A Taxonomy of Safety-Related Requirements

Donald Firesmith
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA. 15213 USA
+1 (412) 268-6874
dgf@sei.cmu.edu

ABSTRACT
As software-intensive systems become more pervasive, more and more safety-critical systems are being developed and deployed. Yet when most people think about safety requirements, they think of safety-critical functional requirements, which are requirements that have critical safety ramifications if not correctly implemented. However, there are actually four major classifications of safety-related requirements: (1) pure safety requirements, (2) safety-significant requirements, (3) safety constraints, and (4) requirements for safety systems. This paper describes a taxonomy of these different kinds of safety-related requirements, and clearly and briefly defines and describes each of the above categories of safety-related requirements.

Keywords
Safety, safety-critical, safety requirement, safety constraint, safety categorization, safety engineering, requirements engineering

1 SAFETY-RELATED REQUIREMENTS

As software-intensive systems become more pervasive, more and more safety-critical systems are being developed and deployed. Yet when most requirements engineers think about safety requirements, they think of safety-critical functional requirements, which are functional requirements that have critical safety ramifications. Such safety-critical functional requirements typically can cause serious accidents if incorrectly implemented, although it is also possible to unintentionally specify requirements that can cause accidents even if they are properly implemented. Many people also incorrectly confuse safety requirements with reliability requirements, even though it is possible to have a safe system that is unreliable (e.g., it never does anything) and an unreliable system that is safe (e.g., its failures do not cause any harm).

A major theme of this paper is that there exists a large taxonomy of safety-related requirements, only a few of which are typically identified, analyzed, and specified on most projects. Most safety and requirements engineers will be familiar with the term ìsafety-critical requirementsî, and some may well be familiar with safety requirements limiting the existence of specific hazards. Unfortunately, there are many other kinds of safety-related requirements that should be considered when developing systems with significant safety implications.

In the taxonomy presented in this paper, there are actually four major classifications of *safety-related requirements*[1]:

1. **Safety requirements** ñ any requirements that specify mandatory amounts of a subfactor of the safety quality factor (i.e., requirements to protect assets from accidental harm, detect safety incidents, and respond to safety incidents).
2. **Safety-significant requirements** ñ any non-safety primary mission requirement that can cause hazards and safety incidents. They include functional requirements, data requirements, interface requirements, and non-safety quality requirements (e.g., accuracy, capacity, performance, reliability, robustness, or security) that are necessary to meet the primary mission of the system.
3. **Safety constraints** ñ any constraints (i.e., engineering decisions that have been selected to be mandated as requirements) that directly impact safety.
4. **Safety system requirements** ñ any requirements for safety systems as major components of systems of systems.

The primary purpose of safety engineering is to protect valuable assets from *accidental* harm[2], which is harm that

---

[1] It is very difficult to find intuitively obvious names that clearly distinguish these four types of safety-related requirements, especially the first two. There is little in the way of standard naming conventions because the different types are often confused with each other or themselves misleading (e.g., all safety-significant requirements are not safety-critical).

[2] This differentiates safety from security, which is concerned with protecting valuable assets from *malicious* and *unauthorized* harm. Also, security must protect valuable assets from ever-present threats, safety

is unplanned and unintended but not necessarily unexpected. Many accidents are caused by problems with system and software requirements, and ìempirical evidence seems to validate the commonly stated hypothesis that the majority of safety problems arise from software requirements and not coding errorsî [2]. Accidents typically arise from the occurrence of rare hazards, which are combinations of conditions that increase the likelihood of accidents causing harm to valuable assets such as people, property, or the environment. Requirements specifications are typically incomplete because they do not specify how the system should avoid or eliminate hazards and how the system should behave when hazards or safety incidents occur. Requirements specifications are also typically incomplete in that they usually do not define how the system should behave in all reasonable combinations of states or how the system should handle exceptional circumstances, such as mandatory exception handling.

The safety team of safety analysts and engineers typically produce a safety program plan and perform various types of safety analyses using such techniques as fault trees analysis (FTA), event tree analysis(ETA), hazard cause and effects analysis (HCEA), and failure mode and effect analysis (FMEA). However, their efforts are usually not integrated into the requirements specifications, and this makes it difficult to ensure that the architecture incorporates the appropriate safeguards.

2 SAFETY REQUIREMENTS

Safety requirements are those safety-related requirements that are specifically engineered to achieve mandatory amounts of a subfactor of the safety quality factor (i.e., requirements to protect assets from accidental harm, detect safety incidents, and respond to safety incidents). Whereas normal requirements specify what the system shall *do* or *make happen*, safety requirements specify what the system shall *not* do or *prevent from happening* [3].

Safety requirements are those requirements specifically engineered to achieve a specific minimum level of the quality attribute ìsafetyî. Whereas normal functional, data, and interface requirements specify what the system shall *do* or *make happen*, safety requirements specify what the system shall *not* do or *prevent from happening* [3].

Many safety requirements are specified to directly protect valuable assets from harm due to accidents resulting from hazards that cause safety risks. Whereas a gram of prevention is worth a kilogram of cure, sometimes accidental harm cannot be prevented and we must resort to cure. Thus, there should typically also be safety requirements for detecting the occurrence of safety incidents (an accident or near miss) and properly reacting to the occurrence of safety incidents:

- **Protect valuable assets requirements:**
  - **Asset/harm requirements**. This type of safety requirement sets a maximum acceptable limit on the amount of harm that may occur to an asset. The specified harm may be restricted to a specific asset or a type of asset. It can also be limited to a specific level (severity) of harm. For example, ìOn average, the automated airport transport system shall not accidentally injure more than one passenger seriously enough to require hospitalization per 50,000 passenger trips.î
  - **Accident requirements**. This type of safety requirement sets a maximum acceptable limit on the number or frequency of safety incidents (especially accidents) or types of safety incidents. For example, ìOn average, the automated airport transport system shall not allow more than one passenger to fall out of an open subway door per 50,000 passenger trips.î
  - **Hazard requirements**. This type of safety requirement sets a maximum acceptable limit on the frequency or duration of hazards (e.g., by type or specific hazard). For example, when specifying requirements for an automated subway system connecting the terminals of an airport, the combination of a moving train with open doors (two conditions) is clearly a hazard where the potential harm is to both passengers and their luggage (two types of valuable assets). A ëprotect valuable assets from hazardí safety requirement could be: ìOn average, the automated airport transit system shall not allow a subway train to be moving when one or more of its doors are open (the quality criterion) for a duration of more than one second more than once per year (the quality measure).î [5]
  - **Safety risk requirements**. This type of safety requirement sets a maximum acceptable limit on the level of risks (e.g., by risk category or by specific risk). For example, ìThe automated airport transit system shall not have any safety risks estimated as intolerable (i.e., safety integrity level = 5).î
- **Detect safety incident requirements:**

  This type of safety requirement specifies how a system must detect safety incidents that occur in spited of the preceding requirements for protecting valuable assets. For example, ìOn average, the automated airport

---

deals with relatively rare hazards, so that security attacks are common whereas safety incidents (i.e., accidents and near misses) should be very infrequent [1]. Thus, security requirements must protect against common dangers whereas safety requirements protect against hopefully very rare dangers, a contrast that has major implications concerning the verifiability of safety requirements.

transport system shall not accidentally injure more than one passenger seriously enough to require hospitalization per 50,000 passenger trips.î

- **React to safety incident requirements:**

  Safety requirements are often relatively reusable, especially within an application domain and across members of a product line. As quality requirements, safety requirements are typically of the form of a system-specific quality criterion together with a minimum or maximum required amount of an associated quality measure [4]. This structure means that safety requirements can often be written as instances of parameterized generic safety requirement templates. For example:

Type of Safety Requirement	**Form of Parameterized Requirement**
Prevention of Accidental Harm to Valuable Asset	The system shall *not* [cause | permit to occur] [amount of a type of harm to a type of asset] more than [a threshold of measurement].
Prevention of Safety Incidents (esp. Accidents)	The system shall *not* [cause | permit to occur] [optional: accident severity] [type of safety incident] more than [a threshold of measurement].
Prevention of Hazards	The system shall *not* [cause | permit to occur] [type of hazard] more than [a threshold of measurement].
Prevention of Safety Risk	The system shall *not* [cause | permit to occur] a [harm severity category] [accident | hazard] with likelihood greater than [probability | accident likelihood category]. *No* credible system [accident | hazard] shall represent a [threshold of measurement = safety risk category] safety risk.
Detection of Violation of Prevention	The system shall detect [accidental harm | safety incident | hazard | safety risk].
Reaction to Violation of Prevention	When the system detects [accidental harm | safety incident | hazard | safety risk], then the system shall [list of system actions].

3 SAFETY-SIGNIFICANT REQUIREMENTS

A *safety-significant requirement* is any non-safety primary mission requirement that can cause hazards and safety incidents (i.e., an accident or near miss). Safety-significant requirements include functional requirements, data requirements, interface requirements, and non-safety quality requirements[3] that are necessary to meet the primary mission of the system. Thus, safety-significant requirements have associated safety risks.

This paper recommends a relatively standard approach to categorizing safety risks in terms of accident severities and likelihoods. The resulting categorizations lay a foundation for the taxonomy of the different kinds of safety-significant requirements.

At the beginning of a project before asset, hazard, and safety risk analysis are performed, the safety team typically categorizes accident/hazard severities, accident/hazard likelihoods, and associated safety risks. These categorizations are then used during the rest of safety engineering as well as during the engineering of various types of safety-significant requirements.

Accident Severity Categorization

The severity of the harm that an accident can cause to a valuable asset varies from inconsequential to catastrophic. To make this continuum of damage manageable, the safety team typically categorizes it into a small number of severity levels. Accident severity levels are typically based on the worst credible impact of a type of accident or of an accident resulting from a given hazard. Instead of emphasizing only health safety, the safety engineers should take care to also include harm to each kind of valuable asset (i.e., life, property, *and* the environment). The actual levels and the boundaries between levels naturally will vary from project to project, although projects within certain application domains may share accident severity levels specified by international, military, or industry standards [6], [7], [8], [9], and [10]. An example of such an accident/hazard severity categorization is found in Table 1.

Accident/Hazard Likelihood Categorization

The probability of accidents can vary from relatively high to essentially zero. However, accident and hazard probabilities are often very difficult if not impossible to accurately and precisely estimate during the development of a complex system. This is especially true if the system contains significant amounts of software because the failure modes are discontinuous and thus difficult to predict. Therefore, accident and hazard likelihoods[4] are often divided into a small number of categories, typically having intuitive if somewhat ambiguous definitions. Cautious safety engineers take care to ensure that their likelihood categories are based on a relevant system-specific

---

[3] Note that a lack of adequate accuracy, capacity, performance, reliability, robustness, and security can all negatively impact the safety of a system

[4] The term ìlikelihoodî is used instead of probability because it better implies the lack of accuracy and precision.

timeframe, such as the estimated operational lifespan of the system. Similarly, accident and hazard likelihoods for individual systems should be differentiated from the much higher likelihoods of collections of similar systems. For example, the accident likelihood for an individual aircraft is less than the combined accident likelihood for all aircraft within a fleet. An example of such an accident/hazard likelihood categorization is found in Table 2.

Safety Risk Categorization

Using accident severity categories and accident frequency categories, the safety engineers typically produce a matrix of safety risk indices (individual cells in the matrix) that can be grouped into a smaller more manageable number of safety risk categories, a.k.a. safety integrity levels (SILs). An example of such a safety risk matrix is documented in Table 3, whereby its associated safety risk categories are defined in Table 4. Note that the actual values of the safety risk indices and the symmetry of the table will vary between projects, application domains, and associated international, military, and application domain specific standards.

Severity Level	Definition in terms of the Level of Harm to Valuable Assets
Catastrophic	• Loss of life (e.g., members of the public, users, and operators) • Life threatening or permanently disabling injury • Loss of system • Property/financial loss exceeding $1,000,000 • Irreversible severe environmental damage that violates a law or regulation
Severe	• Permanent partial disability • Injury or occupational illness requiring hospitalization of at least 3 individuals • Loss of subsystem • Property/financial loss exceeding $200,000 • Reversible environmental damage that violates a law or regulation
Major	• Injury or occupational illness resulting in one or more lost work days • Loss of component • Property/financial loss exceeding $10,000 • Mitigatible environmental damage where restoration can be accomplished without violating a law or regulation
Minor	• Injury not resulting in the loss of a work day • Property/financial loss less than or equal to $10,000 • Minimal environmental damage where restoration can be accomplished without violating a law or regulation
None	No harm is caused to any valuable asset

Table 1. Example Accident Severity Categories

Likelihood Category	Definition of Likelihood Category (for both accidents and hazards for both individual systems and for sets of systems)
High	• An accident will frequently occur during an individual systemís operational lifespan. • Rough estimate that an accident will occur during an individual systemís operational lifespan: 10^{-1} < probability <= 1. • The hazard is continuously present throughout an individual systemís operational lifespan. • Accidents will be continually happening during the combined range of the operational lifespans of all instances of the system.

Table 2. Example Accident/Hazard Likelihood Categories (continued on next page)

Likelihood Category	Definition of Likelihood Category (for both accidents and hazards for both individual systems and for sets of systems)
Moderate	• An accident will occur multiple times during an individual systemís operational lifespan. • Rough estimate that an accident will occur during an individual systemís operational lifespan: 10^{-2} < probability <= 10^{-1}. • The hazard often occurs during an individual systemís operational lifespan. • Accidents will often happening during the combined range of the operational lifespans of all instances of the system.
Low	• An accident is likely to occur once during an individual systemís operational lifespan. • Rough estimate that an accident will occur during an individual systemís operational lifespan: 10^{-3} < probability <= 10^{-2}. • The hazard will occur several times during an individual systemís operational lifespan. • Accidents will happen several times during the combined range of the operational lifespans of all instances of the system.
Remote	• It is highly unlikely although not impossible for an accident to occur during an individual systemís operational lifespan. • Rough estimate that an accident will occur during an individual systemís operational lifespan: 10^{-6} < probability <= 10^{-3}. • The hazard can be reasonably expected to occur during an individual systemís operational lifespan. • An accident is unlikely but may be reasonably expected to occur during the combined range of the operational lifespans of all instances of the system.
Negligible	• It is reasonable to expect that an accident will not occur during an individual systemís operational lifespan. • Rough estimate that an accident will occur during an individual systemís operational lifespan: 0 <= probability <= 10^{-6}. • It is reasonable to expect that the hazard will not occur during an individual systemís operational lifespan. • It is reasonable to expect that an accident will not occur to any system during the combined range of the operational lifespans of all instances of the system.

Table 2. Example Accident/Hazard Likelihood Categories (continued)

Safety Risk		Accident/Hazard Likelihood				
		High	**Moderate**	**Low**	**Remote**	**Negligible**
Accident Severity	**Catastrophic**	Intolerable	Intolerable	Intolerable	Critical	Major
	Severe	Intolerable	Critical	Critical	Major	Moderate
	Major	Critical	Critical	Major	Moderate	Moderate
	Minor	Critical	Major	Moderate	Minor	Minor
	None	None	None	None	None	None

Table 3. Example Safety Risk Matrix

Safety Risk	SIL	Definition / Safety Evidence Assurance Level (SEAL)
Intolerable Risk	5	• The safety risk is too high to be tolerated. • Any associated functional, data, and interface requirements have an intolerable risk and must be rejected.
Critical Risk	4	• The safety risk is very high. • Safety-critical requirements or components have sufficient risk to justify extensive enhancements to the development process. Extensive evidence of their proper implementation is needed to demonstrate adequate safety.
Major Risk	3	• The safety risk is high. • Requirements or components with this SIL have sufficient risk to justify major enhancements to the development process. Significant evidence of proper implementation is needed to demonstrate adequate safety.
Moderate Risk	2	• The safety risk is moderate. • Requirements or components with this SIL have sufficient risk to justify moderate enhancements to the development process. A moderate amount of evidence of proper implementation is needed to demonstrate adequate safety.
Minor Risk	1	• The safety risk is minor. • Requirements or components with this SIL have sufficient risk to justify only minor enhancements to the development process. Only a minor amount of evidence of proper implementation is needed to demonstrate adequate safety
No Risk	0	• There is no safety risk. • The normal development process is adequate. No special evidence is needed to demonstrate adequate safety is.

Table 4. Example Safety Risk Categories and Safety Integrity Levels

Requirements

The preceding categorizations lay the foundation for identifying, analyzing, and specifying the safety-significant requirements.

For example, suppose that you are specifying the requirements for an automated subway system connecting the terminals of an airport. Clearly, there will be functional requirements for opening and closing the subway doors and more functional requirements for starting, moving, and stopping at the next terminal. Although it is perfectly okay and even necessary to have the subway doors open or to have the subway move between terminals, it is clearly a hazard to have the doors open while the subway is moving because passengers and their luggage can fall out. Therefore the requirements for opening and closing doors as well as the requirements for starting, moving, and stopping are safety-significant functional requirements.

One common oversimplification is that safety and requirements engineers may consider only functional requirements, forgetting that data requirements, interface requirements, and certain quality requirements (e.g., reliability) can also result in safety hazards and associated safety incidents if not properly implemented. For example, incorrect implementation of data requirements regarding to the maximum permitted acceleration can cause people to fall. Similarly, incorrect implementation of interface requirements for communicating with the security system might let the subway stop at a station that has been closed do to fire or terrorist attack. Finally, poor reliability resulting in a critical failure or poor security resulting in a successful security breach can both lead to accidents.

Another common oversimplification is that safety-significant requirements are all grouped together and referred to as *safety-critical requirements*. Safety-significant requirements can and should be further classified in terms of the amount of their associated safety risk, which is typically defined as the probability that a hazard exists multiplied by the conditional probability that the hazardous conditions will result in an accident multiplied by maximum harm that the accident can cause. To deal with the resulting continuum of safety risks, they are usually further categorized into a small number of safety integrity levels (SIL) as documented in Table 4. For example, SIL = 5 might mean an intolerably high risk such as a significant probability of catastrophic harm, whereas SIL = 0 might mean that essentially no safety risk exists.

Thus, some functional, data, and interface requirements may have an associated safety risk that is intolerable (e.g., SIL = 5); such requirements should be rejected and thus should not be included as requirements in a requirements specification. Other requirements having a high SIL value (e.g., SIL = 4) are safety-critical, and a safety-critical system is one for which there exists at least one risk with a SIL greater or equal to 4 [8]. Because of this level of risk,

safety critical systems require a very high safety evidence assurance level (SEAL) for their correct implementation. For example, safety-critical requirements may need to be *formally* specified using a mathematically precise specification language to enable automatic source code generation of provably safe software. Similarly, *semi-formally* specification using modeling languages such as state charts and decision trees may be adequate for safety-major (SIL = 3) requirements, whereas *non-formal* specification using structured textual requirements may be adequate for safety-moderate (SIL = 2) and safety-minor (SIL = 1) requirements. Finally, a SIL value of 0 means that there is essentially no associated safety risk, so that such functional, data, and interface requirements can be considered to be safety-independent requirements.

The identification of safety-significant requirements will depend on (1) the prior existence of functional, data, interface, and quality requirements to be categorized by SIL value, (2) the existence of a list of categorized hazards and associated safety risks, and (3) the performance of an associated safety risk analysis (e.g., based on fault-tree, event-tree, and similar techniques) to identify the safety risks associated with the individual functional, data, and interface requirements.

4 SAFETY CONSTRAINTS

A constraint is a business rule or engineering decision that is treated during requirements engineering as if it were a requirement even though it would ordinarily be made during architecture development, design, implementation, integration, or testing. A safety constraint is often a mandated safety policy or a mandated safeguard such as a hardware interlock, barrier around moving parts, handling procedures for toxic materials, and safety procedures. Whereas safety constraints are in many ways no different than other types of constraints, because they are specified for safety reasons, they are subject to safety certification and accreditation like other safety-related requirements. Some safety constraints are required by a relevant regulation, standard, or law. In fact, some safety constraints merely mandate compliance with such a regulation, standard, or law, and therefore act as a way to group the numerous constraints included in the regulation, standard, or law. On the previously mentioned example airport transit system, the following standards might be mandated: ASCE 21-96 [5] or IEEE P1474.1 [11]. Relevant example safety constraints include:

- ìThe airport transit system shall comply with *Automated People Mover Standards, Part 1*, ASCE 21-96.î
- ìSystem safety shall not depend on the correctness of actions taken or procedures used by operating personnel.î
- ìNo credible single point hardware failure, whether self-revealing or non-self-revealing, shall cause an unsafe condition.î
- ìThe doors shall be disabled from opening unless: (1) the train is at a designated stopping point within designated tolerances, (2) zero movement is detected, and (3) the train is constrained against movement.î

5 SAFETY SYSTEM REQUIREMENTS

The three types of safety-related requirements that have been discussed so far (i.e., safety-significant requirements, safety requirements, and safety constraints) typically specify aspects of the primary system to be built. However if there is a significant risk associated with the primary system, then it may well have major mandatory safety subsystems or it may be developed in conjunction with one or more safety systems, whereby these safety systems or subsystem only exist to ensure the safety of the primary system. The classic example of such safety systems is the emergency coolant system of a nuclear power plant. Every functional, data, interface, and quality requirement of the safety system may well have significant safety implications. These requirements for the safety system or subsystem are safety system requirements, the fourth and final category of safety-related requirements in our taxonomy.

6 SAFETY REQUIREMENTS TAXONOMY

As illustrated in Figure 1, the preceding four basic classifications of safety-related requirements can be placed into a larger taxonomy of requirements types. At the lowest level of abstraction near the bottom of the figure, all of the system requirements are classified as functional requirements, data requirements, interface requirements, quality requirements, or constraints. In a separate orthogonal inheritance subtree extending to the right, all system requirements (i.e., all functional, data, interface, and quality requirements as well as constraints) can also be classified into main mission system requirements and **safety system requirements**. Thus, the taxonomy uses multiple classification to produce two independent overlapping inheritance structures.

On the left side of Figure 1, we see that **safety-significant requirements** are all system functional, data, interface requirements that have safety ramifications combined with all non-safety quality requirements (i.e., all other quality requirements not specifically mandating minimal acceptable amounts of safety such as availability, capacity, performance, portability, security, and usability requirements). The figure also clearly shows that all safety-significant requirements are not safety critical requirements, but only those with safety integrity level 4 (or those with the equivalent program-specific safety integrity level). The top branches of the safety requirements taxonomy classify the **safety requirements** (those directly specifying an amount of the safety quality factor) into two independent inheritance trees: one tree addressing the goals of protection, detection, and reaction

while the second tree captures the requirements resulting from the first four types of safety analysis. Finally, **safety constraints** are a subset of constraints that mandate specific safeguards and safety mechanisms.

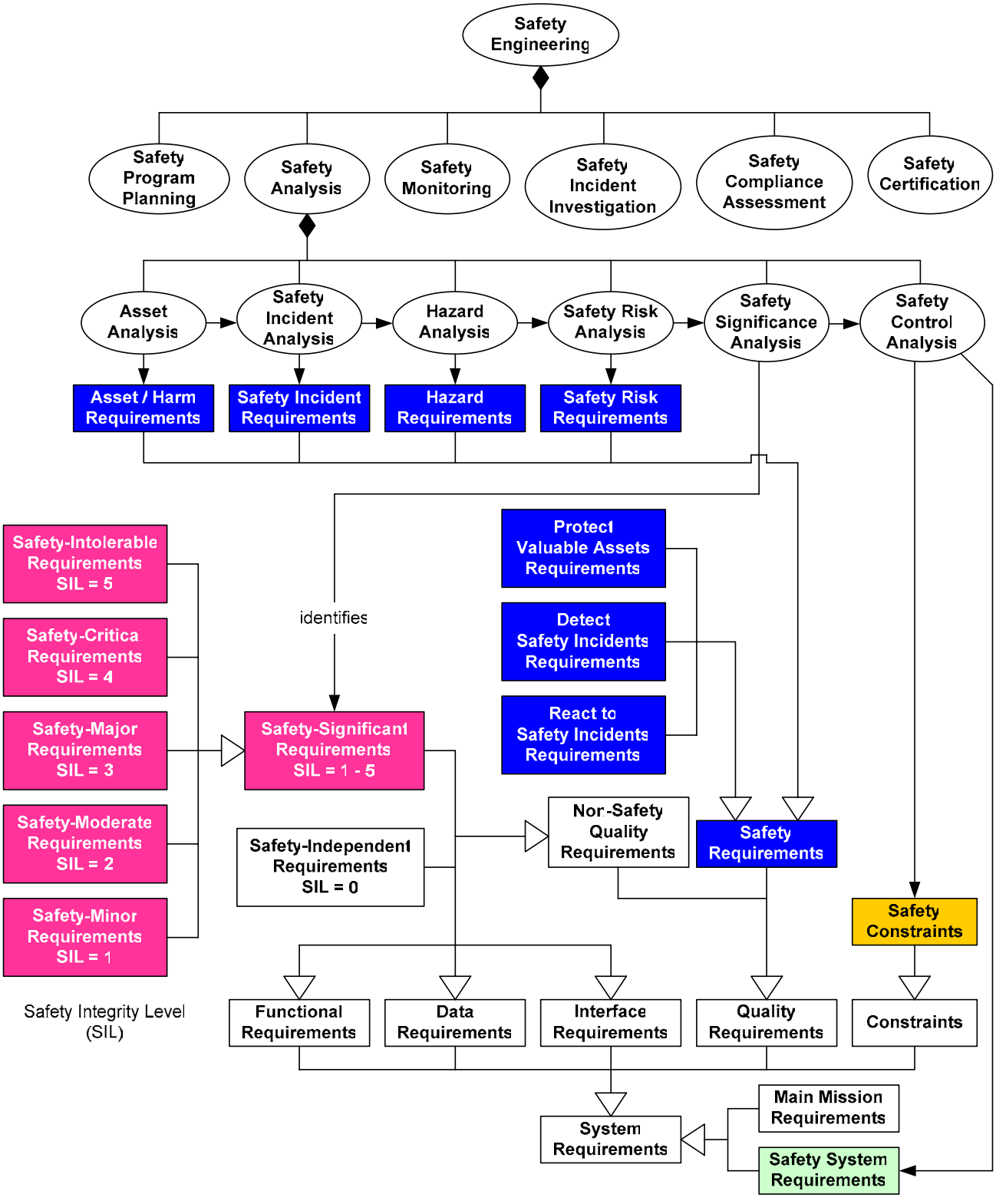

Figure 1. Taxonomy of Safety-Related Requirements

7 CONCLUSION

There are several different kinds of safety-related requirements and using a standard taxonomy to organize them can have the following benefits:

- The taxonomy can help requirements and safety engineers ensure that no significant types of safety-related requirements fall through the cracks during requirements and safety engineering.
- The taxonomy captures different types of safety-related requirements that are currently being engineered on real projects.
- The different types of safety-related requirements in the taxonomy have different sources (e.g., different stakeholders and documents).
- The different types of safety-related requirements are elicited and analyzed differently. Thus, ìlimit harmî safety requirements can be derived from business goals. ìLimit accidents/hazards/risksî safety requirements can be derived from an asset/hazard/risk analysis. Safety significant requirements are typically derived the same way and at the same time as any other functional, data, and interface requirements; it is only after they exist that they are analyzed for their safety implications. Safety constraints typically come from regulations, laws, standards, and best industry practices.
- The different types of safety-related requirements are specified differently. Safety-significant requirements are primarily specified with the other functional requirements (and data and interface requirements). Their safety aspects are specified as metadata (attributes). On the other hand, safety requirements are typically specified with the other quality requirements. Similarly, safety constraints are typically specified with the other constraints. Finally, safety system requirements are specified separately in the requirements specification for the safety system or subsystem.
- The different types of safety-related requirements typically have different reuse potentials. Safety requirements and safety constraints are often very reusable. For example, safety requirements can be reused as parameterized templates based on the quality model used [12]. Safety-significant requirements tend to be much more application specific and less reusable.

FUTURE WORK

This taxonomy is being used as the foundation for an Independent Research and Development (IRAD) project at the SEI which is producing technical notes on reusable safety requirements and a process for engineering safety requirements. Other future work to be considered might include researching the following questions:

- How do the different kinds of safety-related requirement vary in terms of their reuse potential?
- How do the different kinds of safety-related requirements vary in terms of their optimum identification, analysis, and specification techniques?
- How should the efforts of the safety and requirements teams be better coordinated?

ACKNOWLEGEMENTS

I would like to thank my colleagues Peter Capell, Tim Morrow, and Mary Popeck of the SEI as well as Daniel Berry and Paul Tiplady for their useful review comments. Of course, any remaining defects are my own.

STIPULATIONS

The views and conclusions contained in this position paper are solely those of the author and should not be interpreted as representing official policies, either expressed or implied, of the Software Engineering Institute, Carnegie Mellon University, the U.S. Air Force, the U.S. Department of Defense, or the U.S. Government.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

REFERENCES

1. Donald G. Firesmith, *Common Concepts Underlying Safety, Security, and Survivability Engineering,* Technical Note CMU/SEI-2003-TN-033, Software Engineering Institute, Pittsburgh, Pennsylvania, December 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tn033/03tn033.html
2. Nancy G. Leveson, *Software: System Safety and Computers*, Addison-Wesley, 1995.
3. Amer Saeed, RogÈrio de Lemos, and Tom Anderson, ìThe Role of Formal Methods in the Requirements Analysis of Safety-Critical Systems: a Train Set Example,î *Proceedings of the 21st Symposium on Fault-Tolerant Computing*, Montreal, Canada, pp. 478-485.
4. Donald G. Firesmith, ìUsing Quality Models to Engineer Quality Requirements,î *Journal of Object Technology (JOT)*, vol. 2, no. 5, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, September/October 2003, pp. 67-75. http://www.jot.fm/issues/issue_2003_09/column6
5. Automated People Mover Standards Committee, *Automated People Mover Standards, Part 1*, ASCE 21-96, 1996. http://www.apmstandards.org
6. Department of Defense, *Mishap Risk Management*, MIL-STD-882D, 1998.
7. European Community, *Software for Railway Control and Protection Systems*, CENELEC EN 50128, 1997.
8. Radio Technical Commission for Aeronautics (RTCA), *Software Considerations in Airborne Systems and Equipment Certification*, RTCA/DO 178B, 1992.

9. European Space Agency (ESA), *Space Product Assurance: Safety*, ECSS-Q-40A, 1996.

10. International Electrotechnical Commission (IEC), *Medical Electrical Equipment - Part 1: General Requirements for Safety*, IEC 601-1-4, 1996.

11. IEEE, *Communications-Based Train Control (CBTC) Functional and Performance Requirements*, IEEE P1474.1.

12. Donald Firesmith, ìEngineering Safety Requirements, Safety Constraints, and Safety Critical Requirements,î *Journal of Object Technology (JOT)*, vol. 3, no. 3, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, March-April 2004, pp. 27-42. http://www.jot.fm/issues/issue_2004_03/column3