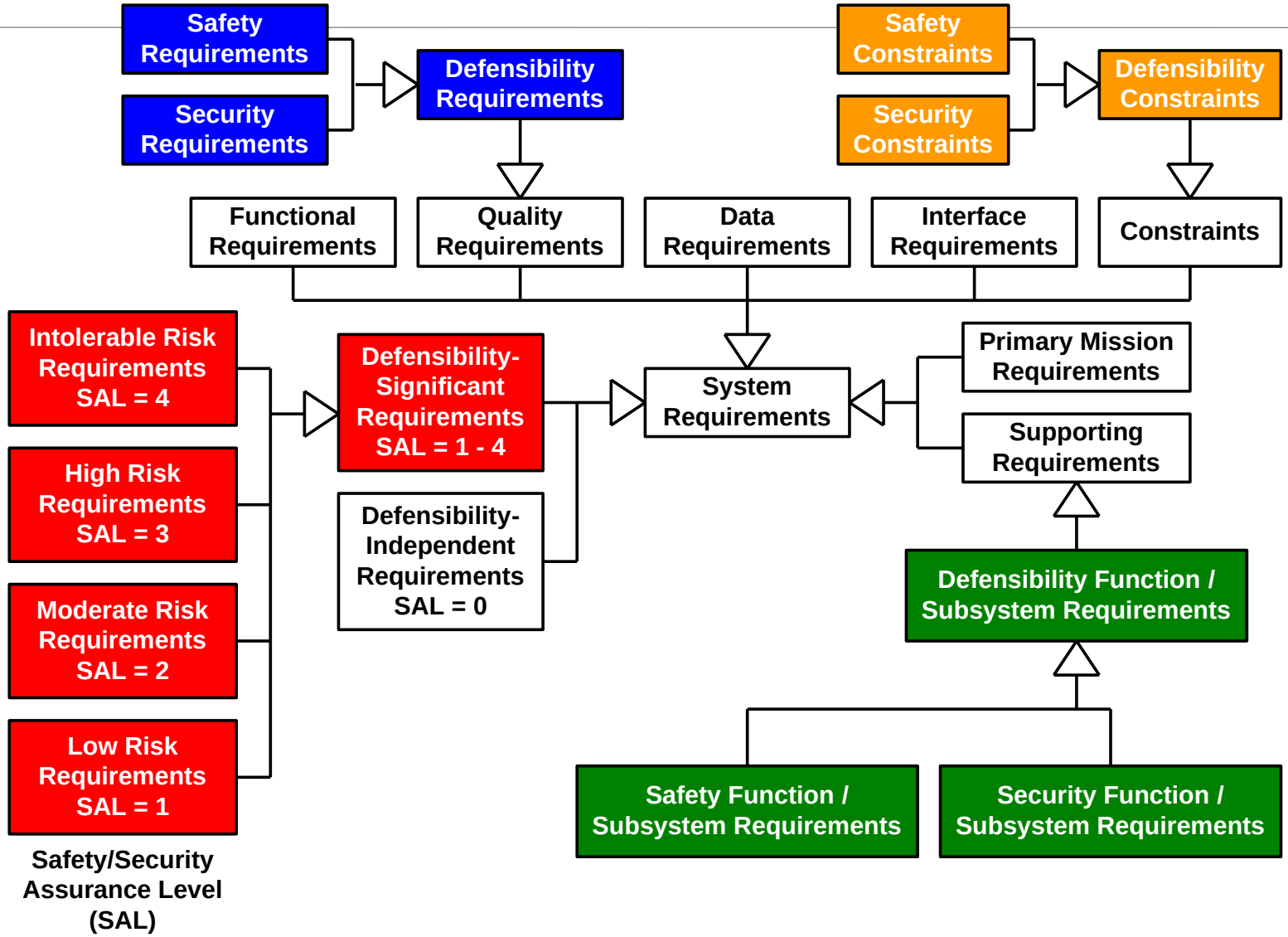
Types of Requirements



Types of Defensibility-Related Requirements



Four Types of Defensibility-Related Requirements



Example Safety- and Security-Related Requirements (Actually Goals – Requirements are more specific)

Safety / Security Requirement :

- "When in mode V, the system shall limit the occurrence of *accidental harm* of type W to valuable assets of type X to an average rate of no more than Y asset value per Z time duration."
- "When in mode X, the system shall *detect misuses* of type Y an average of at least Z percent of the time."

Safety / Security Significant Requirement

- "The system shall automatically transport passengers between stations."
- "The system shall enable users to update their personal information."

Safety / Security Function / Subsystem Requirement

- "The system shall include a fire detection and suppression subsystem."
- "The system shall support the encryption/decryption of sensitive data."

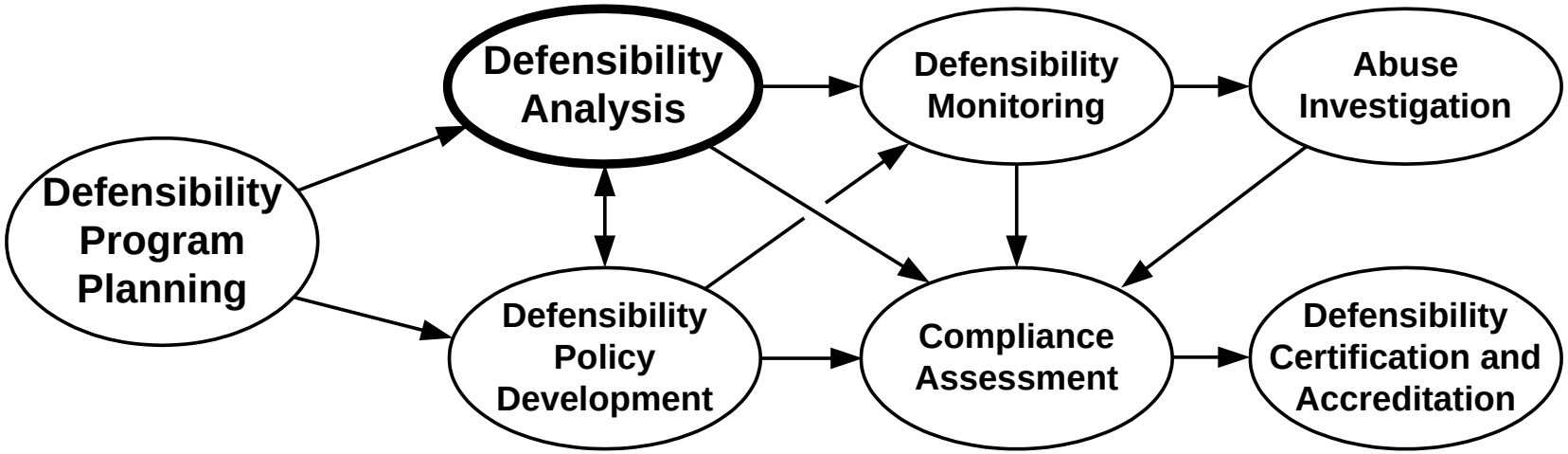
Safety / Security Constraint

- "The system shall not contain any of the hazardous materials in Table X."
- "The system shall use passwords for user authentication."

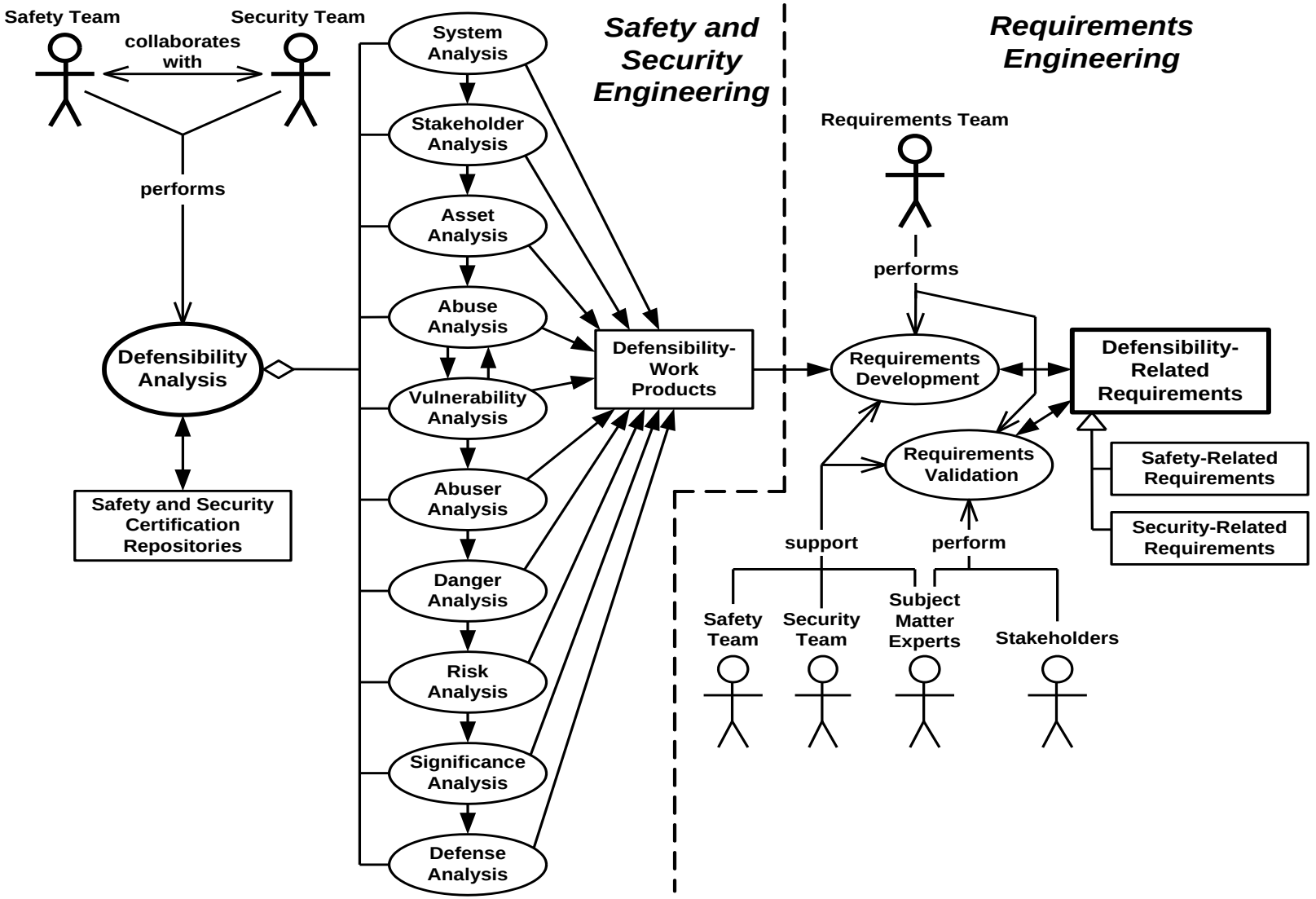
Common Process:

A Basis for Effective Collaboration

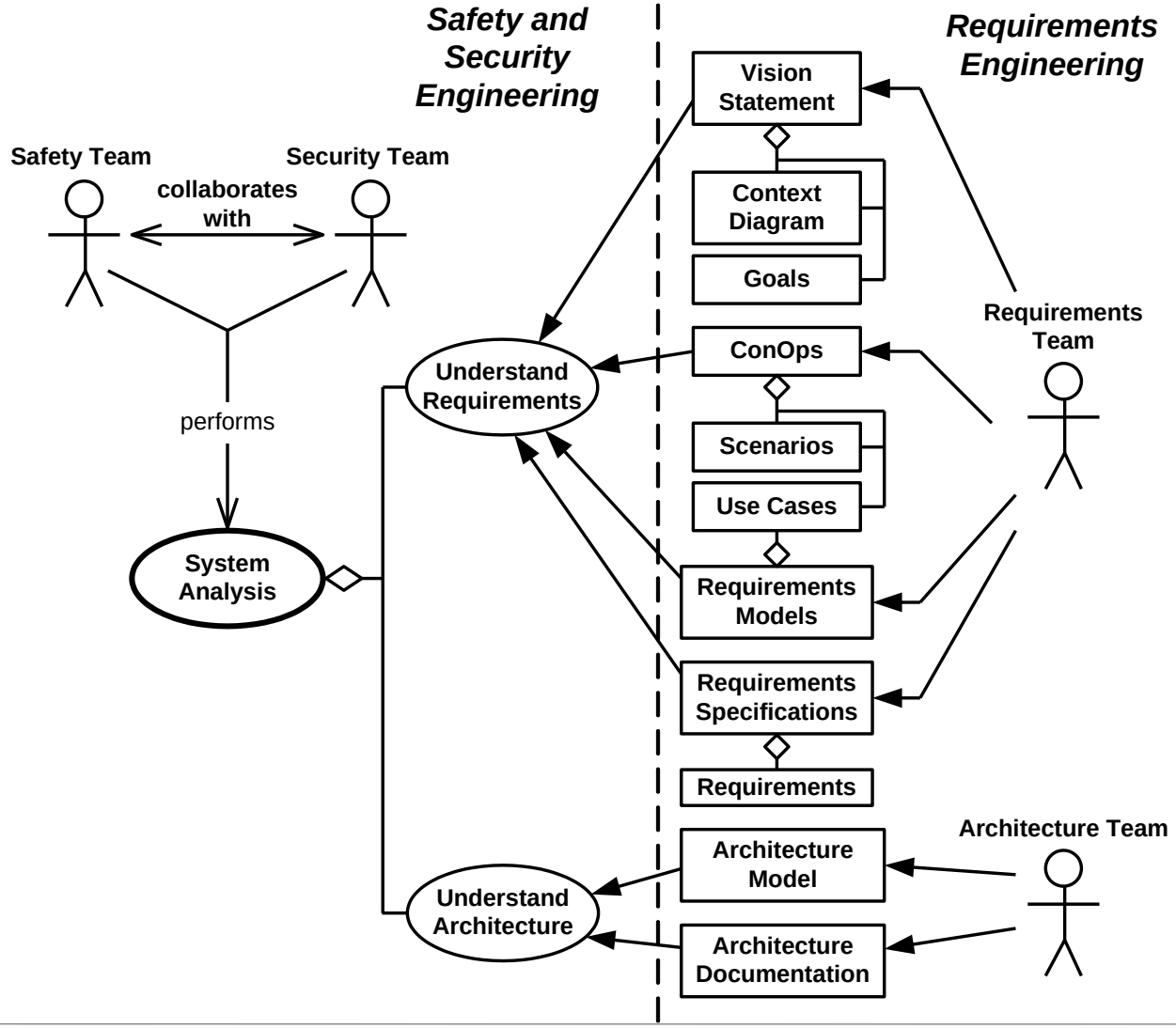
Overall Defensibility Engineering Method



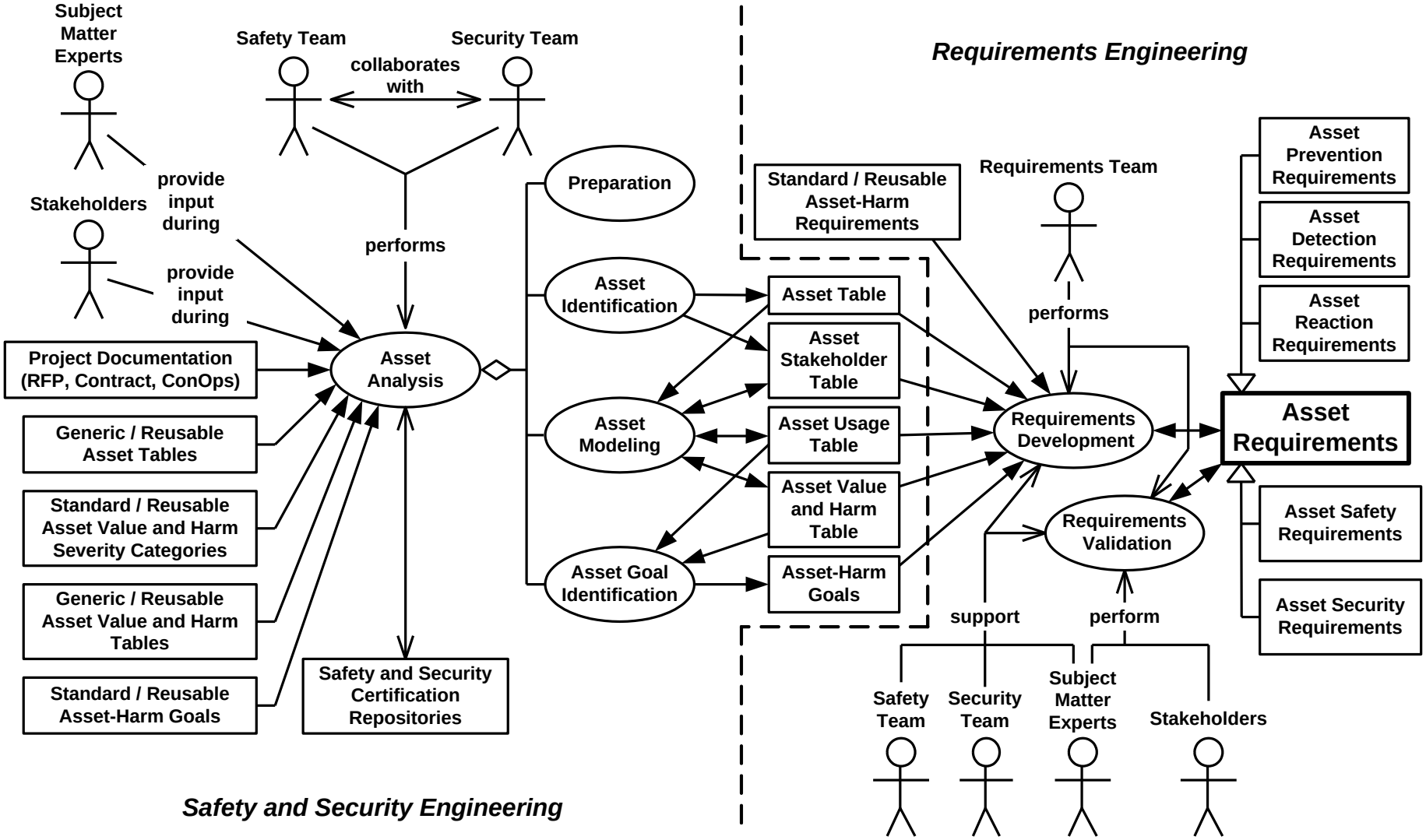
Defensibility Analysis



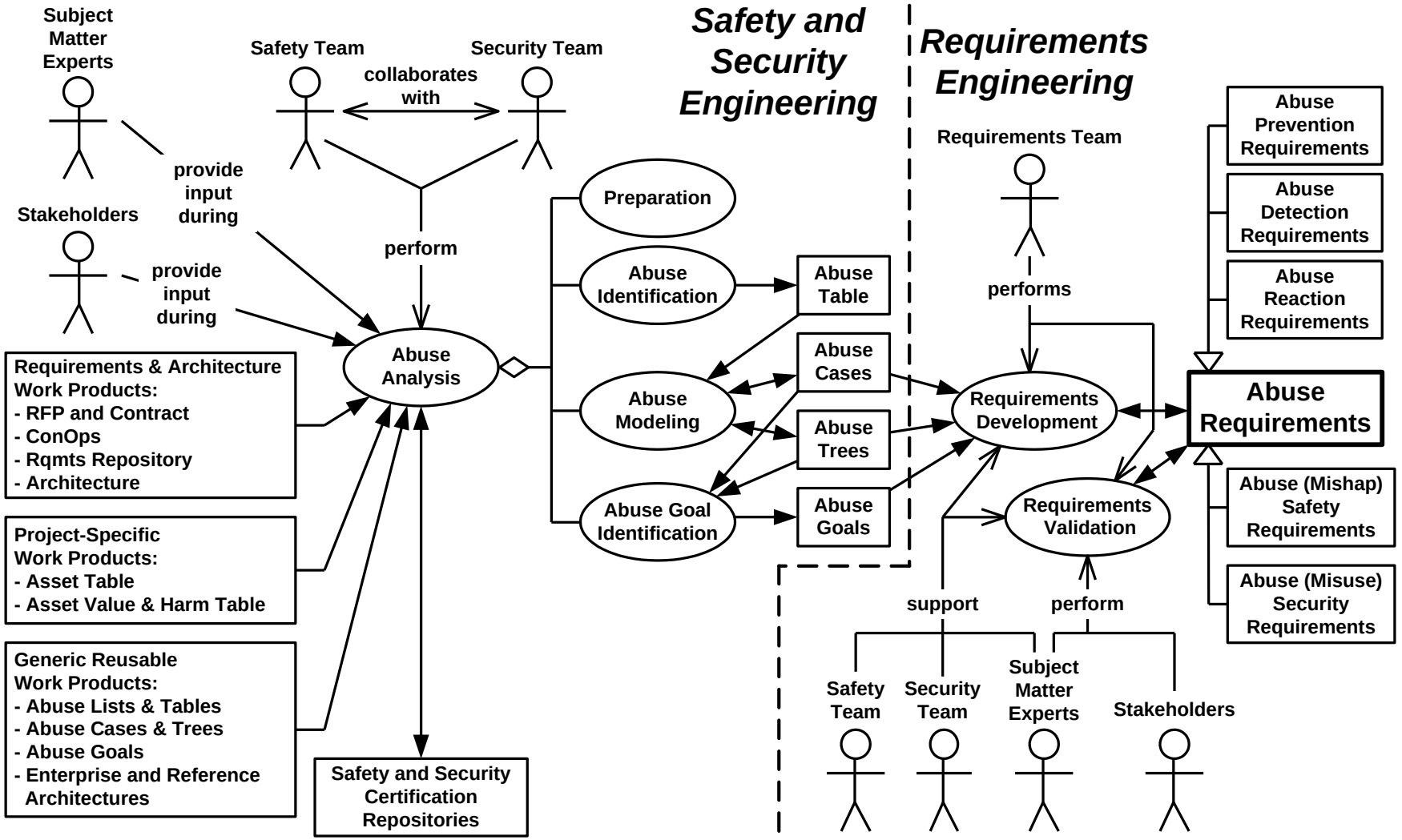
Systems Analysis



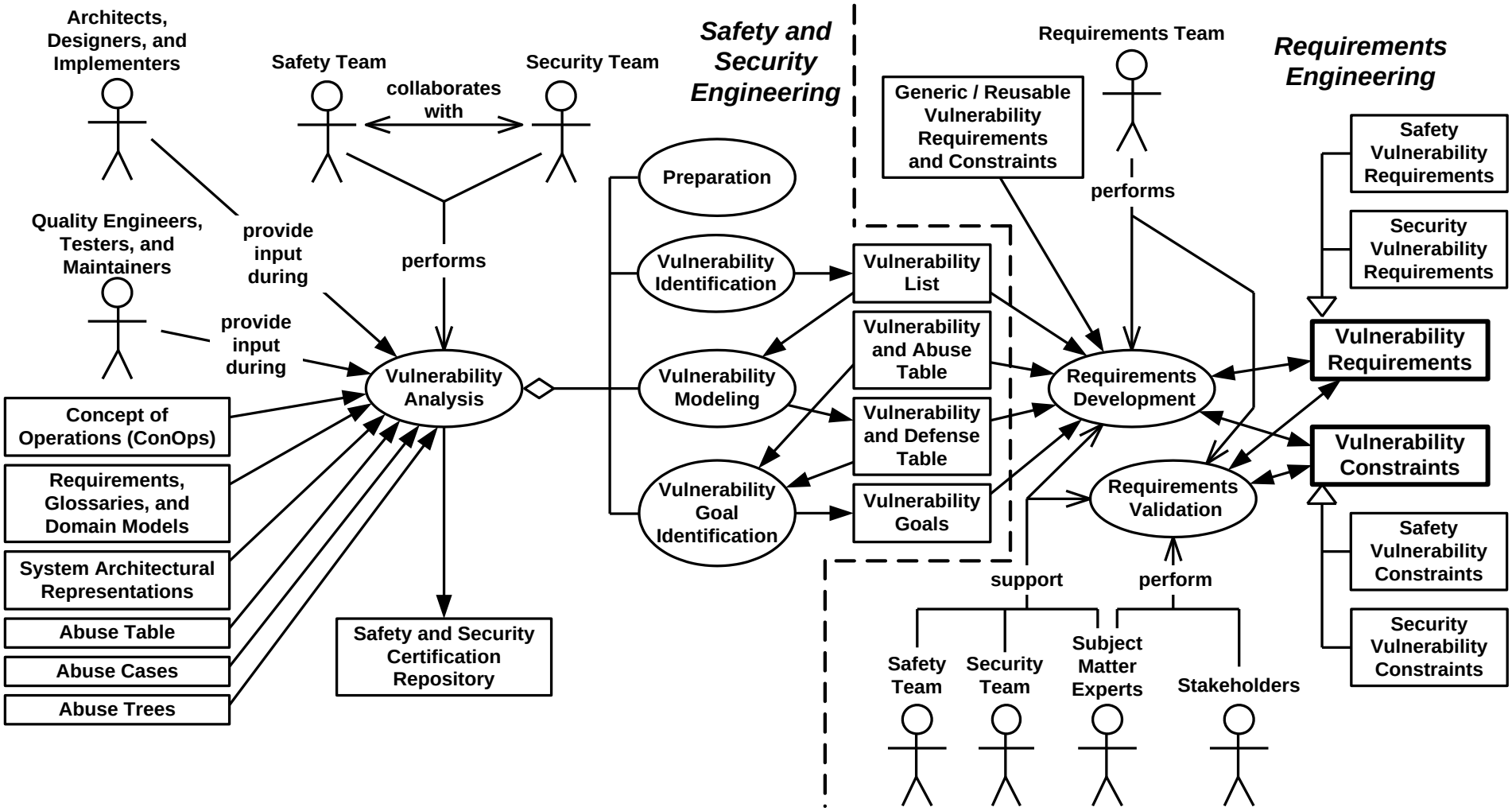
Asset Analysis



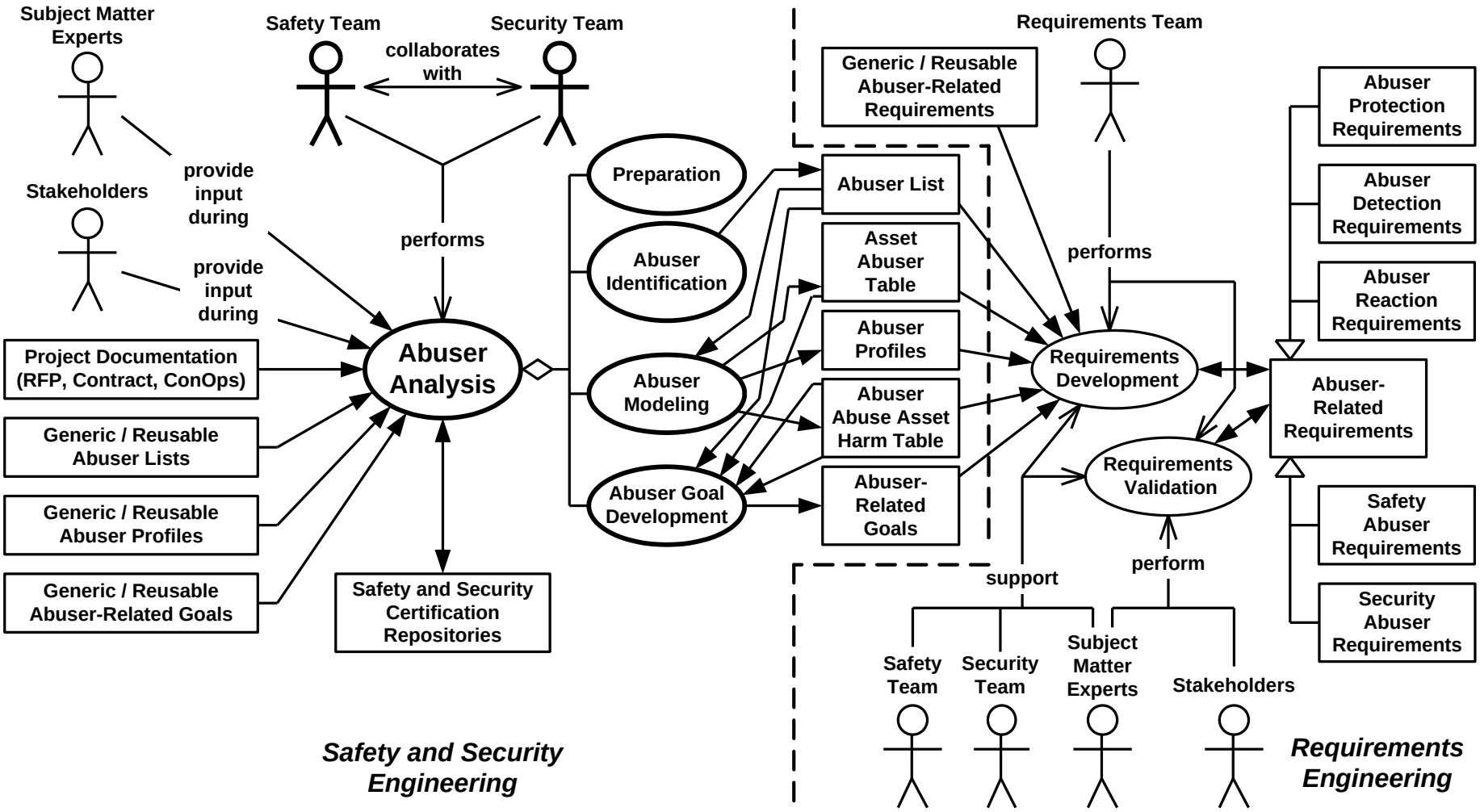
Abuse (Misuse and Mishap) Analysis



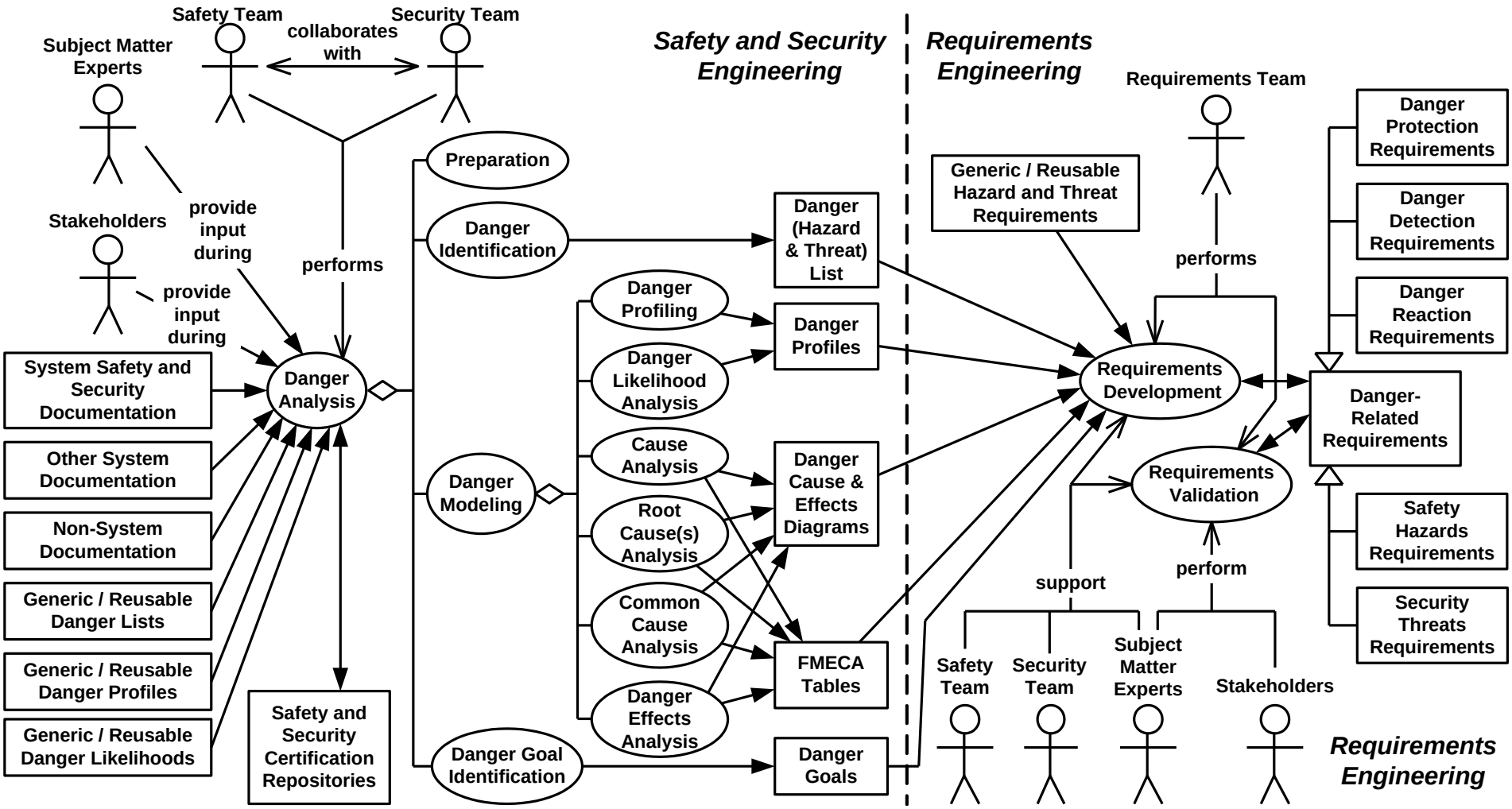
Vulnerability Analysis



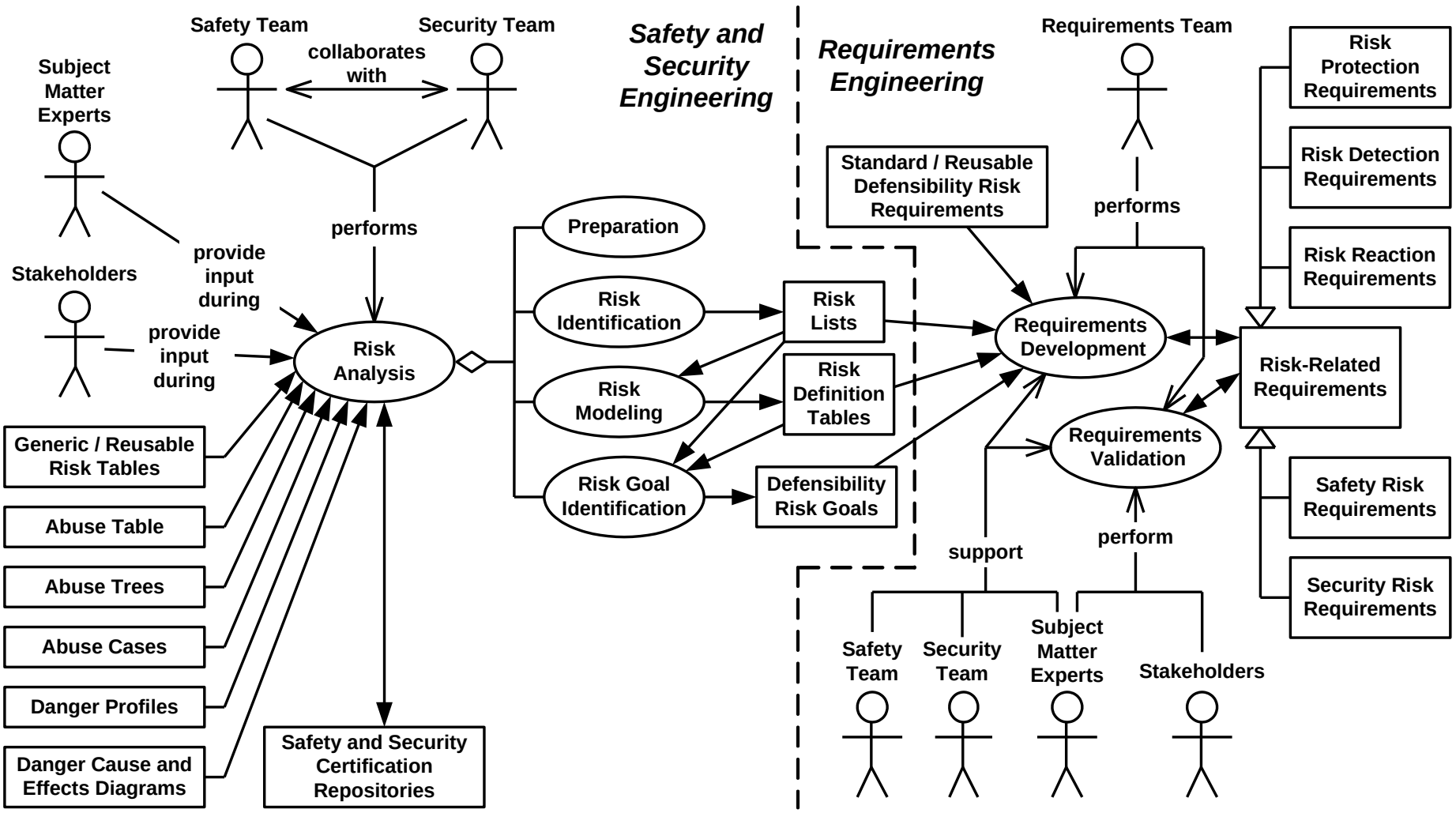
Abuser Analysis



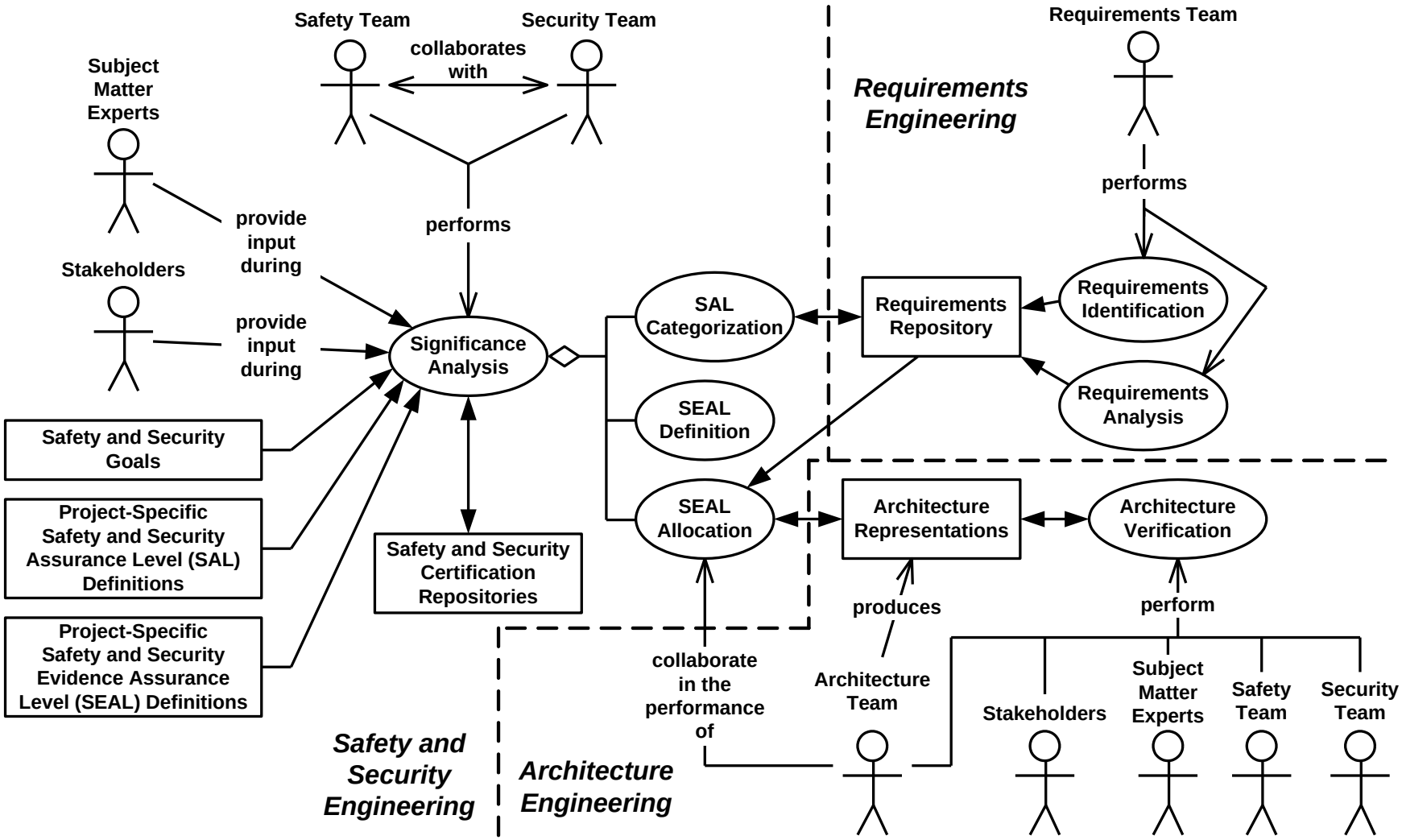
Danger Analysis



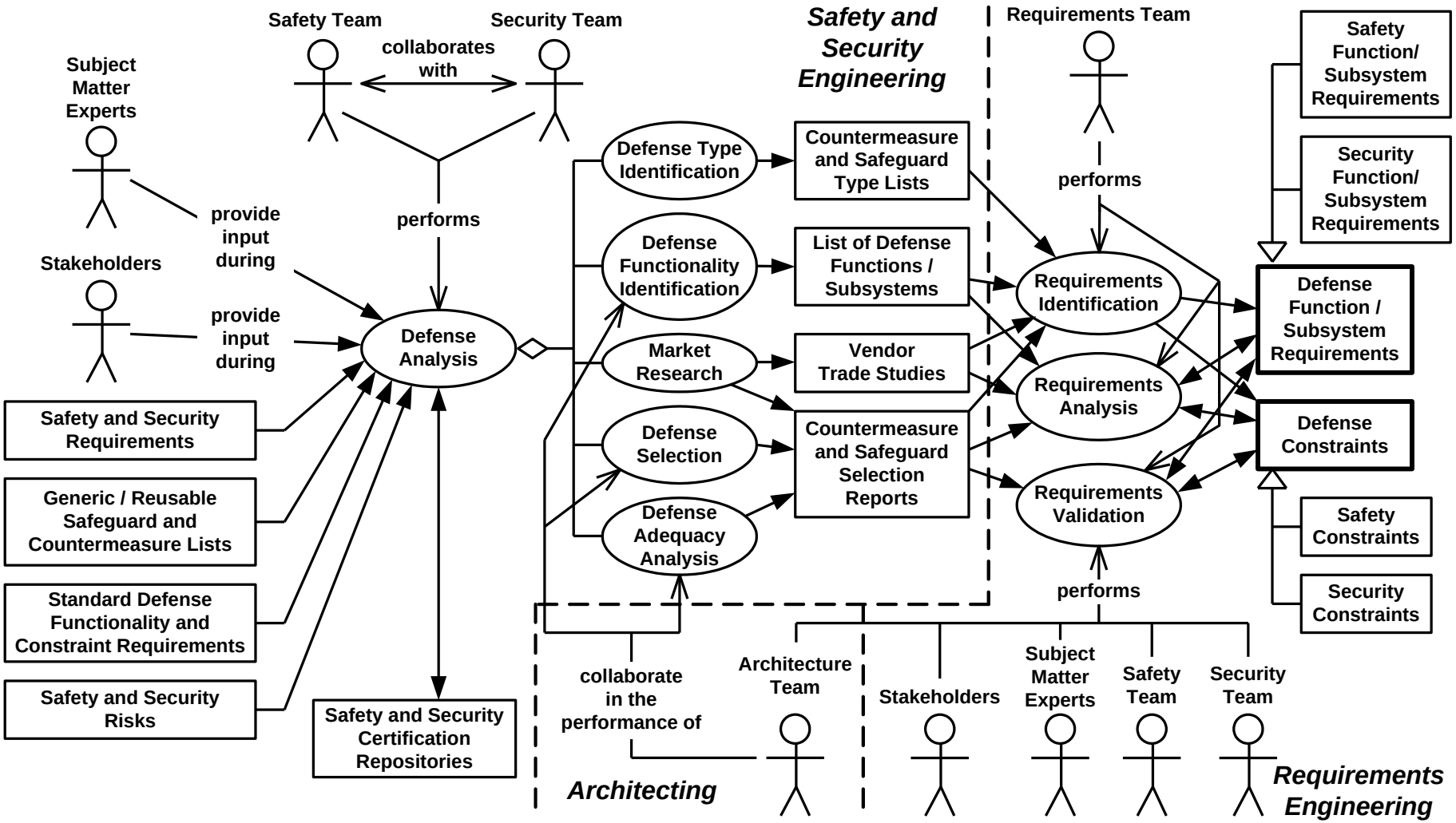
Defensibility Risk Analysis



Defensibility Significance Analysis



Defense Analysis



Conclusion:

Process Improvement Recommendations

Process Improvement Recommendations

Ensure close Collaboration among Safety, Security, and Requirements Teams.

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 proper Properties.

Any Questions?

For more information, contact:

Donald Firesmith
Acquisition Support Program (ASP)
Software Engineering Institute (SEI)
Pittsburgh, Pennsylvania, USA 15213
412-268-6874
dgf@sei.cmu.edu