Software Engineering Institute

An Evaluation of A-SQUARE for COTS Acquisition

Sidhartha Mani
Nancy Mead

May 2014

TECHNICAL NOTE
CMU/SEI-2014-TN-003

CERT Division

http://www.sei.cmu.edu

Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZM
20 Schilling Circle, Bldg 1305, 3rd floor
Hanscom AFB, MA 01731-2125

DM-0001061

Table of Contents

List of Figures

List of Tables

Acknowledgments

Sidhartha Mani acknowledgments: I sincerely thank Dr. Nancy Mead for her continuous guidance throughout the project, Husam Suleiman for providing me with the artifacts from his case study on SQUARE, and the Carnegie Mellon University students who helped me conduct the Architecture Tradeoff Analysis Methodology.

Abstract

Developed by the Software Engineering Institute (SEI) at Carnegie Mellon University, Software Quality Requirements Engineering for Acquisition (A-SQUARE) is a methodology used for eliciting and prioritizing security requirements as part of the acquisition process. In the project described in this paper, we evaluated the effectiveness of the A-SQUARE method by applying it to a COTS product for the advanced metering infrastructure of a smart grid. We evaluated the ability of the A-SQUARE method to identify security requirements for the COTS product; identify candidate COTS products; elicit, categorize, and prioritize security requirements; prioritize COTS products; and select a COTS product. We also evaluated the usability of the A-SQUARE tool using qualitative evaluation criteria.

1 Background

Requirements engineering is the foundation of software engineering. Yet security often is not considered during requirements engineering, even though studies have shown that such consideration would benefit software engineering projects. When organizations acquire commercial off-the-shelf (COTS) software, security is almost never addressed. In this report, we consider the security requirements of COTS components.

1.1 Industry Problem

Industry organizations commonly use COTS products to speed up development of software. COTS product selection is generally based only on features and cost, though scalability and performance are sometimes considered. Security is considered only as an afterthought to functional requirements (features). As a result, companies often end up building vulnerable systems. Sixty-three percent of data breaches are due to bad outsourcing decisions; that is, COTS products are acquired without considering the right security quality attributes [Ashford 2013].

These bad outsourcing decisions arise from the following problems:

- failure to identify security requirements for COTS products
- lack of a well-defined and well-established process for incorporating security requirements engineering into acquiring COTS products
- unprioritized security requirements (Analysis of requirements to understand the relative importance of requirements for the COTS product is not performed or not understood.)
- no established set of principles for specifying requirements for COTS products

1.2 Current Research

There is little current research in the area of incorporating security considerations into requirements engineering for the acquisition of COTS software.

Arlene Minkiewicz describes three measures that can be taken to incorporate security into COTS [Minkiewicz 2005]:

1. wrapping non-secure COTS products in a secure layer
2. acquiring only pre-certified COTS products
3. certifying COTS products internally

Nancy Mead proposes another measure: the Security Quality Requirements Engineering methodology for Acquisition (A-SQUARE) [Mead 2010]. The Security Quality Requirements Engineering (SQUARE) methodology [Mead 2005] is a nine-step method used for eliciting and prioritizing security requirements. A-SQUARE, an adaptation of the SQUARE method to software acquisition needs, guides organizations in eliciting security requirements for commercial off-the-shelf (COTS) software.

In the following sections, we report on the A-SQUARE Case Study project in which we apply the A-SQUARE method to a case study for acquiring a COTS product for the advanced metering infrastructure (AMI) of a smart grid.

2 A-SQUARE Method

A-SQUARE can be applied to three types of acquisition scenarios:

1. The acquisition organization holds a typical client role for newly developed software.
2. The acquisition organization specifies the requirements as part of a request for proposal (RFP) for newly developed software.
3. An organization acquires COTS software.

In the A-SQUARE Case Study project, we focused on the third scenario, the acquisition of COTS software. In this scenario, the organization develops a list of software requirements and compares them to the software packages under consideration.

The steps in the A-SQUARE method for acquiring COTS software are shown in Table 1.

Table 1: Process for Acquiring COTS Software

	Step	Input	Technique	Participants	Output
1	**Agree on definitions**	Candidate definitions from IEEE and other standards	Structured interviews and focus groups	Acquisition organization stakeholders and security specialists	Agreed-to definitions
2	**Identify assets and security goals**	Definitions, candidate goals, business drivers, policies and procedures, and examples	Facilitated work sessions, surveys, and interviews	Acquisition organization stakeholders and security specialists	Assets and Goals
3	**Identify preliminary security requirements**	Assets and goals	Work sessions	Acquisition organization stakeholders and security specialists	Preliminary security requirements
4	**Review COTS software package information and specifications**	Assets, goals, and preliminary security requirements	Studying security features of various packages and documenting them	Acquisition organization stakeholders, security specialists, and COTS vendor	Security features of various packages
5	**Review and finalize security requirements**	Preliminary security requirements and features of various packages	Conducting work sessions where participants use features to refine, review, and modify preliminary security requirements	Acquisitions organization stakeholders and security specialists	Final security requirements
6	**Perform tradeoff analysis**	Final security requirements, a spreadsheet of security features	Conducting a tradeoff analysis of COTS products relative to final security requirements	Acquisitions organization stakeholders and security specialists	Prioritized list of COTS products relative to security requirements
7	**Make the final product selection**	A prioritized list of COTS products relative to security requirements and other COTS product features	Conducting a tradeoff analysis	Acquisitions organization stakeholders	Final product selection

3 System Architecture and Functional Requirements

Before applying the A-SQUARE method to our selection of a COTS component—a database component—for the AMI of a smart grid [Suleiman 2013], we needed to understand the system architecture and functional requirements. In this section, we present the system architecture and a description of the functional requirements of the smart grid and its AMI.

A *smart grid* is an infrastructure of tools, technologies, and topologies that accommodate centralized, distributed, and mobile power generation. A smart grid allows stakeholders to optimally deal with different power generation and distribution needs. The infrastructure consists of various components, domains, sub-domains, and consumers interacting with one another. Communications occurs through two kinds of interfaces—electrical (physical) and communications (logical) interfaces. Each interface maintains privacy while interacting with other interfaces. Figure 1 summarizes the smart grid's communications interfaces and includes the database that the smart grid requires to function.

The database is responsible for maintaining up-to-date and consistent information about access control, electricity demand, electricity consumption, electricity generation, transmission, vendors, consumers, the workforce, and many other kinds of information.

A smart grid is a self-healing system that automatically takes corrective measures in case of a failure, without human intervention. The databases that a smart grid uses also must satisfy this property to work effectively with the smart grid.

An AMI is a measuring system that can read, store, and transmit the usage, demand, and usage trends of electricity for many users. This system is used for load and power control, as well as monitoring. The database of an AMI stores information about electricity usage and demand for every customer as well as other personal information about consumers. If the security of this data is not handled properly, confidential information could be lost or a loss of power (due to modifications of the readings in the data store) could occur and lead to power failure, surge, or associated safety issues.

The following security requirements, specifically for the AMI database, are in addition to the security requirements of the overall smart grid:

Trust - Data is accessed and used only by appropriate users and devices. It is communicated only through appropriate protocols.

Privacy - Access to energy or security-related information by unauthorized sources is prevented.

Communications and Device Security - All devices and communications channels are secure.

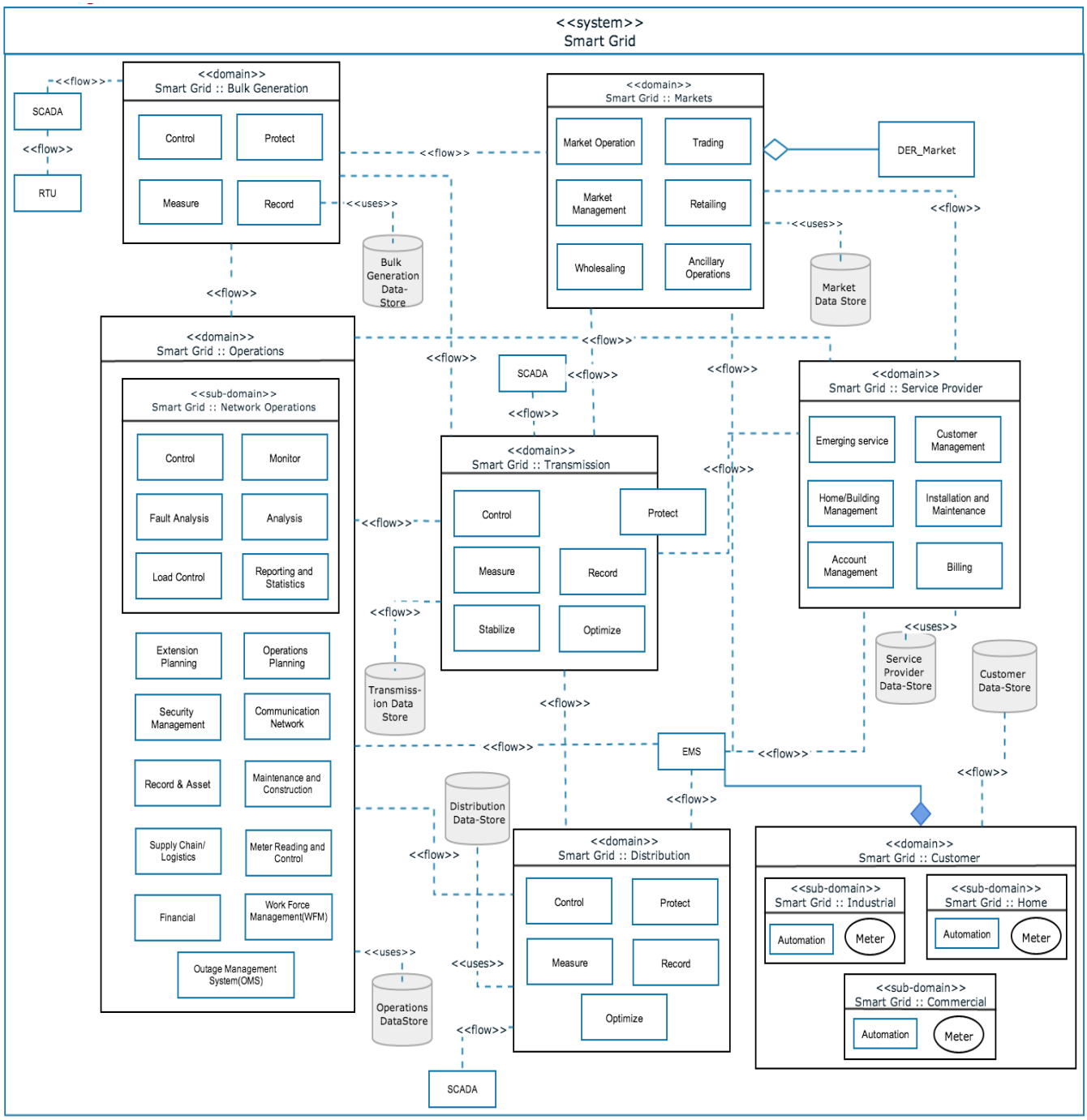

Figure 1: Smart Grid Communications Interfaces[1]

[1] Adapted from the smart grid shown in the September 2013 article, *Evaluating the Effectiveness of Security Quality Requirements Engineering (SQUARE) Method: A Case Study Using Smart Grid Advanced Metering Infrastructure* [Suleiman 2013].

4 Research Method

For this project, we used a case study to perform a holistic investigation. Using the research design proposed by Suleiman and Sventinovic [Suleiman 2013], we investigated *how* and *why* an attacker would target an AMI database. This analysis forms the most important work in this case study. (The data sources for this analysis are cited from academic and industrial publications.)

The research design of this case study is illustrated in Figure 2. We began the design by identifying our *research objectives*. We then identified our *research questions* and *units of analysis*. Next, we prepared, sorted, and organized data during the *data collection* step. This evaluation is performed to determine the effectiveness of the method.

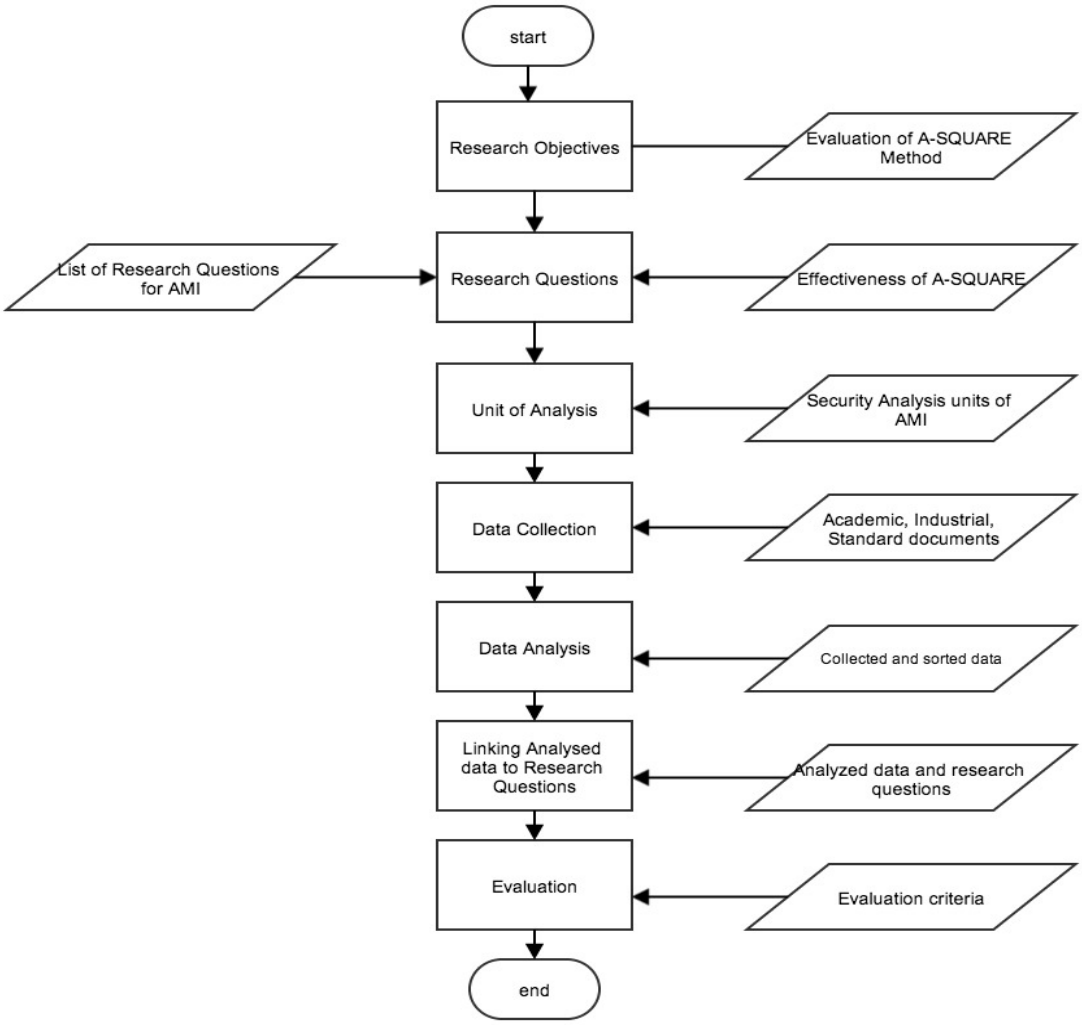

Figure 2: Case Study Design Plan Flowchart

The *research objective* of this project is to evaluate the effectiveness of using the A-SQUARE method to aid selection of a database component for an AMI. The effectiveness of A-SQUARE is evaluated by analyzing the documents produced, which contain threats, vulnerabilities, and risks, and their categorization, requirements, and prioritization. Through these artifacts, the evaluation is done by using the following criteria for evaluating the A-SQUARE method:

Involvement of stakeholders and vendors – the provision that allows all stakeholders to communicate their security concerns about the COTS component

Ease of combining security requirements with other kinds of requirements - the facility provided for specifying other kinds of requirements, such as functional requirements or other quality attributes

Ability to perform tradeoffs between security and other requirements - the ability to analyze and prioritize security while keeping other requirements (functional and other quality requirements) in mind

Support for revision - the ability to add, delete, or update requirements after initially selecting them

Traceability - the clarity of the decision process and the decisions made throughout the A-SQUARE process, which is required to justify the rationale behind a decision

Consistency - the ability of the method to identify that two security requirements are contradictory

We asked the following questions to evaluate the effectiveness of the A-SQUARE method for selecting a database for the AMI. Our questions build on those asked by Suleiman and Svetinovic [Suleiman 2013].

1. What are the main security scenarios and requirements of the database component in the AMI?
2. What are the main threats and risks to a database component in the AMI?
3. What are the requirements that a COTS database product should satisfy to prevent this?
4. How can we categorize and prioritize these requirements?

We also defined the following criteria for evaluating the A-SQUARE tool:

Usability - the ease of setup and use of the A-SQUARE tool and the amount of time a new user would require to understand and to effectively use the tool

Robustness - the reliability of the tool and the results provided by it

Conformance to A-SQUARE - the tool's ability to support different implementations of the A-SQUARE method through required features

Traceability - the support provided by the tool to provide a clear understanding of the decision process and the decisions made throughout the steps of the A-SQUARE method, which is required to justify the rationale behind a decision

Support for various methods - the ability of the tool to support different techniques applied in each of the steps of the A-SQUARE method (e.g., the ability to support various techniques to perform tradeoff analysis)

5 Applying the A-SQUARE Method

The A-SQUARE [Mead 2013] method considers security requirements along with functional requirements and other quality attributes for COTS acquisition. This method consists of the following steps:[2]

1. Agree on definitions.
2. Identify assets and goals.
3. Identify preliminary security requirements.
4. Review COTS package information and specification.
5. Review and finalize security requirements.
6. Perform tradeoff analysis.
7. Make the final product selection.

The A-SQUARE method relies on the following roles:

The *administrator* is a role required only for the use of the A-SQUARE tool. The process itself does not require an administrator. The administrator is responsible for assigning the right access control privileges to the various users of the tool.

The *acquisition organization engineer (AOE)* is the person responsible for understanding, overseeing, and ensuring the correct conduct of the A-SQUARE process. He/she is responsible for identifying goals and sub-goals and is involved throughout the A-SQUARE process.

The *acquisition organization stakeholders* are the people who participate in the selection process and the A-SQUARE steps, but do not necessarily interact with the A-SQUARE tool.

The *security specialist* is a subject matter expert who is knowledgeable about the COTS products on the market and about security.

The *COTS vendor* is the person who provides the COTS component to the acquiring organization.

The *contractor* is the person from the acquiring organization who contracts the development of a COTS product. (For this case study, this role is not relevant.)

For the first step, *agree on definitions,* we drew from the definitions provided by the tool and added some other relevant definitions.

In the second step, *identify assets and goals,* we identified a high-level goal for the case study. We identified sub-goals that would ensure the achievement of the main goal. We also identified artifacts that would map to these sub-goals.

[2] These steps are presented in detail in Table 1.

In the third step, *identify preliminary security requirements,* we used the original requirements for the AMI to identify possible security requirements for the COTS product to be used in the AMI.

For the fourth step, *review COTS package information and specification,* we used a web search to identify candidate products. We shortlisted four COTS products based on their features.

The fifth step, *review and finalize security requirements,* allowed us to revisit the previously elicited requirements and make changes or add new requirements.

For the sixth step, *perform tradeoff analysis,* we conducted interviews and brainstorming sessions.

The *final product selection* was based on the results of the tradeoff analysis step.

6 Case Study Results

In this section, we describe the results of using the A-SQUARE method to elicit and prioritize security requirements for (and ultimately select) a COTS database for the AMI. To apply A-SQUARE successfully, it was important to fully understand the requirements for the AMI. In the following paragraphs, we describe each step and the results we observed. We conducted all of these steps using the A-SQUARE tool.

Step 1: Agree on definitions.

We identified 74 terms that needed to be defined. The tool provided a list from which we could choose definitions relevant to this project.

Step 2: Identify assets and security goals.

We started this step by identifying a single high-level goal, *Acquire a database product that satisfies the functional, quality, and security needs for the advanced metering infrastructure*. The acquisition organization's engineer is responsible for conducting this step, and the tool ensures that only the AOE can perform this step.

We then identified the sub-goals and the artifacts needed to support them. We identified 11 quality sub-goals in addition to the functional sub-goals. We identified four types of artifacts:

1. architecture diagram
2. use cases
3. misuse cases
4. the SEI security quality attribute template

The mapping of security goals to assets and the relative priorities of the sub-goals are shown in Table 2. The prioritization was done in a brainstorming session in which various scenarios were discussed for each goal to understand the importance of the goals.

Table 2: Mapping and Prioritization of Sub-Goals and Assets

Priority	Sub-Goal	Assets
1	Functional Goals	• use cases
2	Confidentiality - Data must be accessible only by authorized entities. Encryption is needed to support confidentiality, and privacy is needed to ensure that this sub-goal is met.	• architecture diagram • misuse cases • security quality attribute template from the SEI
3	Authentication - The identity of the person accessing the data and resources of the system are verified before allowing access.	• misuse cases • security quality attribute template from the SEI • use cases
4	Access Control – Access must be restricted based on criteria other than the identity of the user (e.g., role).	• misuse cases • security quality attribute template from the SEI • use cases

5	Non-Repudiation - The user should not be able to deny something he/she did, including receiving, tampering with, or changing information.	• architecture diagram • misuse cases • security quality attribute template from the SEI • use cases
6	Reliability - The smart grid should be reliable; therefore, the database should also be reliable (by replicability or some other mechanism).	• architecture diagram • security quality attribute template from the SEI
7	Fault Tolerance - The smart grid should be fault tolerant; therefore, the database also should be fault tolerant.	• architecture diagram • security quality attribute template from the SEI
8	Availability - The data should be available to authorized entities whenever needed.	• architecture diagram • security quality attribute template from the SEI
9	Scalability - The database should be able to scale independently and should be modular.	• architecture diagram • security quality attribute template from the SEI
10	Authorization – The database should be able to give a person or a system permission to perform some actions on the database.	• misuse cases • security quality attribute template from the SEI • use cases
11	Capacity - The user requests could be up to a million requests per second. The database should be able to handle this volume by using appropriate mechanisms such as caching.	• architecture diagram • security quality attribute template from the SEI
12	Integrity - The data should be complete and safe from being updated or deleted by unauthorized sources.	• architecture diagram

Step 3: Identify preliminary security requirements.

The *security specialist* is responsible for this step. Access control is ensured by the tool. The security specialist, who understands the needs of the system, identifies and describes in detail a set of requirements that a database COTS component for the AMI must satisfy.

There should be enough detail in the descriptions to understand whether a particular COTS product meets the requirement. We identified 10 security requirements and mapped them to the security sub-goals identified in step 2. The list of security requirements and their mapping to goals for the database are summarized in Table 3.

Table 3: Security Requirements and Goal Mapping

Requirement	Goal
The database product can read and transmit 512-bit key-encrypted data. This ability is designed to ensure that all man-in-the-middle attacks are very hard to carry out. Man-in-the-middle attacks are possible when a monitoring device can be set up at the infrared port of the smart meter.	• Confidentiality
The database should be updatable by a large number of users who can perform CRUD (Create, Read, Update, Delete) requests only on tables created for them or by them. The database should provide each user with his/her own username and password. This feature is used when a large number of smart meters (SMs) are installed in the grid. Every SM should be its own user so that, even if the encryption of one of SM is broken, the rest of the SMs are not affected.	• Integrity • Authorization • Access Control
The database product should support data encryption for the data it stores so that, even if the attacker obtains the data, the data cannot be read and understood. Only users with the right access for that data should be able to decrypt it.	• Access Control • Confidentiality
All operations conducted on the database should be logged. If a user makes a change, such as creating, updating, or deleting, the database should be able to remotely and securely log these actions. The log should contain information such as change made, previous revision, change made by (username or unknown user), user source (IP or MAC address), and change made on (date/time). The log should be created only by the database and no one should be able to change it. The administrator should be allowed to access it (read only).	• Non-repudiation
The database should be able to accommodate large numbers of queries. It should use mechanisms, such as queuing, caching, and distributed processing, to process a large number of queries without breaking down.	• Reliability • Performance • Fault-Tolerance
The database should support the creation of tables and records on the fly. The database should support independent scaling of information regardless of the hardware; that is, it should support a distributed file system. The database could be relational or not.	• Scalability
The database should be available over a network. It should be possible to remotely issue all commands on the database. It should not succumb to threats or attacks.	• Availability
The database should be capable of communicating using any communications protocol and should be able to handle changes in the protocol.	• Functional goal
The database should be able to run on the proprietary operating system developed for the smart grid's AMI.	• Functional goal
The database should support virtualization that enables the CPU and memory allocated to it to be easily increased or decreased. It should have the facility to run over physically distributed servers.	• Functional goal

Step 4 - Review COTS package information and specification.

In this step, we identified a set of COTS products that satisfied the functional requirements and the quality requirements for the data component of the AMI. The identification was done using only internet research of COTS vendor websites and information sites to find candidates and study their features. We conducted further research to gain consensus on the usability, support, and robustness of the product. This research involved exploring information other than that pro-

vided by vendors; we reviewed information about the candidate products on online forums such as *stackoverflow* and *quora*.

The COTS products were rated based on the research. The following table with the COTS products shown anonymously and their ratings resulted. The left column contains the goals. The numbers between 1 and 3 in the right columns signify the usability and community support for the particular goal for that particular product (1 is a poor rating, 2 is a mediocre rating and 3 is a good rating). We quantified the candidate products using these ratings.

Table 4: COTS Products and Their Ratings

	Product A	Product B	Product C	Product D
Encrypted Communications	3	3	2	2
Authorization	3	2	2	2
Access Control	3	2	2	2
Encryption	3	3	2	2
Non-Repudiation	3	2	2	2
Functional Requirements	3	2	2	1
Other Quality Attributes	3	2	2	2
Availability	3	2	2	1

Step 5 - Review and Finalize Requirements.

This step is conducted only by the *security specialist*; the tool ensures that access to this step is provided only to the *security specialist*. The *security specialist* reviews every requirement while comparing the requirements with the features provided by each of the tools. If a requirement needs to be changed, the requirement is submitted for review. If the requirement does not need to be changed, it is considered to be approved. In this case study, all of the COTS products supported all of the requirements. Therefore, all of the requirements were approved.

Step 6 - Perform Tradeoff Analysis.

This step is conducted to understand the relative priorities of the COTS products. Each COTS product was given a score based on its support of each of the goals. The COTS products were ranked based on the total count of these values. The scoring scheme is presented in the explanation for *Step 4 - Review COTS package information and specification.*

The rankings we obtained are summarized in Table 5:

Table 5: Rankings of COTS Products (1 = high)

COTS Package	Ranking
Product A	1
Product B	2
Product C	3
Product D	4

Step 7 - Make the final product selection.

In this step, a final product is selected from the candidate COTS packages. The final product selection can be based on criteria such as community support, cost, training required, or setup costs. We made the selection using community support as the criteria and found that the community support is the greatest for Product A. Therefore, our final choice was Product A.

7 Evaluation

In this section, we evaluate the application of the A-SQUARE method to selecting a COTS database product for an AMI. We specified nine use cases and developed a security quality attribute template and system architecture diagram. We identified 11 quality requirements, 8 of which were security requirements. The entire process of applying the A-SQUARE method for selecting a database product for an AMI took us 12 person-weeks, including all of the steps prior (data collection) and after (report generation).

7.1 Evaluating the A-SQUARE Method

The criteria used to evaluate the A-SQUARE method are discussed below:

Involvement of stakeholders. Three roles participate in this case study: the security specialist, acquisition organization engineer, and COTS vendor. The acquisition organization engineer performs *Step 2 - Identify assets and goals.* The assets and goals for the software system being acquired are identified in this step. This step is performed by the acquisition organization engineer. In *Step 3 - Identify preliminary security requirements,* the initial security requirements for the COTS product are identified by the security specialist. In *Step 4 - Review COTS package information and specification,* the COTS vendor describes the security features provided by the COTS component. In this way, all of the roles' concerns are taken into account.

Ease of combining security requirements with other requirements. In S*tep 2 - Identify assets and goals,* there is no emphasis on any kind of requirements. The requirements are identified through the goals; therefore, both security and non-security quality requirements are identified.

Ability to perform tradeoff between security and other kind of quality attributes. In *Step 6 - Perform tradeoff analysis,* we prioritized the various quality attributes; based on these quality attributes, we prioritized the COTS components. These results show that A-SQUARE has the ability to perform tradeoff analysis between security and other kinds of quality attributes.

Support for revision. In A-SQUARE, S*tep 5 - Review and finalize security requirements* allows requirements to be reviewed and revised.

Traceability. Each of the steps in A-SQUARE requires an exit criteria artifact. Each decision can be traced back to its rationale using the artifacts.

Consistency. In A-SQUARE, *Step 5 - Review and finalize requirements* ensures consistency. If two requirements are contradictory, the discrepancy can be rectified in this step.

7.2 Evaluating the A-SQUARE Tool

The evaluation criteria for the A-SQUARE tool are discussed below.

Usability. The tool is available as a free download from the SEI website. There are instructions provided to use the tool. The installation process involves manually starting the server and setting up the databases. The server is an Apache web server, and the database is MySQL. This tool expects the database name to have a space, whereas MySQL did not allow us to create a database

with a space in its name. Therefore, we changed the configuration file for the tool to expect a different database name. Once this was done, the tool worked as expected. The tool has several user types, and the instruction manual was helpful in understanding the access rights of the different types of users. It is intuitive and user friendly for a sophisticated user who is well versed in software engineering. This tool would not be considered easy to use by an inexperienced user.

Robustness. The tool did not lose data. After multiple restarts, the data persisted.

Conformance to A-SQUARE. We could perform all of the A-SQUARE steps using the tool. The tool even made it easy to perform these steps, laying out relevant information in an easy format throughout.

Traceability. The tool provides ways to document rationale behind important decisions such as priority assignments and final product selection.

Support for various techniques. The tool records the results of each step and the rationale behind decisions without regard to technique.

8 Conclusion

In conclusion, we found that each step of the A-SQUARE method provided the analyst with a good representation of the results. For example, the developed artifacts—use cases, the architecture diagram, and the security quality template from the SEI—helped us to understand the system well. The A-SQUARE tool helped us to neatly document and visualize the results along with the rationale. The tool provided us ample facilities to prioritize, categorize, and display security requirements.

The A-SQUARE tool also provided us with the steps for performing tradeoff analysis so that the relative priorities between the different types of requirements could be understood.

The A-SQUARE method had some disadvantages; for example, it did not provide a template to elicit, categorize, or prioritize requirements. These steps can be done in any fashion that the user decides, which can affect the accuracy of the final results. The A-SQUARE tool also lacked a function to check the completeness of the requirements elicitation process and a framework for final product selection.

There are some areas of research that the A-SQUARE method would benefit from:

- Techniques for measuring the security of COTS components
- Prioritization of COTS components for a project

These areas would greatly benefit the A-SQUARE method. The application of pairwise comparisons to prioritize and make decisions about the priorities of COTS components would fall under the second area of research.

Other prioritization techniques should be applied to the A-SQUARE method, and a comparative study of the best prioritization technique for various types of projects should be found. The support for various techniques should be provided by the A-SQUARE tool if new techniques are added to the method.

The tool serves its purpose of assisting the user of the A-SQUARE method well. No changes are recommended at this time.

References/Bibliography

URLs are valid as of the publication date of this document.

[Ashford 2013]
Ashford, Warwick. "Bad outsourcing decisions cause 63% of data breaches." *Computer Weekly*. February 15, 2013. http://www.computerweekly.com/news/2240178104/Bad-outsourcing-decisions-cause-63-of-data-breaches

[CC 2006]
Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 1 (CCMB-2006-09-001). September 2006. http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R1.pdf

[Doan 2006]
Doan, Dung. *Commercial Off the Shelf (COTS) Security Issues and Approaches* (Master's Thesis). Naval Postgraduate School, Monetery, CA, 2006. http://handle.dtic.mil/100.2/ADA456996

[Giorgini 2006]
Giorgini, P.; Mouratidis, H.; & Zannone, N. "Modelling Security and Trust with Secure Tropos." 2006. http://www.cs.toronto.edu/~zannone/publication/gior-mour-zann-06-IDEA.pdf

[Kazman 2000]
Kazman, Rick; Klein, Mark; & Clements, Paul. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004). Software Engineering Institute, Carnegie Mellon University, 2000. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5177

[Mead 2005]
Mead, Nancy; Hough, Eric; & Stehney II, Ted. *Security Quality Requirements Engineering* (CMU/SEI-2005-TR-009). Software Engineering Institute, Carnegie Mellon University, 2005. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=7657

[Mead 2010]
Mead, Nancy R. "Adapting the SQUARE Method for Security Requirements Engineering to Acquisition." Software Engineering Institute, Carnegie Mellon University. February 2010. http://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_51613.pdf

[Mellado 2006]
Mellado, D.; Fernández-Medina, E.; & Piattini, M. "Applying a Security Requirements Engineering Process," 192-206. *Proceedings of the 11th European Symposium on Research in Computer Security*. Hamburg, Germany, 2006. http://www.springerlink.com/content/91l8q364jk2p5421/

[Minkiewicz 2005]
Minkiewicz, A. "Security in COTS-Based Systems." *CrossTalk 18*, 11 (November 2005): 23-27. http://www.crosstalkonline.org/storage/issue-archives/2005/200511/200511-Minkiewicz.pdf

[OWASP 2009]
Open Web Application Security Project. *OWASP Secure Software Development Contract Annex*. 2009. https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex

[Polydys 2009]
Polydys, M. & Wisseman, S. "Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise" (Information Resources Management College Occasional Paper). National Defense University Press, Washington, DC, 2009. https://buildsecurityin.us-cert.gov/sites/default/files/publications/SwAinAcquisition%20MitigatingRisks%20to%20Enterprise.pdf

[Suleiman 2013]
Suleiman, H. & Sventinovic, D. "Evaluating the Effectiveness of Security Quality Requirements Engineering (SQUARE) Method: A Case Study Using Smart Grid Advanced Metering Infrastructure." *Requirements Engineering Journal 18*, 3 (September 2013).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	May 2014	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
An Evaluation of A-SQUARE for COTS Acquisition	FA8721-05-C-0003

6. AUTHOR(S)

Sidhartha Mani, Nancy Mead

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2014-TN-003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Developed by the Software Engineering Institute (SEI) at Carnegie Mellon University, Software Quality Requirements Engineering for Acquisition (A-SQUARE) is a methodology used for eliciting and prioritizing security requirements as part of the acquisition process. In the project de-scribed in this paper, we evaluated the effectiveness of the A-SQUARE method by applying it to a COTS product for the advanced metering infrastructure of a smart grid. We evaluated the ability of the A-SQUARE method to identify security requirements for the COTS product; identify candidate COTS products; elicit, categorize, and prioritize security requirements; prioritize COTS products; and select a COTS product. We also evaluated the usability of the A-SQUARE tool using qualitative evaluation criteria.

14. SUBJECT TERMS	15. NUMBER OF PAGES
A-SQUARE, acquisition, requirements	27

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL