Safety-Related Requirements And Their Relationships To Other Types Of System Requirements

SW Certification Consortium Workshop
Nuclear Regulatory Commission (NRC)
Washington, DC
29 October 2013

Donald Firesmith
Client Technical Solutions Directorate
Software Engineering Institute

Topics

Safety and Security Ontology:

- Asset, Harm, Abuse, Vulnerability, Abuser, Hazard, Risk
- Scope of Safety/Security Analysis

Safety and Safety Engineering

Quality Model:

- Quality Characteristics
- Quality Attributes

Types of Requirements

Safety- and Security-Related Requirements

Safety/Security Analysis (SA) – Ontology

Foundational Concepts:

- Asset, Harm, Abuse (Mishap/Misuse), Vulnerability, Abuser, Danger (Hazard/Threat), Risk

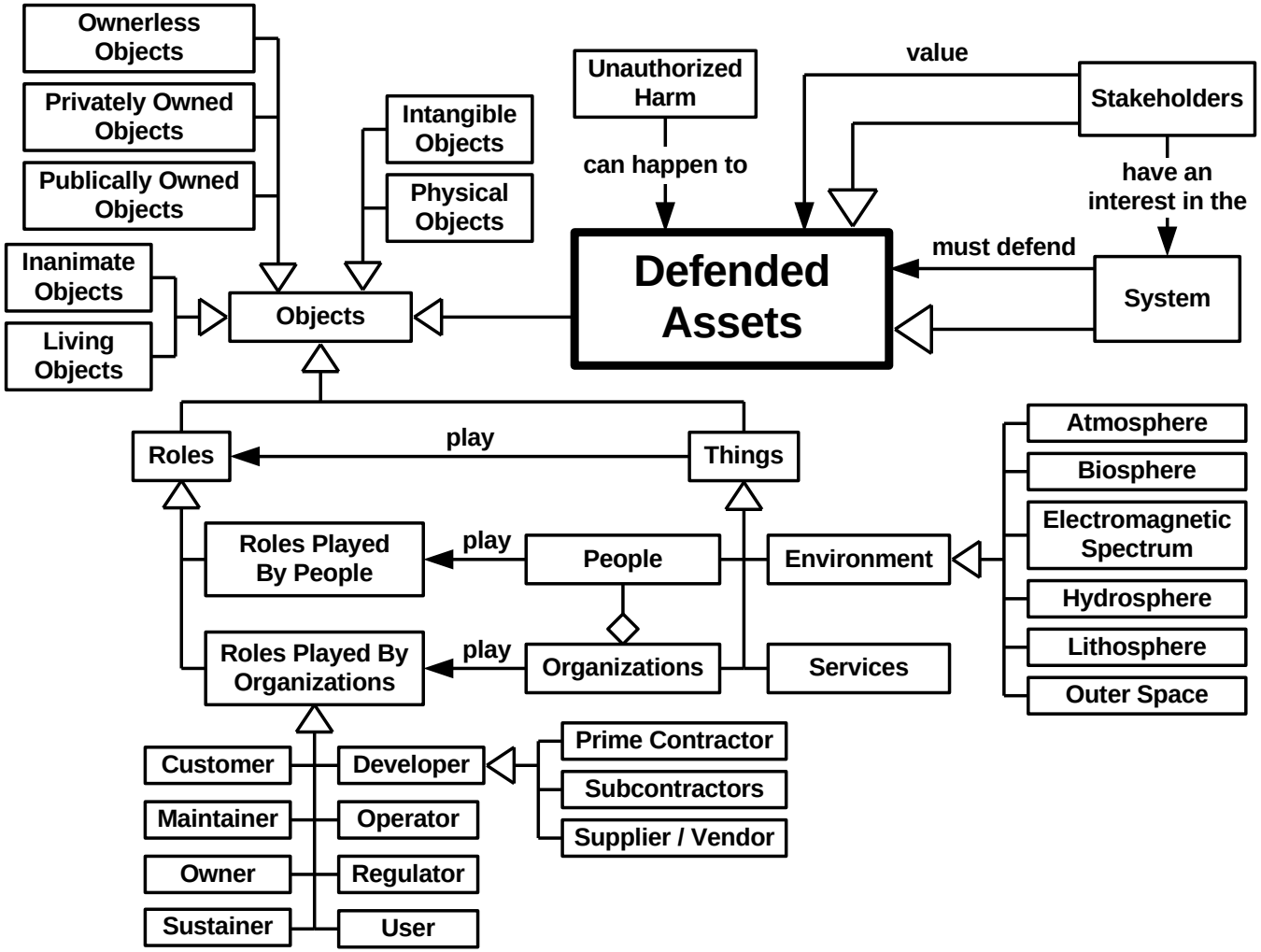
Foundation of:

- Safety/Security Analysis
- Safety/Security Requirements

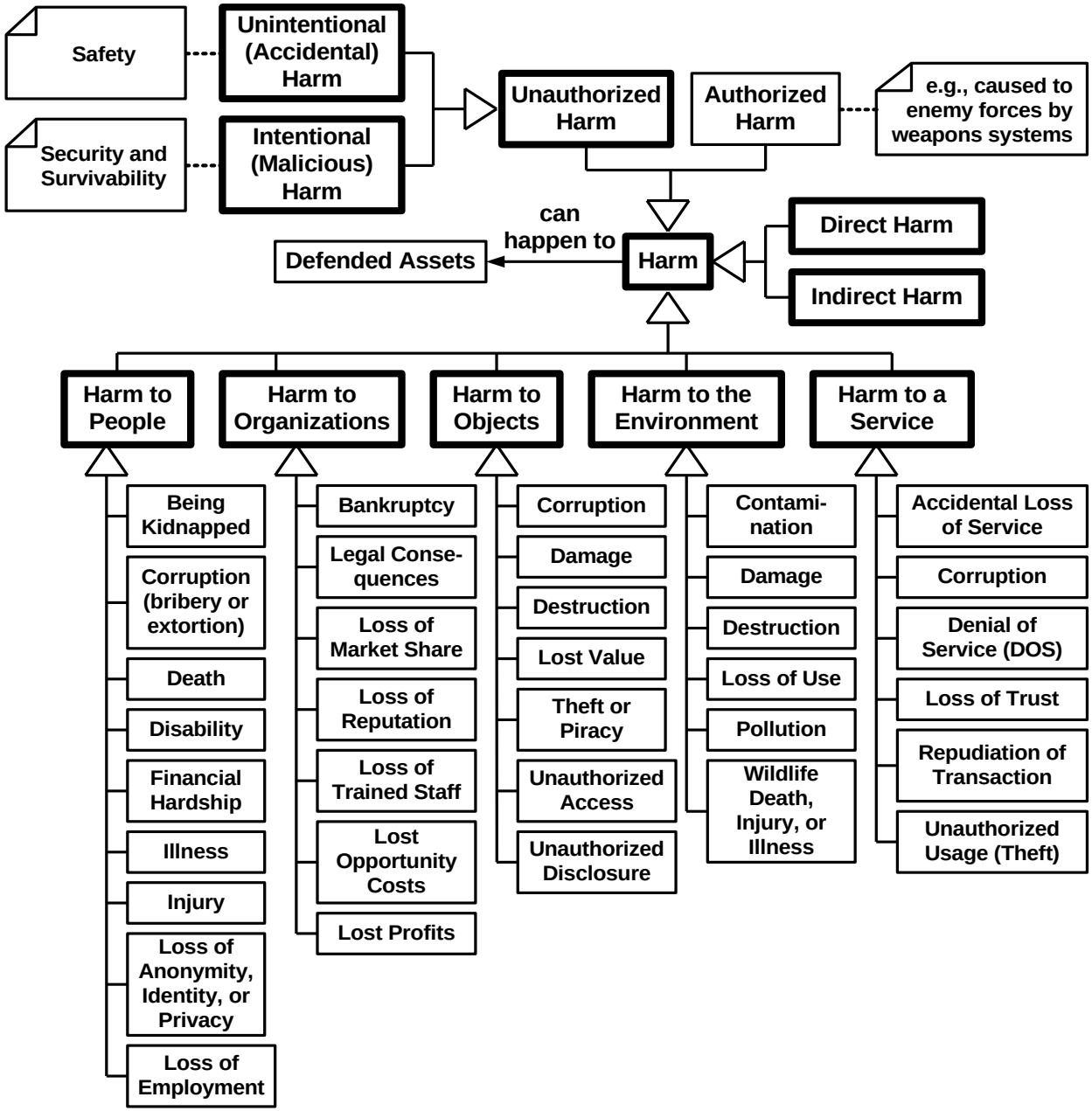
People are often not careful in their usage of these terms:

- Confuse Hazard and Abuse/Vulnerability
- Confuse Threat and Abuser

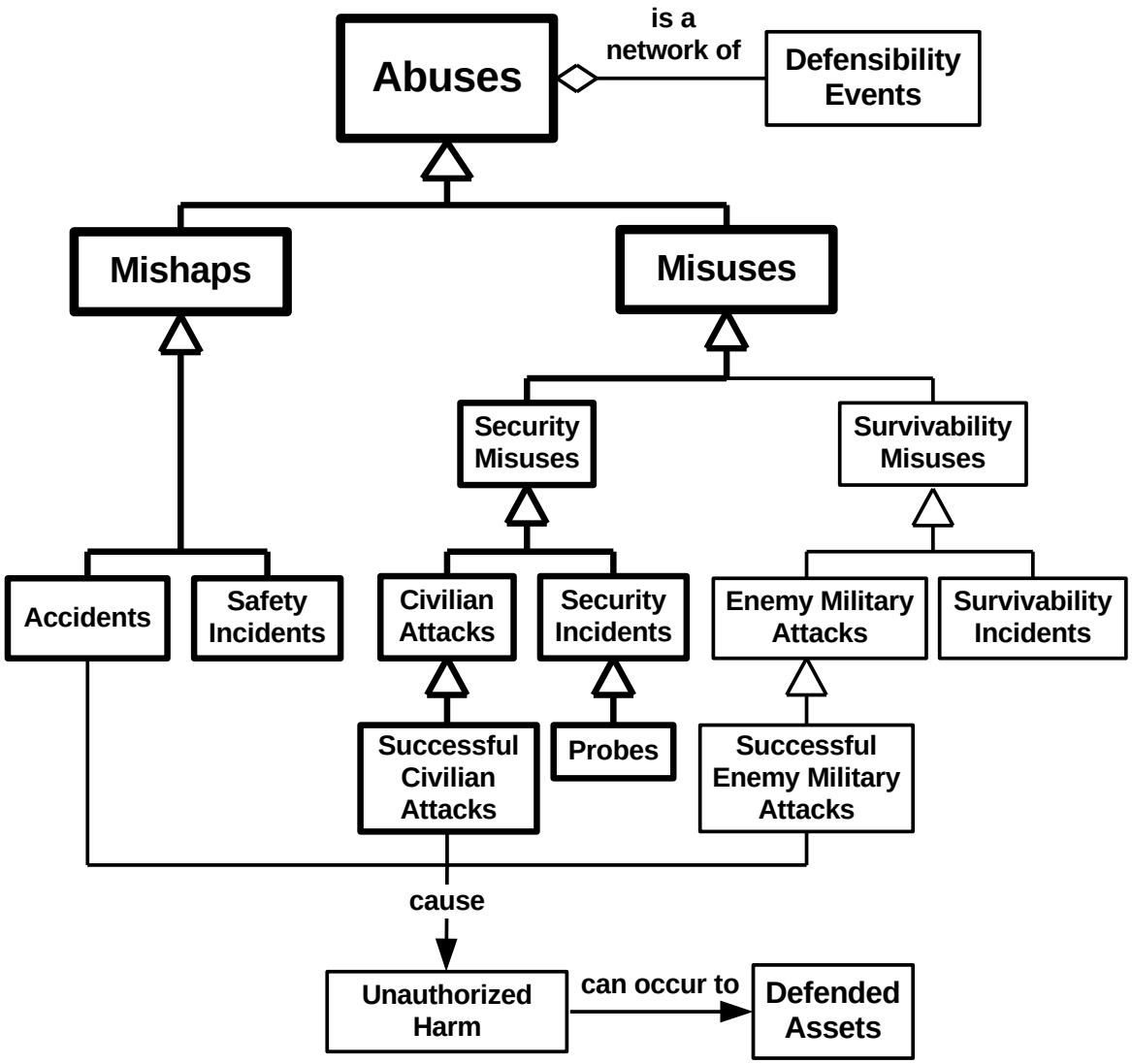
SA – Assets



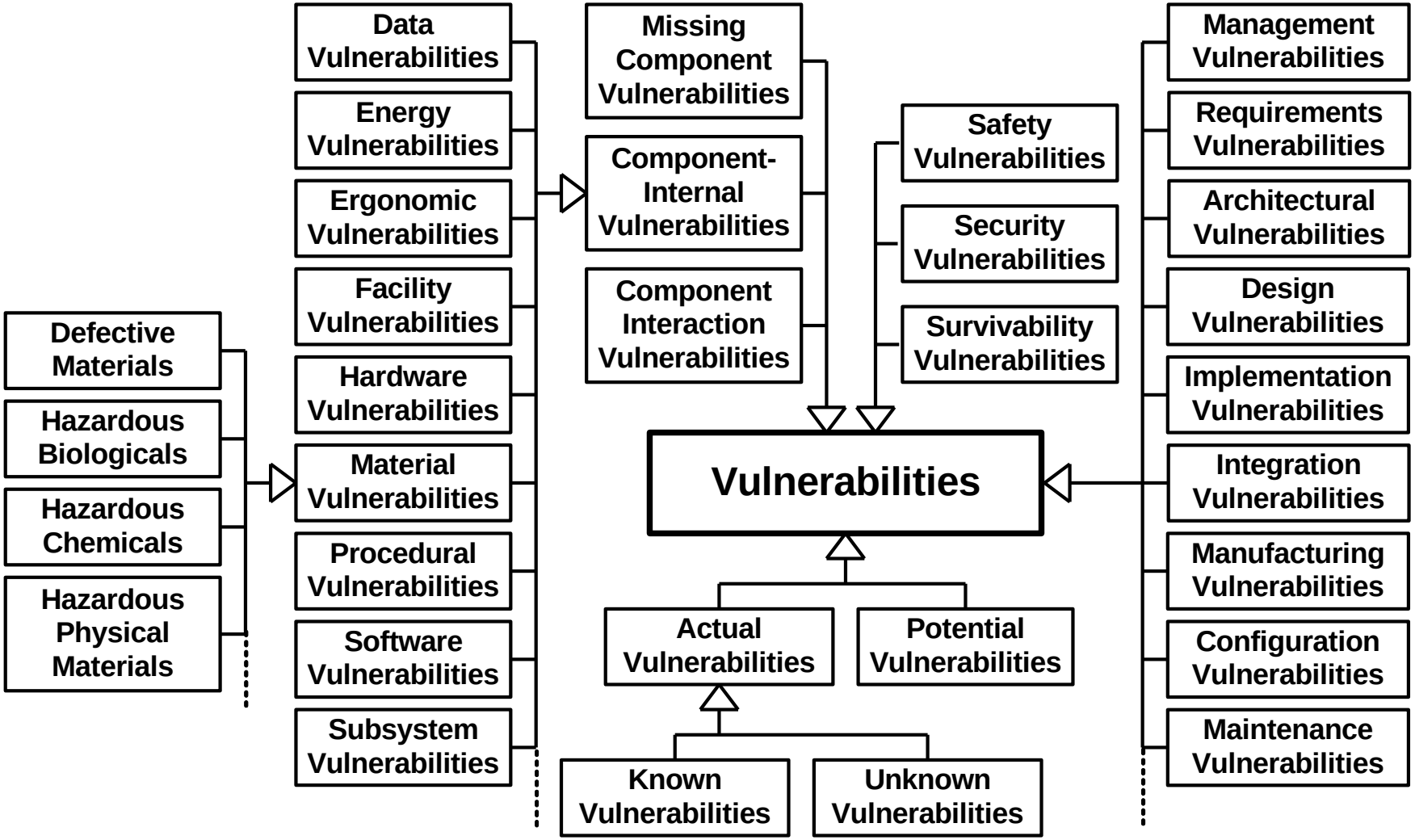
SA – Harm



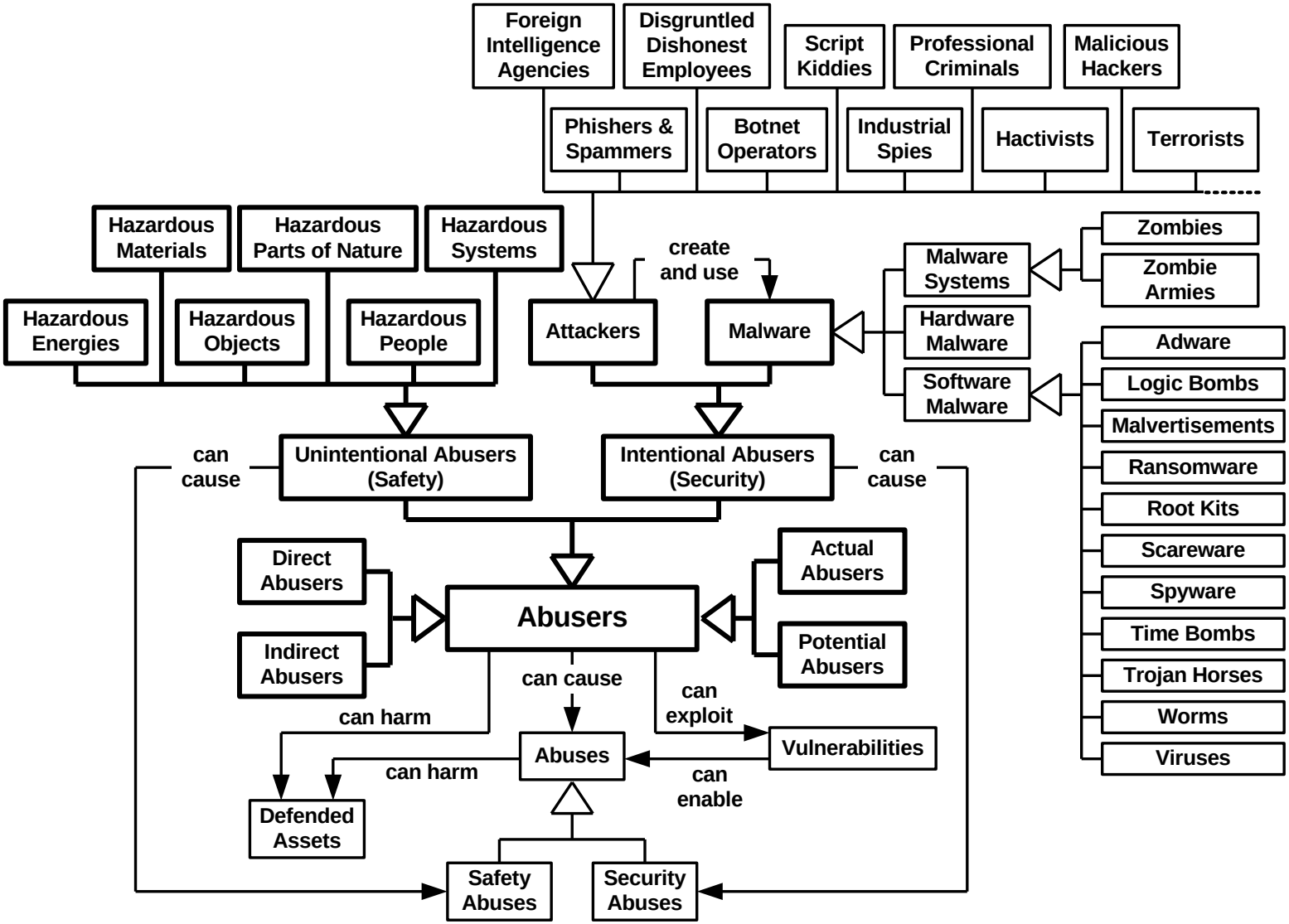
SA – Abuses



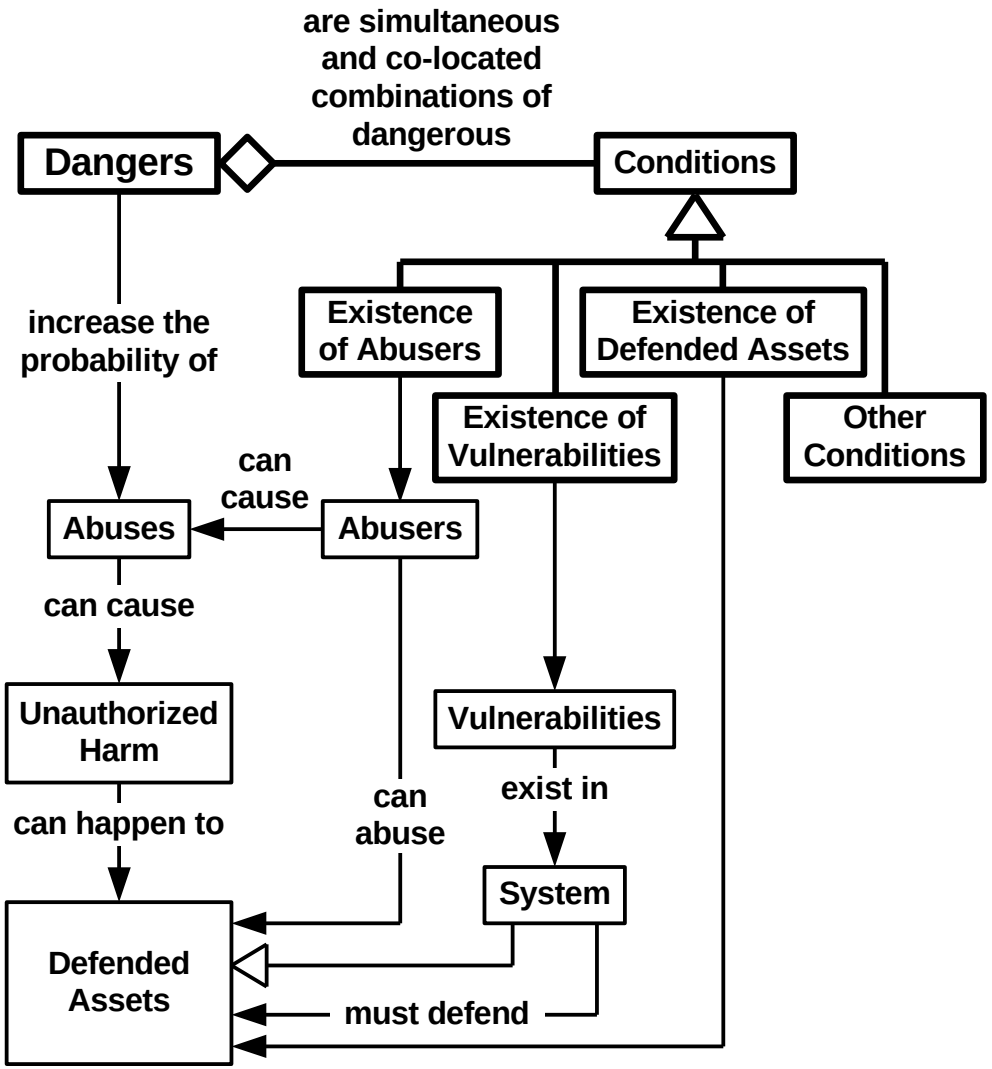
SA – Vulnerabilities



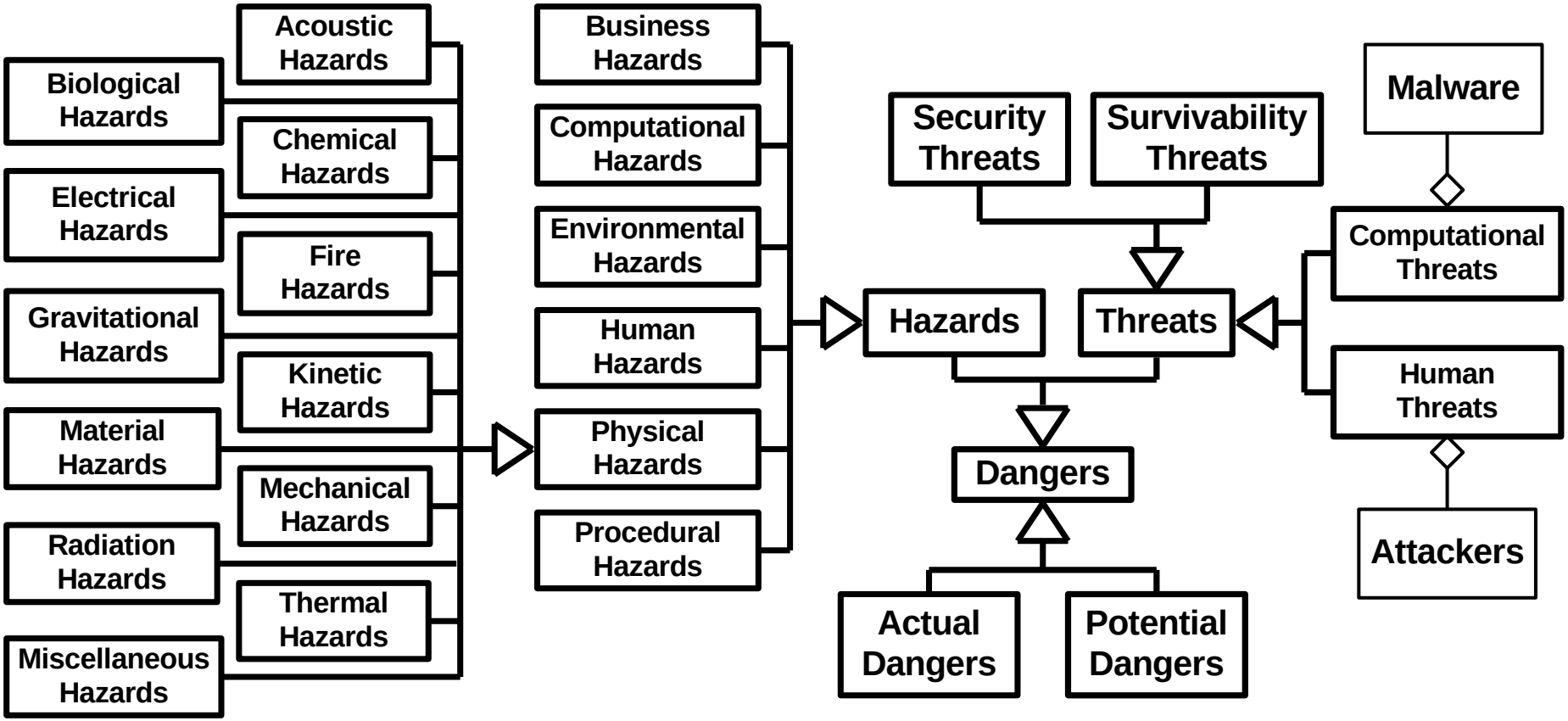
SA – Abusers



SA – Hazards 1



SA – Hazards 2 – Categories of Dangers



Safety as a Quality Characteristic

Safety is *degree* to which:

- the system:
  - *Prevents (eliminates, mitigates, makes sufficiently rare)*
  - *Detects*
  - *Reacts to*
- the following:
  - *Accidental harm* to defended assets
  - *Safety abuses (mishaps such as accidents and near misses)*
  - *Safety abusers (people, systems, and the environment)*
  - *Safety vulnerabilities*
  - *Hazards* (conditions including the existence of non-malicious abusers who unintentionally exploit system vulnerabilities to accidentally harm vulnerable defended assets)
  - *Safety risks*

Security (civilian) and **Survivability** (military) can be defined analogously.

Defensibility, Safety, and Security

Safety Engineering

the systems engineering discipline concerned with lowering the risk of *unintentional* (i.e., *accidental*) *unauthorized* harm to defended 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 safety incidents), system-internal vulnerabilities, system-external *unintentional* abusers, *hazards*, and *safety* risks

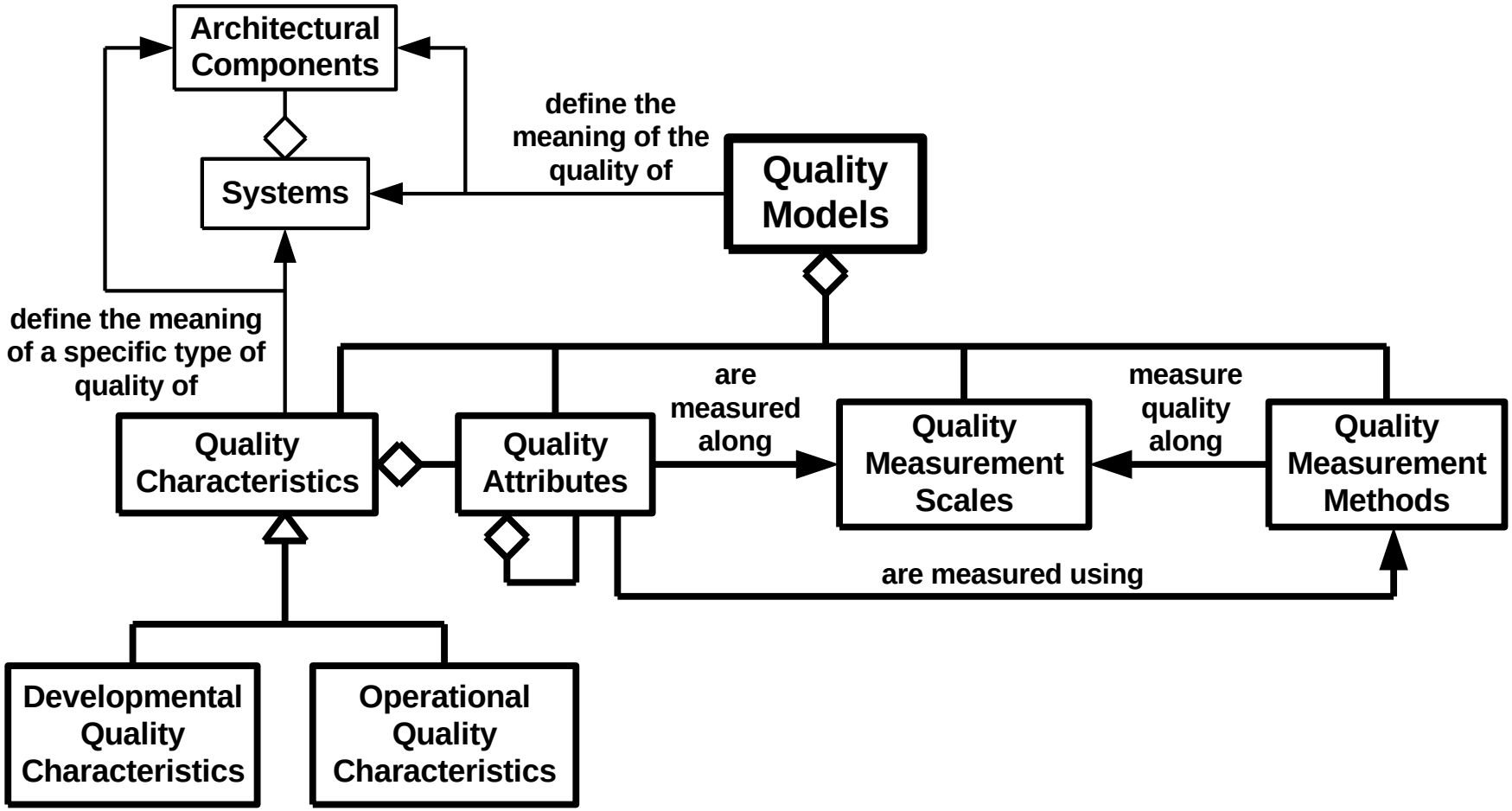
Security Engineering

the systems engineering discipline concerned with lowering the risk of intentional (i.e., malicious) unauthorized harm to defended assets to a level that is acceptable to the system's stakeholders by preventing, detecting, and properly reacting to such harm, civilian misuses (i.e., attacks and security incidents), system-internal vulnerabilities, system-external intentional civilian abusers, threats, and security risks

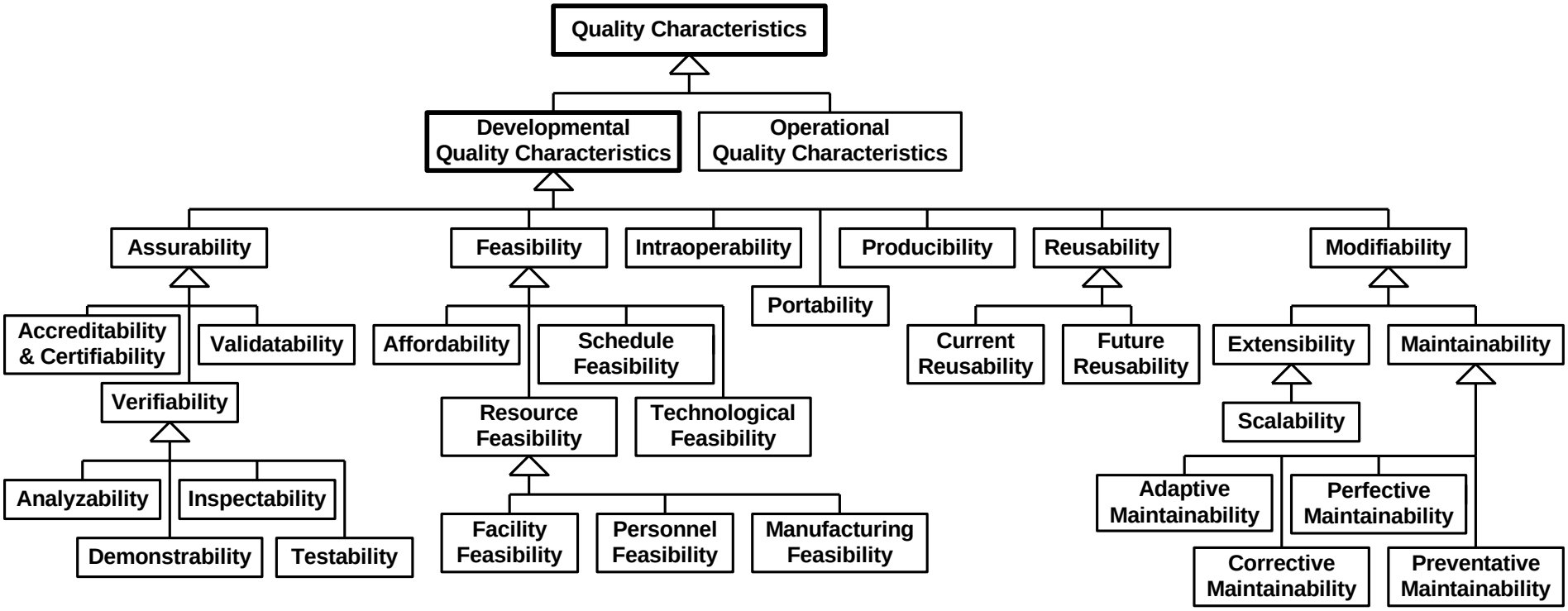
Survivability Engineering

the systems engineering discipline concerned with lowering the risk of *intentional* (i.e., *malicious*) unauthorized harm to defended assets to a level that is acceptable to the system's stakeholders by preventing, detecting, and properly reacting to such harm, *military misuses* (i.e., *attacks* and survivability incidents), system-internal vulnerabilities, system-external *intentional military* abusers, threats, and *survivability* risks

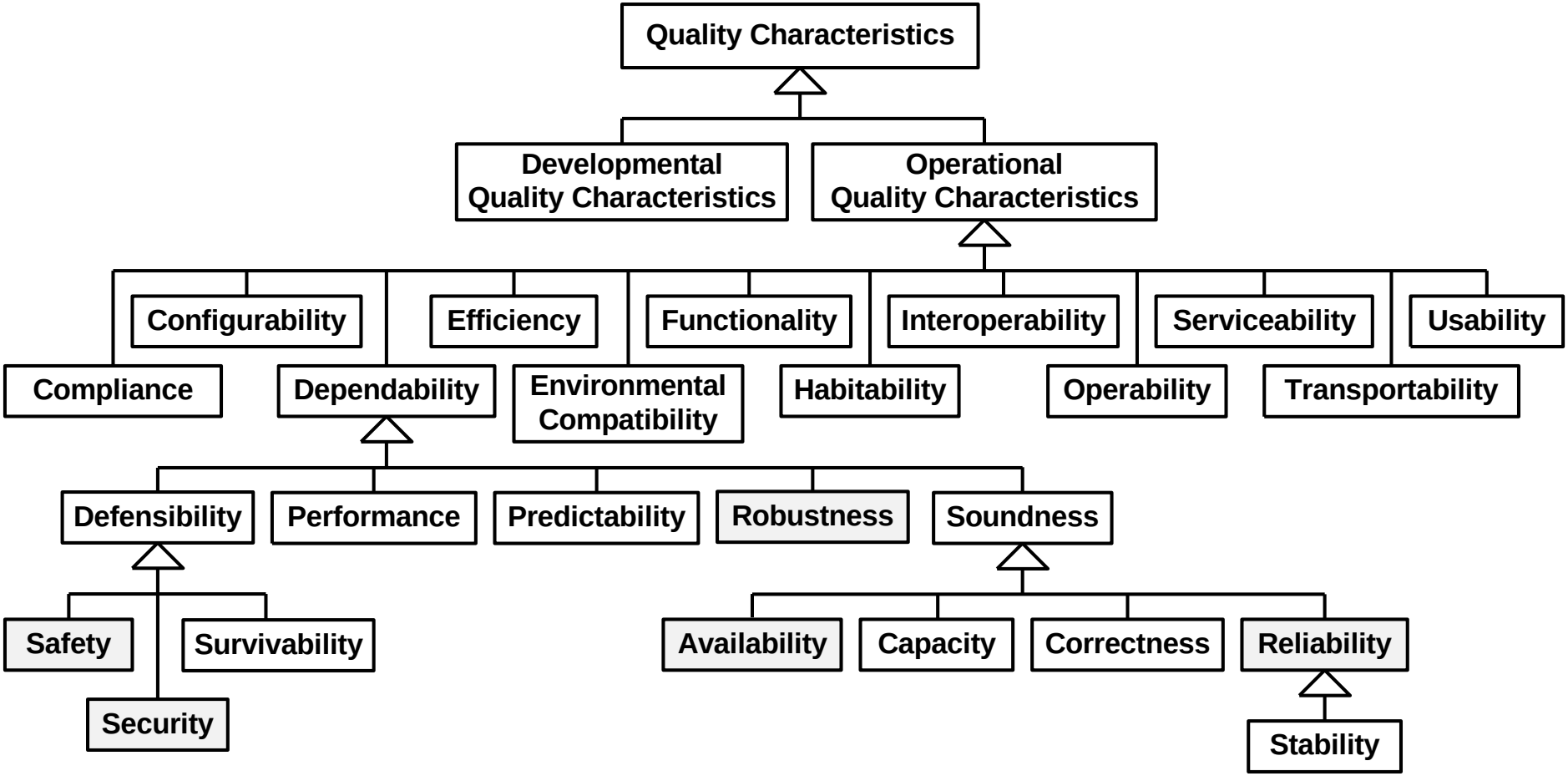
Quality Model (ISO Standard)



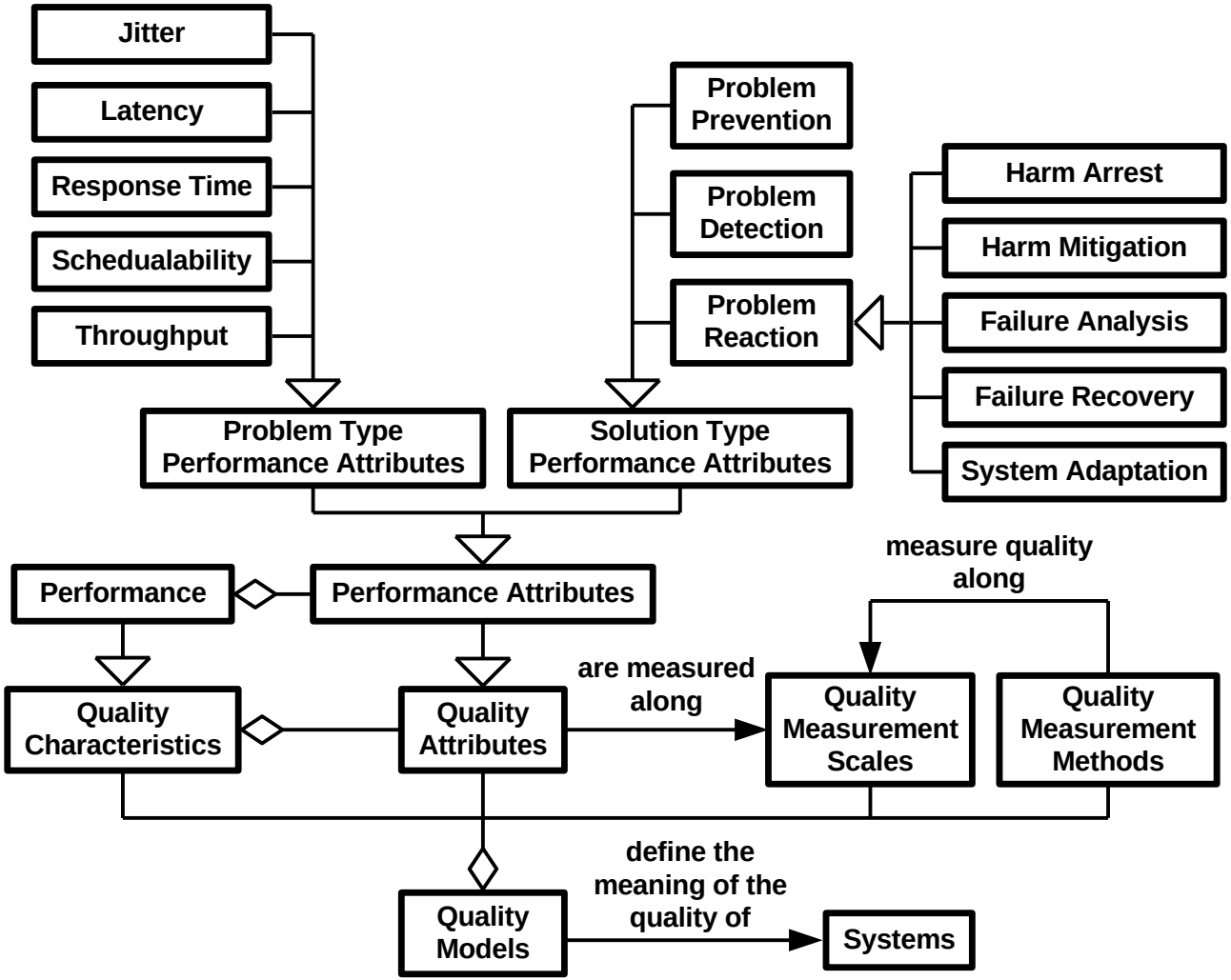
Developmental Quality Characteristics



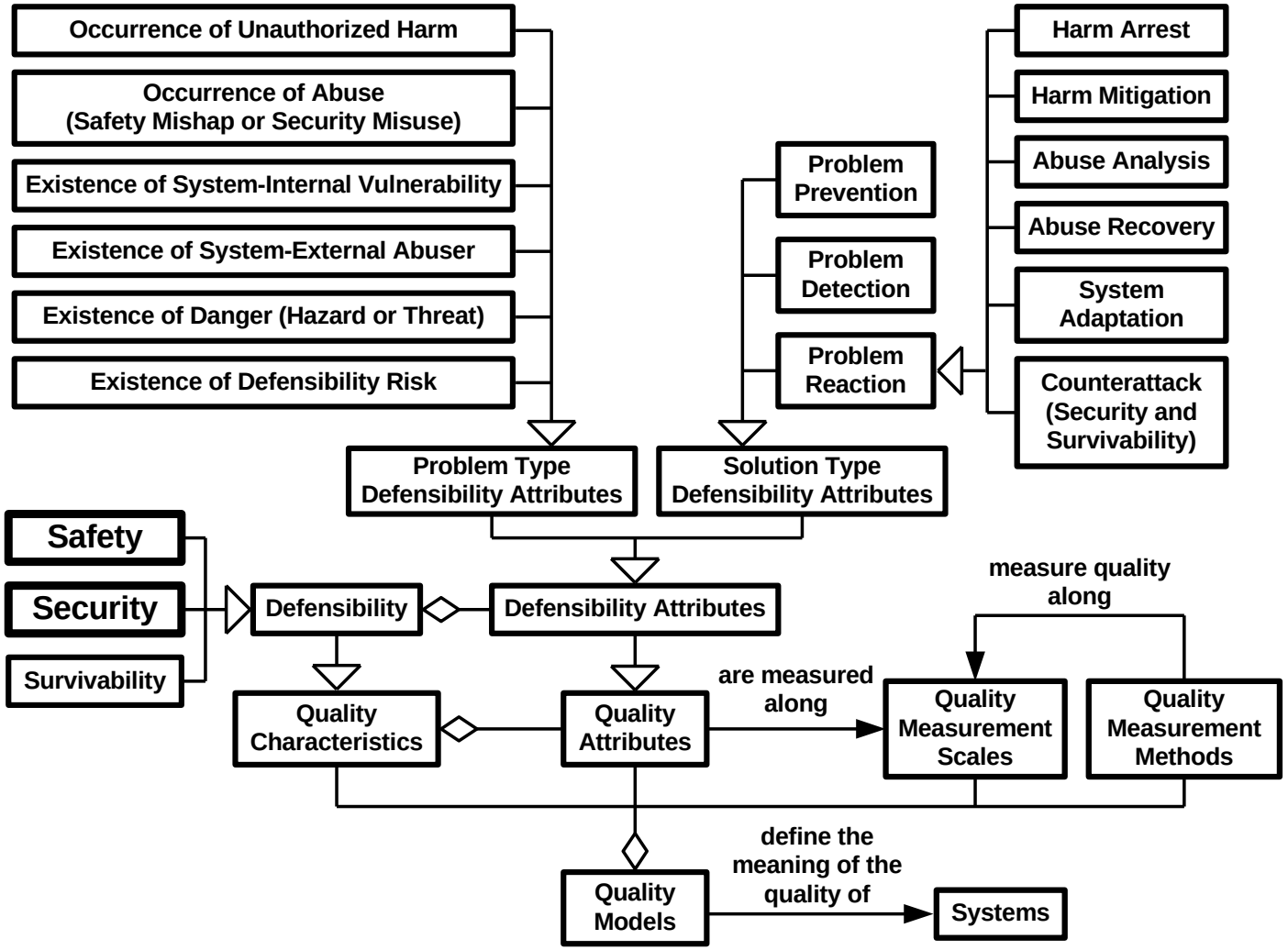
Operational Quality Characteristics



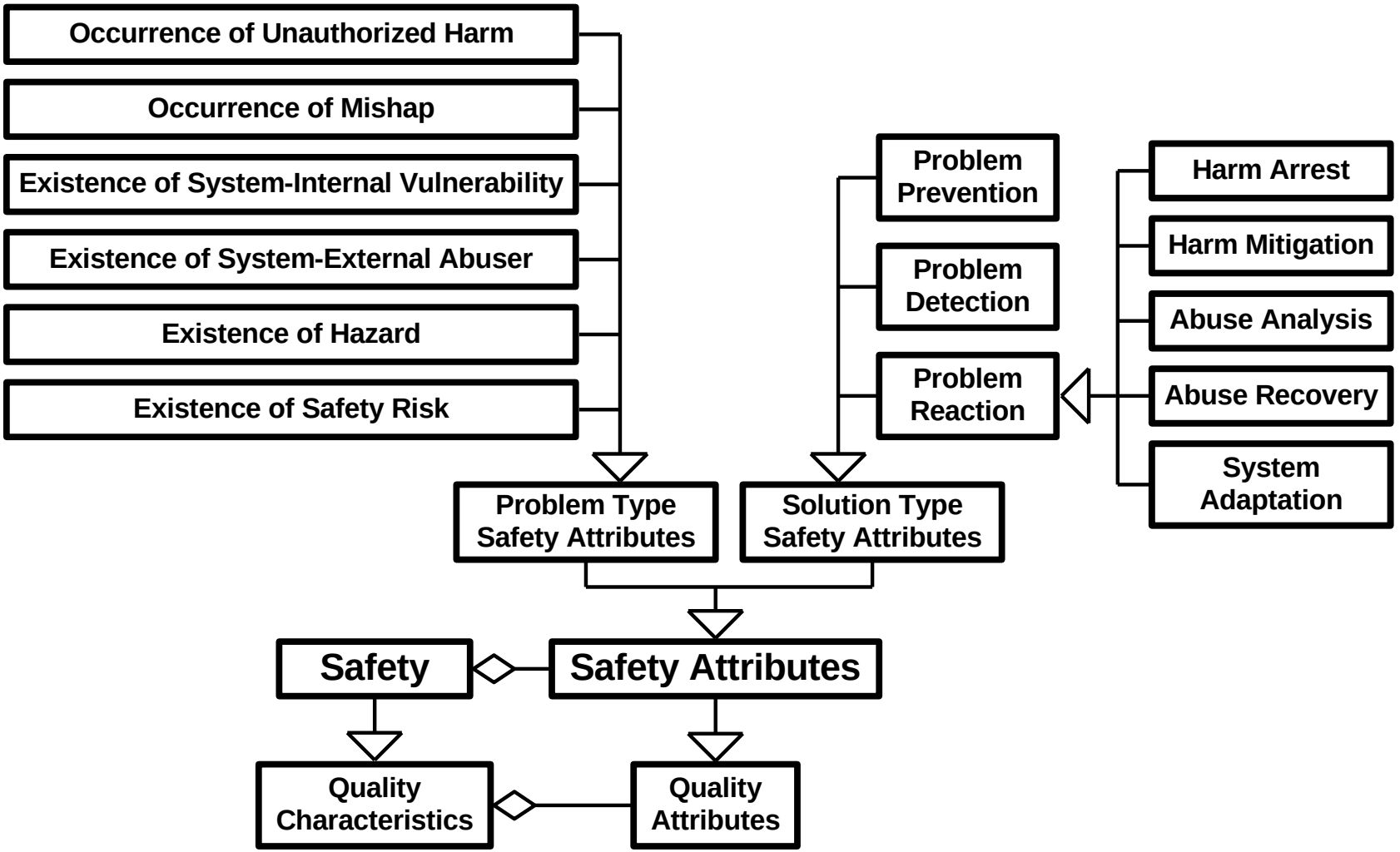
Performance Attributes



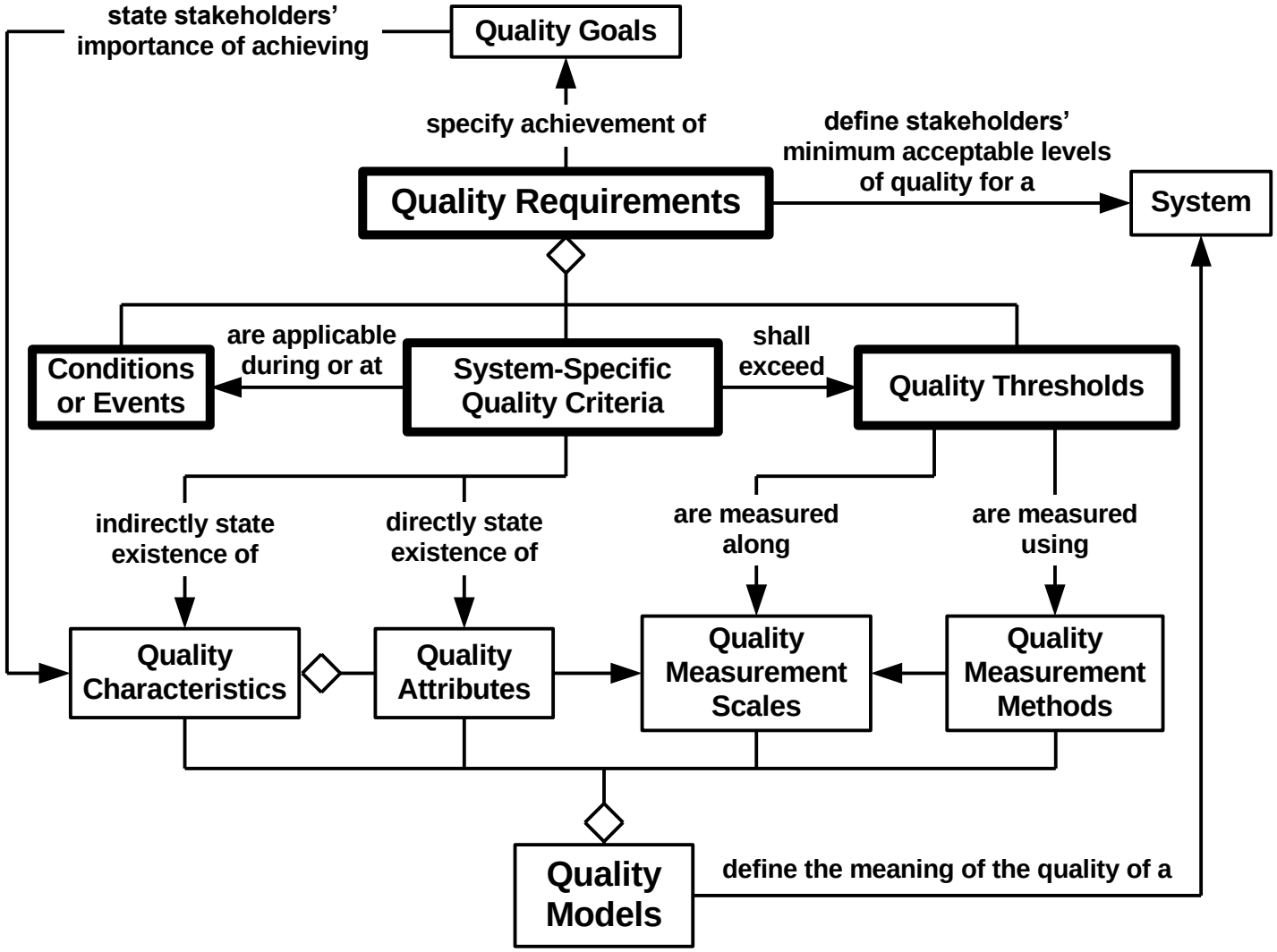
Defensibility Attributes



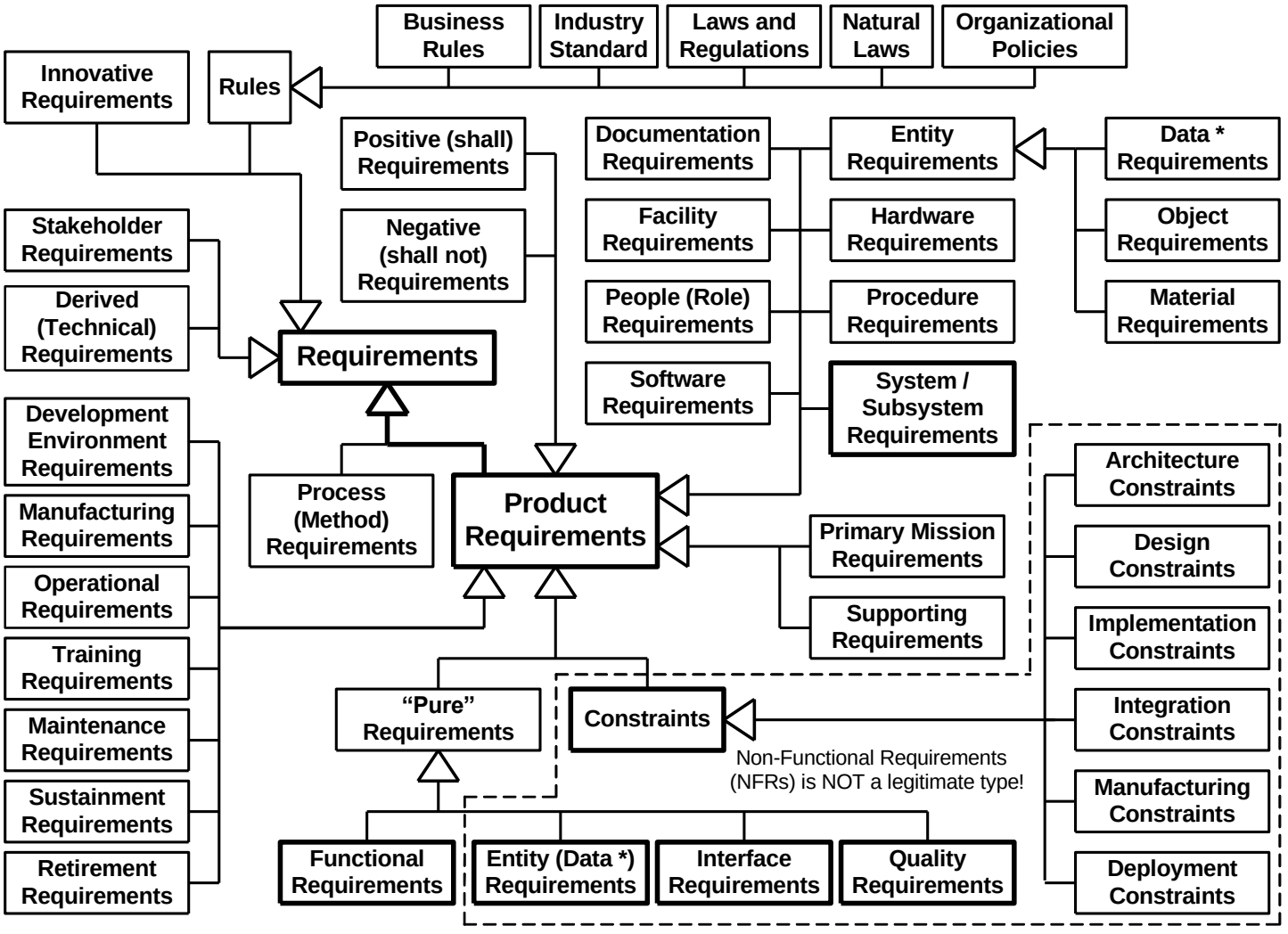
Safety Attributes



Components of a Quality Requirement



Types of Requirements



Defensibility-Related Requirements
Safety- and Security-Related Requirements

Defending Requirements:

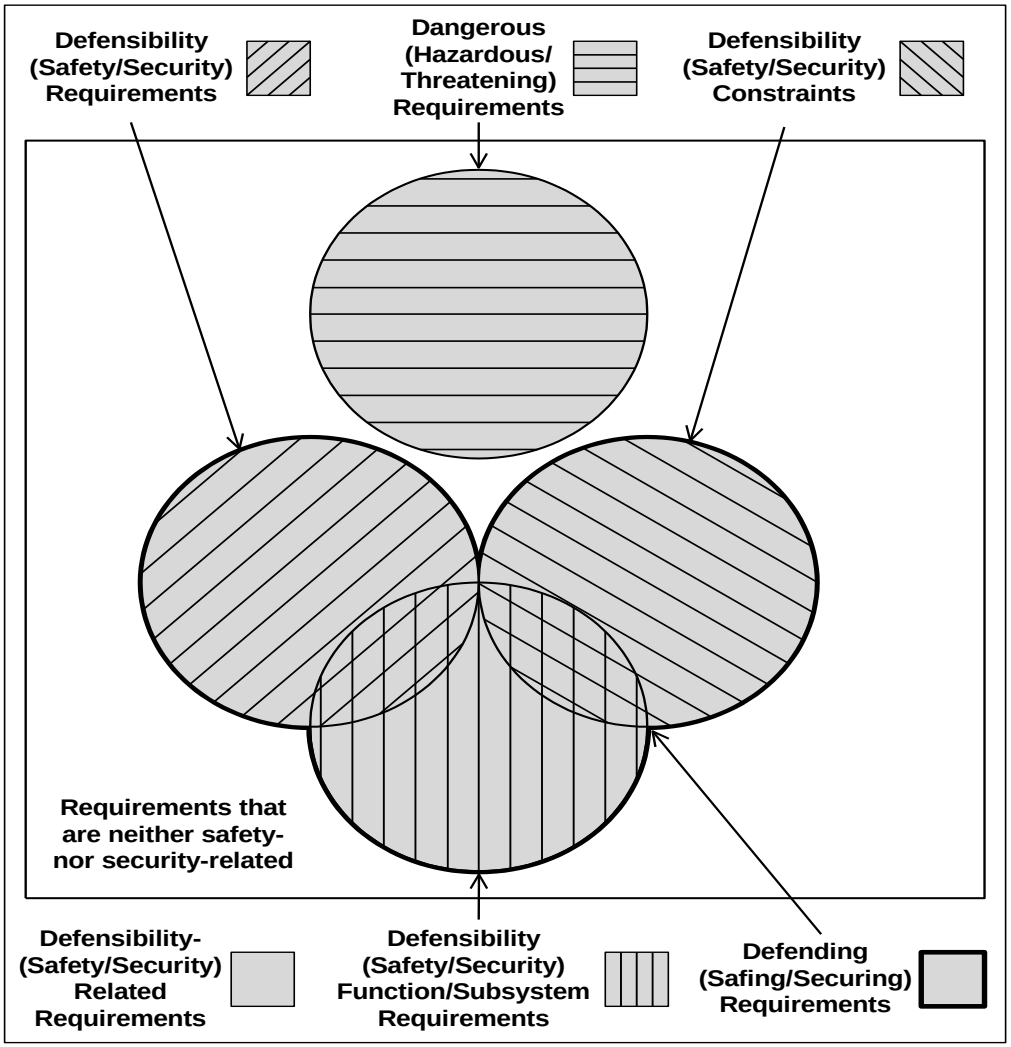
Specifically intended to make the system more safe or secure

- **Defensibility Requirements**
  Quality (Safety/Security) Requirements specifying how safe in terms of protected assets, harm to these assets, mishaps/misuses, vulnerabilities, abusers, hazards/threats, and safety/security risks
- **Defensibility Function/Subsystem Requirements**
  Functional/data/interface/quality Requirements specifying a defensibility function or subsystem
- **Defensibility (Safety/Security) Constraints**
  Architecture, design, implementation, integration, and configuration constraints specifying defenses (safeguards and countermeasures)

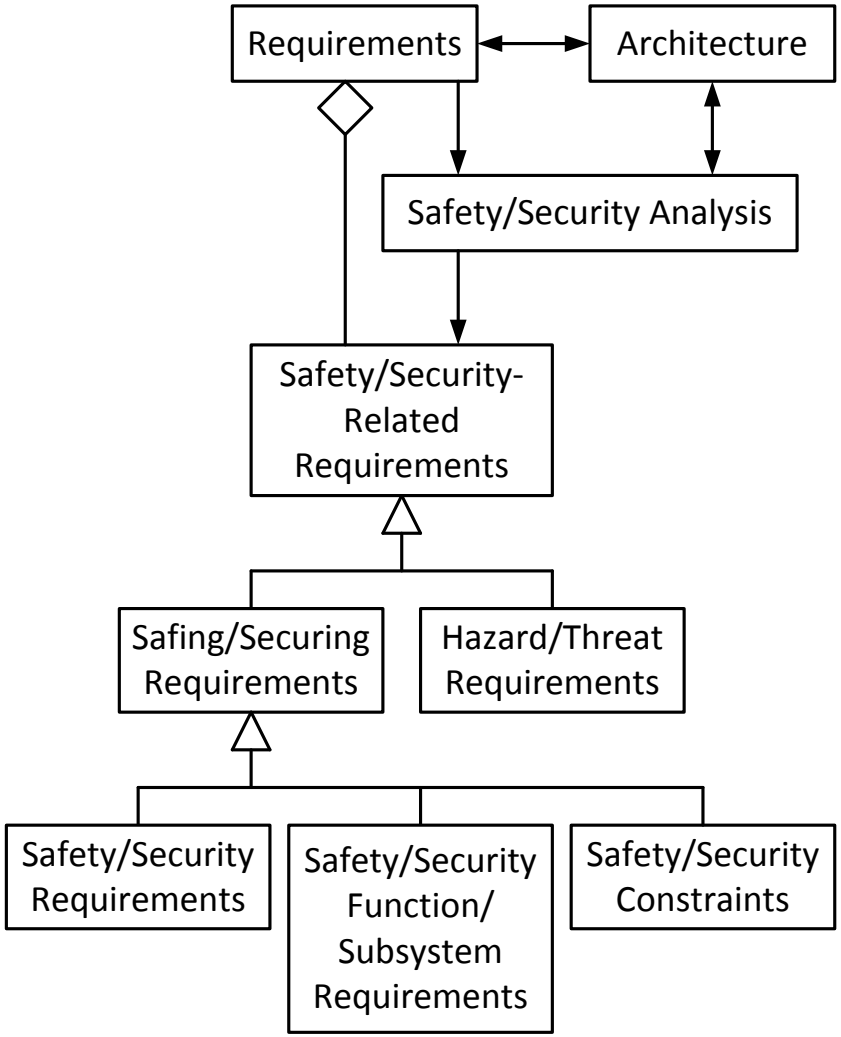
Dangerous (Hazardous/Threatening) Requirements

May make the system less safe or secure if not implemented right (Safety/Security Assurance Levels)

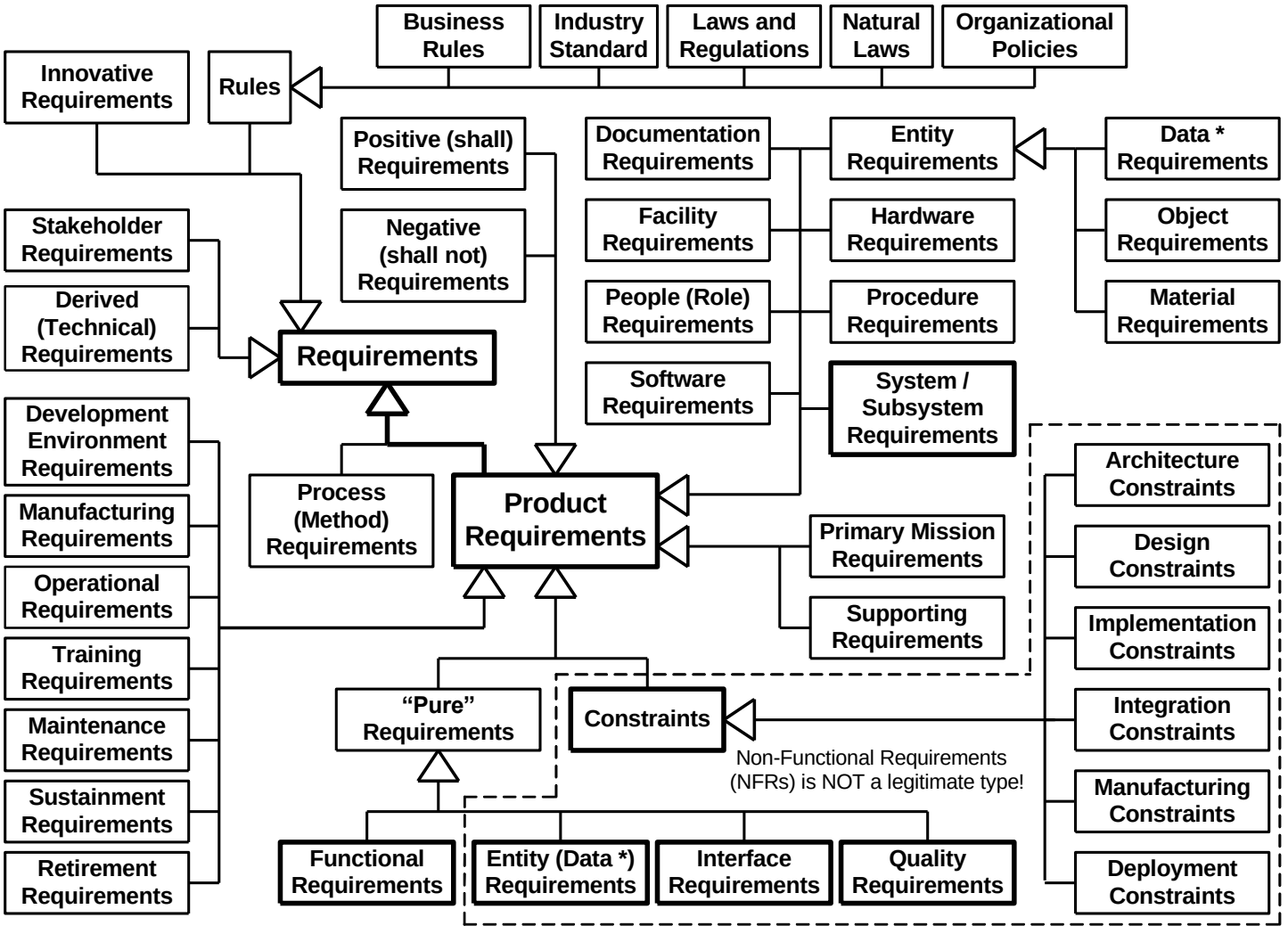
Types of Defensibility-Related Requirements (Safety- and Security-related Requirements)



Requirements, Architecture, and Danger (Safety/Security) Analysis



Types of Requirements



Conclusion

Top-level architecture drives safety/security analysis and vice versa.

Safety/security analysis drives safety-related requirements.

Safety/security engineering cannot be separate from requirements and architecture engineering.

- Safety/security engineering cannot be ignored until after requirements and architecture engineering.

Requirements, architecture, and hazard/threat analysis must be done incrementally, iteratively, and concurrently during the entire development and life cycle.

There are many types of requirements:

- There are several types of safety/security-related requirements.
- Many projects address only one or two of them.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.