

Requirements Elicitation and Analysis Processes for Safety and Security Requirements

Nancy R. Mead

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA. 15213 USA
+1 412 268 5756
nrm@sei.cmu.edu

ABSTRACT

This paper describes a process framework for the elicitation of safety and security requirements and early experience in applying the framework. A larger research project that provides context for the process is briefly described.

Keywords

Computer safety requirements, information security requirements, requirements elicitation, requirements analysis, requirements elicitation process

1 INTRODUCTION

The elicitation and analysis process framework is part of a larger research project, System Quality Requirements Engineering (SQUARE), aimed at improving safety and security requirements engineering practices. The goal for the process framework is to identify and/or develop a recommended process and set of associated techniques for eliciting and analyzing safety and security requirements. Some of the steps to accomplish this goal include the following:

- Identify current processes and techniques for eliciting and analyzing quality requirements.
- Determine the relative strengths and weaknesses of these processes and techniques.
- Determine why these processes and techniques are not being widely used.
- Develop a candidate process and associated set of techniques based on the study of existing processes and techniques.
- Potentially adjust this process based on analysis of SEI experience in security and safety.
- Prototype the use of this process and its techniques.
- Iterate and document the resulting recommended process and set of techniques.

An earlier report has provided a summary of current research in requirements engineering for survivable systems [1]. Earlier papers have also provided research

insights [2]. In this paper I describe an elicitation and analysis process that I believe will be useful for safety and security requirements.

2 DRAFT REQUIREMENTS ELICITATION AND ANALYSIS PROCESS

In this process I draw on successful tried and true requirements elicitation methods, as well as newer research methods of the past few years [3, 4, 5]. Significant work has been done recently in the area of risk assessment for survivable systems. A brief discussion of each of the steps follows, and the steps are summarized in Table 1.

Step 1: Candidate definitions for standard security terms, such as confidentiality, integrity, and availability, are presented to stakeholders. The stakeholders have the option to add to the definitions and then to select a final set of definitions for the project. This heads off a debate on the meaning of each term and prevents the confusion that can result when one stakeholder has his or her own idea of what a definition should be and another stakeholder has a different idea.

Step 2: Using business drivers and goals, mission statements, and other artifacts, develop a set of security and survivability goals. This can be a challenge because stakeholders do not necessarily have written business goals, so they may have to be developed first before security goals can be addressed.

Step 3: Select the techniques to be used for requirements elicitation. There is a wide variety of elicitation techniques, and their effectiveness can vary depending on the size of the organization and the formality of their development process. It's a good idea to pick a few techniques and then narrow them down before proceeding with the next steps.

Step 4: Develop the artifacts necessary to support the elicitation technique. Use cases and misuse cases are popular tools to support security requirements elicitation. Threat models and attack patterns also provide useful input. For reuse, templates can be introduced at this point in the process. This can be a very time-consuming activity, as

many organizations don't have the needed artifacts for the "normal" case, such as use cases and architecture diagrams.

Step 5: Elicit the actual requirements using the techniques selected in step 4 and the support of the artifacts developed in step 4. One difficult area is the phrasing of the requirements so that they represent real, testable requirements rather than architecture/design constraints or lofty goals that cannot be met.

Step 6: Categorize the requirements and assess whether they are really requirements or other kinds of constraints, as noted in step 5.

Step 7: Perform a risk assessment activity. There is a wide variety of choices, including traditional software risk assessment, OCTAVE [3], or other methods [4, 5].

Step 8: Select a method for prioritizing requirements and go through the prioritization process. Priorities could be based on the likelihood that the risk will become a reality, cost/benefit analysis, areas of particular concern for the stakeholders, etc.

Step 9: Inspect the requirements using a standard inspection or review process to ensure that they are consistent, complete, testable, etc.

This summer's work, described below, shows that the draft process can be applied to systems concerned with security properties, and it may be applicable to systems with safety properties as well.

3 EXPERIENCE WITH THE DRAFT PROCESS

I am in the early stages of applying the process with real clients. So far, I am finding that organizations are very interested in the work that I am doing and they want to improve their processes. However, they are very busy with development work and have little time for process improvement. It is also the case that I have to start with the basics, such as defining terms, developing architectural diagrams, and helping to identify business goals and mission statements. Small to mid-size companies typically do not have the infrastructure to just add a new process and run with it. Thus, I am spending significant time to get the organizations to the point where they can benefit from new processes and tools. It remains to be seen whether the ideal time to apply this process is after a system architecture exists or prior to architecture development. Initial findings suggest that an architecture needs to be in place at some level in order for this process to be successful.

This summer, I supervised seven graduate students in applying the process and developing some rudimentary tools. Five students worked as a team with a client, under my supervision. Two student interns worked on extending the process descriptions and developing some basic Web-based tools, also under my supervision.

A client organization we were working with initially had to withdraw due to the pressure of tight deadlines, a problem that is all too common in today's business environment. The current client organization has been very responsive and supportive of the work, and has stayed with us through completion of the pilot exercise. The current client is also willing to continue in-depth analysis of some of the steps in the fall, with a different student group. Some of the current students may continue with independent studies or further internships rather than the team project. This client exercise focused on security requirements.

Results and Lessons Learned from the Summer Experience

The student project was very successful in providing feedback on the method and in providing the client with suggested improvements in their information security posture.

Here's a synopsis of the findings:

The client, a small start-up company, was in the process of expanding the market for an existing product to include users who might be installing the product on networks with Internet access. The current users typically install the product on isolated networks, where security concerns are minimal. Hence the client was motivated to review and improve their security posture.

As a first step, I described the research project to the client, and they were interested in working with the students. I requested copies of existing documentation, which were provided to the students, and arranged a subsequent student/client meeting.

It quickly became clear that the client organization had user documentation but little, if any, architecture or requirements documentation. This is a fairly typical state of affairs for many organizations. Hence, the students had to spend a fair amount of time eliciting and documenting the architecture, business goals, and other requirements. Since there was a large team of students, many activities took place in parallel, rather than sequentially.

There is not sufficient definition of the process itself. Each process step needs to be broken down into substeps, with explanatory notes. This is an exercise for me, with review by research colleagues. The students needed a lot of guidance because the steps were not sufficiently detailed or self-explanatory.

For step 1, a set of candidate definitions was developed. Definitions were recommended to the client, and the client picked a final set of definitions to be used for the project. A similar exercise took place in identifying the business goals, in this case associated with security.

Most of the time this summer was focused on steps 3 through 5, which were executed in an iterative process. Step 3, selecting elicitation techniques, was not really done.

The students just iterated with the client to identify use and misuse cases. This occurred in part because there was one individual who served as the primary client, and a second who provided higher level review. As a result, some of the difficulties that occur with large groups of stakeholders did not take place, and surveys and other sophisticated techniques were not needed. Additionally, it may be the case that stakeholder selection is important at each step, and different stakeholders may need to be involved at different steps in the process. This is an area that could use more study in the fall.

One student subteam developed a set of use cases based on working on the actual client product at the client location. Another student subteam developed a set of misuse cases. Much time was spent iterating on the misuse cases to decide the best way to document them. The use cases and misuse cases were ultimately documented in both tabular form and in Visio. In addition to the misuse cases, some attack trees were developed to test completeness of the misuse cases. Unfortunately, there was insufficient time to develop a complete set of attack trees, so this activity will be picked up in the fall. It has become clear that a threat model, based on attack trees or some other method, is needed in order to develop a good set of requirements [6].

Once the use and misuse cases were done, architectural and policy recommendations for improving security were developed. These were traced to the misuse cases. From these recommendations, the students developed a set of implementation choices for the client. At this point, the students were ready to abstract these recommendations into a set of requirements. Ideally we wanted to trace the security goals to requirements, which in turn would trace to architectural recommendations and implementation choices. This turned out to be a difficult task, in part because the students had no experience with requirements engineering, and in part because we had difficulty writing a security requirement that is free of implementation details. We believe that development of the threat model will help with this.

One of the students did a risk assessment (step 7), but this was relatively independent of the other activities. The risk assessment would also help in defining requirements, so this is another loop that needs to be closed.

In parallel with this, the students attempted to prioritize the misuse cases, with input from the client, do a cost/benefit analysis, and use the cost/benefit analysis to help with the prioritization. Initially the students and the client prioritized the misuse cases as high/medium/low (there were no cases categorized as low in the end). The high priority cases were further addressed with the cost/benefit analysis. Several prioritization and analysis models were considered before arriving at the final selection.

In the end, there was insufficient time for the inspection process. This will also be picked up in the fall.

Considering that there were seven students working for 12 weeks at an average of 30 hours a week, it's clear that the process is quite time consuming. However, some of the time was spent in learning about the requirements area and also in documenting the current goals and architecture on behalf of the client.

RHAS Workshop

The students have completed their work for the summer, and there is good feedback on the usefulness of the process, as well as further documentation of the process. The students produced two large reports – one report for the client with the actual results of the process, and another report providing feedback on the process itself.

I am hoping that the workshop participants will provide professional feedback on the content of the process, whether it applies equally to safety and security and, if not, what the differences are. I am also hopeful that the workshop participants will provide advice on how to identify appropriate clients for applying the draft process (that is, what would be the characteristics of clients who would be able to apply new processes without having to start with fundamentals).

4 FUTURE WORK

My intent is to refine the process and gain more experience with a variety of clients. Toward the end of the summer we uncovered a number of new references and practices that would help to identify better security requirements [7]. These will be factored into our thinking. I will continue to track the literature for new approaches to safety and security requirements elicitation and analysis. Research in this area is needed to improve the safety and security of critical systems.

•

REFERENCES

1. Mead, N. R. *Requirements Engineering for Survivable Systems* (CMU/SEI-2003-TN-013, ADA418410). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tn013.html>
2. Linger, R. C.; Mead, N. R.; & Lipson, H. F. "Requirements Definition for Survivable Systems," 14-23. *Third International Conference on Requirements Engineering*, Colorado Springs, CO, April 6-10, 1998. Los Alamitos, CA: IEEE Computer Society, 1998.
3. Alberts, C. & Dorofee, A. *Managing Information Security Risks: The OCTAVE Approach*. New York: Addison Wesley, 2003.

4. Butler, S. A. & Fischbeck, P. "Multi-Attribute Risk Assessment." *SREIS 2002, Second Symposium on Requirements Engineering for Information Security*, Raleigh, NC, October 16, 2002. Lafayette, IN: CERIAS, Purdue University, 2002.
5. Feather, M. S. "A Risk-Centric Decision Process." *Software Engineering for High Assurance Systems (SEHAS) 2003*, Portland, OR, May 9-10, 2003. <http://www.sei.cmu.edu/community/sehas-workshop/feather/>
6. Howard, M. & Lipner, S. "Inside the Windows Security Push." *IEEE Security & Privacy* 1, 1 (2003): 57-61.
7. Moffett, J. D.; Haley, C. B.; & Nuseibeh, B. *Core Security Requirements Artefacts*. Technical Report 2004/23, ISSN 1744-1986, Open University, UK. <http://computing.open.ac.uk>

Table 1 Security & Safety Requirements Elicitation and Analysis Process

Step No.	Step	Input	Techniques	Participants	Output
1	Agree on definitions	Candidate definitions from IEEE and other standards	Structured interviews, focus group	Stakeholders, requirements team	Agreed-to definitions
2	Identify safety and security goals	Definitions, candidate goals, business drivers, policies and procedures, examples	Facilitated work session, surveys, interviews	Stakeholders, requirements engineer	Goals
3	Select elicitation techniques	Goals, definitions, candidate techniques, expertise of stakeholders, organizational style, culture, level of safety and security needed, cost benefit analysis, etc.	Work session	Requirements engineer	Selected elicitation techniques
4	Develop artifacts to support elicitation technique	Selected techniques, potential artifacts (e.g., scenarios, misuse cases, templates, forms)	Work session	Requirements engineer	Needed artifacts: scenarios, misuse cases, models, templates, forms
5	Elicit safety and security requirements	Artifacts, selected techniques	Joint Application Design (JAD), interviews, surveys, model-based analysis, safety analysis, checklists, lists of reusable requirements types, document reviews	Stakeholders facilitated by requirements engineer	Initial cut at safety and security requirements
6	Categorize requirements as to level (system, software, etc.) and whether they are requirements or other kinds of constraints	Initial requirements, architecture	Work session using a standard set of categories	Requirements engineer, other specialists as needed	Categorized requirements
7	Perform risk assessment	Categorized requirements, target operational environment	Risk assessment method, analysis of anticipated risk against organizational risk tolerance, including hazard/threat analysis (OCTAVE, Shawn Butler, Martin Feather)	Requirements engineer, risk expert, stakeholders	Risk assessment results, added mitigation requirements to bring exposure into acceptable level
8	Prioritize requirements	Categorized requirements and risk assessment results	Prioritization methods such as Triage, Win-Win, etc.	Stakeholders facilitated by requirements engineer	Prioritized requirements
9	Requirements inspection	Prioritized requirements, candidate formal inspection technique	Inspection method such as Fagan, peer reviews, etc.	Inspection team	Initial selected requirements, documentation of decision making process and rationale