The Challenges of Engineering Safety- and Security-Related Requirements

System Engineering Challenges Workshop
RuSEC'2010 Moscow, 23-24 September 2010

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Donald Firesmith
23 September 2010

Contents

Related Disciplines:

Requirements, Safety, Security, and Survivability Engineering

Defensibility Engineering

Safety Engineering

the systems engineering discipline 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 properly reacting to such harm, mishaps (i.e., accidents and incidents), vulnerabilities, abusers, hazards, and safety risks

Security Engineering

the systems engineering discipline 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 properly reacting to such harm, misuses (i.e., attacks and incidents), vulnerabilities, *civilian* abusers, threats, and security risks

Survivability Engineering

the systems engineering discipline 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 properly reacting to such harm, misuses (i.e., attacks and incidents), vulnerabilities, *military* abusers, threats, and survivability risks

Similarities and Differences

Similarities:

- *Systems* engineering rather than *software* engineering
- Lower risks to *acceptable* levels:
  - Never zero – never perfectly safe, secure, or survivable
- Prevention, *detection*, and *reaction*:
  - Unauthorized harm to valuable assets (people, property, environment, services)
  - Abuses (safety mishaps and security misuses)
  - Vulnerabilities
  - Abusers
  - Dangers (safety hazards and security threats)

Differences:

- Unintentional vs. intentional (malicious) abusers
- Civilian vs. military abusers
- Terminology

Requirements Engineering

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)

Safety- and security-related requirements are primarily system-level requirements.

System Engineering 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 *work products*

Some of these differences are unnecessary and represent detrimental redundancies:

- Underlying concepts and terminologies
- Tasks and techniques
- Work products (models and documents)

Challenges$_2$

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)

Safety and security are too often tested into existing systems (reactively) rather than adequately built into systems during development (proactively)

Challenges$_3$

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

- *Poor* safety- and security-related requirements that are often:
  - Vague, unverifiable, and infeasible capabilities or goals rather than true requirements
  - Potentially inappropriate architectural and design constraints
  - Inadequate to drive architecture, design, and implementation
  - Specified and managed separately from all other requirements
  - Relegated to “second class” “supplementary” specifications
  - Ignored by requirements engineers and architects
  - Produced too late to drive architecture and testing

Challenges$_4$

"Erroneous specification is a major source of defects and subsequent failure of safety-critical systems. Many failures occur in systems using software that is perfect, it is just not the software that is needed because the specification is defective."

> John C. McKnight, "Software Challenges in Aviation Systems," *Proceedings of the 21st International Conference on Computer Safety, Reliability and Security*, 2002

"Software-related accidents almost always are due to misunderstandings about what the software should do."

> Kathryn Anne Weiss, "An Analysis of Causation in Aerospace Accidents," *Proceedings of the 2001 Digital Avionics Systems Conference*, updated 7 September 2004

Challenges$_5$

"Software-related accidents are **usually caused** by flawed requirements. Incomplete or wrong assumptions about the operation of the controlled system can cause software related accidents, as can incomplete or wrong assumptions about the required operation of the computer. Frequently, omitted requirements leave unhandled controlled-system states and environmental conditions."

Nancy G. Leveson, *Safeware: System Safety and Computers*, 2003

"Software-related accidents are **almost all caused** by flawed requirements:

- Incomplete or wrong assumptions about the operation of the controlled system or required operation of the computer
- Unhandled controlled-system states and environmental conditions."

Nancy G. Leveson, *A new Approach to Ensuring Safety in Software and Human Intensive Systems*, July 2009

Challenges$_6$

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

- Current requirements lack practical and precise thresholds.
- Defenses (safeguards and countermeasures) should be commensurate with risk.
- Without well-defined thresholds, how can:
  - Architects perform engineering trade-offs?
  - Testers set test completion criteria for the testing of safety- and security-related requirements?

Many mandated security "requirements" are actually constraints such as:

- Security subsystems
- Industry "best practices"

Challenges$_7$

What about safety and security requirements for preventing, detecting, and reacting to:

- Harm to valuable assets?
- Abuses (safety mishaps and security misuses)?
- Vulnerabilities?
- Abusers (unintentional and intentional)?
- Dangers (safety hazards and security threats)?
- Risks (safety and security)?

Challenges$_8$

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

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 a major “root cause” of many (or most) accidents involving software-intensive systems.

Challenges$_9$

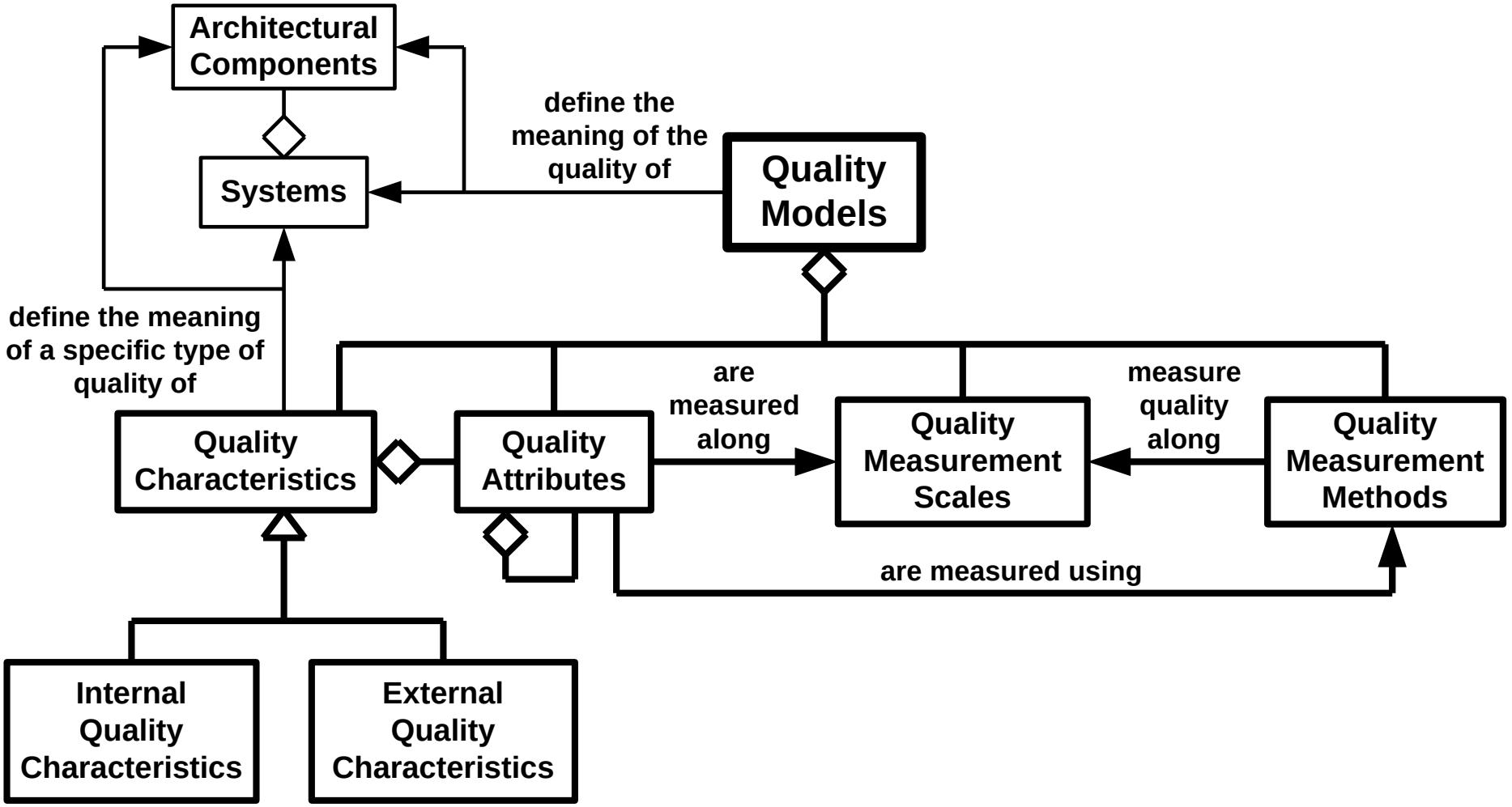
Situation cries out for process improvement:

- 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
- Easier certification and accreditation
- Better safety- and security-related requirements
- Fewer accidents and successful attacks that cause less harm to valuable assets

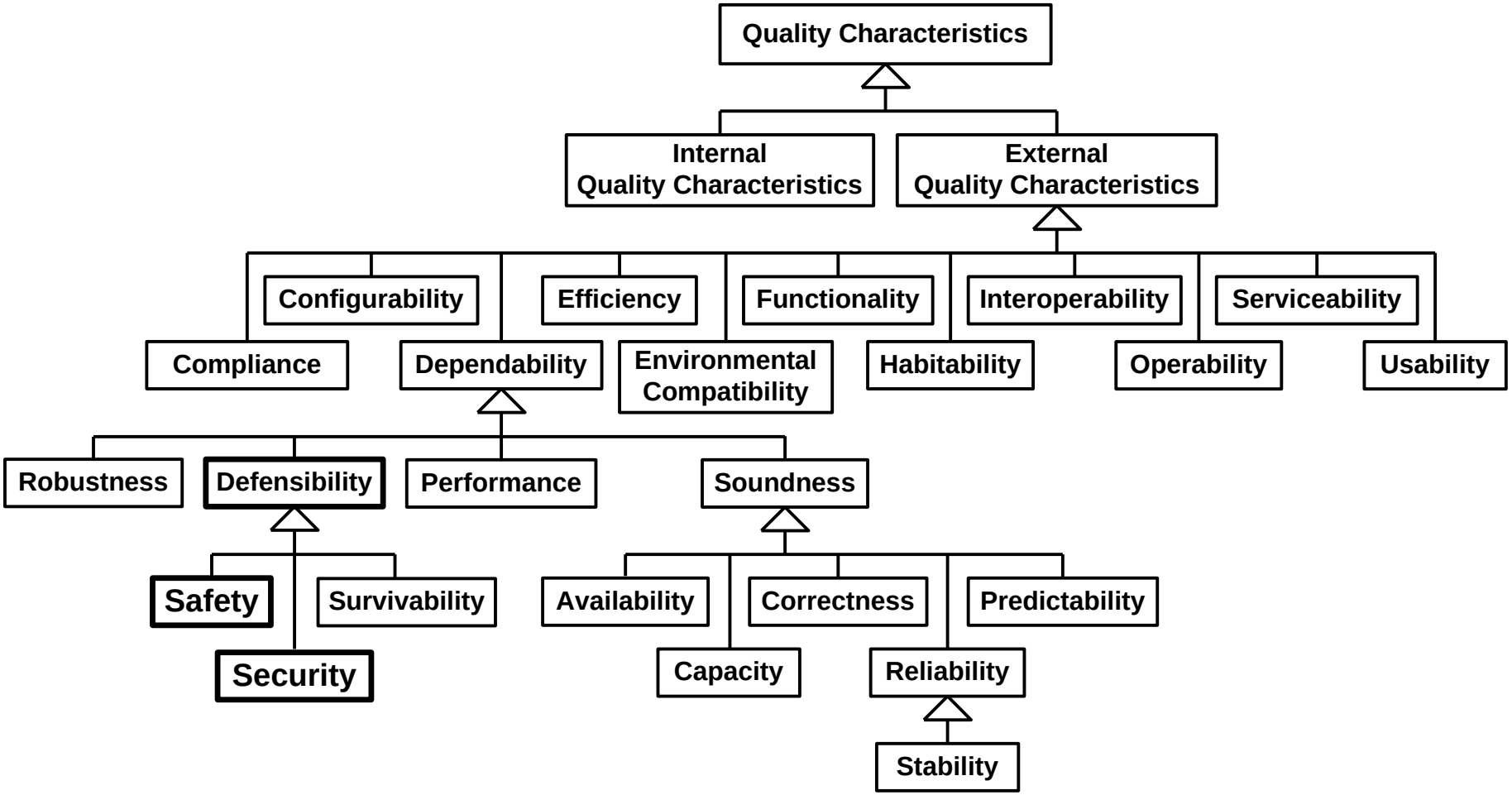
Fundamental Concepts:

A Foundation for Understanding

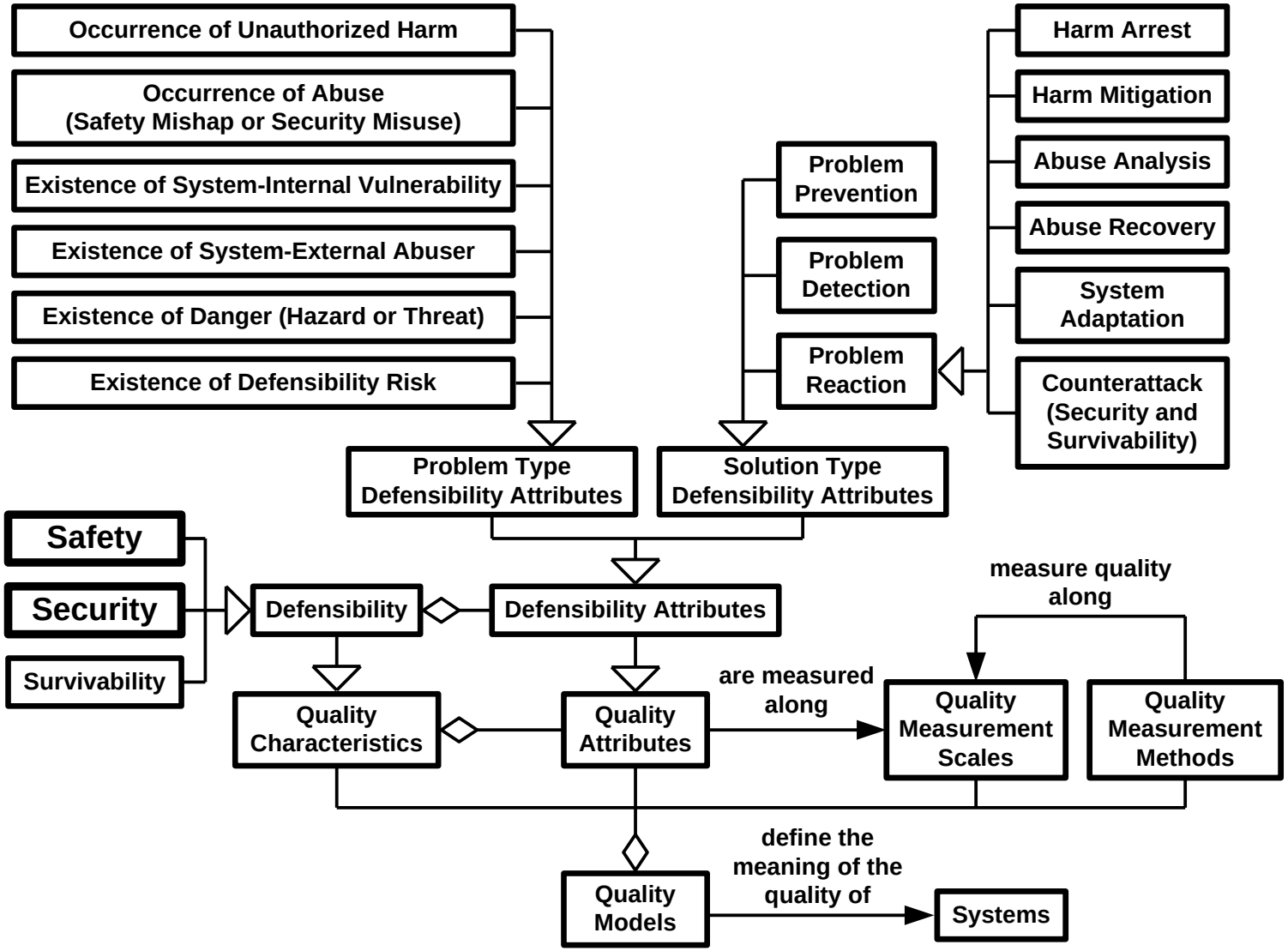
Quality Models



Quality Characteristics (External)



Defensibility Quality Attributes



Defensibility

Defensibility

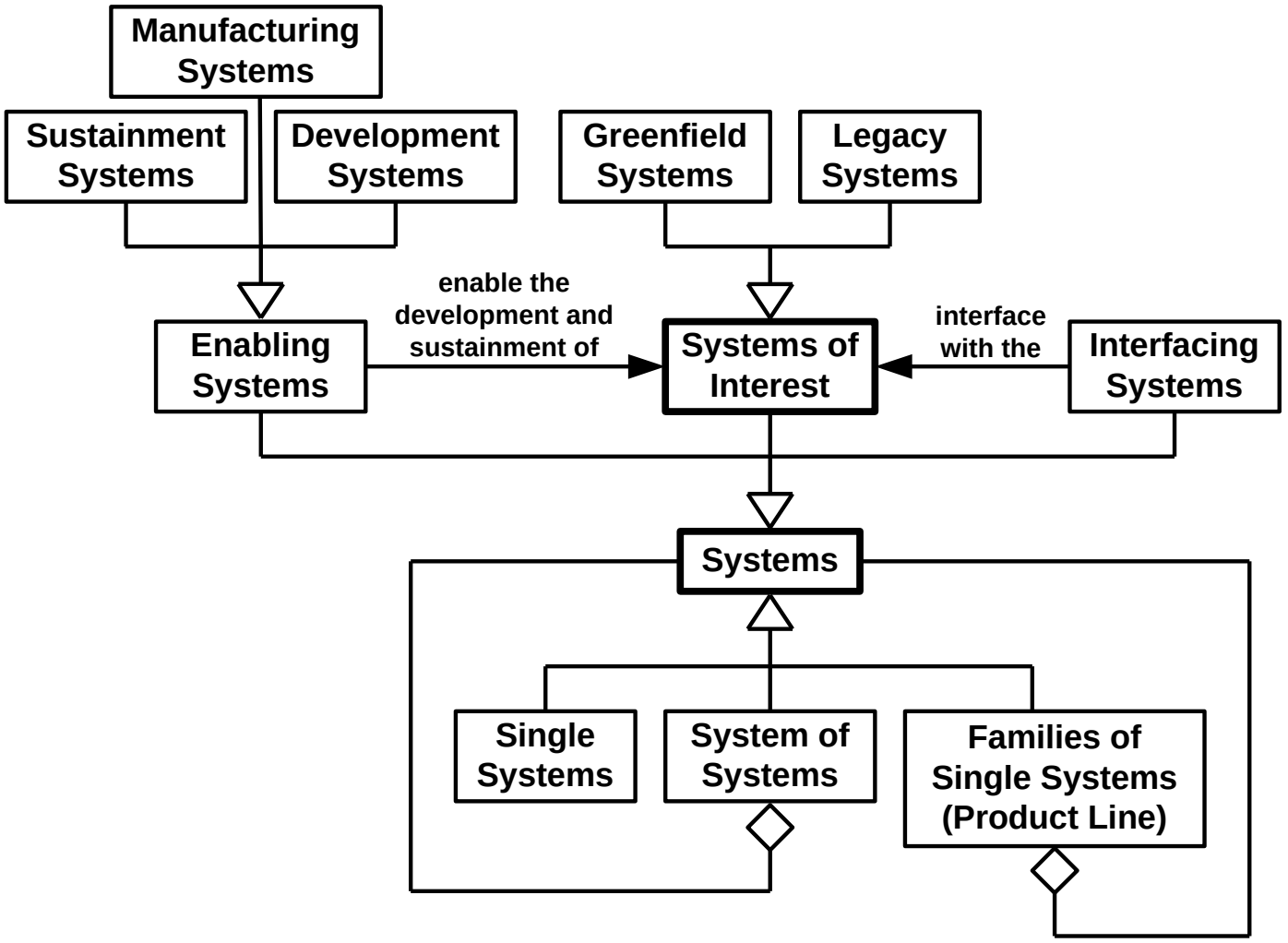
the quality characteristic capturing the degree to which the system:

- Properly prevents, detects, reacts to, and adapts to:
  - Unintended and unauthorized *harm* to *valuable assets*
  - *Abuses (mishaps and misuses)*
  - *Abusers*
  - *Vulnerabilities*
  - *Dangers (hazards and threats)*
- Has *defensibility risks* that are acceptably low to its *stakeholders*

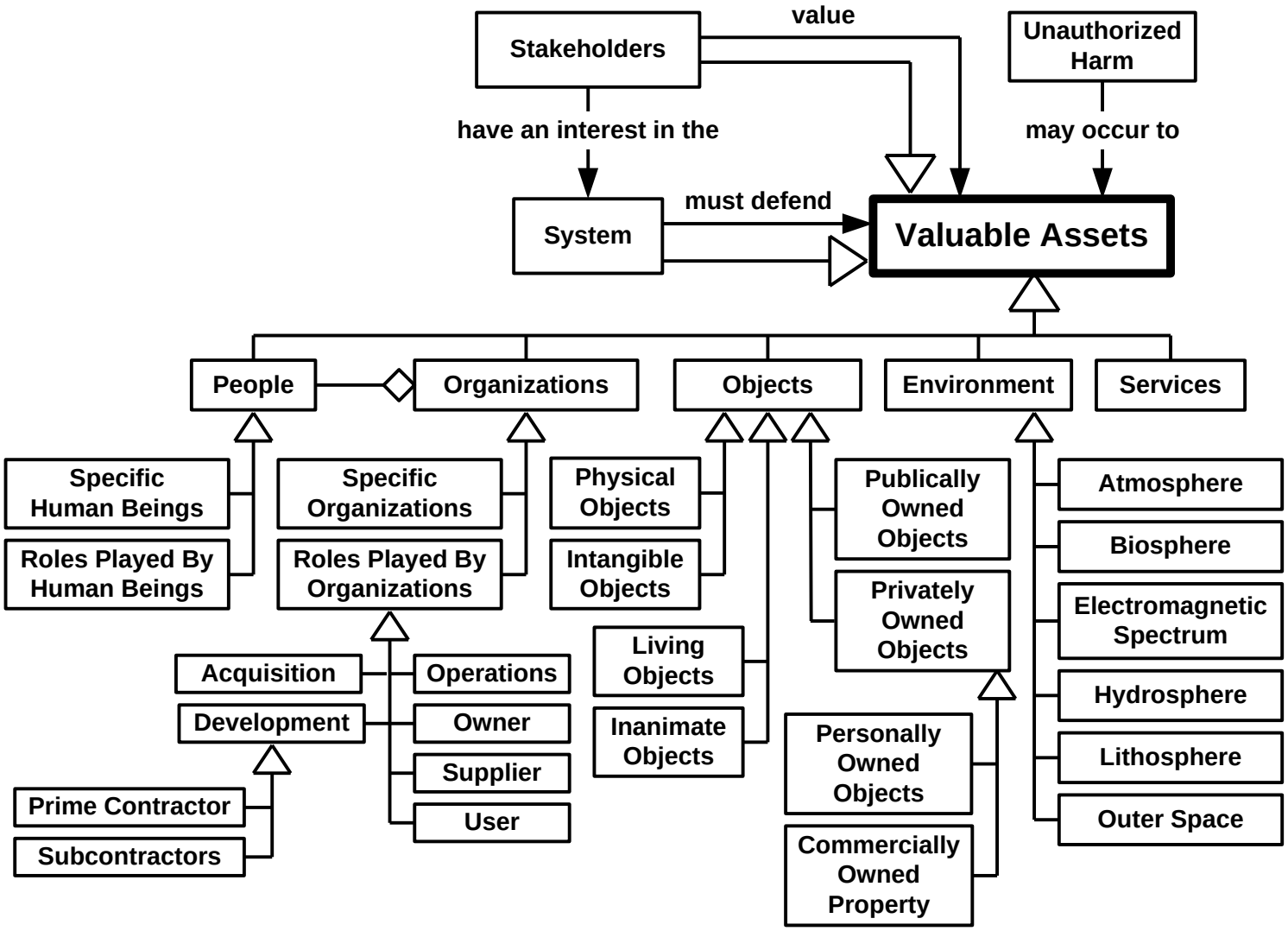
Safety and security are defined in a similar manner by replacing:

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

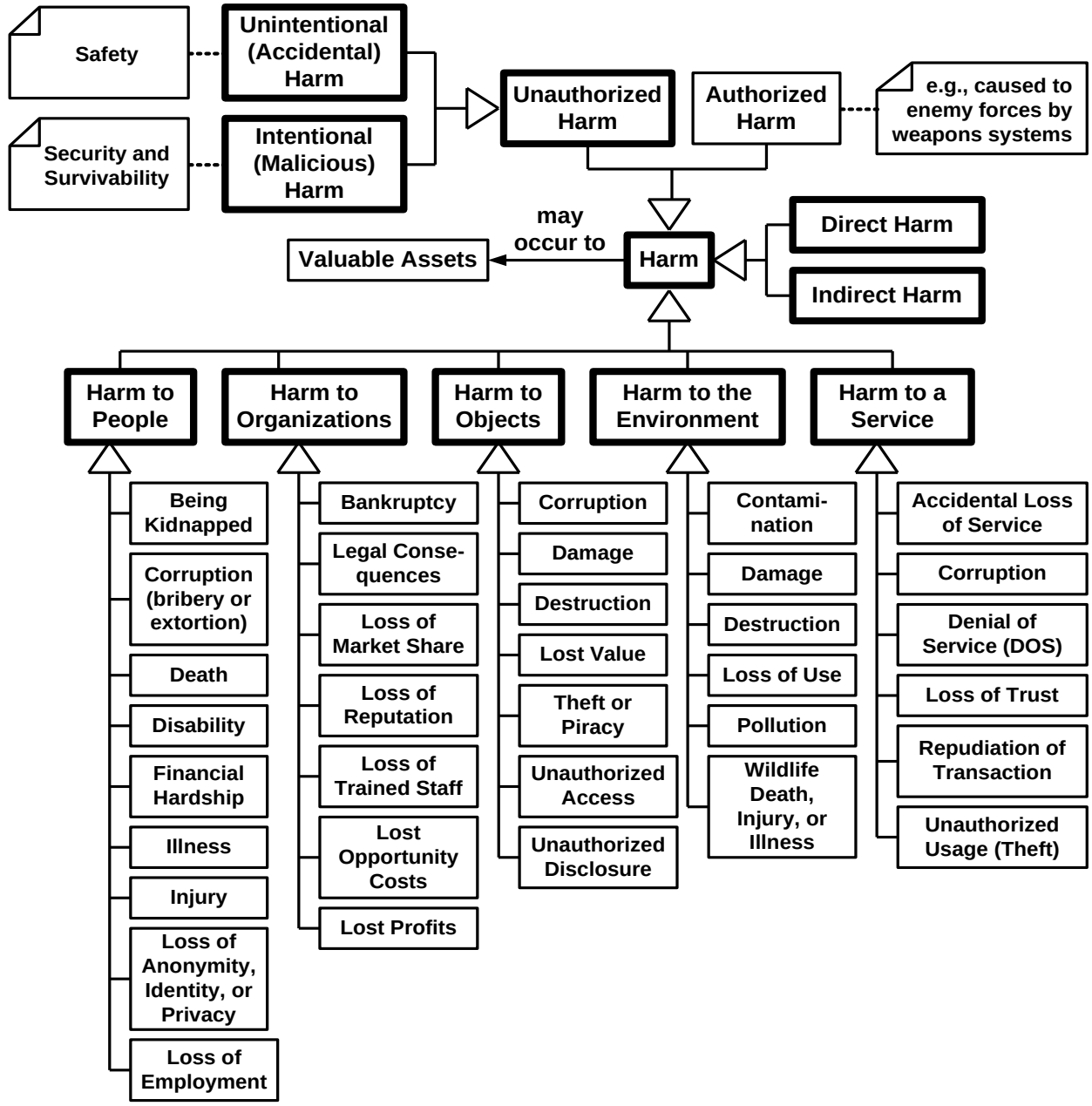
Types of Systems



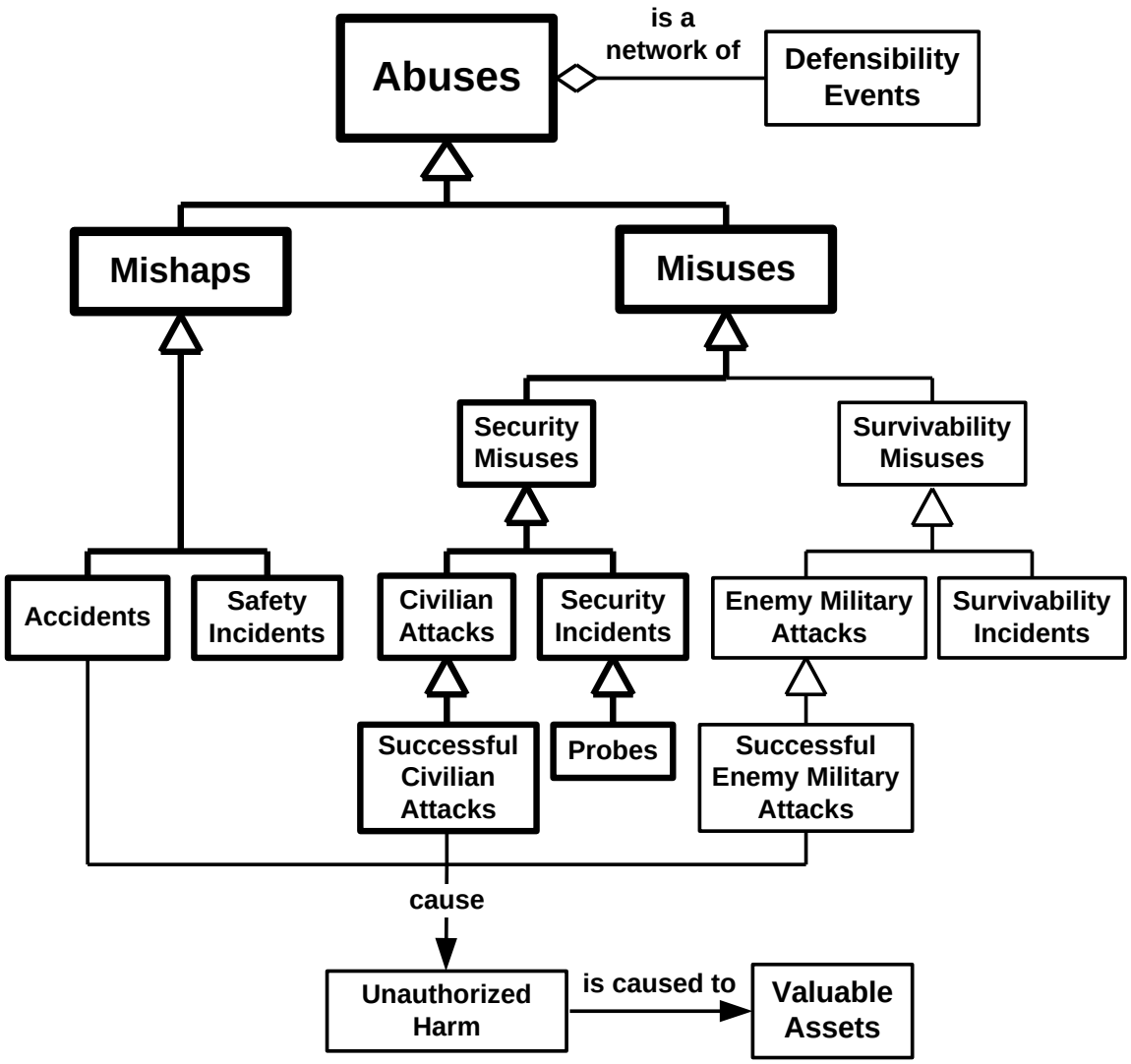
Types of Valuable Assets



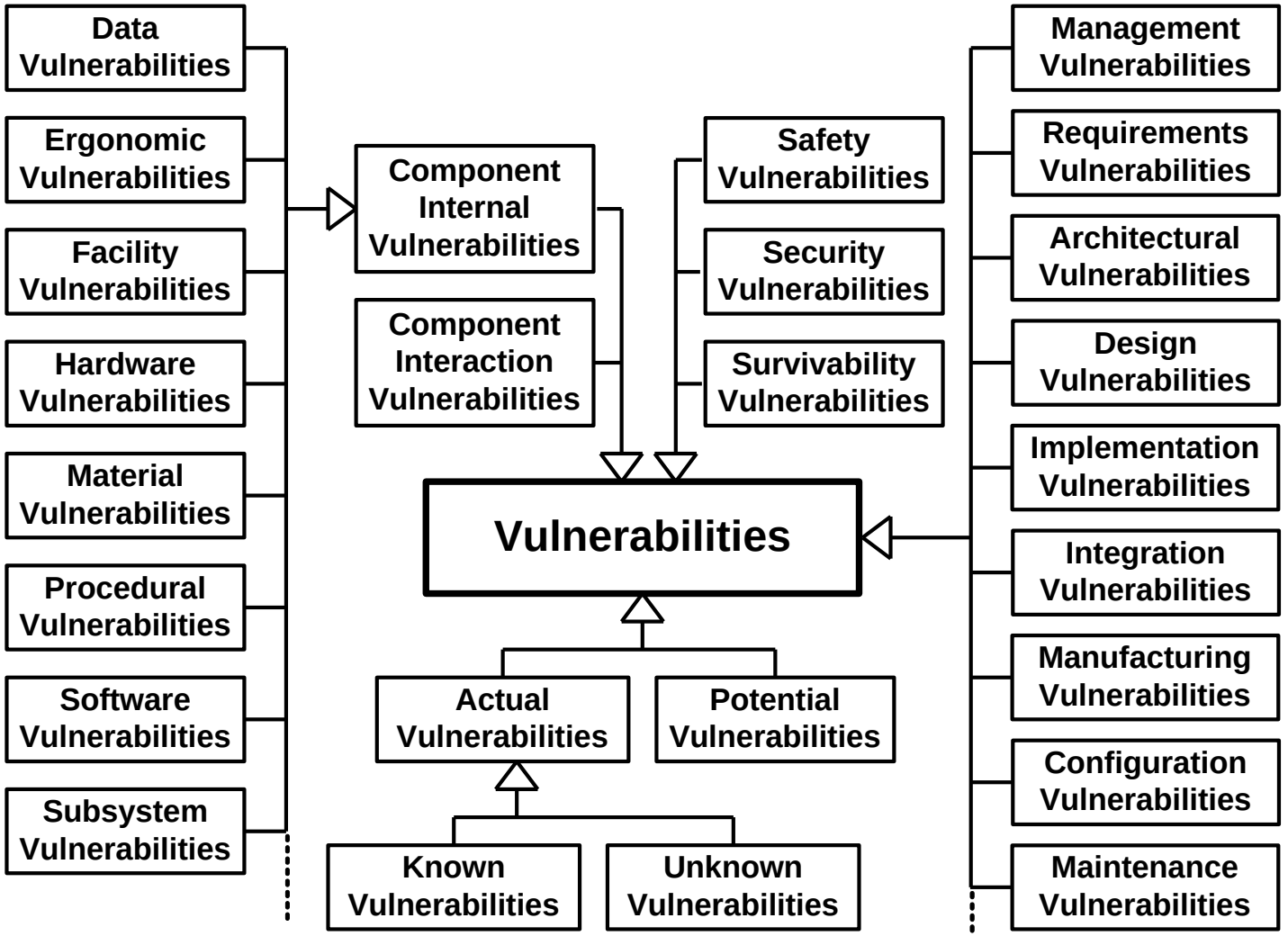
Types of Harm



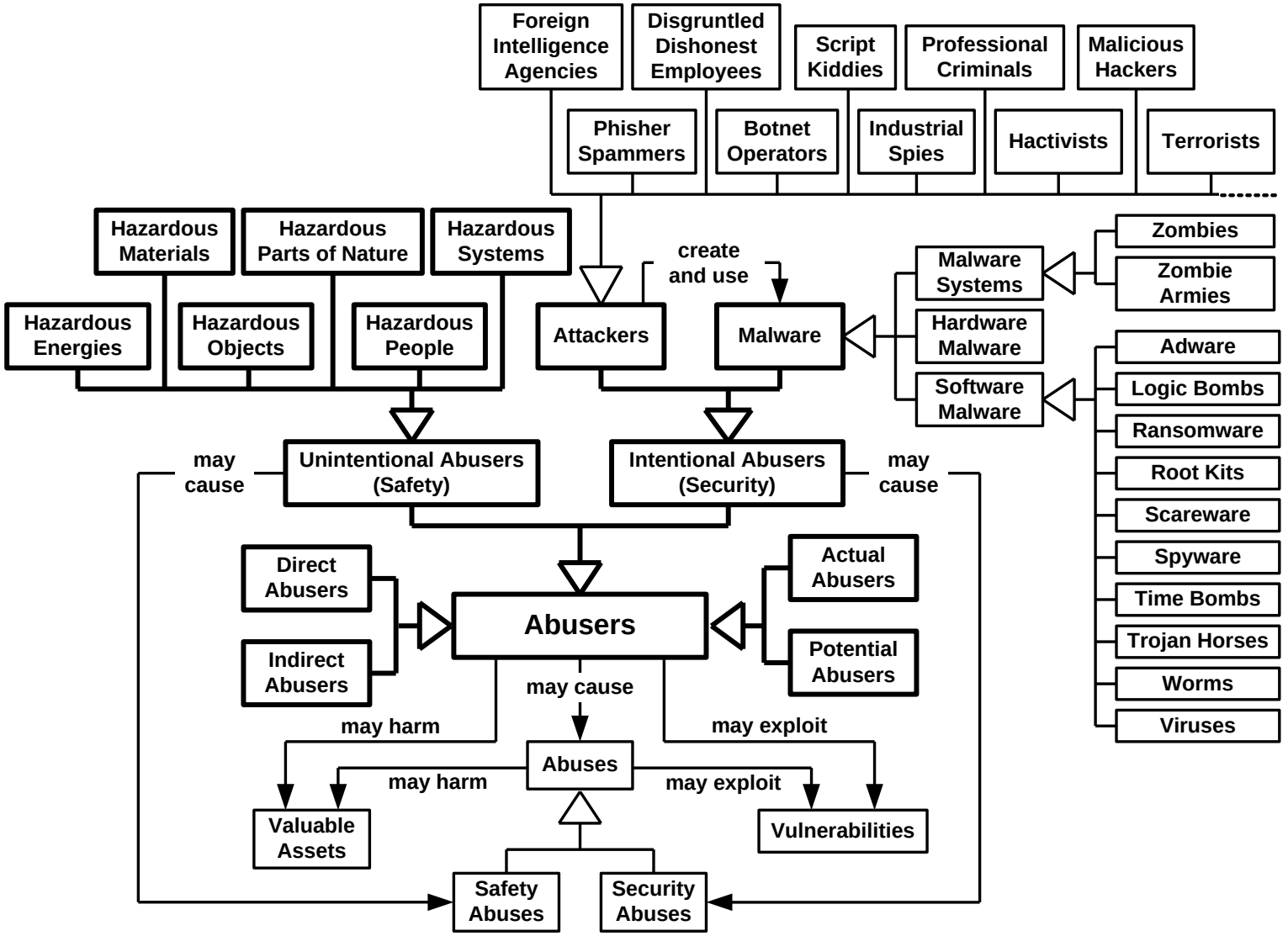
Types of Abuses



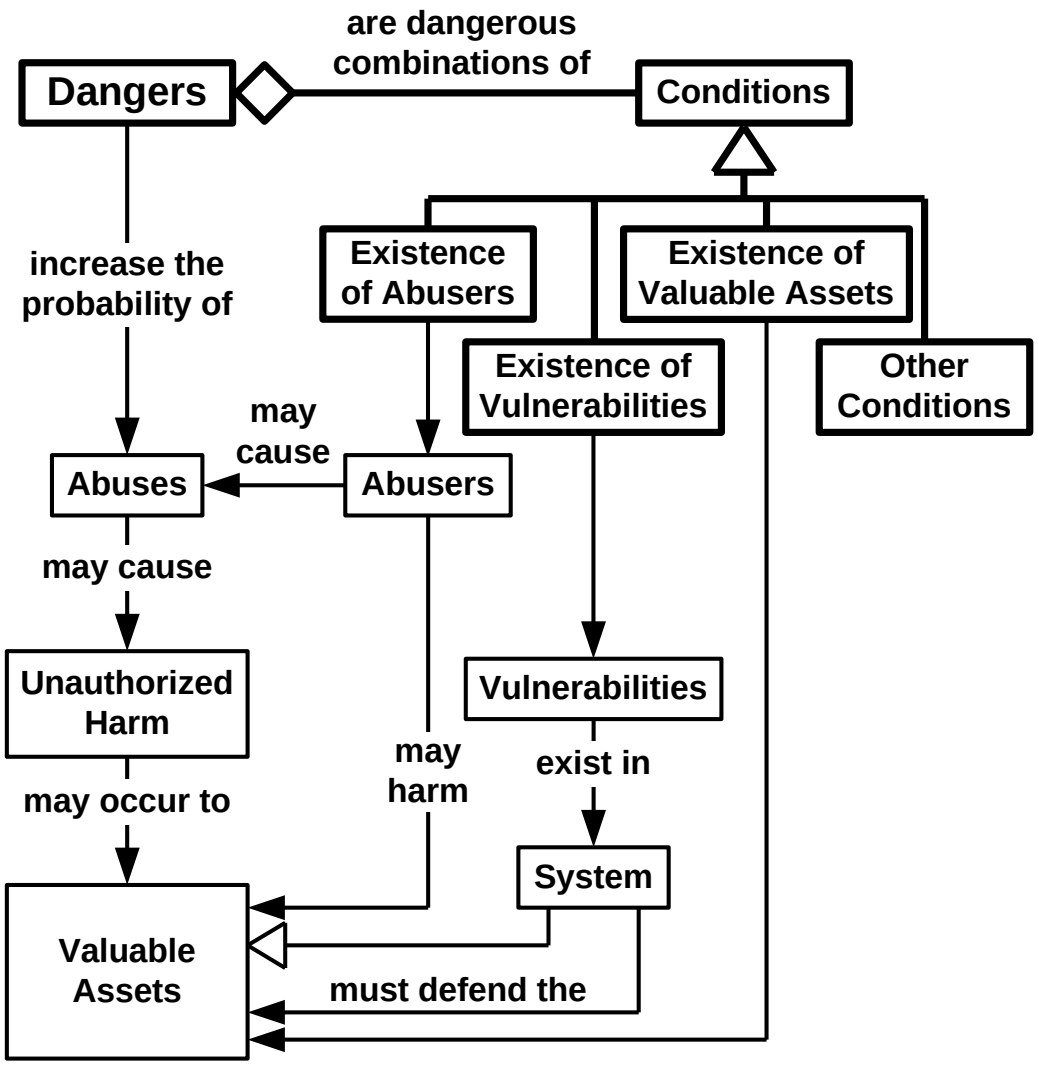
Types of Vulnerabilities



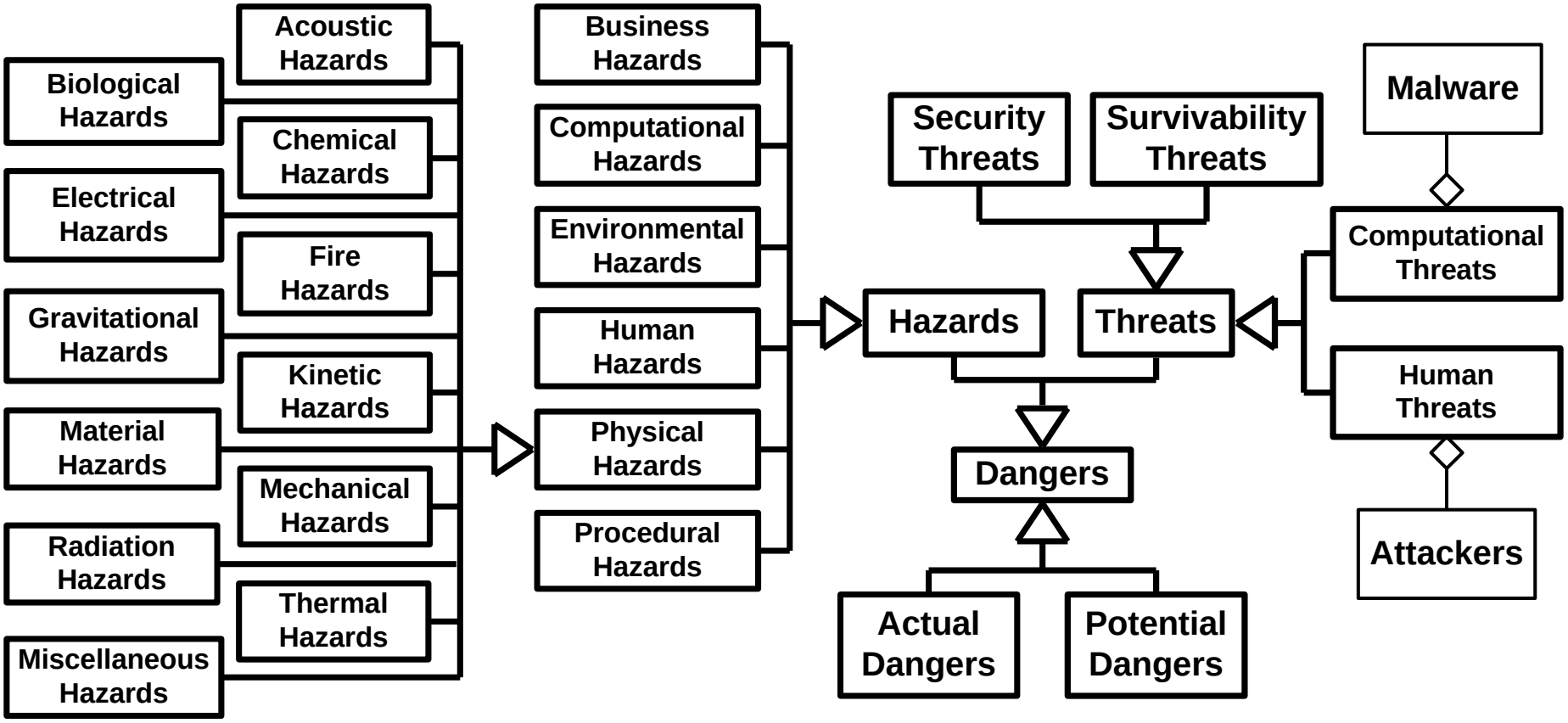
Types of Abusers



Components of Dangers

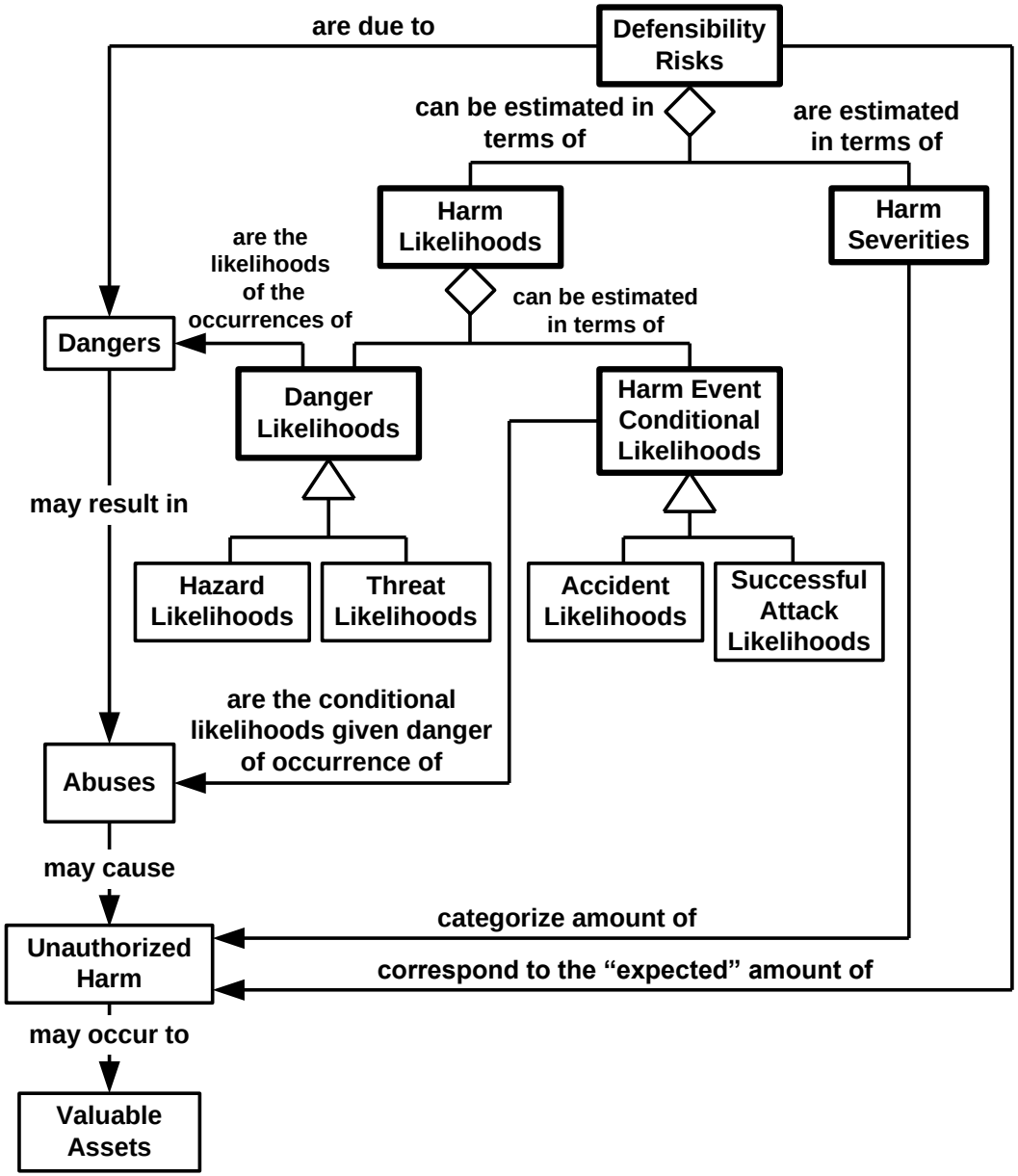


Types of Dangers



Defensibility Risks

Expected amount of harm to valuable assets



Safety- and Security-Related Requirements

Types of Safety- and Security-Related Requirements

Too often only a single type of requirements is considered.

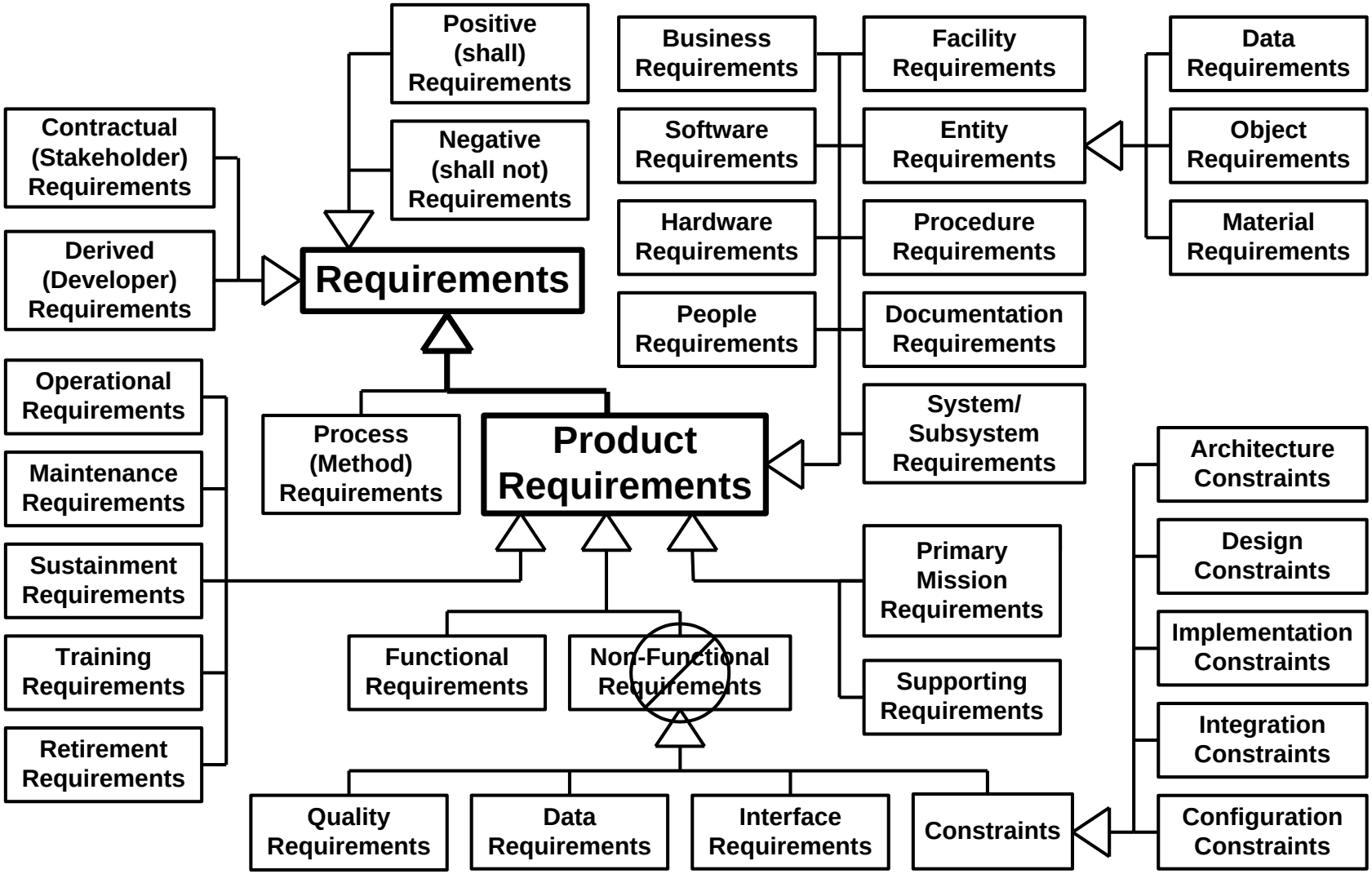
Not just:

- Specific types of non-functional requirements (NFRs):
  - Safety and security requirements are quality requirements
- Functional, data, and interface requirements with safety or security ramifications
- Architecture and design constraints
- Safety and security functions/subsystems
- Software requirements
- Constraints on functional requirements

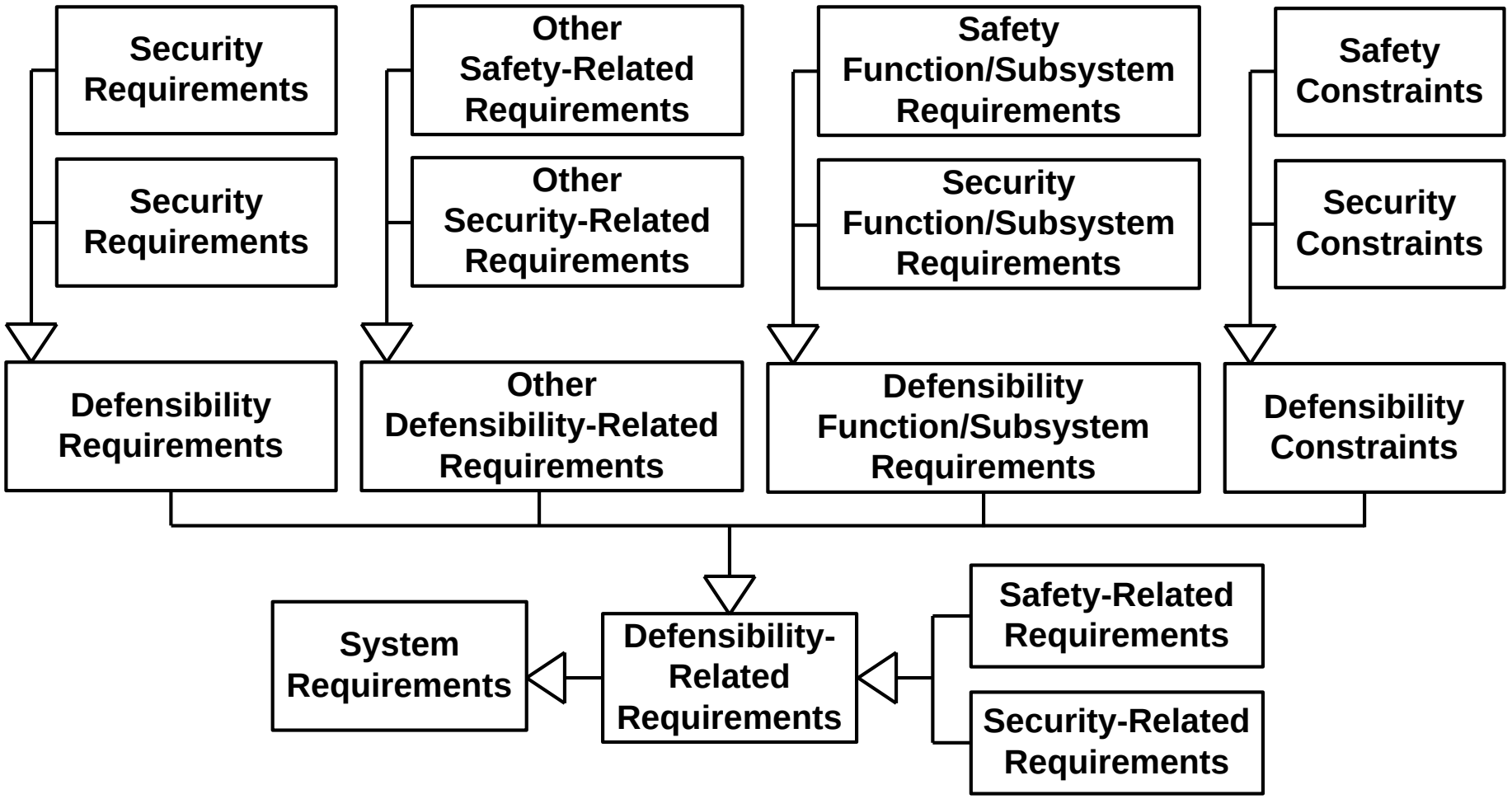
Reason for presentation title:

Safety- and security-related requirements for software-intensive systems

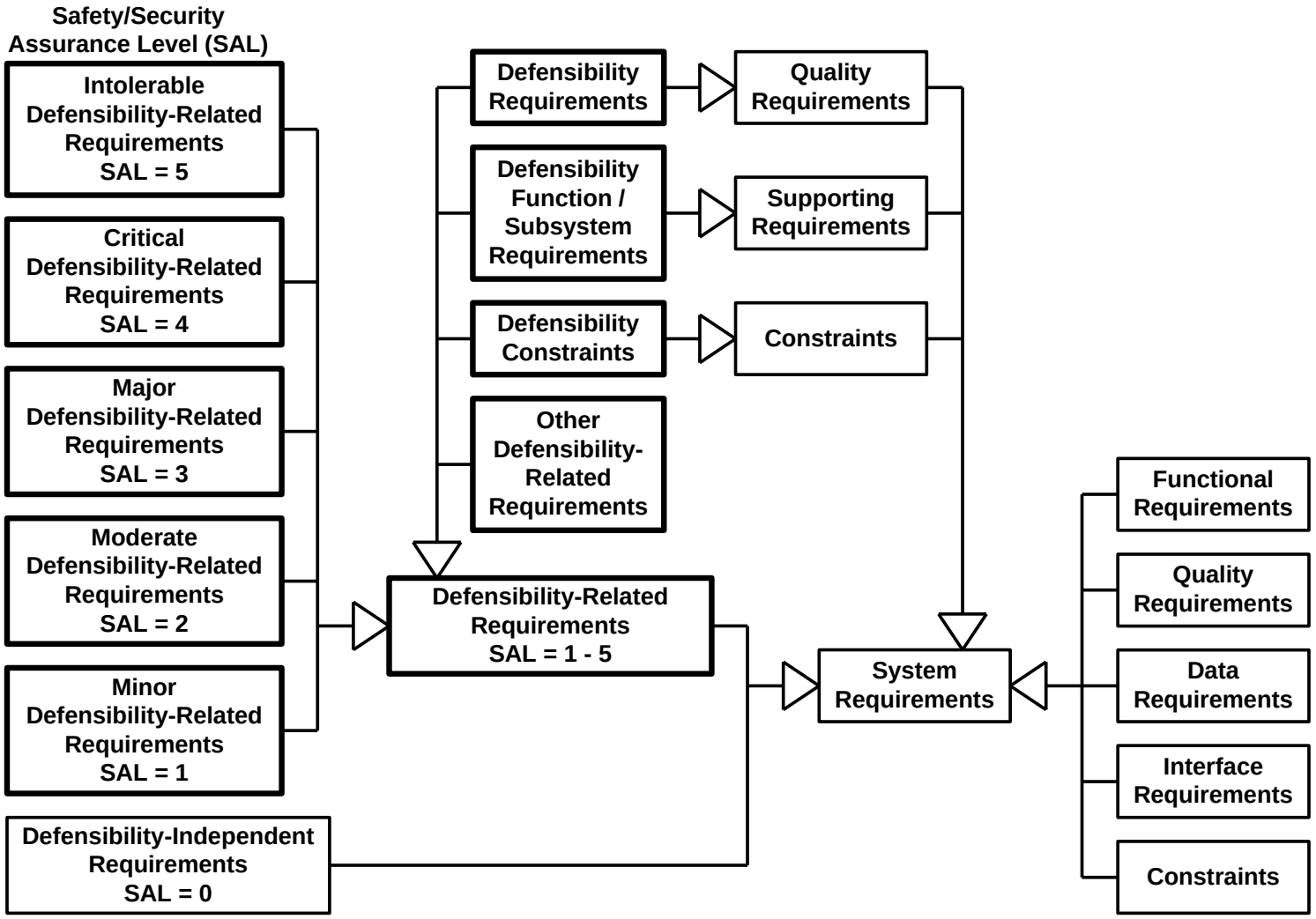
Types of Requirements



Types of Defensibility-Related Requirements



Four Types of Defensibility-Related Requirements



Example Safety- and Security-Related Requirements

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."

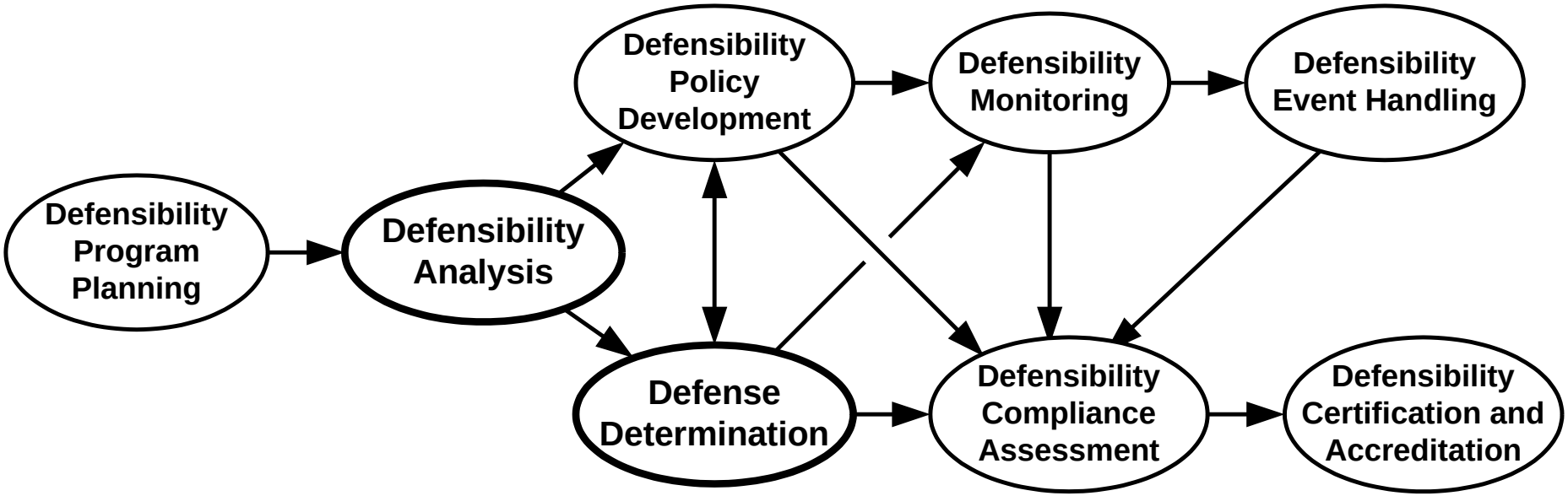
Safety / Security Constraint

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

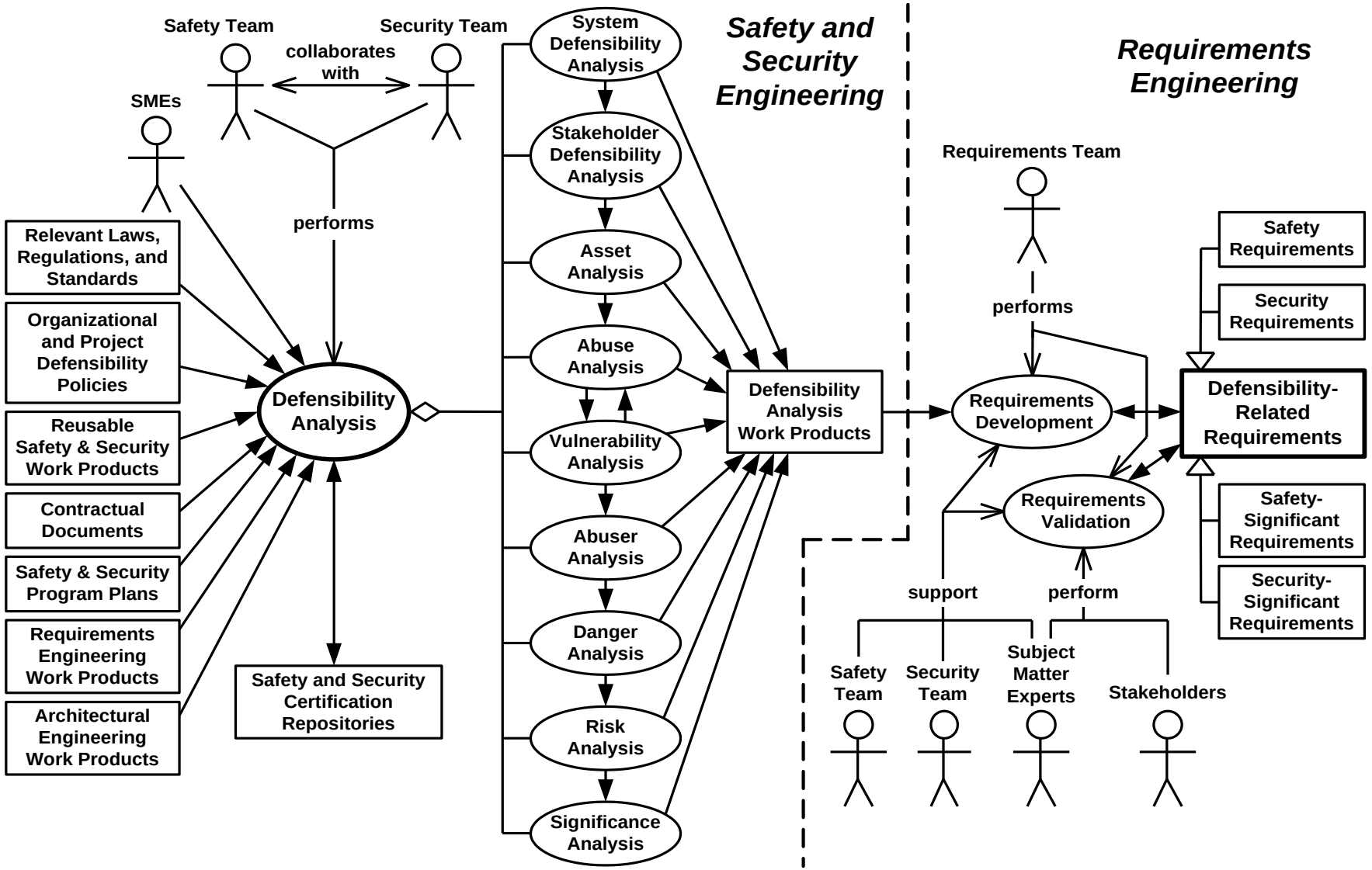
Consistent Common Method:

A Basis for Effective Collaboration

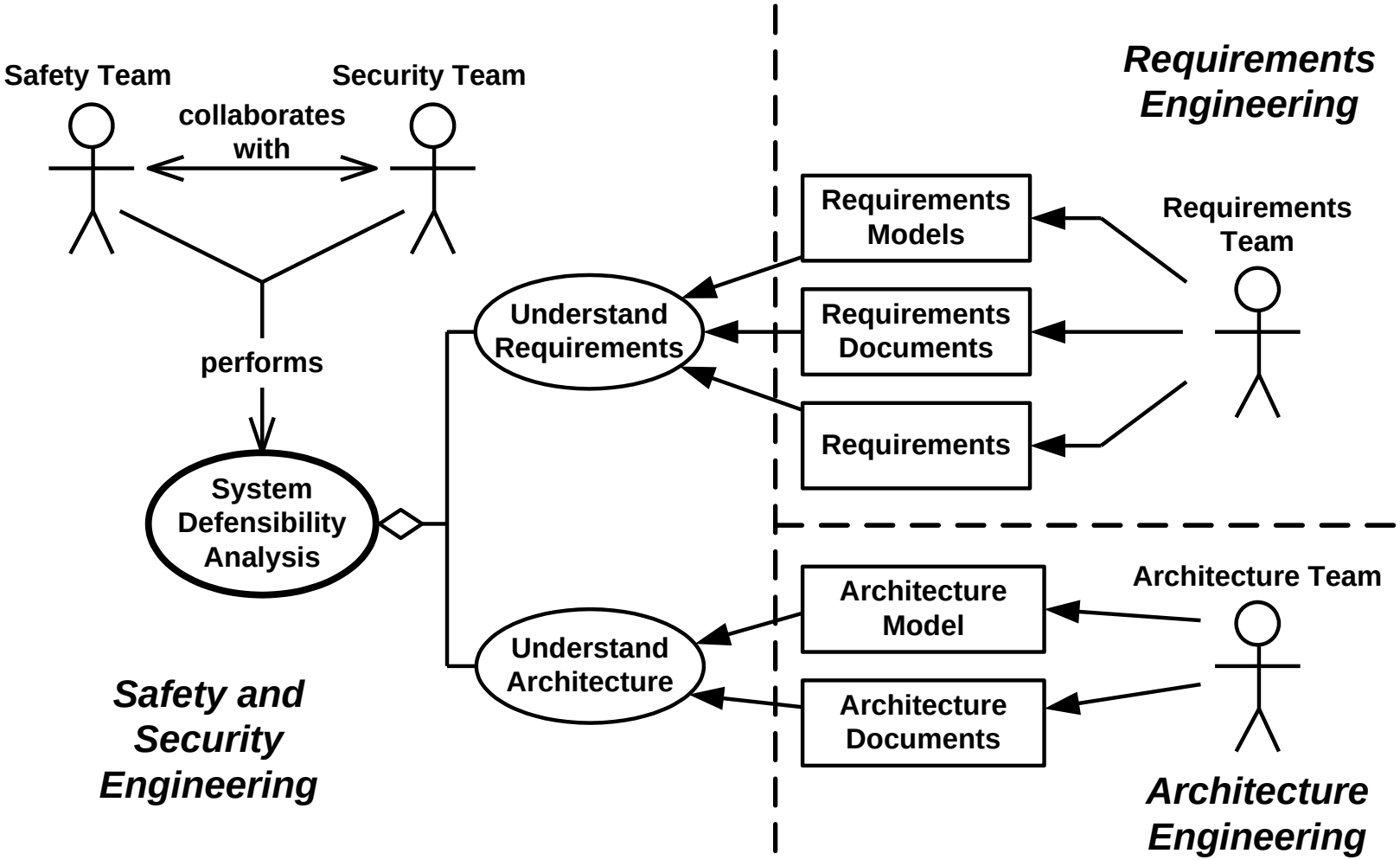
Overall Defensibility Engineering Method



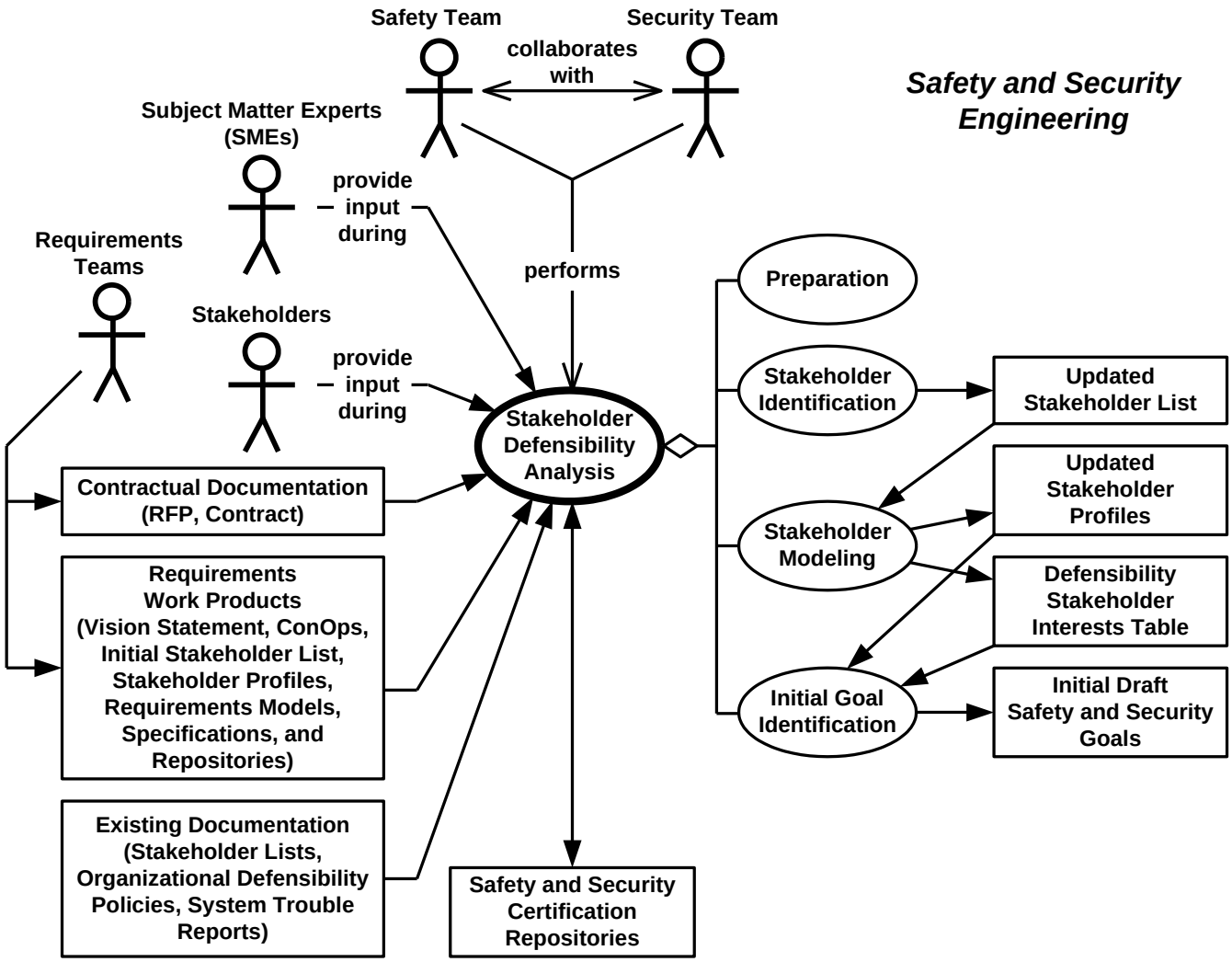
Defensibility Analysis



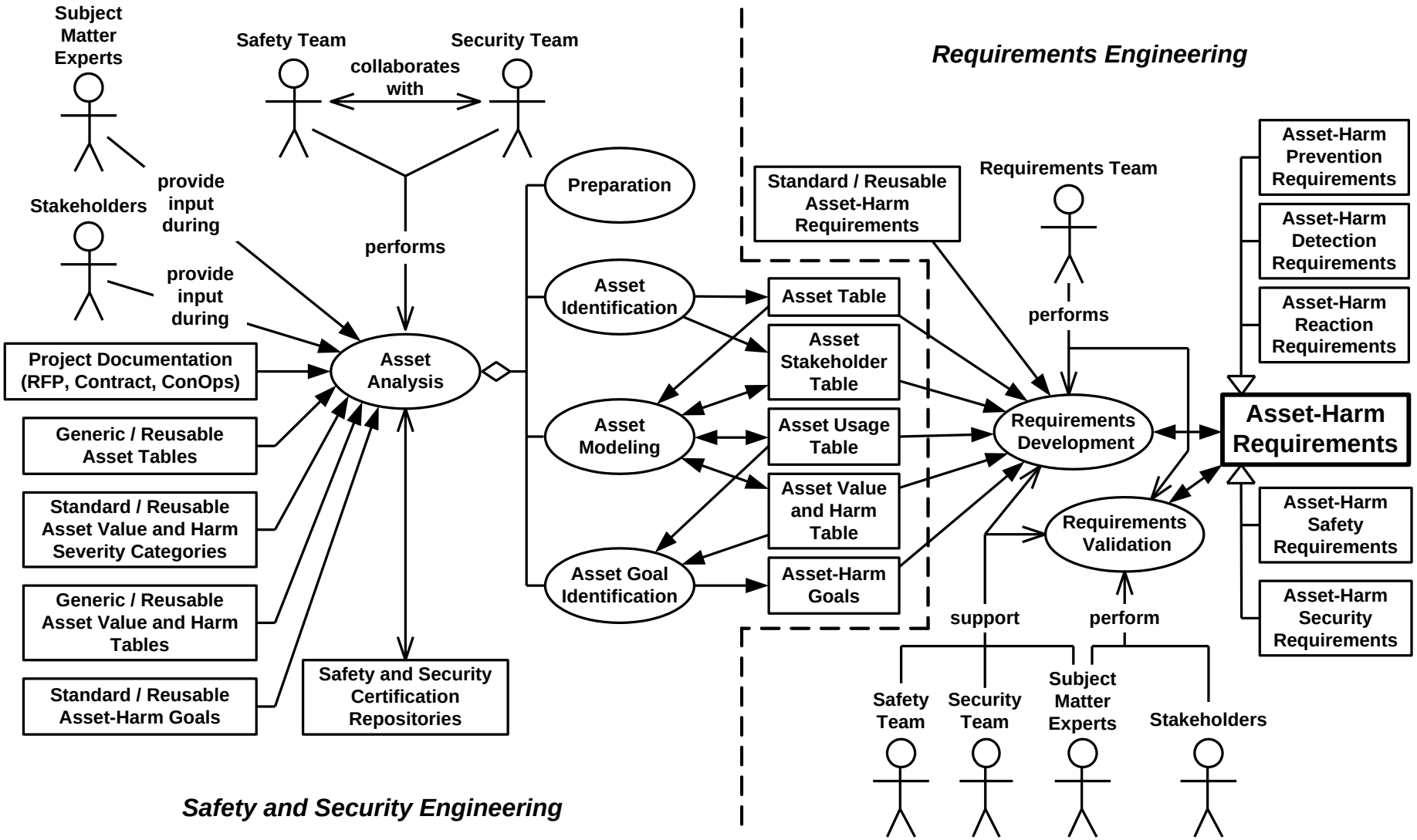
System Defensibility Analysis



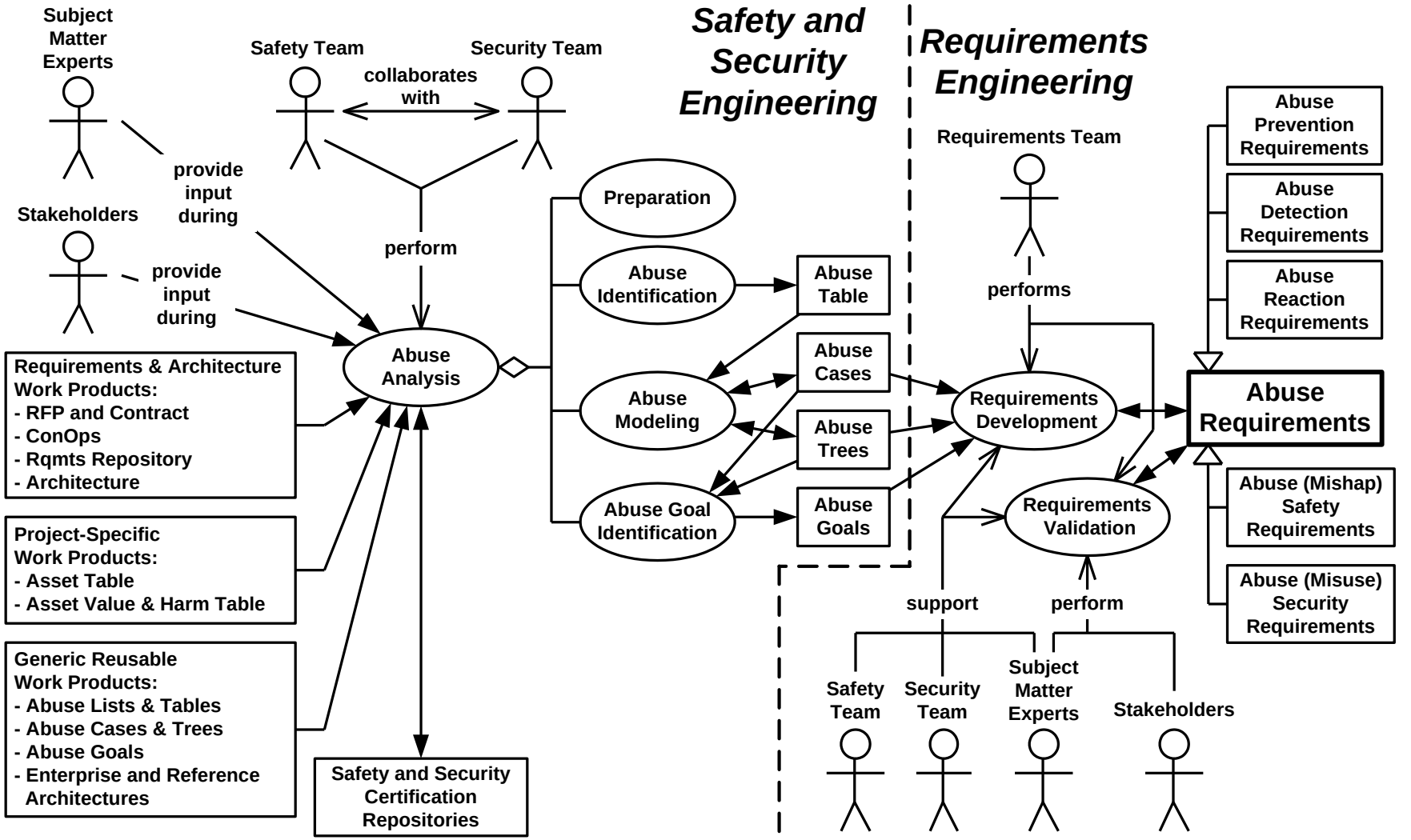
Stakeholder Defensibility Analysis



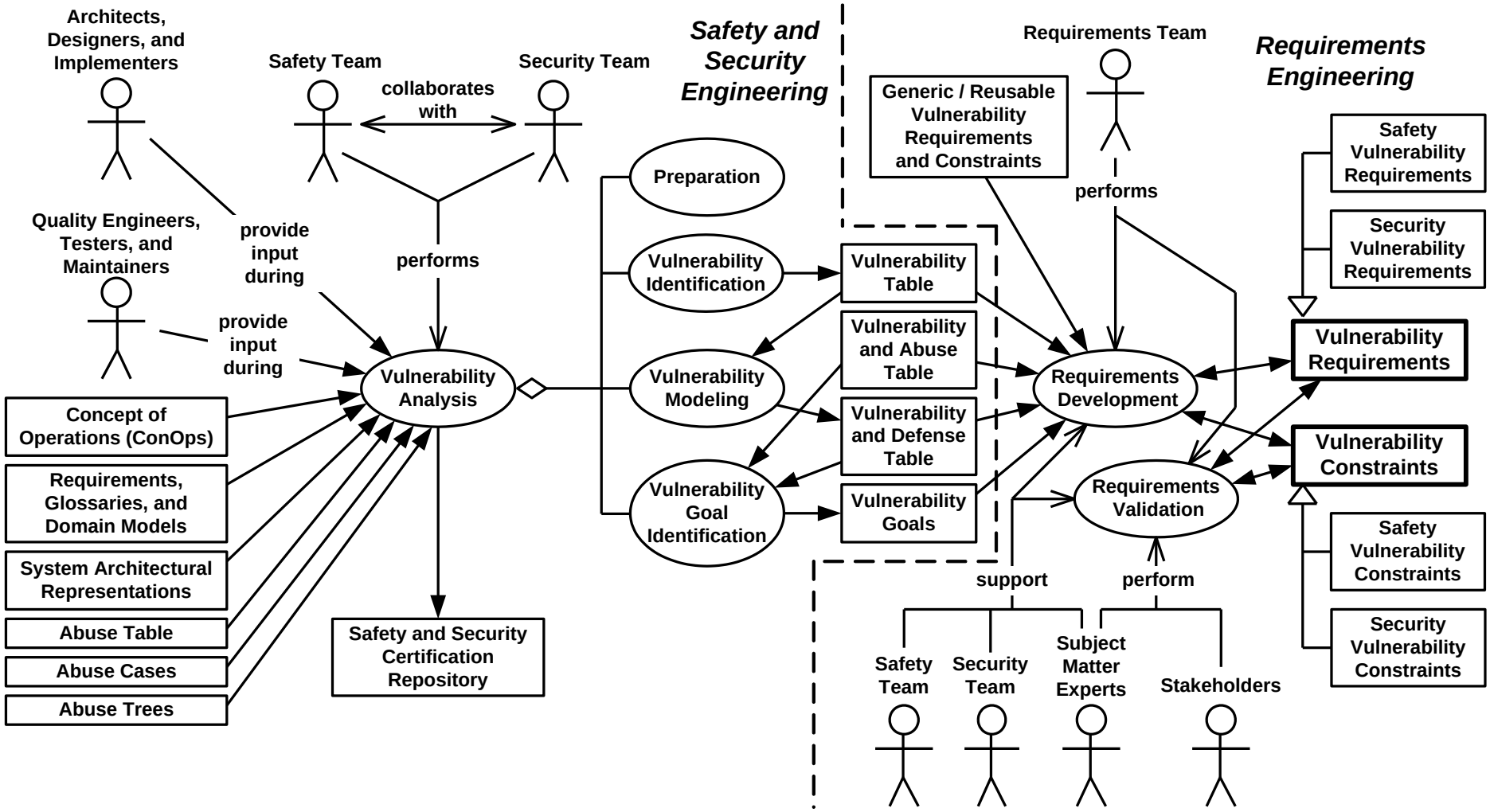
Asset Analysis



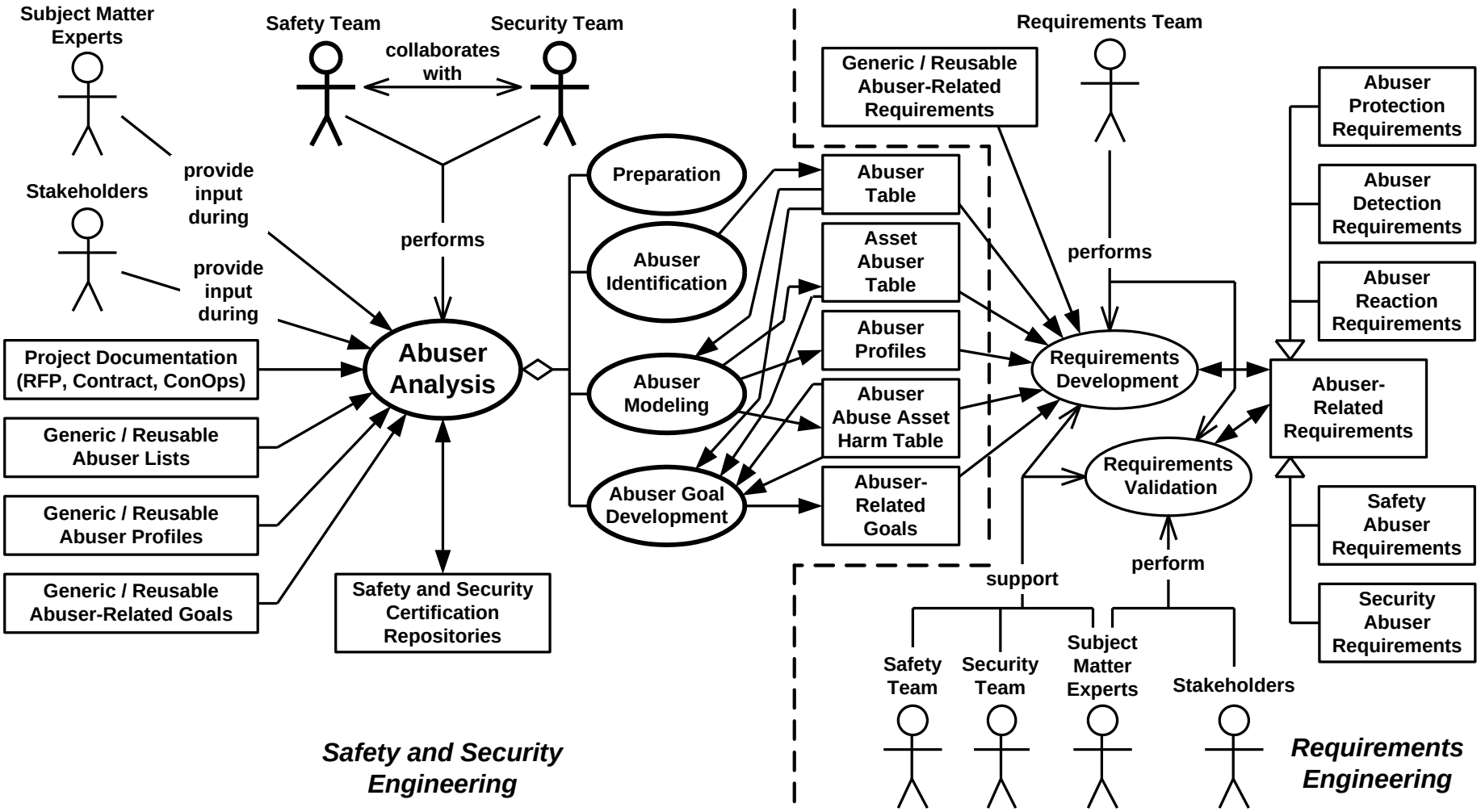
Abuse (Misuse and Mishap) Analysis



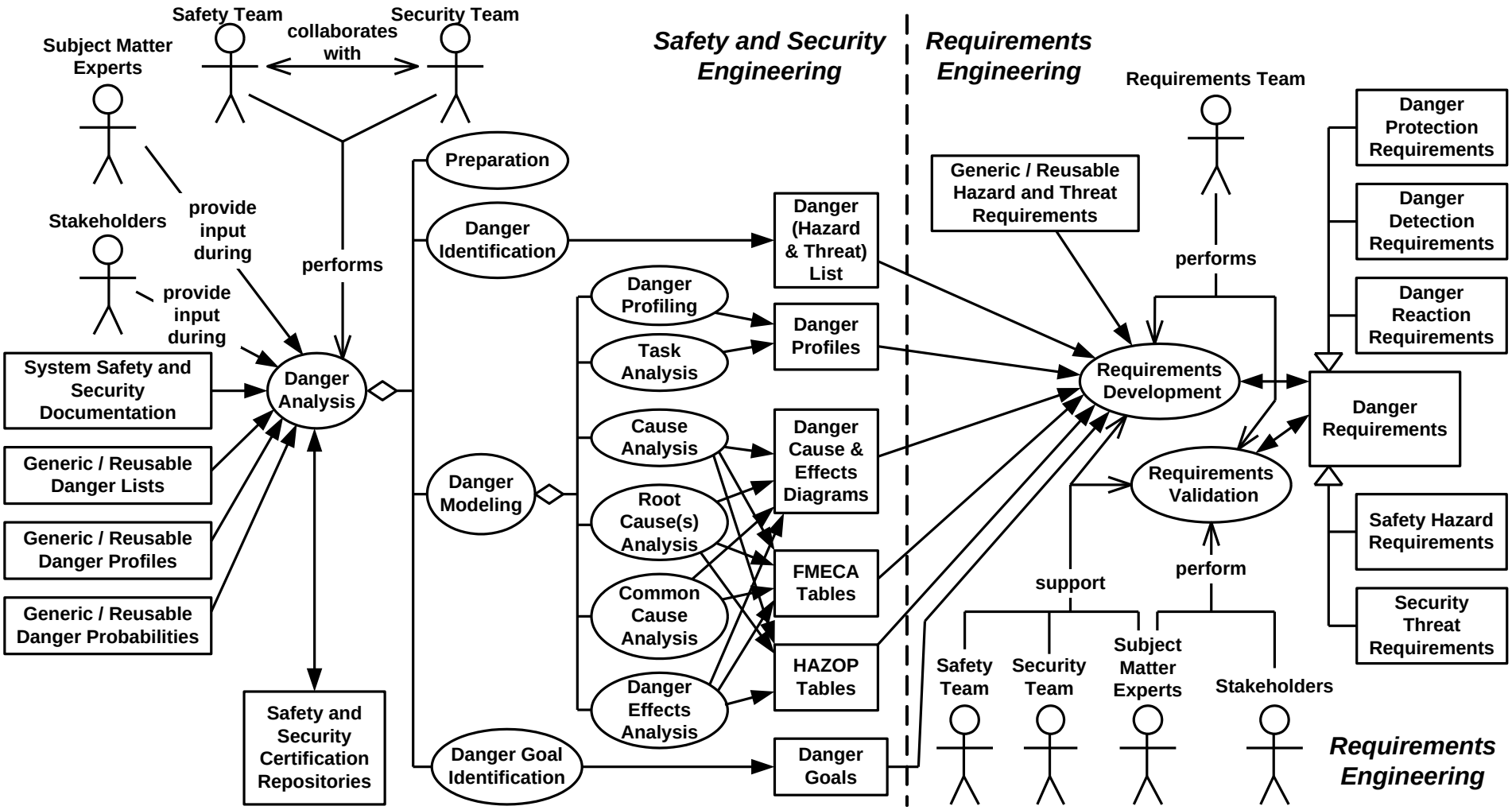
Vulnerability Analysis



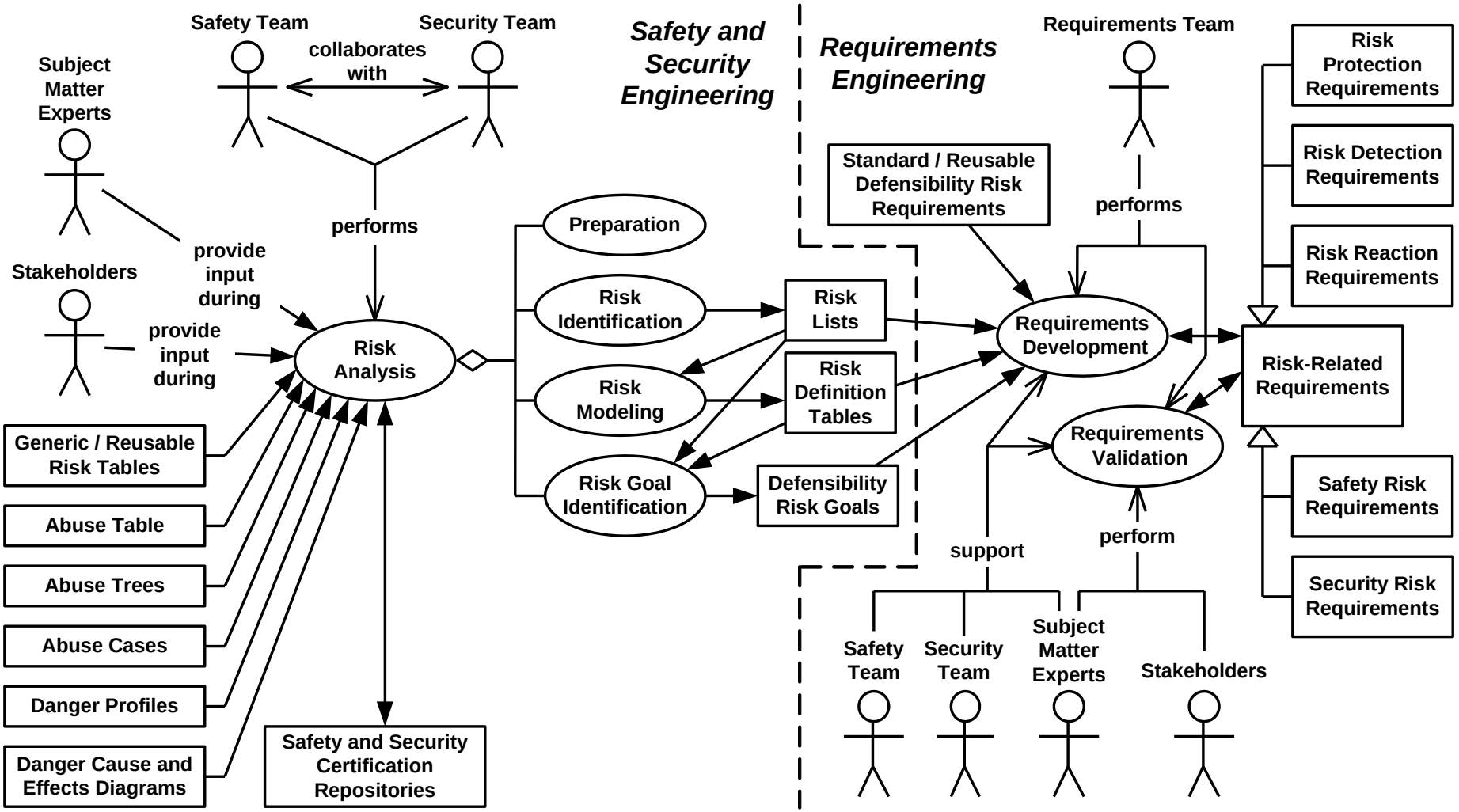
Abuser Analysis



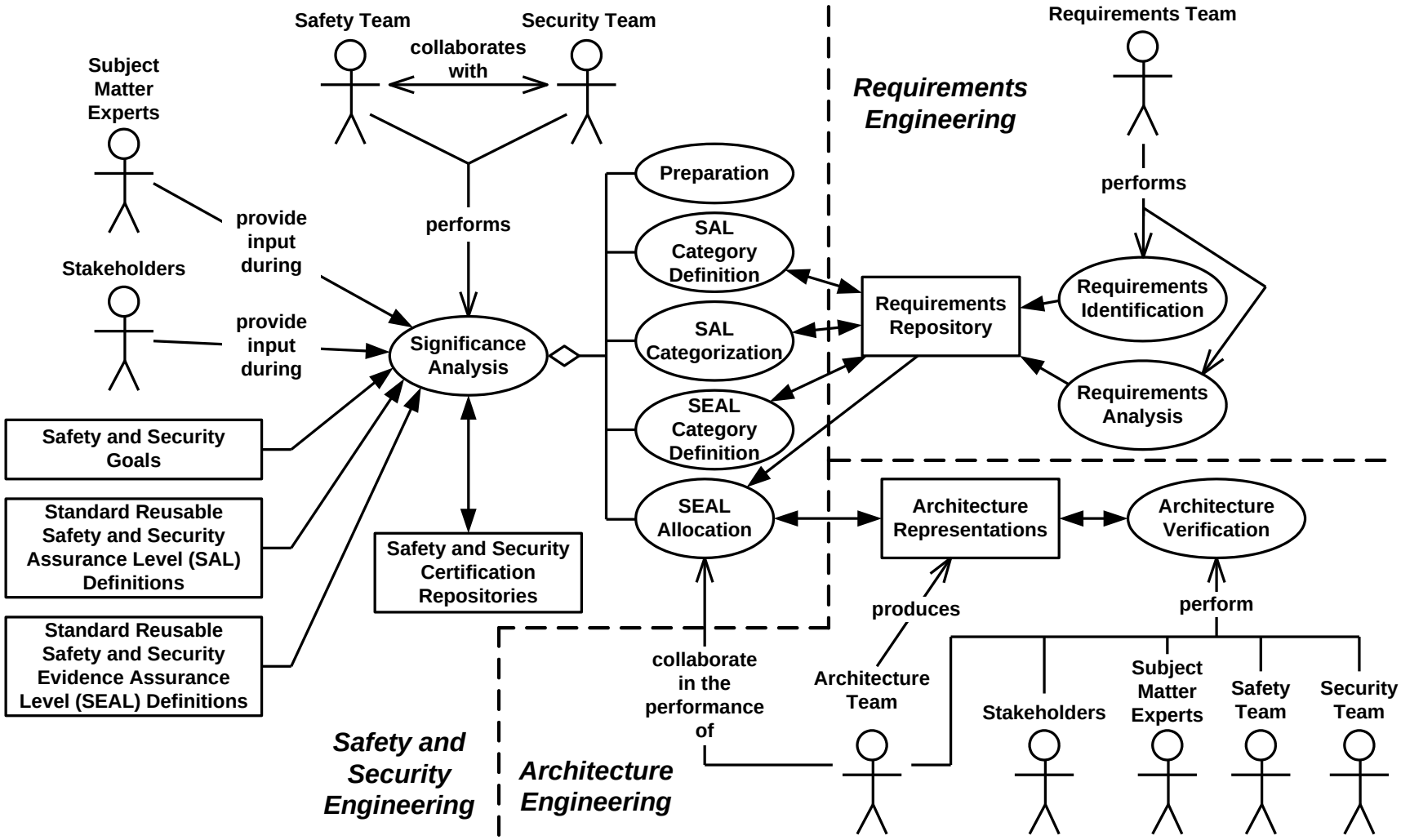
Danger Analysis



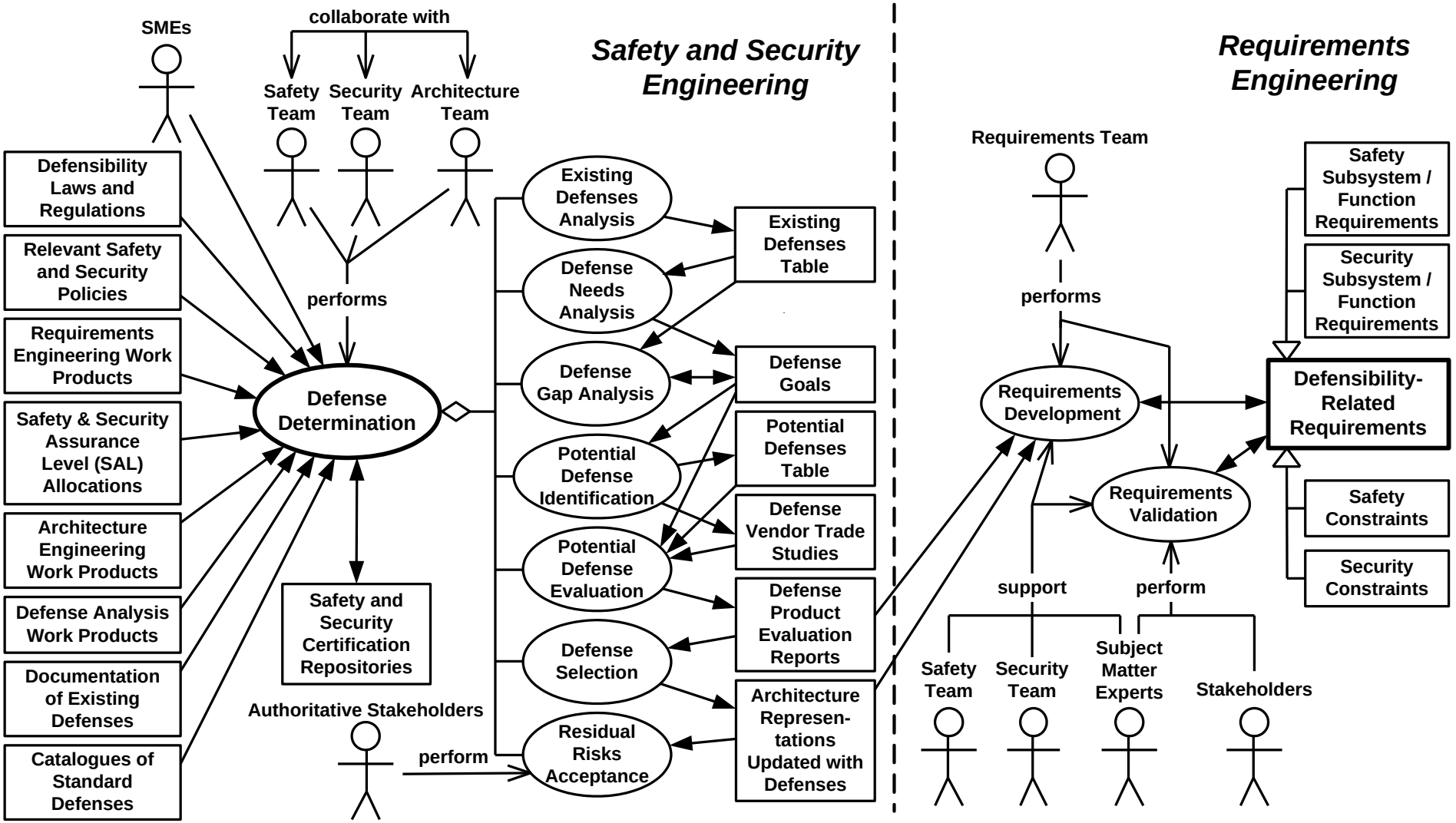
Defensibility Risk Analysis



Defensibility Significance Analysis



Defense Determination



Conclusion:

Process Improvement Recommendations

Process Improvement Recommendations

Ensure close collaboration among safety, security, and requirements teams.

Better *integrate safety and security* methods:

- Concepts and terminology
- Techniques and work products
- Provide cross training

Better *integrate defensibility* methods with *requirements* method:

- Early during development cycle (build safety/security in)
- Clearly define team responsibilities
- Provide

Develop all types of safety- and security-related requirements.

Ensure that these requirements have the proper characteristics.

Final Thoughts

Most of my publications (including half day, full day, and two day versions of these slides) can be downloaded from www.donald.firesmith.net

Look for my next book, *Engineering Safety- and Security-related Requirements,* which will be published in Spring 2010 by Auerbach.

I have brought a draft manuscript to the workshop for anyone who would like to see it.

I am looking for volunteers to be technical reviewers.
Please contact me if you are interested and able.

For More Information

Donald Firesmith
Senior Member of the Tech. Staff
Acquisition Support Program
Telephone: +1 412-268-6874
Email: dgf@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

World Wide Web:
www.sei.cmu.edu
www.sei.cmu.edu/contact.html

Customer Relations
Email: customer-relations@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257