Software Engineering Institute

Adapting the SQUARE Process for Privacy Requirements Engineering

Ashwini Bijwe, Carnegie Mellon University
Nancy R. Mead, Software Engineering Institute

July 2010

TECHNICAL NOTE
CMU/SEI-2010-TN-022

CERT® Program

http://www.cert.org

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about SEI publications, please visit the library on the SEI website (www.sei.cmu.edu/library).

Table of Contents

List of Figures

List of Tables

Executive Summary

Privacy is the ability of an individual to control his or her own information. As software systems become more distributed and complex, maintaining privacy of data and ensuring data integrity remain challenges for software practitioners. Developing such systems poses not only technical challenges but also demands compliance with privacy laws. Engineering precise privacy requirements is an important activity in building these software systems, and it is an activity that requires a disciplined approach. The nine-step Security Quality Requirements Engineering (SQUARE) process, which was developed for security requirements engineering, can be adapted for privacy requirements engineering in software development as shown below.

1. Agree on definitions. Stakeholders and the requirements team create a comprehensive list of terms that will foster effective communication and reduce ambiguity. Participants define the terms in the list so that all stakeholders understand their basic scope.
2. Identify assets and privacy goals. Stakeholders and requirements engineers initiate the discussion of the project's assets and overall privacy goals to achieve a common understanding. The purpose of this step is to initiate a discussion among the stakeholders of the assets and their overall privacy goals for the project.
3. Develop artifacts. Requirements engineers generate artifacts that relate to privacy, such as system architecture diagrams, use case scenarios and diagrams, and misuse case scenarios and diagrams.
4. Perform risk assessment. Requirements engineers, risk experts, and stakeholders identify risks using parameters such as the probability of risk materialization and impact of the risk. They then prioritize the risks and define mitigation strategies.
5. Select elicitation technique. Requirements engineers evaluate the various requirements elicitation techniques—such as structured or unstructured interviews and use and misuse cases—and select one.
6. Elicit security requirements. Stakeholders, facilitated by requirements engineers, elicit security requirements. Computer-aided tools such as the Privacy Requirements Elicitation Technique (PRET) can be used along with a privacy requirement questionnaire.
7. Categorize requirements. Requirements engineers, and other specialists as needed, systematically group requirements to prepare for the next step of the process, requirements prioritization. Categorization also enables the team to compare and contrast the privacy requirements with project constraints.
8. Prioritize requirements. Stakeholders, facilitated by requirements engineers, prioritize the requirements to meet the triple constraints of effort, time, and quality. This enables the project manager and teams to see what privacy requirements are part of the system to be developed.
9. Inspect requirements. The inspection team reviews the privacy requirements and removes any defects or ambiguities. This step produces the final privacy requirements document that is accepted by the requirements team and the stakeholders.

Although we have implemented some small case studies in privacy requirements engineering as part of our research effort, it is our intention to seek out larger realistic case studies to go beyond a proof of concept. In addition, we intend to enhance the nine-step SQUARE process so that by 2012 it will support security requirements engineering, privacy requirements engineering, or both.

Abstract

As software systems become more distributed and complex, maintaining privacy of data and ensuring data integrity remain challenges for software practitioners. Developing such systems not only poses technical challenges but also demands compliance with privacy laws. Engineering precise privacy requirements is an important step in building these software systems. This technical note explores the use of a disciplined approach to identifying privacy requirements, primarily how the Security Quality Requirements Engineering (SQUARE) process, which was developed for security requirements engineering, can be adapted for privacy requirements engineering in software development.

1 Introduction

At present, concerns with privacy of personal data in software systems are widespread and increasing. Companies providing all kinds of services, such as credit card companies, stock brokerages, and insurance companies, give their customers statements on privacy. Customers are also asked to sign privacy agreements with health care providers, internet service providers, and many other contracted services. Much of our private, personal information resides in databases associated with software systems. Consequently, privacy needs to be considered early in the software development process. Privacy requirements engineering is an important area that needs additional attention given the increasing availability of private, personal data on the internet and in other automated systems. In this technical note we discuss the challenges of creating privacy requirements and explore a way of adapting a security requirements engineering process for privacy requirements.

Privacy

For this report, we define privacy as the ability of an individual to control his or her own information [Turkington 1990].

The principal mechanisms for ensuring privacy protection are not only technical but also legislative and administrative. The significant difference between security and privacy protection is that threats to individual privacy often arise from authorized users of the system rather than from unauthorized ones. In such cases, security is not breached, but privacy is. A strong privacy protection policy would keep authorized users from making unauthorized use of personal information.

A number of privacy guidelines, such as Sarbanes-Oxley, the Health Insurance Portability and Accountability Act (HIPAA), and Organization for Economic Cooperation and Development (OECD), have been defined to protect personal information in different businesses and domains. Privacy requirements should comply with these laws, standards, and service policies.

Privacy Requirements Engineering Techniques

Following is a list of existing requirements engineering techniques that have been used for privacy requirements engineering.

Goal-Based Requirements Analysis Method

The Goal-Based Requirements Analysis Method [Antón 2001a] is a systematic approach to identifying system and enterprise goals and requirements. It is useful for identifying the goals that software systems must achieve, managing tradeoffs among the goals, and converting them into operational requirements.

Pattern-Based Approach

A pattern-based approach [Barcalow 1997, Schumacher 2003, Fernandez 2001, Kienzle 2002, Konrad 2003, Mouratidis 2005] has been incorporated into software engineering as a method for object-based reuse. With this approach, security patterns are essentially best practices presented in

a template format. This format helps designers identify and understand areas of security concerns and implement appropriate corrective measures.

E-Commerce Personalization Approach

Cranor proposed a number of approaches that may help identify privacy requirements, depending on the functionality of e-commerce personalization systems [Cranor 2003]. Although the author mentions that there is no simple, universal formula for designing a privacy-protective e-commerce personalization system, Cranor does offer some useful rules of relationship between design of a personalization system and privacy principles:

- Pseudonymous profiles are a good approach when personalization information needs not be tied to personally identifiable information.
- Client-side profiles are useful when personalization services can be performed on the client.
- Task-based personalization may be appropriate when knowledge of a user's historical profile does not significantly enhance a personalization service.
- Interfaces that put users in control of the collection and use of their data as well as the types of personalization provided can make most personalization systems more privacy friendly.

Following these rules is a good approach to identifying privacy requirements. However, the rules are limited to e-commerce personalization systems.

Because all three of the above elicitation techniques are generic, they pose a number of problems when used to elicit privacy requirements. All these techniques require a detailed understanding of privacy laws, standards, and policies. Software engineers frequently find it difficult to understand legal language and intricacies, and these misunderstandings can cause a gap in requirements. Also, a systematic methodology for developing privacy requirements suitable for all software environments does not exist.

In the following section we discuss Security Quality Requirements Engineering (SQUARE) [Mead 2005], an existing technique for engineering security requirements. We then adapt this technique for privacy requirements engineering and use it in conjunction with the Privacy Requirements Elicitation Technique (PRET) [Miyazaki 2008] to establish a process for engineering privacy requirements.

SQUARE Process

Security Quality Requirements Engineering (SQUARE) is a methodology an organization can use to engineer security requirements. It was created by the CERT Program at the Software Engineering Institute (SEI), part of Carnegie Mellon University. The SQUARE process provides a means for eliciting, categorizing, and prioritizing verifiable security requirements during the early stages of a software development project. In addition to producing a set of verifiable and prioritized security requirements, the SQUARE methodology is useful for documenting and analyzing the security aspects of various systems. The CERT website (http://www.cert.org/sse/square.html) provides more information about SQUARE, including downloads of SQUARE reports, academic lecture material, workshop material and case studies, and a robust tool that supports the SQUARE process. The SEI Webinar series includes an overview of SQUARE [Mead 2009].

The SQUARE process consists of nine steps:

1. Agree on definitions.
2. Identify assets and security goals.
3. Develop artifacts.
4. Perform risk assessment.
5. Select elicitation techniques.
6. Elicit security requirements.
7. Categorize requirements.
8. Prioritize requirements.
9. Inspect requirements.

2 SQUARE for Privacy

The following sections describe each of SQUARE's nine steps and discuss the modifications needed to adapt the process for privacy requirements engineering.

Step 1 – Agree on Definitions

In this step, participants create a comprehensive list of terms that will aid effective communication and reduce ambiguity. Differences in perspective within a team can produce two kinds of problems [Mead 2005]:

- Certain terms may have multiple meanings among the participants.
- Ambiguity may exist in the level of detail assumed for a particular term.

Agreeing on a set of definitions will help the team solve these problems. To speed up the process, Table 1 provides a set of terms for privacy [Solove 2006, Common Criteria 2007, Antón 2001b, Wang 2003, Mont 2006].

Table 1: Terms for Privacy

access aggregation anonymity anonymous application of denial of service application modification appropriation authentication authorization blackmail client-side profiles collection limitation contact	confidentiality cookie credential theft data breach data controller data exposure data privacy data quality disclosure distortion exclusion exposure fair information practice	functional manipulation identification identity fraud increased accessibility information aggregation information collection information monitoring information personalization information storage information transfer insecurity interrogation intrusion	network credential theft network denial of service network exposure openness privacy privacy act privacy policy privacy protection right to privacy pseudonymity pseudonymous profile secondary use surveillance

As suggested by the SQUARE process, the list of terms should include suggested definitions for each term as well as its source, as shown in Table 2 [Mead 2005]. This will help the stakeholders understand the basic scope of each term and select one of its definitions.

Table 2: Example Term with Suggested Definition [Mead 2005]

Confidentiality	☐ The property that information is not made available or disclosed to unauthorized individuals, entities, or processes. (i.e., to any unauthorized system entity)	[SANS 2003]
	☐ The property that information is not made available or disclosed to unauthorized individuals, entities, or processes	[ISO/IEC 2005]
	☐ A quality or condition accorded to information as an obligation not to transmit that information to an unauthorized party	[National Research Council 1993]
	☐ Other:	

Step 2 – Identify Assets and Privacy Goals

The second step in the SQUARE process is to identify assets and security goals. For privacy requirements engineering, the basic idea of this step is the same, only the requirements engineering team and the stakeholders agree on a set of assets and prioritized privacy goals instead of security goals. The purpose of this step is to initiate a discussion among the stakeholders regarding their assets and overall privacy goals for the project.

Because privacy policy is driven by laws and regulations, a number of privacy goals are derived from laws like the HIPAA, Public Law 104-191, the OECD Guidelines on the Protection of Privacy and Trans-border Flows of Personal Data, and the Personal Information Protection Act (PIPA).

The following are some examples goals for privacy:

- Ensure that personal data is collected with the user's permission.
- Ensure that the data collected for a specific purpose is not used for other purposes without appropriate authorization.
- Ensure that the user is aware of the purpose for which personal data is collected.

Step 3 – Collect Artifacts

In this step, participants collect the relevant artifacts for the system being developed. These artifacts may clarify an existing system or clarify the purpose and environment for the proposed system.

With respect to privacy, some of the relevant artifacts include

- system architecture diagrams
- use case scenarios and diagrams
- misuse case scenarios and diagrams
- attack trees
- user-role hierarchies

System Architecture Diagrams

System architecture diagrams provide an overview of the system as it exists. A dynamic perspective of a system can show how data flows among the different components. Because privacy is concerned with vulnerabilities with respect to data, the architecture diagrams can determine data-flow connections that could be vulnerable to attack as well as connections between components and their data-flow dependencies.

A system architecture diagram can also help determine how the system stores data and how secure those data stores are.

Use Case Scenarios/Diagrams

Privacy use cases will mostly be related to how the system handles user data and how the system components interact with each other. They help the stakeholders and the requirements engineering team gain a better understanding of the system and its requirements.

Misuse Case Scenarios/Diagrams

Misuse cases identify the vulnerabilities of the system and can be used to make the system more resistant to such attacks. They also identify the risks that the system faces.

Consider the misuse case shown in Figure 1. It identifies the connections in communication that may be vulnerable to attack.

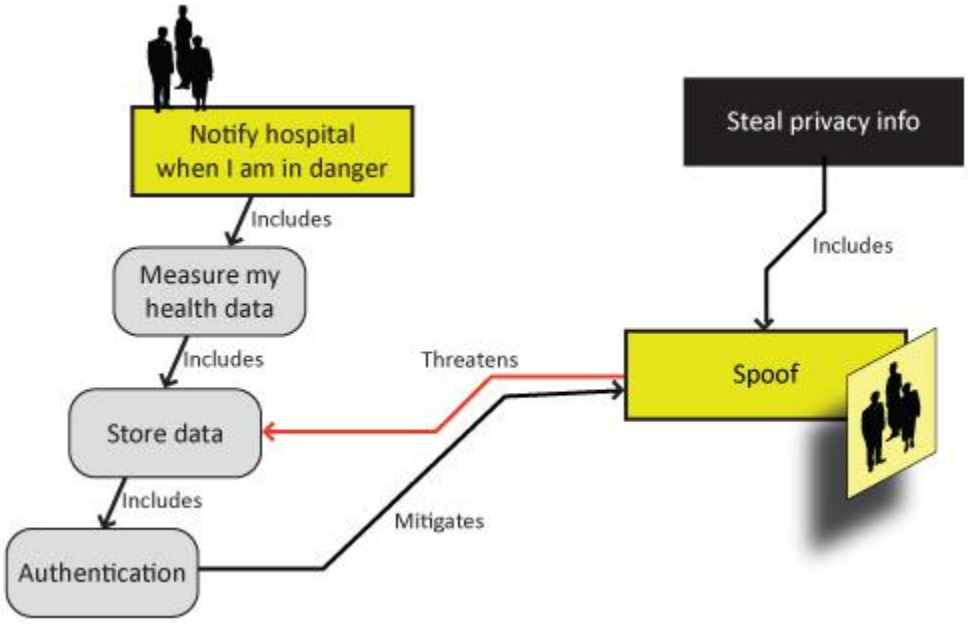


Figure 1: Misuse Case [Miyazaki 2008]

Some of the requirements that can be derived from the above misuse case include the following:

- The system network communications must be protected from unauthorized information gathering and eavesdropping.
- The system shall provide a data backup mechanism.
- The system shall have functional audit logs and usage reports that do not disclose identity information.
- The system shall have strong authentication measures in place at all system gateways and entrance points.

Attack Trees

The purpose of an attack tree is to model threats to the system by focusing on the attackers and the ways they may attack the system [Schneier 2000]. The goal of the attack is represented as the root node, and leaf nodes describe the different ways in which that goal may be achieved. Figure 2 shows an example attack tree.

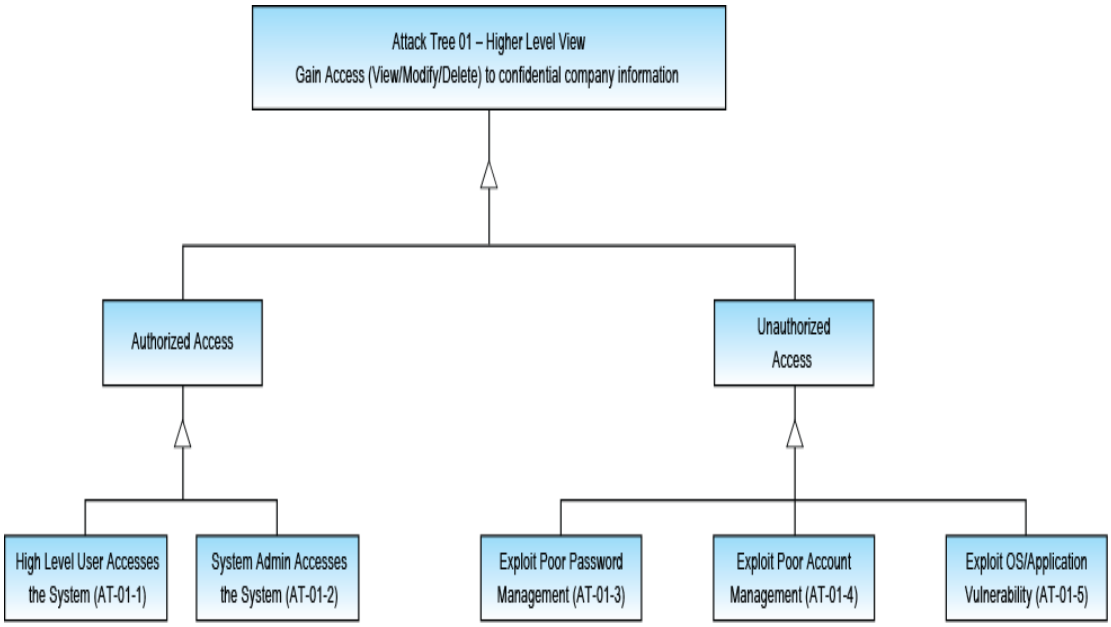


Figure 2: Example Attack Tree [Mead 2010]

Using this knowledge, the stakeholders and the requirements engineering team can determine the ways the system can be protected from potential attacks.

User-Role Hierarchies

Privacy-related systems are required to implement a role-based access control mechanism. Because data is the central point for privacy, it is critical to determine who can access which data. For this purpose, a role-based hierarchy for the system can determine the access control requirements.

Step 4 – Risk Assessment

Risk assessment for privacy and security requirements identifies the vulnerabilities and threats that the system faces, the likelihood that the threats will materialize as real attacks, and the potential consequences of an attack, if any. Risk assessment establishes a rationale for choosing and implementing the privacy requirements. Identification and prioritization of risks also help to prioritize privacy requirements later in the elicitation process [Mead 2005].

Privacy risk assessment identifies vulnerabilities with respect to data and how it can be compromised. As such, it takes into account the policies, regulations, and laws for privacy. Because security risk assessment does not necessarily consider laws and regulations, the goals of privacy risk assessment tend to be different from the goals of security risk assessment.

A number of different privacy laws govern different industries and domains. Some of these laws and regulations provide certain guidelines that can be used to assess privacy risks. For example, the HIPAA addresses privacy concerns of health information systems by enforcing data exchange standards. Privacy Impact Assessment (PIA) is a comprehensive process for determining the privacy, confidentiality, and security risks associated with the collection, use, and disclosure of personal information [Abu-Nimeh 2009].

The privacy risk assessment focuses on the following [Abu-Nimeh 2009]:

- nature of data collected
- purpose of data collection
- procedures for obtaining an individual's consent
- compliance to regulations
- necessity and accuracy of data

Furthermore, the privacy risk assessment checks and analyzes the following [Abu-Nimeh 2009]:

- authorization and authentication requirements
- risk of theft
- third-party vulnerabilities

According to Boehm, any risk assessment should take into account these three steps [Boehm 1991]:

1. risk identification. A list of potential risks should be generated using available project information and requirements.
2. risk analysis. After risks have been identified, they need to be analyzed with respect to their probability of occurrence and their potential impact on the project or system.
3. risk prioritization. The risks then need to be ranked by importance based on the probability of occurrence and degree of impact.

Currently, there exists a body of privacy literature and laws that focuses on some privacy areas of interest. Requirements engineers, risk experts, and stakeholders can classify these works by analyzing what steps of risk assessment process they address [Panusuwan 2009]. Using this classification, participants can select the risk assessment methods that suit their requirements.

A report on privacy risk assessment published by the CERT Program illustrates case studies of two projects that used different risk assessment techniques to identify, analyze, and prioritize risks [Panusuwan 2009]. Another paper describes how PIA [Flaherty 2000] and the HIPAA can be used to assess privacy risks in conjunction with security risk assessment techniques that are used in the SQUARE methodology [Abu-Nimeh 2009].

Step 5 – Select Elicitation Technique

In this step, the requirements engineering team selects one elicitation technique that is suitable for the project and the clients and that elicit all the requirements from the stakeholders [Mead 2005]. Some of the techniques that they consider are

- structured/unstructured interviews
- use/misuse cases [Jacobson 1992]
- facilitated meeting sessions, such as joint application development and the accelerated requirements method [Hubbard 1999, Wood 1989]
- soft systems methodology [Checkland 1990]
- issue-based information systems [Kunz 1970]
- quality function deployment [QFD Institute 2005]

- feature-oriented domain analysis [Kang 1990]

To adapt SQUARE for privacy requirements elicitation, we suggest that the requirements engineering team use the PRET, a computer-aided technique that helps the requirements engineering team elicit and prioritize privacy requirements more efficiently [Miyazaki 2008]. This technique uses a database of privacy requirements based on privacy laws and regulations such as the OECD and PIPA. Using a questionnaire and a decision process, the tools create a list of privacy requirements and their priorities. The PRET makes it faster and easier to elicit requirements and prevent leaks when the team is not familiar with the laws and regulations. However, the PRET currently does not contain all the privacy laws, and the database needs to be updated as the laws change. Also, because the PRET is a generic tool, the requirements are general, and the requirements engineering team needs to verify and tailor them to the specific needs of the project.

Step 6 – Elicit Security Requirements

The PRET can also be used in this step, during which privacy requirements are elicited. The requirements engineering team guides the stakeholders through the five-page PRET questionnaire, part of which is shown in Figure 3.

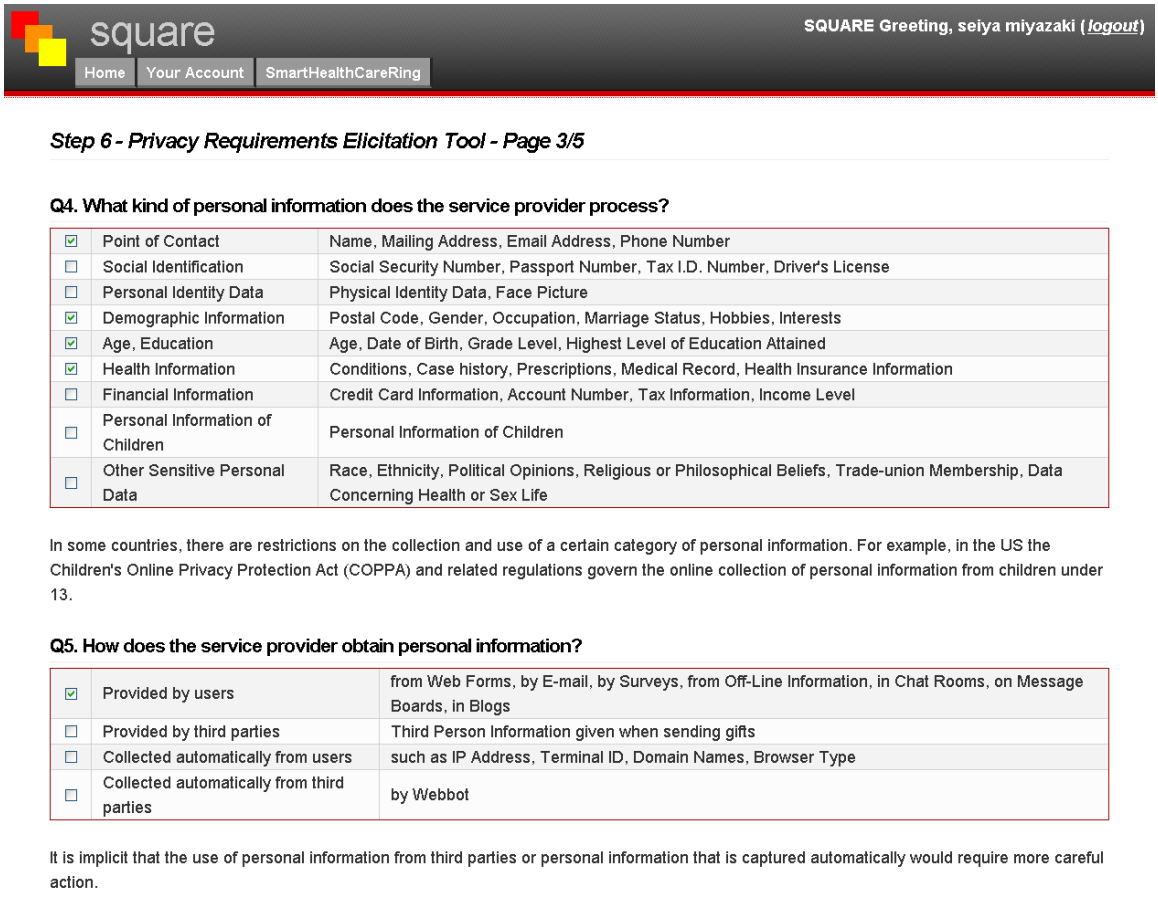
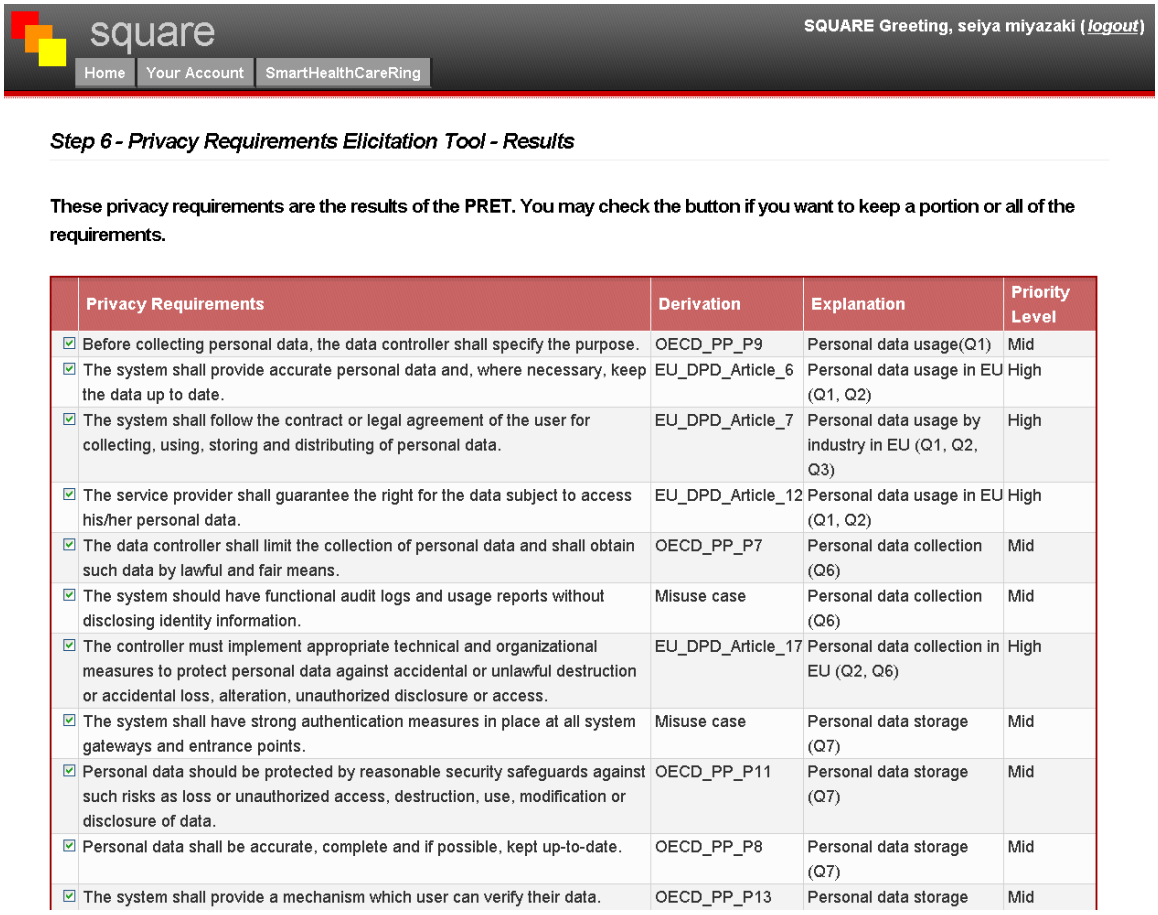


Figure 3: Page 3 of the PRET Questionnaire [Miyazaki 2008]

After the stakeholders complete the questionnaire, the PRET tool results page shows some requirements, their derivation, their derivation explanations, and their priority levels (see Figure 4).



	Privacy Requirements	Derivation	Explanation	Priority Level
☑	Before collecting personal data, the data controller shall specify the purpose.	OECD_PP_P9	Personal data usage(Q1)	Mid
☑	The system shall provide accurate personal data and, where necessary, keep the data up to date.	EU_DPD_Article_6	Personal data usage in EU (Q1, Q2)	High
☑	The system shall follow the contract or legal agreement of the user for collecting, using, storing and distributing of personal data.	EU_DPD_Article_7	Personal data usage by industry in EU (Q1, Q2, Q3)	High
☑	The service provider shall guarantee the right for the data subject to access his/her personal data.	EU_DPD_Article_12	Personal data usage in EU (Q1, Q2)	High
☑	The data controller shall limit the collection of personal data and shall obtain such data by lawful and fair means.	OECD_PP_P7	Personal data collection (Q6)	Mid
☑	The system should have functional audit logs and usage reports without disclosing identity information.	Misuse case	Personal data collection (Q6)	Mid
☑	The controller must implement appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access.	EU_DPD_Article_17	Personal data collection in EU (Q2, Q6)	High
☑	The system shall have strong authentication measures in place at all system gateways and entrance points.	Misuse case	Personal data storage (Q7)	Mid
☑	Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorized access, destruction, use, modification or disclosure of data.	OECD_PP_P11	Personal data storage (Q7)	Mid
☑	Personal data shall be accurate, complete and if possible, kept up-to-date.	OECD_PP_P8	Personal data storage (Q7)	Mid
☑	The system shall provide a mechanism which user can verify their data.	OECD_PP_P13	Personal data storage	Mid

Figure 4: Results Page of the PRET Tool [Miyazaki 2008]

After verifying the requirements, the team needs to select the desired requirements.

Using the PRET tool makes it easy for the team to come up with a general set of privacy requirements. After eliciting the initial set in this way, the team can elaborate on the requirements through other techniques.

Step 7 – Categorize Requirements

The aim of this step is to systematically categorize requirements to help in the next step of the process, requirements prioritization. This step also facilitates team discussion of the requirements and separates requirements from the constraints for the project.

In this step, the requirements engineering team guides the stakeholders to categorize requirements through discussion. The requirements engineering team provides the stakeholders with a set of basic categories and explains the process of categorization. The stakeholders may use the given set or add new categories to the set.

Table 3 is an example of a matrix that can be used to categorize requirements.

Table 3: Requirements Categorization Matrix [Mead 2005]

	System Level	Software Level	Architectural constraints
Essential			
Non-essential			

The above matrix provides a generic way of categorizing requirements. However, because a number of privacy-related requirements have legal implications, the team may want to use a categorization that suits privacy requirements. One method for prioritizing legal requirements uses the following categories [Massey 2009]:

- nonlegal requirements
- legal requirements needing further refinement
- implementation-ready legal requirements

Step 8 – Prioritize Requirements

In most software projects, limits on time, resources, and acceptable cost prevent implementation of all requirements. Requirements prioritization helps stakeholders arrange the elicited requirements in the desired implementation order. A good requirement prioritization has some advantages, such as the following [Karlsson 1996, Karlsson 1997]:

- It clarifies for the project manager which requirements are important and which are mere embellishments.
- It provides a means to make tradeoffs between conflicting goals such as quality, cost, and time to market.
- It helps the project manager plan releases that will meet the customers' expectations.

These advantages demonstrate the criticality of requirements prioritization in the requirements engineering process. This prioritization effectively determines what requirements eventually get built into the software. The results of the risk assessment performed in Step 4 and the categorization of Step 7 are important inputs for this step, the output of which is a list of requirements, along with their priorities, that all the stakeholders have agreed upon.

Many unstructured and structured techniques can be used for the process of requirements prioritization. Unstructured methods involve simple discussions between the stakeholders with a goal of consensus on priorities for requirements. The following section briefly explains some of the structured techniques that can be used to prioritize privacy requirements.

Pair-Wise Comparison Method [Karlsson 1996]

The pair-wise comparison method is based on the analytical hierarchy process (AHP) [Saaty 1980] and derives the relative importance of one requirement over another. Given a set of *n* requirements, the method requires *n*(n-1)/2* comparisons. Using the values given for each comparison, mathematical formulas can be used to derive the prioritization for the *n* requirements.

Using pair-wise comparison facilitates the prioritization task. In general it is easier to decide the relative priority of two requirements than of many. Also, the priorities are derived as percentages, and the total of all priorities is always 100 percent; a requirement with a priority of 40 percent

would represent 40 percent of the total importance. These priority percentages can be used to calculate the customers' satisfaction with the requirements delivered within the software.

Also, pair-wise comparison data can be used to evaluate the consistency of the comparisons. Good comparison consistency implies consistent prioritization and establishes a level of confidence for the prioritization process.

This method of prioritization can be combined with a cost-value analysis approach to provide a better way of prioritizing requirements [Karlsson 1997]. The AHP method uses pair-wise comparison to calculate the relative implementation costs. The calculated cost and value can be plotted on a diagram to create a cost-value plot. The stakeholders can then use this diagram to determine the priorities based on a cost-value analysis.

A Method for Prioritization of Legal Requirements [Massey 2009]

A number of privacy-related requirements have legal implications. Laws and regulations affect privacy requirements prioritization because noncompliance carries high cost penalties and because prioritizing requirements demands a considerable amount of domain knowledge. This method of prioritizing requirements has two steps:

1. Find legal implications. In this step, we use the required legal text as input. The main aim of this step is to map the requirements to the subsections in the legal text with the help of legal-domain experts.
2. Calculate a prioritization score for every requirement. This step uses the mapping from the first step to calculate a prioritization score based on the following formula:

$$P = \sum_{R=1}^{n} (S_M(R) + C(R) + E(R) + S_C(R)$$

Where

P	prioritization score
n	number of requirements
R	particular requirement
S_M	number of subsections mapped
S_C	number of subsections contained (by the subsections to which a requirement is mapped)
C	number of cross-references
E	number of exceptions

A lower value of P indicates a greater readiness for implementation, whereas a higher score indicates a need for further refinement of the requirements.

This method prioritizes requirements based on their legal implementation readiness. Because this method deals with only the legal aspect, we suggest using other methods, such as the pair-wise method using the AHP, to prioritize requirements based on other criteria [Massey 2009].

Step 9 – Inspect Requirements

Requirements inspection is the last step in the process and a very important one. Inspections remove defects and clear ambiguities in the requirements. Reports suggest that more defects are introduced in the requirements-gathering phase of the software development life cycle than during any other phase [Kelly 1992], so inspections are a critical step in the requirements-gathering process.

Inspections can be informal or formal. There exist a number of methods to carry out inspections, ranging from ad hoc to use of checklists and even Fagan reviews and scenario-based inspections. Various experiments show that scenario-based inspection methods provide a better defect detection rate than checklist or ad hoc inspections [Porter 1995].

The requirements engineering team can guide the stakeholders to use any of the available inspection methods to perform this step in the process.

The outcome of this process is a final privacy requirements document that has been agreed upon and verified by all the stakeholders and the requirements engineering team.

3 Further Work

Through our analysis, we have seen that small modifications can be made to adapt the SQUARE process for engineering privacy requirements. To assess the effectiveness of this adaption of SQUARE, we need to implement case studies and evaluate their results. During our privacy research work, we have implemented several small case studies, but these do not provide the level of confidence that medium-to-large case studies in the field would provide.

Second, to facilitate use of SQUARE for privacy requirements engineering, the tool needs to be modified. Although the steps may appear to be similar on the surface, different techniques come into play in the automated support provided by the tool. Although a prototype version of the SQUARE tool with the PRET exists, a robust, integrated tool is needed that will support security, privacy, or both. In particular, the PRET steps must be merged into the SQUARE tool. Further, the reliability of the PRET tool itself needs to be improved. The PRET database must be enhanced to cover other laws such as the Leech-Bliley Act, the Financial Privacy Act, the Electronic Communications Privacy Act, PIPA (which applies in Canada and other countries), and the Cable Communications Policy Act. As the database size increases, the PRET tool's priority calculation algorithm will also need to be updated to produce a verifiable set of privacy requirements. The robust SQUARE privacy tool will be developed by a Carnegie Mellon University Master of Software Engineering Studio team from fall 2010 through fall 2011. The new tool will support security, privacy, or both. The expected public release of the tool will be in spring 2012, when it will be available for download.

References

[Abu-Nimeh 2009]
Abu-Nimeh, Saeed & Mead, Nancy R. "Privacy Risk Assessment in Privacy Requirements Engineering," 17-18. *Second International Workshop on Requirements Engineering and Law* (ReLAW '09). Atlanta, GA, Sept. 2009. IEEE, 2009.

[Antón 2001a]
Antón, A. I.; Carter, R. A.; Dagnino, A.; Dempster, J. H.; & Siege, D. F. "Deriving Goals from a Use-Case Based Requirements Specification." *Requirements Engineering Journal 6*, 1 (February 2001): 63-73.

[Antón 2001b]
Antón, Annie I. & Earp, Julia B. *A Taxonomy for Web Site Privacy Requirements* (NCSU Technical Report TR-2001-14). North Carolina State University at Raleigh, 2001.

[Barcalow 1997]
Barcalow, J. & Yoder, J. "Architectural Patterns for Enabling Application Security." *Proceedings of the Fourth Conference on the Pattern Languages of Programs* (Washington University Technical Report wucs-97-34). Monticello, IL, Sept. 1997.

[Boehm 1991]
Boehm, B. W. "Software Risk Management: Principles and Practices." *IEEE Software 8*, 1 (January 1991): 32-41.

[Checkland 1990]
Checkland, Peter. *Soft System Methodology in Action.* John Wiley & Sons, 1990.

[Common Criteria 2007]
Common Criteria. *Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components (CC), Version 3.1, Revision 2* (CCMB-2007-09-002). September 2007. http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R2.pdf

[Cranor 2003]
Cranor, Lorrie Faith. "'I Didn't Buy it for Myself': Privacy and Ecommerce Personalization" 111-117. *Proceedings of the Workshop on Privacy in the Electronic Society* (WPES '03). Washington, D.C., Oct. 2003. Association for Computing Machinery, 2003.

[Fernandez 2001]
Fernandez, E. & Pan, R. "A Pattern Language for Security Models." *Proceedings of the Eighth Conference on the Pattern Languages of Programs.* Monticello, IL, Sept. 2001. The Hillside Group, 2001.

[Flaherty 2000]
Flaherty, D. H. "Privacy Impact Assessments: An Essential Tool for Data Protection." *22nd Annual Meeting of Privacy and Data Protection Officials.* Venice, Italy, September 2000. Reprinted

in *Privacy Law & Policy Reporter 45* (2000).
http://www.austlii.edu.au/au/journals/PLPR/2000/45.html

[Hubbard 1999]
Hubbard, R. "Design, Implementation, and Evaluation of a Process to Structure the Collection of Software Project Requirements." PhD diss., Colorado Technical University, 1999.

[ISO/IEC 2005]
International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC). *ISO/IEC 27001:2005 Information technology — Security techniques — Information security management systems — Requirements*. ISO/IEC, 2005.

[Jacobson 1992]
Jacobson, Ivar. *Object-Oriented Software Engineering: A Use Case Driven Approach.* Boston, MA: Addison-Wesley, 1992.

[Kang 1990]
Kang, Kyo C.; Cohen, Sholom G.; Hess, James A.; Novak, William E.; & Peterson, A. S. *Feature-Oriented Domain Analysis (FODA) Feasibility Study* (CMU/SEI-90-TR-021). Software Engineering Institute, Carnegie Mellon University, 1990.
http://www.sei.cmu.edu/library/abstracts/reports/90tr021.cfm

[Karlsson 1996]
Karlsson, Joachim. "Software Requirements Prioritizing," 110-116. *Proceedings of the International Conference on Requirements Engineering* (ICRE '96). Colorado Springs, CO, April 1996. IEEE, 1996.

[Karlsson 1997]
Karlsson, Joachim & Ryan, Kevin. "A Cost-Value Approach for Prioritizing Software Requirements." *IEEE Software 14*, 5 (September/October 1997): 67-74.

[Kelly 1992]
Kelly, J. C.; Sherif, J. S.; & Hops, J. "An Analysis of Defect Densities Found During Software Inspection." *Journal of Systems Software 17*, 2 (February 1992): 111-117.

[Kienzle 2002]
Kienzle, Darrell M. & Elder, Matthew C. *Final Technical Report: Security Patterns for Web Application Development* (DARPA Contract # F30602-01-C-0164). Security Patterns, 2002.
http://www.scrypt.net/~celer/securitypatterns/final%20report.pdf

[Konrad 2003]
Konrad, Sascha; Cheng, Betty H. C.; Campbell, Laura A.; & Wassermann, Ronald. "Using Security Patterns to Model and Analyze Security Requirements," 13-20. *Proceedings of the International Workshop on Requirements for High Assurance Systems* (RHAS03). Monterey Bay, CA, Sept. 2003. IEEE Press, 2003.

[Kunz 1970]
Kunz, Werner & Rittel, Horst W. J. *Issues as Elements of Information Systems* (Working Paper No. 131). Institute of Urban and Regional Development, University of California, 1970. http://iurd.berkeley.edu/sites/default/files/wp/131.pdf

[Massey 2009]
Massey, Aaron K.; Otto, Paul N.; & Antón, Annie I. "Prioritizing Legal Requirements," 27-32. *Second International Workshop on Requirements Engineering and Law* (ReLAW '09). Atlanta, GA, Sept. 2009. IEEE, 2009.

[Mead 2005]
Mead, Nancy R.; Hough, Eric D.; & Stehney, Theodore R., II. *Security Quality Requirements Engineering (SQUARE) Methodology* (CMU/SEI-2005-TR-009). Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm

[Mead 2009]
Mead, Nancy R. *SQUARE Up Your Security Requirements Engineering with SQUARE.* Software Engineering Institute, Carnegie Mellon University, 2009. http://www.sei.cmu.edu/library/abstracts/webinars/14may2009.cfm

[Mead 2010]
Mead, Nancy R. *SQUARE Series Lecture 3, Detail (Part 1).* Software Engineering Institute, Carnegie Mellon University, 2010. http://www.cert.org/sse/square/square-description.html

[Miyazaki 2008]
Miyazaki, S.; Mead N. R.; & Zhan, J. "Computer-Aided Privacy Requirements Elicitation Technique," 367-372. *Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference* (APSCC). Yilan, Taiwan, December 2008. IEEE, 2008.

[Mont 2006]
Mont, M. C. & Thyne, R. "Privacy Policy Enforcement in Enterprises with Identity Management Solutions." *Proceedings of the 2006 international Conference on Privacy, Security and Trust: Bridge the Gap between PST Technologies and Business Services*. Markham, Ontario, Oct. 2006. Association for Computing Machinery, Vol. 380, 2006.

[Mouratidis 2005]
Mouratidis, H.; Weiss, M.; & Giorgini, P. "Security Patterns Meet Agent Oriented Software Engineering: A Complementary Solution for Developing Security Information Systems," 225-240. *Proceedings of the Twenty-Fourth International Conference on Conceptual Modeling* (ER05). Klagenfurt, Austria, Oct. 2005. In *Lecture Notes in Computer Science 3716*, Springer, 2005.

[National Research Council 1993]
National Research Council and Social Science Research Council. *Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics.* Edited by G.T. Duncan, T.B. Jabine, and V.A. de Wolf. Commission on Behavioral and Social Sciences and Education, Committee on National Statistics. Washington, DC: National Academy Press, 1993.

[Panusuwan 2009]
Panusuwan, Varokas & Batlagundu, Prashanth. *Privacy Risk Assessment Case Studies in Support of SQUARE* (CMU/SEI-2009-SR-017). Software Engineering Institute, Carnegie Mellon University, 2009. http://www.sei.cmu.edu/library/abstracts/reports/09sr017.cfm

[Porter 1995]
Porter, Adam A.; Votta, Lawrence G., Jr.; & Basili, Victor R. "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment." *IEEE Transactions on Software Engineering 21,* 6 (June 1995): 563-575.

[QFD Institute 2005]
QFD Institute. *Frequently Asked Questions About QFD*. http://www.qfdi.org/what_is_qfd/faqs_about_qfd.htm (2005).

[SANS 2003]
The SANS Institute. "SANS Glossary of Terms Used in Security and Intrusion Detection." http://www.sans.org/resources/glossary.php (2003).

[Saaty 1980]
Saaty, T. L. *The Analytic Hierarchy Process.* McGraw-Hill, 1980.

[Schneier 2000]
Schneier, Bruce. *Secrets and Lies: Digital Security in a Networked World.* John Wiley & Sons, 2000.

[Schumacher 2003]
Schumacher, Markus. *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*. Springer-Verlag, 2003.

[Solove 2006]
Solove, Daniel J. "A Taxonomy of Privacy." *University of Pennsylvania Law Review 154,* 3 (January 2006): 477-560.

[Turkington 1990]
Turkington, R. C. "Legacy of Warren and Brandeis Article: The Emerging Unencumbered Constitutional Right to Information Privacy." *Northern Illinois University Law Review 10,* 3 (Summer 1990): 479.

[Wang 2003]
Wang, H. & Wang, C. "Taxonomy of Security Considerations and Software Quality." *Communications of the ACM 46,* 6 (June 2003): 75-78.

[Wood 1989]
Wood, Jane & Silver, Denise. *Joint Application Design: How to Design Quality Systems in 40% Less Time*. Wiley & Sons, 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	July 2010	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Adapting the SQUARE Process for Privacy Requirements Engineering	FA8721-05-C-0003

6. AUTHOR(S)

Ashwini Bijwe, Nancy R. 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-2010-TN-022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

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

13. ABSTRACT (MAXIMUM 200 WORDS)

As software systems become more distributed and complex, maintaining privacy of data and ensuring data integrity remain challenges for software practitioners. Developing such systems not only poses technical challenges but also demands compliance with privacy laws. Engineering precise privacy requirements is an important step in building these software systems. This technical note explores the use of a disciplined approach to identifying privacy requirements, primarily how the Security Quality Requirements Engineering (SQUARE) process, which was developed for security requirements engineering, can be adapted for privacy requirements engineering in software development.

14. SUBJECT TERMS	15. NUMBER OF PAGES
security requirements engineering, software security	32

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