SEI Insights

SEI Blog

The Latest Research in Software Engineering and Cybersecurity

A Collaborative Method for Engineering Safety- and Security-Related Requirements

POSTED ON SEPTEMBER 26, 2011 BY **DONALD FIRESMITH** [/AUTHOR/DONALD-FIRESMITH] IN **ACQUISITION** [HTTPS://INSIGHTS.SEI.CMU.EDU/SEI_BLOG/ACQUISITION/]

Background: In our **research** [http://www.sei.cmu.edu/research/] and **acquisition** [http://www.sei.cmu.edu/acquisition/] work on commercial and Department of Defense (DoD) programs, we see many systems with critical safety and security ramifications. With such systems, safety and security engineering are used to managing the risks of accidents and attacks. Safety and security requirements should therefore be engineered to ensure that residual safety and security risks will be acceptable to system stakeholders. The **first post** [http://insights.sei.cmu.edu/post.cfm/the-importance-of-safety-and-security-related-requirements] in this series explored problems with quality requirements in general and safety and security requirements in particular. The **second post** [http://insights.sei.cmu.edu/post.cfm/obstacles-in-engineering-safety-and-security-related-requirements-second-in-a-three-part-series], took a deeper dive into key obstacles that acquisition and development organizations encounter concerning safety- and security-related requirements. This post introduces a collaborative method for engineering these requirements that overcomes the obstacles identified in earlier posts.

Anyone involved in building safety- and security-critical systems needs to consider the following:

1. Are you building a safety-critical system or one that must be secure from attack?
2. Do your safety and security engineers begin their work only after the architecture is engineered, rather than building it in from the start via safety- and security-related requirements?
3. Do your safety and security engineers develop their work products (documents and models) independently of each other and requirements engineers?
4. Do your requirements specifications largely ignore safety, security, or both?
5. Are many of your safety and security requirements so general that they are meaningless, such as "The system shall be safe and secure from attack?"
6. Are most of your safety- and security-related requirements merely architecture and design constraints that prevent safety- and security-engineers from collaborating with architects to create innovative solutions?
7. Is use-case modeling or structured analysis your primary or only requirements-analysis method, even when engineering safety- and security-related requirements?

If you answer yes to any of these questions, then your safety, security, and requirements engineers can benefit from a better way of engineering their requirements. To achieve this goal, an appropriate safety- and security-requirements analysis method is needed.

We propose using the **Engineering Safety- and Security-related Requirements (ESSR) method** [http://www.sei.cmu.edu/library/abstracts/presentations/icse-2010-tutorial-firesmith.cfm], which consists of the following analysis-based tasks.

- **Stakeholder analysis** determines the stakeholders who have a vested interest in the safety and security of the system and the appropriate sources for eliciting safety and security goals and requirements. Safety- and security-engineers collaborate to identify the safety- and security-related stakeholders in the system and the assets that the system must defend from accidental and malicious harm. These stakeholders are modeled by producing stakeholder profiles and creating an initial partial list of the stakeholder's safety- and security-goals.
- **Asset analysis** determines the assets that must be protected from unauthorized harm and the harm that these assets must be protected from. Safety- and security-engineers collaborate to identify the assets that the system must protect from harm. They model each defended asset by categorizing it, determining its value, identifying the types and severities of harm that it may suffer, and determining its stakeholders.
- **Abuse analysis** examines the ways that the system and the assets for which the system is responsible can be abused. Specifically, this task identifies the different types of abuses including safety mishaps (accidents and safety incidents) and security misuses (attacks and security incidents) that can occur. Abuse analysis also identifies which assets that the abuses can harm,

in what manner, and to what degree. Safety- and security-engineers model these abuses using appropriate techniques (e.g., **abuse case modeling**, **attack trees**) and create abuse profiles.

- **Vulnerability analysis** determines the existence of the system-internal weaknesses or defects that can enable abuses (mishaps and misuses) to occur. Safety- and security engineers identify the credible potential system-internal vulnerabilities (e.g., defects and weaknesses) that could enable the abuses that may harm the defended assets. They also model these vulnerabilities using appropriate techniques such as **STAMP-Based Process Analysis (STPA)**, **Event Tree Analysis (ETA)**, **Fault Tree Analysis (FTA)**, or **Failure Modes and Effects Analysis (FMEA)**.
- **Abuser analysis** determines the system-external people and things that can accidentally or maliciously abuse the system and the assets that it must defend from unauthorized harm. Safety- and security engineers identify the credible potential abusers that could exploit the vulnerabilities and thereby cause the abuses that may harm the defended assets. They model these abusers using appropriate techniques (e.g., **STPA**, **abuse case modeling**, **task analysis**, or **user profiling**).
- **Danger analysis** determines the dangers (i.e., safety hazards and security threats), which are cohesive sets of conditions involving the existence of abusers, vulnerabilities, and assets that could increase the probability of abuses occurring. When restricted to safety and security, danger analysis is often called hazard analysis or threat analysis, even though they typically include all of these types of analysis. Safety- and security engineers model these safety hazards and security threats using appropriate techniques (e.g., **operator task analysis**, **ETA**, **FTA**, and **FMEA**).
- **Risk analysis** determines the maximum acceptable residual safety and security risks as well as the specific types of assets, harm, vulnerabilities, abusers, and dangers that are associated with these risks. Safety- and security engineers model these risks using appropriate techniques (e.g., **calculating risk level as the product of probability times harm severity**, using degrees of software control instead of probabilities, and **risk matrices**).
- **Safety- and security-significance analysis** identifies the goals and requirements that have safety and security ramifications so the corresponding parts of the system can be implemented using a process having the appropriate level of rigor and completeness, e.g., to justify the use of a more powerful (and therefore more expensive) development process. Safety- and security engineers categorize requirements into **safety/security assurance levels (SALs)**, such as safety-critical and security-critical, based on the degree to which the requirements have safety and security ramifications. They collaborate with requirements engineers to update the requirements repository by annotating requirements with their SALs. Based on how these categorized requirements are allocated to architectural components, they assign the components safety/security evidence assurance levels (SEALs) that determine the degree of completeness and rigor to be used when architecting, designing, implementing, integrating, and testing these

components. In other words, components with high SEALS should be as small as practical to minimize the increased effort, cost, and schedule needed to develop them. Finally, they update the certification repository with the results of safety- and security-significance analysis.

- **Defense determinatio**n determines the appropriate defenses (i.e., controls including safeguards and security countermeasures) that are needed to defend the system and its associated defended assets from unauthorized harm. Safety- and security engineers perform a gap analysis to identify potential new defenses. They then evaluate these potential defenses using appropriate techniques (e.g., engineering analyses, product and vendor trade studies).

Where appropriate (except for the safety- and security-significance analysis task), safety- and security engineers create safety and security goals for each type of analysis and then collaborate with the requirements engineers to transform these goals into requirements to prevent, detect, and react to it. They then update the certification repository with the results of the analysis. Also, where appropriate, they collaborate with requirements engineers to transform these informal restraints into official requirements. Finally, where appropriate, this information is stored in the certification repository to eventually support the system's safety and security accreditation and certification.

The above tasks result in the engineering of multiple types of associated safety and security requirements (e.g., prevention, detection, and reaction requirements as well as safety and security constraints). All such possible requirements, however, are rarely appropriate for most systems. The harm severity and likelihood of the associated mishaps and misuses may not justify the cost of producing and using the resulting safety- and security-defenses. Some requirements make others unnecessary, e.g., a requirement preventing the existence of a vulnerability may eliminate the need for a requirement to prevent an abuse enabled by that vulnerability. On the other hand, high-level requirements associated with the early analysis steps (e.g., prevent harm to a defended asset) may be used to derive lower-level requirements associated with later analyses steps (prevent vulnerability that enables abuse to harm the defended asset).

The tasks of ESSR described above are best performed in an evolutionary (i.e., incremental, iterative, and concurrent) manner. Due to the evolutionary nature of ESSR, the temporal ordering of the preceding sequence of analyses is merely a logical simplification to improve understandability; a waterfall approach to safety and security is neither intended nor recommended. Safety, security engineers, and requirements engineers should also perform these tailorable tasks in a collaborative manner. At the end of this process, comprehensive safety and security analyses will have been performed and documented, safety and security goals will have been turned into their corresponding requirements, and the certification repository will contain the analysis- and requirements-related safety and security evidence needed for accreditation and certification.

The preceding ESSR method for collaboratively engineering safety- and security-related requirements is described in considerably more detail in **tutorials** [http://www.sei.cmu.edu/library/abstracts/presentations/icse-2010-tutorial-firesmith.cfm], a **class** [http://www.sei.cmu.edu/training/P64.cfm], and a book to be published early in 2012.

Additional Resources:

For more information, please visit
www.sei.cmu.edu/library/abstracts/presentations/icse-2010-tutorial-firesmith.cfm
[http://www.sei.cmu.edu/library/abstracts/presentations/icse-2010-tutorial-firesmith.cfm]

About the Author

Donald Firesmith

Contact Donald Firesmith [https://www.sei.cmu.edu/contact.cfm]
Visit the SEI Digital Library for **other publications by Donald**
[http://resources.sei.cmu.edu/library/author.cfm?authorID=4637]
View **other blog posts by Donald Firesmith**
[/author/donald-firesmith]

Defense (DoD). It is operated by Carnegie Mellon University.