Security Quality Requirements Engineering (SQUARE) Methodology

Nancy R. Mead
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213
1-412-268-5756
nrm@sei.cmu.edu

Ted Stehney
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
tstehney@andrew.cmu.edu

ABSTRACT

Requirements engineering, a vital component in successful project development, often neglects sufficient attention to security concerns. Further, industry lacks a useful model for incorporating security requirements into project development. Studies show that upfront attention to security saves the economy billions of dollars. Industry is thus in need of a model to examine security and quality requirements in the development stages of the production lifecycle.

In this paper, we examine a methodology for both eliciting and prioritizing security requirements on a development project within an organization. We present a model developed by the Software Engineering Institute's Networked Systems Survivability (NSS) Program, and then examine two case studies where the model was applied to a client system. The NSS Program continues to develop this useful model, which has proven effective in helping an organization understand its security posture.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications– e*licitation methods, methodologies.*

General Terms

Management, Measurement, Documentation, Design, Security.

Keywords

Requirements Engineering, Software Engineering, Requirements Elicitation, Process.

1. INTRODUCTION

It is well recognized in industry that requirements engineering is critical to the success of any major development project. Several authoritative studies have shown that requirements engineering defects cost 10 to 200 times as much to correct once fielded than if they were detected during requirements development. Other studies have shown that reworking requirements defects on most software development projects costs 40 to 50 percent of total project effort, and the percentage of defects originating during requirements engineering is estimated at more than 50 percent. The total percentage of project budget due to requirements defects is 25 to 40.

A recent study found that the return on investment when security analysis and secure engineering practices are introduced early in the development cycle, ranges from 12 to 21 percent, with the highest rate of return occurring when the analysis is performed during the application and design [1]. NIST reports that software that is faulty in security and reliability costs the economy $59.5 billion annually in breakdowns and repairs [2]. The costs of poor security requirements show that there would be a high value to even a small improvement in this area. By the time that an application is fielded and in its operational environment, it is very difficult and expensive to significantly improve its security.

Requirements problems are the single number one cause of why projects

- are significantly over budget,
- are significantly past schedule,
- have significantly reduced scope,
- deliver poor-quality applications,
- are not significantly used once delivered,
- are cancelled.

Requirements engineering typically suffers from the following major problems:

- Requirements identification typically does not include all relevant stakeholders and does not use the most modern or efficient techniques.
- Requirements analysis typically is either not performed at all (identified requirements are directly specified without any analysis or modeling) or analysis is restricted to functional requirements, ignoring quality requirements, other non-functional requirements, and architecture, design, implementation, and testing constraints.
- Requirements specification is typically haphazard, with specified requirements being ambiguous, incomplete (e.g., non-functional requirements are often missing), inconsistent, not cohesive, infeasible, obsolete, neither testable nor validatable, and not usable by all of their intended audiences.

Software Engineering for Secure Systems -- Building Trustworthy Applications (SESS'05), May 15-16, 2005, St Louis, MO,USA. 1-59593-114-7/05/05

- Requirements management is typically weak with poor storage (e.g., in one or more documents rather than in a database or tool), missing attributes, and limited to tracing, scheduling, and prioritization.

A model is thus needed (and has been developed) to help address these current problems facing industry.

2. INITIAL SQUARE MODEL

System Quality Requirements Engineering (SQUARE) is a model developed at Carnegie Mellon by Nancy Mead as part of a research project with Donald Firesmith, and Carol Woody of the Software Engineering Institute. This process provides a means for eliciting, categorizing, and prioritizing security requirements for information technology systems and applications. The focus of this methodology seeks to build security concepts into the early stages of the development lifecycle. The model may also be useful for documenting and analyzing the security aspects of fielded systems, and could be used to steer future improvements and modifications to these systems.

SQUARE, as initially conceived, comprised the nine steps shown in Table 1.

Table 1: Initial Security Requirements Elicitation and Analysis Process [3]

Step Number	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

Step Number	Step	Input	Techniques	Participants	Output
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

The original SQUARE model outlines each step in terms of the necessary inputs, the recommended participants and methods to be followed, and the step's final output. The output from each step then flows to the sequential steps that follow. The participants for each step vary depending on the organization under study. Generally, a requirements engineer is tasked with each step, and should consider the input of all relevant stakeholders with respect to the environment of the organization and the study.

SQUARE begins when an organization agrees upon a common base that will serve the methodology to follow. The first task for the organization is to agree upon a common set of security definitions, followed by the definition of organizational security goals. Once the organization has defined a common ground, it can begin to transform its ideas about security goals into an actionable security requirements deliverable. Next, the organization chooses from various elicitation techniques, and then can begin documenting important functional information in order to develop artifacts (such as network maps and diagrams, attack tree diagrams, and use and misuse cases). These artifacts are then used to develop initial requirements, which are subsequently categorized to meet the needs of the organization's business goals. Risk assessment allows for the organization to discover how the combination of impact and likelihood of various threats affect the organization's risk tolerance with regard to each categorized requirement. Following this prioritization, a final list of requirements is produced and is inspected by all relevant stakeholders. The final output of SQUARE is a security requirements document that is designed to satisfy the security goals of the organization.

Once SQUARE was developed, it became important to discover its usefulness to industry. The next logical step was to analyze the methodology in the form of case studies in an educational and business environment.

3. CASE STUDIES

The initial SQUARE model was tested by graduate students at Carnegie Mellon University in 2004 in two consecutive case studies. Carnegie Mellon students, under the mentorship of Nancy Mead, partnered with an IT firm, Acme Corporation, in order to apply the model to one of the firm's fielded systems.

3.1 Acme Corporation

Acme Corporation (Acme)[1] is a private company headquartered in Pittsburgh. It provides technical and management services to various public sectors and a number of diversified private sectors. Its product under study, the Asset Management System (AMS)[2], provides a tool for companies to make strategic allocations and planning of their critical IT assets. It provides specialized decision

[1] Acme Corporation (Acme) is an alias used to protect the identity of the client under study.

[2] Asset Management System (AMS) is an alias used to protect the identity of the client under study.

support capabilities via customized views. AMS provides a graphical interface to track and analyze the state of important assets. The security requirements surrounding the AMS are the subject of these graduate case studies.

It is important to note here that the AMS is a fielded system, undergoing major upgrades, so the results from these case studies may not be a perfect fit for determining SQUARE's usefulness in a pre-production environment. However, the willingness of the client to participate was an important factor in its selection. Further, the results of these case studies are important in beginning to understand the effectiveness of the initial nine steps of the SQUARE process.

3.2 Case Study 1

The first case study was conducted by seven graduate students at Carnegie Mellon University, under the mentorship of Nancy Mead, during the summer of 2004. The team followed the SQUARE model with two goals in mind:

1. Complete a security requirements deliverable for Acme, and
2. Provide feedback to the NSS Program regarding both difficulties encountered and recommendations for incorporation into the model.

The work from this case study produced [4,5].

The team attempted to address each of the nine steps in the model. Within the time allotted, the team was unable to give all nine steps the full attention needed to provide complete results to NSS. For this reason, a second iteration was eventually needed. However, this team completed a great deal of important work that was critical to the success of the second iteration. In short, this case study served as the information-gathering workhorse – its results were analyzed in the second case study to provide the final outputs from SQUARE.

Most of the meaningful work produced from this iteration came in the form of artifacts developed as well as the delivery of other meaningful documentation. The team laid the groundwork in defining business and security goals and had Acme agree to a list of security definitions. Then, it completed use case and misuse case work, a preliminary attack tree analysis, and completed a final deliverable to the client consisting of architectural and policy recommendations. The team did not have enough time to complete a true security requirements deliverable, but instead provided architectural and policy recommendations, along with cost data, as a meaningful product for Acme. What was lacking was a more succinct document that focused more on requirements and less on recommendations. Further, the final deliverable did not easily map back to Acme's business or security goals. A second case study would be conducted to refine the output from this case study.

3.3 Case Study 2

During the fall of 2004, a second team, comprised of four Carnegie Mellon graduate students, began a new iteration of the case study that built upon the deliverables from the first iteration. Two main goals were identified:

1. Provide a deeper examination of certain aspects of the 9-step model, and
2. Provide a more focused security requirements deliverable for the client.

Work from this case study produced [6,7].

The initial work for this case study began with a more in-depth analysis of artifact generation. Here, the team worked to fill in the gaps in the documentation provided from the previous iteration and to forge new ground in untested artifact generation.

A more comprehensive set of use cases (and corresponding diagrams) were generated. Not only did this work help to more fully characterize the system, but also allowed the group to become familiar with AMS.

Attack trees were reexamined and a more robust set of attack trees was created. These attack trees were then compared to the misuse cases provided by the previous team. This comparison served as a sanity check – all misuse cases and attack trees resolved, and the team moved forward confident that a reasonable set of possible attacks had been considered and documented.

Upon recommendations from the first case study, this team borrowed from the Survivable Systems Analysis model in characterizing essential services and assets. This was completed as an additional sanity check to be utilized in the prioritization stage to follow.

Alongside this work, the team worked with Acme to determine a refined set of security goals that could be represented in a hierarchy. The team first outlined Acme's business goal, and then determined three high-level security goals. From here, nine lower level security requirements were drawn from the various architectural and policy recommendations provided as input from the first team's work. The hierarchy allows for varying levels of abstraction, and provides a means for mapping a low-level recommendation to Acme's security goals and ultimately its business goals (see Figure 1).

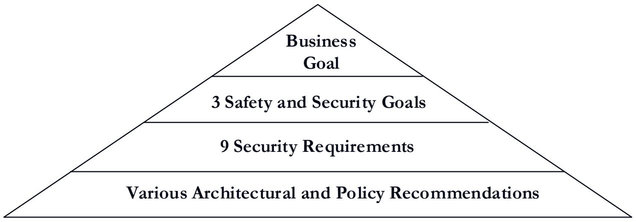
Business Goal
3 Safety and Security Goals
9 Security Requirements
Various Architectural and Policy Recommendations

Figure 1.

To examine risk assessment, the team first completed a literature review of eight industry models aimed at analyzing risk. Based on various suitability and feasibility criterion, the team selected two models to field test within the boundaries of the case study. More specifically, the team tested the Risk Filtering and Ranking Methodology created by Yacov Haimes, as well as the NIST's risk assessment technique outlined in its Special Publication 800-30. Risk assessment results from these two field tests were then combined and used as an input into requirements prioritization.

The results of risk assessment were then used to prioritize the categorized requirements provided from the output of step 6. Each of the nine security requirements were labeled essential, conditional, or optional based on how well they protected against likely and important threats. Essential asset and service identification served as an added sanity check to ensure that the requirements truly fulfilled all security goals.

For the requirements inspection portion of the research, the team kept a peer review log to keep formal documentation of bugs and defects. This tool provided a useful way for the team to communicate and manage a wide range of documents.

4. OUTPUT FROM SQUARE STEPS

In each case study, the student teams focused part of their efforts in researching various methods to conduct each step. In some cases, redundant work was completed to determine which methods might lend themselves better to SQUARE. In order to provide concrete examples of the 9 SQUARE steps, we present here a sample of the output from each individual step (all taken from the two case studies) to demonstrate how SQUARE looks in action.

Step 1: Agree on Definitions

The student teams worked with the client to agree on a common set of security definitions in order to create a common base of understanding. The following is a minute subset of the definitions that were agreed upon:

- *Access Control*: Access Control ensures that resources are only granted to those users who are entitled to them.
- *Access Control List:* A table that tells a computer operating system which access rights or explicit denials each user has to a particular system object such as a file directory or individual file.
- *Anti-virus software:* A class of program that searches hard drive and floppy disk for any known or potential viruses.

The full set of definitions was drawn from resources such as Carnegie Mellon University, industry, and dictionaries.

Step 2: Identify Safety and Security Goals

Here, the project team worked with the client to flesh out safety and security goals that mapped to the company's overall business goal. The business and security goals were defined as follows:

- **Business Goal of AMS:** To provide an application that supports asset management and planning.
- **Safety and Security Goals**: Three high-level safety and security goals were derived for the system:

1. Management shall exercise effective control over the system's configuration and usage,
2. The confidentiality, accuracy, and integrity of the AMS shall be maintained, and
3. The AMS shall be available for use when needed.

Step 3: Select Elicitation Techniques

For this step, student teams were tasked with testing various elicitation techniques and models for the overall benefit of SQUARE. Since there were only 3 stakeholders, and all were members of Acme's development team, structured interviews were the primary elicitation technique. In the future, it would be desirable to have a client with a broader variety of stakeholders.

Step 4: Developing Artifacts

Architectural diagrams, use cases, misuse cases, attack trees, and essential assets and services were documented in this portion of SQUARE. For instance, an attack scenario was documented in the following way:

System Administrator accesses confidential information

1. By being recruited OR
    a. By being bribed OR
    b. By being threatened OR
    c. Through social engineering OR
2. By purposefully abusing rights.

This step creates a volume of important documentation that serves as a vital input into following steps.

Steps 5 and 6: Elicit and Categorize Safety and Security Requirements

Nine security requirements were derived, and then organized to map to the 3 higher level security goals. Two of the 9 requirements are depicted here:

- Req 1: The system is required to have strong authentication measures in place at all system gateways/entrance points (maps to goals 1 and 2).
- Req 3: It is required that a continuity of operations plan (COOP) be in place to assure system availability (maps to goal 3).

The nine security requirements drove made up the heart of the security requirements document that was ultimately delivered to the client.

Step 7: Perform Risk Assessment

For example, the risk management techniques that were field tested were selected after a literature review was completed. This literature review examined the usefulness and applicability of eight risk assessment techniques:

1. General Accounting Office Model
2. National Institute of Standards Model
3. NSA's INFOSEC Assessment Methodology
4. Shawn Butler's Security Attribute Evaluation Method
5. CMU's Vendor Risk Assessment and Threat Evaluation
6. Yacov Haimes' Risk Filtering, Ranking, and Management Model
7. CMU's Survivable Systems Analysis Method
8. Martin Feather's Defect Detection and Prevention Model

Each method was ranked in four categories:

1. Suitability for small companies,
2. Feasibility of completion in the time allotted
3. Lack of dependence on historical threat data, and
4. Suitability in addressing requirements.

After averaging scores from the four categories, NIST's and Haimes' model were selected as useful techniques for risk assessment step. Many threat scenarios were brainstormed during this step – some of this input came from the attack tree and misuse case documentation provided from step 4. The two independent risk assessment analyses produced a meaningful risk profile for the company's system. The two most meaningful findings were:

1. Insider threat poses the most important risk to the AMS
2. Because of weak controls, it is easy for an insider or passerby to defeat authentication.

All findings from the risk assessment, along with the findings from the essential services and asset identification process completed in the artifact generation stage, were used to determine the priority level associated with each of the nine requirements.

Step 8: Prioritize Requirements

The nine security requirements were prioritized based on the following qualitative rankings:

- Essential: product will be unacceptable absent these requirements.
- Conditional: Requirement would enhance safety and security, but would not be unacceptable in its absence.
- Optional: Requirement may or may not be necessary.

Req 1, which dealt with authentication at borders and gateways, was deemed essential because of its importance in protecting against the authentication-related risks outlined as a major risk in the risk assessment. Req 3, dealing with continuity of operations planning, is still seen as an important element and worth considering, but was found to be an optional requirement relative to the other eight requirements. That is, though COOP plans are valuable, the risk assessment phase found that the greater threats to the system were those that dealt with unauthorized disclosure of information, not on availability attacks.

Step 9: Requirements Inspection

Each team member played a role in inspecting the quality of the team's work and deliverables. A peer review log was created to document what had been reviewed, and was used to maintain a log of all problems, defects or concerns. Each entry in the log was numbered and dated, addressing the date, origin, defect type, description, severity, owner, reviewer, and status. Each piece of documentation was assigned to an owner, who was held responsible for making sure that defects were fixed. This step was used as a sanity check to ensure that the team's work was meeting the group's goals and expectation.

Managing and Assessing SQUARE:

The final output to the client was a security requirements document that began by addressing the business goal, followed by the three security goals that supported this business goal, the nine categorized security requirements that supported the higher level security goals, and a list of application and configuration-specific recommendations to meet these security requirements. From here, a responsible firm would use this document in the early stages of the production lifecycle to make sure security requirements are built into the planning of the project. Once a system has been deployed, the firm can now look back to its requirements documentation to analyze whether or not it is meeting is requirements and is thus satisfying its security goals to protect the system's business function. As change occurs – be it a configuration concern in the system, the organization's risk profile, or overall business goal, the process can be reused to plan how the changing environment will affect the security concerns of the system. SQUARE is thus easily repeated within an organization as needed.

Because the key players involve a dedicated task force with knowledge of security who team with a group of knowledgeable client personnel, conducting a SQUARE assessment only requires that a firm have the time and human resources available to assist a group of outside analysts. Further, a firm knowledgeable in security could be in a position to conduct SQUARE analysis without outside help. The first graduate team spent a decent amount of time with the client in helping them develop documentation. Many firms may have this step complete before the SQUARE analysis begins. The second phase of the case made use of this documentation, and was able to complete its assessment with very little client/analyst interaction. The SQUARE analysis was very lightweight and unobtrusive to the client in this regard.

5. RECOMMENDED CHANGES

The two case studies produced two independent assessments of SQUARE. Some of the important findings from the first iteration were the need to define system architectures before outlining security goals, to handle use cases differently concerning fielded versus non-fielded systems, to use attack trees and misuse cases as a sanity check for one another, and to generate architectural and policy recommendations that are tied to cost data.

The second team felt that SQUARE might be better if applied to a to-be system as opposed to a fielded system. The team felt a constant pressure to generate recommendations as opposed to requirements, and commented that SQUARE's application should be limited to requirements generation. Next, because of the order in which the team proceeded through the SQUARE steps, and because of the inputs used for each step, the team found that the steps dealing with eliciting, categorizing, and prioritizing requirements could be realigned into one step. Because risk assessment output is needed to prioritize requirements, the team recommended that this step be moved prior to the newly aligned requirements generation step. Finally, the team recommended minor realignments to step names (by providing verbs for clarity, such as *Generate Artifacts* instead of merely *Artifacts*), small adjustments to inputs and outputs of the various steps, and made recommendations on methods that worked or have potential (as identified from the literature review and the field tests) to be realized in subsequent case studies.

6. CURRENT SQUARE MODEL

Upon receiving feedback from the two case studies, NSS has modified the SQUARE model from its initial inception described in Section 2. Here, definitions and security goals continue to be handled up front, as they continue to be the basis for the work to follow. However, artifact development is moved to an earlier point in the process, as its outputs are used throughout subsequent processes. After artifacts are generated, risk assessment is performed. This is followed by the selection of elicitation techniques and the initial elicitation of requirements, which are then categorized and prioritized based upon security goals and risk assessment results. The tasks of the final step remain the same: inspect the work completed throughout the previous eight steps. The current working model will be used in an upcoming case study. The new 9-step process consists of:

Step 1: Agree on definitions

Step 2: Identify security goals

Step 3: Develop supporting artifacts

Step 4: Perform risk assessment

Step 5: Select elicitation techniques

Step 6: Elicit security requirements

Step 7: Categorize requirements

Step 8: Prioritize requirements

Step 9: Requirements inspections

7. FUTURE WORK

NSS continues to research SQUARE and its effectiveness in the security requirements engineering arena. CMU's CyLab has recently awarded the project a seed funding grant to support a graduate student researcher. In addition, a third case study is planned for the summer of 2005. Here, a new group of graduate students at Carnegie Mellon are expected to begin and complete a full application of the newly updated SQUARE model. The client and the target system have not yet been selected. It is unclear if Acme Corporation will again act as a participant in the case study, but its continued participation will provide both important benefits and some limiting externalities.

Should it remain a partner in this research project, the third iteration will begin farther along in the learning curve, drawing from the firm's heightened understanding of its security goals, from its artifacts, and from the availability of other relevant documentation (for instance, its cost data). The students in this third case study would not need to reinvent the wheel concerning the preliminary SQUARE steps 1-4. The team could instead focus its work on the heart of the requirements engineering steps of SQUARE. Also, the potential for time savings provided from the firm's learning curve could allow the team to complete a full application of steps 5-9 in the team's short three-month window. This could prove a huge benefit, as the results would demonstrate the first cradle-to-grave application of SQUARE (as it would not extend over multiple case studies).

However, a third study with Acme would be subject to certain biases that could limit the effectiveness of its results. For instance, this study would still not provide any evidence of SQUARE's usefulness in a pre-production environment if the same target system were analyzed. Further, this study could fall victim toward biases that surround this client. Perhaps certain variables surrounding the firm or its environment (including firm size, particular industry, or threat actors to which the firm is exposed) have pushed the results of the first two studies in a misleading direction. Any new insights that are affected by these hidden particularities would not come to fruition. Perhaps a fresh start with a new client/target system will provide new insights not yet recognized. In the coming months, NSS will decide on the details surrounding the third case study. Its decisions will in part be influenced by the availability of willing and accessible clients.

The model created by NSS, as applied in the two case studies described in this paper, has proven useful to Acme's understanding of its security posture. As the model continues to be refined, it shows promise for helping solve industry's inability to adequately address security in the production lifecycle. The SQUARE model has been refined following the results of the first two case studies. A third case study should continue to provide progress in generating a model that will benefit industry.

8. ACKNOWLEDGMENTS

Acknowledgements are made to the members of both case study teams and student interns for their contributions to the development of the SQUARE methodology and associated tools: Peter Chen, Marjon Dean, Dan Gordon, Lilian Lopez, Don Ojoko-Adams, Hassan Osman, Will Salamon, Neha Wattas, Nick Xie, and Eugene Yu.

9. REFERENCES

[1] Soo Hoo, K, Sudbury, J.W., Jaquith, J.R. "Tangible ROI Through Secure Software Engineering", Secure Business Quarterly, Volume 1, Number 2, @stake, 2001.

[2] National Institute of Standards and Technology, "Software Errors Cost U.S. Economy $59.5 Billion Annually" (NIST 2002-10). http://www.nist.gov/public_affairs/releases/n02-10.htm (2002).

[3] Chen, P., Mead, N. R., Dean, M., Lopez, L., Ojoko-Adams, D., Osman, H. Xie, N. *SQUARE Methodology: Case Study on Asset Management System* (CMU/SEI-2004-SR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

[4] Chen, P., Mead, N. R., Dean, M., Lopez, L., Ojoko-Adams, D., Osman, H. Xie, N. *SQUARE Methodology: Case Study on Asset Management System* (CMU/SEI-2004-SR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

[5] Mead, N. "Requirements Elicitation and Analysis Processes for Safety & Security Requirements", 4th International Workshop on Requirements for High Assurance Systems, September 6, 2004, Kyoto, Japan, proceedings published by SEI: http://www.sei.cmu.edu/community/rhas-workshop/#papers.

[6] Student report to be sanitized and published by the SEI.

[7] Student report to be sanitized and published by the SEI.