SEI Insights

SEI Blog

The Latest Research in Software Engineering and Cybersecurity

The Importance of Safety- and Security-related Requirements, First of a Three-Part Series

POSTED ON JUNE 27, 2011 BY **DONALD FIRESMITH** [/AUTHOR/DONALD-FIRESMITH]

In our research and acquisition work on commercial and Department of Defense (DoD) programs ranging from relatively simple two-tier data-processing applications to large-scale multi-tier weapons systems , one of the primary problems that we see repeatedly is that requirements engineers tend to focus almost exclusively on functional requirements and largely ignore the so-called nonfunctional requirements, such as data, interface, and **quality requirements** [http://www.computer.org/portal/web/swebok/html/ch11], as well as technical constraints. Unfortunately, this myopia means that requirements engineers overlook critically important, architecturally-significant, quality requirements that specify minimum acceptable amounts of qualities, such as availability, interoperability, performance, portability, reliability, safety, security, and usability. This blog post is the first in a series that explores the engineering of safety- and security-related requirements.

Quality requirements are essential to a system's architecture and its acceptability by stakeholders. There are several reasons, however, why quality requirements are rarely well specified. Functional requirements are central to how stakeholders tend to think about the system (i.e., what functions the systems performs for its users). Popular requirements engineering techniques, such as **use case modeling** [http://en.wikipedia.org/wiki/Use_case], are effective for identifying and analyzing functional requirements. Unfortunately, these techniques are inadequate and inappropriate for non-functional requirements, which include quality requirements, as well as interface requirements, data requirements, and architecture/design constraints. By specifying how well the system performs its functions, quality requirements logically follow functional requirements.

Most acquisition programs do not explicitly use quality models, which are used to define the different types of quality, their units of measure, and associated metrics (e.g., as defined in **ISO/IEC 9126-1 Software Engineering - Product Quality - Quality Model** [http://en.wikipedia.org/wiki/ISO/IEC_9126]). Without relatively complete quality models, stakeholders and developers are often unaware of--and tend to overlook--the many types of quality. It is also hard for many stakeholders to specify the required level of these qualities. Requirements engineers rarely receive any training in identifying and specifying quality requirements and thus have far less experience engineering them because they are often considered the responsibility of specialty engineering groups, such as reliability, safety, security, and usability (human factors).

While many types of quality requirements are important, safety and security requirements are two of the most vital; almost all major commercial and DoD systems have significant safety and security ramifications, and many are safety- and security-critical. It is far better to build safety and security into a system than to add them once a system's architecture has been completed, much less after the system exists and has been fielded. Yet system requirements rarely specify how safe and secure a system must be to adequately defend itself and its associated assets (people, property, the environment, and services) from harm. Far too often, requirements do not specify what accidents and attacks must be prevented, what types of vulnerabilities the system must not incorporate, what hazards and threats it must defend against, and what the maximum acceptable safety and security risks are.

How big is this problem? On production projects, poor requirements lead to budget and schedule overruns, missed or incorrectly implemented functionality, and systems that are delivered but never used. According to **Nancy Leveson** [http://sunnyday.mit.edu/] , a respected expert in software safety, **up to 90 percent of all accidents are caused, at least in part, by poor requirements** [http://sunnyday.mit.edu/safer-world/safer-world.pdf] . For example, conventional requirement documents typically do not specify what systems should do in unlikely situations when

- valuable assets are harmed
- accidents and attacks occur
- system internal vulnerabilities exist
- system external abusers exploit these vulnerabilities

- safety hazards and security threats exist and
- safety and security risks are high

Requirements also rarely specify that the system must detect when these safety- and security-related events occur or conditions exist. Similarly, the requirements often don't specify what the system must do when it detects them. To summarize, requirements typically do not adequately specify the safety and security-related problems the system must prevent, the system's detection of these problems, and how the system must react to their detection.

The next post in this series will discuss the obstacles that acquisition and development organizations face when engineering safety- and security-related requirements. Our final post in the series will present a collaborative method for engineering these requirements based on

- a common ontology of the concepts underlying safety and security
- a clear model (including proper definitions) of the different types of safety- and security-related requirements and
- a shared set of safety and security analysis techniques that is useful to engineers from both disciplines.

Additional Resources:

To view a tutorial on engineering safety-and security-related requirements for software-intensive systems, 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]

To read an SEI technical note on the common concepts underlying safety, security, and survivability engineering, please visit
www.sei.cmu.edu/library/abstracts/reports/03tn033.cfm
[http://www.sei.cmu.edu/library/abstracts/reports/03tn033.cfm]

To read an SEI technical report on the Security Quality Requirements Engineering (SQUARE) method for engineering security requirements, please visit
www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm
[http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm]

< Previous Article
Next Article >

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]