Experiences in Eliciting Security Requirements

Dr. Nancy R. Mead
CERT, Software Engineering Institute

There are many requirements elicitation methods, but we seldom see elicitation performed specifically for security requirements. One reason for this is that few elicitation methods are specifically directed at security requirements. Another factor is that organizations seldom address security requirements elicitation specifically and instead lump them in with other traditional requirements elicitation methods. This article describes an approach for doing trade-off analysis among requirements elicitation methods. Several case studies were conducted in security requirements elicitation; the detailed results of one case study and brief results of two other case studies are presented here.

A largely neglected area in requirements engineering is that of elicitation methods for security requirements. Many organizations, if they use an elicitation method at all for security requirements, use one that they have previously used for ordinary, functional (end-user) requirements. Alternatively, they may decide to use a brainstorming approach. Such methods are usually not oriented towards security requirements and do not result in a consistent and complete set of security requirements.

Carnegie Mellon University (CMU) graduate students, working with me, selected and applied several elicitation methods in a series of case studies [1]. In this article, I describe a trade-off analysis that we used to select a suitable requirements elicitation method and present results detailed from a case study of one method and a series of two other methods, used in a series of case studies. While results may vary from one organization to another, the discussion of our selection process and the example results should apply to all.

Elicitation Methods
The following is a sample of methods that could be considered for eliciting security requirements.

Misuse Cases
Misuse cases apply the concept of a negative scenario — that is, a situation that the system’s owner does not want to occur — in a use-case context. For example, business leaders, military planners, and game players are familiar with analyzing their opponents’ best moves as identifiable threats. By contrast, use cases generally describe soft problems that are more technology oriented. Examples of soft problems are how to deal with homelessness, how to manage disaster planning, and how to improve Medicare. Eventually, technology-oriented problems may emerge from these soft problems, but much more analysis is needed to get to that point.

Quality Function Deployment (QFD)
QFD is an overall concept that provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development and production [4]. The distinguishing attribute of QFD is the focus on customer needs throughout all product development activities. By using QFD, organizations can promote teamwork, prioritize action items, define clear objectives, and reduce development time.

Controlled Requirements Expression (CORE)
CORE [5, 6] is a requirements-analysis and specification method that clarifies the user’s view of the services to be supplied by the proposed system. In CORE, the requirements specification is created by the user and the developer, not one or the other. The problem to be analyzed is defined and broken down into user and developer viewpoints. Information about the combined set of viewpoints is then analyzed. The last step of CORE deals with constraints analysis such as the limitations imposed by that system’s operational environment in conjunction with some degree of performance and reliability investigation.

Issue-Based Information Systems (IBIS)
IBIS, developed by Horst Rittel, is based on the principle that the design process for complex problems, which Rittel terms wicked problems, is essentially an exchange among the stakeholders in which they bring their personal expertise and perspective to the resolution of design issues [7]. Any problem, concern, or question can be an issue and may require discussion and resolution in order for the design to proceed. The IBIS model centers on the discussion and resolution that is an integral part of the design process.

Joint Application Development (JAD)
JAD [8] is specifically designed for the development of large computer systems. The goal of JAD is to involve all stakeholders in the design phase of the product via highly structured and focused meetings. In the preliminary phases of JAD, the requirements-engineering team is tasked with fact finding and information gathering. Typically, the outputs of this phase — as applied to security requirements elicitation — are security goals and artifacts. The actual JAD session is then used to validate this information by establishing an agreed-upon set of security requirements for the product.

Feature-Oriented Domain Analysis (FODA)
FODA is a domain analysis and engineering method that focuses on developing reusable assets [9]. By examining related software systems and the underlying theory of the class of systems they represent, domain analysis can provide a generic description of the requirements of that class of systems in the form of a domain model, and a set of approaches for their implementation.

The FODA method was founded on two modeling concepts: abstraction and refinement. Abstraction is used to create domain models, as described above, from the specific applications in the domain. Specific applications in the domain are developed as refinements of the domain
**1. REPORT DATE**
DEC 2006

**2. REPORT TYPE**

**3. DATES COVERED**
00-00-2006 to 00-00-2006

**4. TITLE AND SUBTITLE**
Experiences in Eliciting Security Requirements

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**6. AUTHOR(S)**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Carnegie Mellon University, Software Engineering Institute, 4500 Fifth Ave, Pittsburgh, PA, 15213

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSOR/MONITOR’S ACRONYM(S)**

**11. SPONSOR/MONITOR’S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

**15. SUBJECT TERMS**

**16. SECURITY CLASSIFICATION OF:**

<table>
<thead>
<tr>
<th>a. REPORT</th>
<th>b. ABSTRACT</th>
<th>c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

**17. LIMITATION OF ABSTRACT**
Same as Report (SAR)

**18. NUMBER OF PAGES**
6

**19a. NAME OF RESPONSIBLE PERSON**

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
models. The example domain used in the report [9] is that of window-management systems.

**Critical Discourse Analysis (CDA)**
CDA uses sociolinguistic methods to analyze verbal and written discourse [10]. In particular, CDA can be used to analyze requirements elicitation interviews and to understand the narratives and stories that emerge during requirements elicitation interviews.

**Accelerated Requirements Method (ARM)**
ARM is a facilitated requirements elicitation and description activity process [11]. Overall, there are three phases of the process: Preparation phase, Facilitated Session phase, and Deliverable Closure phase. The ARM process is similar to JAD, but it also has certain significant differences with respect to the baseline JAD method. In the ARM process, the facilitators are content-neutral, the group dynamic techniques used are different from those used in JAD, the brainstorming techniques used are different, and the requirements are recorded and organized using different conceptual models.

**Evaluation Criteria**
The following are example evaluation criteria (project participants need to have a common understanding of what they mean in order to use them in selecting an elicitation method):

- **Adaptability.** The method can be used to generate requirements in multiple environments. For example, the elicitation method works equally well with a software package that is near completion as with a project in the planning stages.
- **Computer-aided software engineering (CASE) tool.** The method includes a CASE tool. (The Software Engineering Institute (SEI) defines a CASE tool as a computer-based product aimed at supporting one or more software engineering activities within a software development process [1].)
- **Stakeholder acceptance.** The stakeholders are likely to agree on the elicitation method in analyzing their requirements. For example, the method is not too invasive in a business environment and can be implemented in a reasonable amount of time.
- **Easy implementation.** The elicitation method is not overly complex and can be properly executed easily.
- **Graphical output.** The method produces readily understandable visual artifacts.
- **Quick implementation.** The requirements engineers and stakeholders can fully execute the elicitation method in a reasonable amount of time.
- **Shallow learning curve.** The requirements engineers and stakeholders can fully comprehend the elicitation method within a reasonable length of time.
- **High maturity.** The elicitation method has experienced considerable exposure and analysis in its vetting by the requirements engineering community.
- **Scalability.** The method can be used to elicit the requirements of projects of different sizes, from enterprise-level systems to small-scale applications.

**Ranking Against the Criteria**
The elicitation methods can be ranked against the criteria using a tabular form. In Table 1, we have filled in the values that the student team decided on for the sample methods. Each method was rated according to the desired features, and the rankings were simply added to provide a summary result. A weighted average could also have been used if some features were considered to be more important than others. For example, availability of a CASE tool might be more important than graphical output. A typical weighting scheme could consider criteria to be essential with weight 3, desirable with weight 2, and optional with weight 1. This sort of evaluation is subjective, particularly since the students worked under time constraints and did not have prior experience with this, so results may vary. Each organization or project should develop its own comparison criteria and its own ratings.

In our case studies, we decided to use JAD, ARM, and IBIS on three different projects. These three methods were subjectively ranked to be the most suitable candidates for the case studies, given our constraints. The student team’s allotted time was constrained, since this was a one-semester project. It was also the case, however, that the clients had limited time to devote to this exercise. Therefore, time constraints are mentioned several times.

It is important to note that although students did the elicitation, the projects studied were real industry and government projects, not software projects developed by students in an academic setting. It is also possible that a combination of methods may work best. This should be considered as part of the evaluation process.

**Security Requirements Elicitation Results**
In this section, we present brief results for IBIS and JAD and detailed results for ARM. Detailed results for all three methods can be found in the Requirements Engineering section of the BuildSecurityIn Web site [12] and in the case study report [1].

**Brief Results for IBIS**
The effectiveness of IBIS in eliciting security requirements depends on the quality of the interview questions. To the greatest extent possible, the scope of questions must cover the entire range of security requirements that could possibly involve the system. We found that the interviewer must be persistent in encouraging the stakeholders to explain their rationale when proposing a solution to an issue. By explaining why they have chosen such a position, the stakeholders can naturally discuss the pros and cons among themselves. In addition to proper question selection, we found that the success of IBIS is directly proportional to the variety and level of participation of stakeholders in the project. In fact, in our case study,
IBIS worked best when different stakeholders presented opposing viewpoints, which is common in large-scale projects. The Compendium software tool associated with IBIS was easy for the student team to use and effective in generating IBIS maps. To avoid displaying extremely large maps (which our stakeholders found difficult to read), we recommend exploiting the nested maps feature in Compendium. This feature enables the user to hide some of the lower-level details of the maps by nesting them inside other map elements, while maintaining the ability to drill down into the details if requested. In fact, a comment received from the stakeholders indicated that such a hierarchical map structure would have been more beneficial in handling some of the larger maps.

**Brief Results for JAD**
Due to time constraints, we did not define the work flow, data elements, screens, and reports of the project, so the JAD method turned out to be very similar to an unstructured interview process. Although unstructured interviews were used in an earlier case study, we did not attempt to do a direct comparison of the JAD results with those earlier case study results. In essence, the team just asked the stakeholders some questions about the project. Thus, the team did not use the full capability of the JAD method, which may have biased the results. The JAD session phase was designed for developing functional (end user) requirements; there was no specific way to discuss quality arrangements. Participants should read the memorandum before the Session phase to understand the content, expectation, and goals of the method. The overall goal of the memorandum is to increase the quality of the Session phase.

**Detailed Results for ARM**
Results obtained using ARM on a government project are described here. This is not a military project, but the security concerns, such as access control, are similar to the typical security concerns of military projects. ARM is designed to elicit, categorize, and prioritize security requirements. As noted earlier, the ARM method includes three phases.

**Preparation Phase**
As the name implies, this phase is used to prepare for the Session phase. There are six steps in the Preparation phase:
1. Define goals, objectives, and project success criteria (PSC) of the project.
2. Define objectives and preliminary scope of the session.
3. Establish partitions and identify participants.
4. Determine environmental and logistical aspects.
5. Establish expectations for participants.
6. Communicate with participants.

One way to obtain this information is to use a feedback form composed of questions for the stakeholders. The list of questions can be found in the case study report [1]. The stakeholders should be given a few business days to complete and return the form. In the meantime, the requirements engineering team can prepare a memorandum containing goals, objectives, PSC, preliminary scope, partition definitions, participants, and logistic arrangements. Participants should read the memorandum before the Session phase to understand the content, expectation, and goals of the method. The overall goal of the memorandum is to increase the quality of the Session phase.

**Session Phase**
The Session phase is the heart of the ARM process. It includes six steps:
1. Executive sponsor commentary.
2. Scope closure.
3. Brainstorm, organize, and name (BON).
4. Details.
5. Prioritization.
6. Participant feedback.

Before the Session phase meeting, logistical arrangements should be made to ensure that the meeting goes smoothly. The detailed list of logistical items can be found in the case study report [1].

**Executive Sponsor Commentary**
This step allows executive sponsors to provide introductory remarks to the participants regarding the planned session. Depending on the project organization, this step may or may not be necessary.

**Scope Closure**
The purpose of this step is to define what is in or out of scope. When eliciting security requirements, participants might need to familiarize themselves with security issues ahead of time to make this determination.

**BON**
The BON step provides an efficient way to elicit the candidate requirements from participants. The requirements engineering team can start by asking the participants the focus question, which should be crafted to tie to the goals, objectives, and scope of the project together. For example: *An important security requirement of the beta application is...* Based on their professional experience and security knowledge, the participants are then asked to write down seven important security requirements within seven minutes.

Next, the participants are asked to write their top three or four security requirements on cards within three minutes. The requirements engineering team then collects the cards and displays the candidate security requirements. The candidate security requirements produced in this example are listed in the following 24 initial requirements produced in ARM:
1. The ability to securely transmit data to remote sources.
2. The preservation of data integrity.
3. The enforcement and usability of an access control system.
4. Manageable security (and not hinder business where possible).
5. A strong, reliable authentication process.
6. Private information (from the outside world).
7. Consistent application program interfaces (APIs).
8. Data integrity.
9. Authentication and access control.
10. Strong authentication.
11. Risk reduction or elimination of risk of inappropriate behavior.
12. Granular access to data for users (operators) and customers.
13. Accountability (who did what, when, how...).
15. Indelibility (deletions and retractions are noted/logged).
16. Integrity.
17. Access control.
Experiences in Eliciting Security Requirements

Get Your Free Subscription

Fill out and send us this form.

517 SMXS/MXDEA
6022 Fir Ave
BLDG 1238
HILL AFB, UT 84056-5820
FAX: (801) 777-8069 DSN: 777-8069
PHONE: (801) 775-5555 DSN: 775-5555
Or request online at www.stsc.hill.af.mil

NAME: ___________________________

RANK/GRADE: __________________

POSITION/TITLE: ___________________

ORGANIZATION: _____________________________

ADDRESS: _____________________________

STATE: ___________________ ZIP: ___________

PHONE: (______)_________________________

FAX: (______)_________________________

E-MAIL: _______________________________

Check Box(es) To Request Back Issues:
Aug2005 [ ] SYS: Field Capabilities
Sept2005 [ ] Top 5 Projects
Oct2005 [ ] Software Security
Nov2005 [ ] Design
Dec2005 [ ] Total Creation of SW
Jan2006 [ ] Communication
Feb2006 [ ] New Twist on Technology
Mar2006 [ ] PSP/TSP
Apr2006 [ ] CMII
May2006 [ ] Transforming
June2006 [ ] Why Projects Fail
July2006 [ ] Net-Centricity
Aug2006 [ ] Ada 2005
Sept2006 [ ] Software Assurance
Oct2006 [ ] Star Wars to Star Trek
Nov2006 [ ] Management Basics

To Request Back Issues on Topics Not Listed Above, Please Contact GTSC Customer Service at customerservice@hill.af.mil.

December 2006

www.stsc.hill.af.mil
In the BON step of ARM, the information in the system must be indelible. The system shall implement indelibility.

In the final ...

Figure 1: Ranked Score of the Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>A1</th>
<th>A2</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>C1</th>
<th>C2</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
<th>E4</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

1. The system shall utilize cryptographically strong authentication.
2. The information in the system must be kept private from unauthorized users.
3. The system shall implement selectively secure communication with outside entities.
4. The system shall utilize and enforce an access control system.
5. The system will attempt to reduce or eliminate risks of inadvertent behavior.
6. The system shall provide granular access to data for users (operators) and customers.
7. The system shall provide role-based, restricted view, edit, and action access (e.g., summary report information, public for particular people). (tied with)
8. The system shall represent and support segmented disclosure.
9. The system shall implement auditing capabilities.
10. The system shall provide accountability of users’ actions. (tied with)
11. The system will be available 24/7 via remote authenticated access.
12. The system shall maintain a partitioned data store that is public read only and private read/write. (tied with)
13. The system shall implement a key action audit (e.g., attribution of who pressed the publish button and from where, and what changes were made).
14. The system shall implement indelibility. (tied with)

Where possible, the system’s security features must be manageable and not hinder business.

Future Plans

These case studies are part of the securi-
ty quality requirements engineering (SQUARE) project [13]. Current plans call for a comparison of SQUARE with other security requirements engineering methods, experimental combination of elicitation and prioritization methods (for example, combining ARM for elicitation with AHP for prioritization), development of supporting tools, and development of tutorial materials.

Acknowledgement
This work was part of a more extensive case study project by CMU graduate students Lydia Chung, Frank Hung, Eric Hough, and Don Ojoko-Adams [1]. This work is supported by the Army Research Office through grant number DAAD19-02-1-0389 (“Perpetually Available and Secure Information Systems”) to CMU’s CyLab.

References

About the Author
Nancy R. Mead, Ph.D., is a senior member of the technical staff in the Networked Systems Survivability Program at the SEI and is also a faculty member in the Master of Software Engineering and Master of Information Systems Management programs at CMU. Her research interests are in the areas of information security, software requirements engineering, and software architectures. Mead has more than 100 publications and invited presentations. She has a doctorate in mathematics from the Polytechnic Institute of New York and bachelor and master degrees in mathematics from New York University.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213
E-mail: nrm@sei.cmu.edu

SSTC is the premier forum in the Department of Defense (DoD) to enhance attendee’s professional skills and knowledge of systems and software technologies and policies, enabling them to improve the capabilities they provide to the warfighter.

18-21 June 2007 • TAMPA BAY

WHO should attend –
Acquisition Professionals, Program/Project Managers, Programmers, System Developers, Systems Engineers, Process Engineers, Quality and Test Engineers

WWW.SSTC-ONLINE.ORG