SQUARE for Acquisition: Case Study 2

June 2010

This work was funded by

# 

Copyright 2010 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use. This document may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Table of Contents

[Introduction 1](#_Toc265158764)

[The Companies Involved 1](#_Toc265158766)

[The Acquisition Organization 1](#_Toc265158767)

[The Contracted Organization 1](#_Toc265158768)

[The Roles 2](#_Toc265158769)

[The SQUARE Process 3](#_Toc265158770)

[Step 1 – Agree on Definitions 4](#_Toc265158771)

[Step 2 – Identify Assets and Security Goals 6](#_Toc265158772)

[Step 3 – Develop Artifacts 7](#_Toc265158773)

[Step 4 – Perform Risk Assessment 9](#_Toc265158774)

[Step 5 – Select Elicitation Techniques 11](#_Toc265158775)

[Step 6 – Elicit Security Requirements 13](#_Toc265158776)

[Step 7 – Categorize Requirements 14](#_Toc265158777)

[Step 8 – Prioritize Requirements 16](#_Toc265158778)

[Step 9 – Inspect Requirements 18](#_Toc265158779)

[Step 10 – Review of Requirements by Acquisition and Contractor Organizations 20](#_Toc265158780)

[References 21](#_Toc265158781)

# Introduction

This document presents a case study for use in a course or workshop. The case study shows how to adapt the Security Quality Requirements Engineering (SQUARE) process for a typical software acquisition. In this case, the acquisition organization defines the requirements as part of the Requirements for Proposal (RFP) process, taking care to ensure the requirements are not so low-level that they constrain the contractor organization. The acquisition organization then delivers the defined requirements to the contractor organization for further work. This case study differs from Case Study 1 in which the acquisition organization also has the client role but the contractor is responsible for the requirements identification.

The student participants will use SQUARE as the underlying process framework for defining the security requirements. This document provides background on the case study, elaborates each step of the SQUARE process, and gives guidance on how to complete each process step.

## The Companies Involved

### The Acquisition Organization

The acquisition organization is a growing IT services company that provides banking solutions and software support worldwide. It provides a comprehensive range of financial services, including personal financial services, commercial banking, corporate and investment banking, private banking, and consumer finance, and other related services.

As part of its financial growth and to sustain the growing number of employees working in the company, the organization wishes to improve its human resource management systems by acquiring a company that specializes in providing these same services as well as customer relationship management software. This acquisition would help the company better manage their operations, such as

* managing employee payrolls
* maintaining a management information system for employee records
* managing employee skill sets

### The Contracted Organization

The contracted, or contractor, organization provides human resource management and customer relationship management software. Its clients are typically large organizations with a large employee base. The software is deployed using a client-server style and allows the company to access the functionalities provided by the software from a web browser. The software is generally written in a proprietary scripting language (for example, PeopleCode) and has the ability to interface with an SQL Database.

## 

## The Roles

The group will be split into two teams:

* acquisition team**—**This team represents people from the acquisition organization acting as the client. A brief description about your company and the kind of software it is acquiring has been explained in *The Companies Involved* section.
* contractor team**—**This team represents people who are involved in building and developing the software. This team ultimately gets acquired by the acquisition team as part of the software acquisition process. This team will have different roles and responsibilities as the SQUARE process is implemented.

In this case study, the acquiring organization is defining the requirements. The contractor organization will do the remaining software development activities. For the purpose of this case study, we use SQUARE as our underlying process framework to help define the security requirements. It is assumed that the contract award has been made and the contractor is on board. Table 1 describes the stakeholders’ roles in the implementation of the SQUARE process.

Table 1: Acquisition and Contractor Organization Roles in the SQUARE Process

|  |  |  |  |
| --- | --- | --- | --- |
| **Step Number** | **Step** | **Acquisition Organization** | **Contractor** |
| 1 | Agree on definitions | ✓ | 🗶 |
| 2 | Identify assets and security goals | ✓ | 🗶 |
| 3 | Develop artifacts to support security requirements definition | ✓ | 🗶 |
| 4 | Perform Risk Assessment | ✓ | 🗶 |
| 5 | Select elicitation techniques | ✓ | 🗶 |
| 6 | Elicit security requirements | ✓ | 🗶 |
| 7 | Categorize requirements as to level(system, software etc) and whether they are requirements or other kinds of constraints | ✓ | 🗶 |
| 8 | Prioritize requirements | ✓ | 🗶 |
| 9 | Review and inspect requirements | ✓ | 🗶 |
| 10 | Review of requirements by acquisition and contractor organizations | ✓ | ✓ |

# The SQUARE Process

In the remainder of this document are descriptions of each of the ten steps in the adapted SQUARE process for a typical software acquisition, tasks for workshop participants, and samples to help with the tasks. The participants may use some other technique or methodology to complete the steps; the examples are only meant to serve as a guideline.

## Step 1 – Agree on Definitions

To establish a common understanding among the stakeholders in the acquisition organization, they must first agree on a common set of definitions to be used throughout the SQUARE process. The purpose of this step is to ensure that there are no ambiguities in terms and that everyone has the same perspective on these terms.

|  |
| --- |
| **Task**  This task for the acquisition organization is to develop the first set of definitions. The exit criterion for this task is a well-documented, agreed-to set of definitions. These definitions will be used throughout the entire SQUARE process.  Sample definitions and guidance are below.  Time: 20 minutes |

Sample Definitions

As a starting point, participants can refer Table 2, which was part of the initial SQUARE case study [Mead 2005, Table 24]. Standard definitions are also available in popular public resources such as IEEE [IEEE 1990], the Software Engineering Book of Knowledge [IEEE 2005], and Wikipedia [Wikipedia 2010].

Table 2: Sample Terms to be Defined

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| access control | access control list | antivirus software | artifact | asset |
| attack | auditing | authentication | availability | back door |
| breach | brute force | buffer overflow | cache cramming | cache poisoning |
| confidentiality | control | corruption | cracker | DoS attack |
| disaster recovery plan | disclosure | disgruntled employee | downtime | disruption |
| encryption | espionage | essential services | exposure | fabrication |
| fault line attacks | fault tolerance | firewall | hacker | honey pot |
| HTTP header manipulation | impact | incident | incident handling | integrity |
| interception | interruption | intrusion | intrusion detection system | intrusion prevention system |
| liability | luring attack | malware | man-in-the-middle attack | masquerade |
| modification | non-essential services | non-repudiation | patch | penetration |
| penetration  testing | physical  security | port scanning | privacy | procedure |
| recognition | recovery | replay attack | resilience | resistance |
| risk | risk assessment | security policy | script kiddies | spoof |
| SQL injection | stakeholder | stealthing | survivability | target |
| threat | threat assessment | threat model | toolkits | Trojan |
| trust | uptime | victim | virus | vulnerability |
| worm |  |  |  |  |

Examples of a few definitions are

* Trojan—anon-self-replicating malware that appears to perform a desirable function for the user but instead facilitates unauthorized access into the user’s computer system
* man-in-the-middle attack—a kind of an eavesdropping technique in which an attacker makes independent conversations with victims and transmits messages between them, making them believe they are directly talking over a private secure connection, when, in fact, the entire conversation is being controlled by the attacker
* confidentiality—ensuring that information is accessible only to authorized individuals and that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity)
* authentication—the process of attempting to verify the digital identity of the sender of a communication, such as a request to log in
* auditing—the information gathering and analysis of assets to ensure such things as policy compliance and security are protected from vulnerabilities

## Step 2 – Identify Assets and Security Goals

The purpose of this step is to agree on a set of assets and security goals for the project. The goals are the origin from which the security requirements of the project can be mapped. The business goal or objective sets the tone for the security goals. In this step, assets are identified and a high-level business goal is first defined and then broken down into its constituent security goals.

|  |
| --- |
| **Task**  Using the information in *The Companies Involved* section, the stakeholders in the acquisition organization will agree on common business goal, identify assets, and prioritize security goals. The exit criteria for this process are assets, a single business goal, and several prioritized security goals for the project.  Samples of a business goal, assets, and a prioritized set of security goals follow.  Time: 15 minutes |

Business Goal Example

This tool provides the means to make an informed decision about a particular employee in the company.

Sample Assets

* confidential data of all employees
* list of critical systems being used in the project
* access control rights of all users on high security systems
* security and inspection procedures for using the tools

Security Goals Example

These goals are prioritized from 1 (highest) to 5 (lowest).

|  |  |
| --- | --- |
| **Security Goal** | **Priority** |
| Management shall exercise effective control over the system’s configuration and usage. | 1 |
| The confidentiality, accuracy, and integrity of the tool’s data shall be maintained. | 2 |
| The system shall be available for use when needed. | 3 |
| System performance should not be significantly impacted by the new security measures. | 4 |
| The system should prevent unauthorized access to an employee’s sensitive information. | 5 |

## Step 3 – Develop Artifacts

The purpose of this step is to collect a complete set of artifacts for the system. Typically, the artifacts are system architecture diagrams, use-case scenarios, misuse case scenarios, and attack trees. This serves as a useful knowledge repository to be used later for reference on any specific information that needs to be procured for the project.

|  |
| --- |
| **Task**  This task for the acquisition organization is to generate as many system artifacts as possible. The exit criterion for this process step is a set of complete artifacts generated by the acquisition organization.  A sample artifact of an attack tree is in Figure 1 for reference.  Time: 15 minutes |



Figure 1: Attack Tree

This attack tree shows the vulnerability of unauthorized login to a server due to various reasons, and what kinds of attacks are possible due to the unauthorized access [Mead 2005].

****

Figure 2: Misuse Case

Figure 2 presents an example of a misuse case diagram. This shows how an actor (maybe an attacker in this case) gains access to the system illegally and, through the development tool (Edit Plus), gains access into the central server of the system. This misuse case shows how the actor can interact with the system and perform unwanted operations such as tweaking of input validations [Mead 2005].

## Step 4 – Perform Risk Assessment

The next step is to perform risk assessment. The goal of this step is to determine a risk assessment method, elicit risks in an elicitation session and also from the existing data such as misuse cases, scenarios, and goals, and to come out with a risk assessment.

Selecting Risk Assessment Techniques

There are numerous risk assessment techniques. The following is a possible list from [Mead 2005], with references for further information:

* General Accounting Office’s (GAO’s) models [USGA 1999]
* National Institute of Standards and Technology’s (NIST’s) models [Stoneburner 2002]
* National Security Agency’s INFOSEC Assessment Methodology (IAM) [NSA 2004]
* Shawn Butler’s Security Attribute Evaluation Method (SAEM) [Butler 2002]
* CERT/CC’s Vendor Risk Assessment & Threat Evaluation (V-RATE) [Lipson 2001]
* Yacov Haimes’ Risk Filtering, Ranking, and Management (RFRM) Framework [Haimes 2004]
* CERT/CC’s Survivable Systems Analysis (SSA) Method [CERT/CC 2002]
* Martin Feather’s Defect Detection and Prevention (DDP) Process [Cornford 2004]

Since there are numerous techniques that can be used for risk assessment, a structured decision-making technique can help in picking the right assessment method. Multi-criteria decision analysis (MCDA) is one technique that can help. The following are the steps of this process:

1. Define the problem.
2. Find all possible risk assessment methods.
3. Generate a set of criteria to be used.
4. Evaluate the feasibility of the criteria, including coverage of all concerns.
5. Create a decision matrix to be used.
6. Add weighting to each criterion.
7. Reach a consensus on the score of each criterion.
8. Compute the score based on each criterion and determine the best weighting.

The overall score is computed by a summation of scores in all criteria, as shown in Table 3. The notation in parentheses under the criteria numbers reflect the weights, as mentioned in Step 6 above in the MCDA technique.

Table 3: Template for Score Computation in the Multi-Criteria Decision Analysis Technique

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Criteria 1 (W1)** | **Criteria 2 (W2)** | **…** | **Criteria N (WN)** | **Score** |
| Solution 1 |  |  |  |  |  |
| Solution 2 |  |  |  |  |  |
| … |  |  |  |  |  |
| Solution N |  |  |  |  |  |

The criteria in Table 4 were used to evaluate the techniques in the case study in [Mead 2005]. 1 corresponds to *most suitable*; 4 corresponds to *least suitable*.

Table 4: Sample Evaluation of Risk Assessment Techniques

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Suitable for small  companies** | **Feasible to complete this quarter** | **Does not require additional data collection** | **Suitable for requirements** | **Average Score** |
| **Methodologies** | GAO | 2 | 4 | 2 | 2 | 2.50 |
| NIST | 2 | 2 | 1 | 1 | 1.50 |
| NSA/IAM | 3 | 3 | 2 | 2 | 2.50 |
| SAEM | 4 | 4 | 4 | 4 | 4.00 |
| V-Rate | 3 | 4 | 4 | 4 | 3.75 |
| Haimes | 2 | 2 | 2 | 2 | 2.00 |
| SSA | 2 | 2 | 2 | 4 | 2.50 |
| DDP/Feather | 3 | 4 | 2 | 4 | 3.25 |

As seen in Table 4, the NIST and Haimes RFRM models suited the case best.

|  |
| --- |
| **Task**  This is a task for the acquisition organization. Knowing some of the threats and vulnerabilities of both the acquisition and contractor organization, you will develop a set of risks evaluating likelihood, impact, and thereby priority of the risks. When assessing impact, it is important to consider cost and performance. For any given risk, you will determine the severity and probability of the risk, as well as the cost and likelihood of success of the mitigation. The exit criterion Isa documented list of risks that identifies all the vulnerabilities and threats, classified according to their likelihood and impact.  A set of sample risk statements is in Table 5 for reference.  Time: 15 minutes |

Table 5: Sample Risk Statements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **Consequence** | **Impact** | **Probability** | **Mitigation Plan** |
| The software system is using an outdated technological solution. | It could expose new security vulnerabilities in the system. | High | High | There should be thorough regression testing of the software before release. |
| Developers working on the product are new and lack the required expertise. | The development effort may increase, resulting in significant slippage of schedule and milestones. | High | High | Rigorous training should be introduced for the resources. Expert technical advisor should be associated with the team to provide timely guidance and assistance as required. |

## Step 5 – Select Elicitation Techniques

The goal of this step is to select an appropriate requirements elicitation technique that the acquisition organization can be use to elicit security requirements. The technique chosen should be consistent with the stakeholders of the project, the expertise and skill level within the team, and the scope of the project.

Following is a set of sample elicitation techniques the acquisition organization should consider in this step.

Elicitation Techniques to Consider

**Misuse Cases**Misuse cases are described in Step 3.

**Soft Systems Methodology (SSM)**   
SSM deals with problem situations in which there is a high social, political, and human activity component [Checkland 1990]. The SSM can deal with “soft problems” that are difficult to define, rather than “hard problems” that are more technology-oriented. Examples of soft problems include how to deal with homelessness, how to manage disaster planning, and how to improve Medicare. Eventually technology-oriented problems may emerge from these soft problems, but much more analysis is needed to reach that point.

The primary benefit of SSM is that it provides structure to soft problem situations and enables their resolution in an organized manner. In addition, it compels the developer to discover a solution that goes beyond technology.

**Quality Function Deployment (QFD)**   
QFD is “an overall concept that provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development and production” [QFD 2005]. The distinguishing attribute of QFD is the focus on customer needs throughout all product development activities. By using QFD, organizations can promote teamwork, prioritize action items, define clear objectives, and reduce development time [QFD 2005].

**Issue Based Information Systems (IBIS)**  
Developed by Horst Rittel, the IBIS method is based on the principle that the design process for complex problems, which Rittel calls “wicked” problems, is essentially an exchange among the stakeholders in which each stakeholder brings his or her personal expertise and perspective to the resolution of design issues [Kunz 1970]. Any problem, concern, or question can be an issue and may require discussion and resolution for the design to proceed.

**Joint Application Development (JAD)**   
The JAD methodology is specifically designed for the development of large computer systems [Wood 1995]. Its goal is to involve all stakeholders in the design phase of the product via highly structured and focused meetings. In the preliminary phases of JAD, the requirements engineering team is charged with fact-finding and information-gathering tasks. Typically, the outputs of this phase, as applied to security requirements elicitation, are security goals and artifacts. The actual JAD meetings are then used to validate the security goals and artifacts by establishing an agreed-on set of security requirements for the product.

**Feature-Oriented Domain Analysis (FODA)**   
FODA is a domain analysis and engineering method that focuses on developing reusable assets [Kang 1990]. By examining related software systems and the underlying theory of the class of systems they represent, domain analysis can provide a generic description of the requirements of that class of systems in the form of a domain model and a set of approaches for the implementation of the requirements.

**Accelerated Requirements Method (ARM)**   
The ARM process is a facilitated requirements elicitation and description activity [Hubbard 2000]. It includes three phases

1. preparation phase
2. facilitated session phase
3. deliverable closure phase

The ARM process is similar to JAD but has certain significant differences from the baseline JAD method; these differences contribute to its uniqueness. For example, in this process, the facilitators are content neutral, the group dynamic techniques used are different from those used in JAD, the brainstorming techniques used are different, and the requirements are recorded and organized using different conceptual models.

The MCDA technique used to identify a risk assessment technique, described in the *Selecting Risk Assessment Techniques* section, can also be applied here to select the most appropriate requirements elicitation technique.

|  |
| --- |
| **Task**  This is a task for the acquisition organization. Based on the available elicitation techniques, the team must choose one particular technique and document the rationale behind choosing this technique.  Time: 15 minutes |

## Step 6 – Elicit Security Requirements

This step is the heart of the SQUARE process. Based on the elicitation technique selected Step 5, the acquisition organization can now implement the technique and elicit the required security requirements. Care has to be taken to ensure that the requirements are clear and unambiguous. Also, care needs to be taken to ensure that the requirements are not documented as technical or business constraints. The requirements should speak about what the system intends to do and not how it is to be done.

|  |
| --- |
| **Task**  This is a task for the acquisition organization. Based on the elicitation technique chosen in Step 5, the team needs to come up with a list of security requirements. It is not necessary that the requirements are final. However, they need to be documented.  Sample requirements, along with placeholders for revising the sample requirements, are in   Table 6.  Time: 15 minutes |

Table 6: Sample Security Requirements

|  |  |
| --- | --- |
| **Requirement #** | **Requirement** |
| R01 | The system is required to have strong authentication measures in place at all system gateways/entrance points.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| R02 | The system must prevent unauthorized access to viewing or tampering with any employee-specific sensitive data. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| R03 | It is required that the designated security personnel be able to audit the status and usage of system resources (including security devices).  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| R04 | It is required that the system’s network communications be protected from unauthorized information gathering and/or eavesdropping by encryption and other reasonable techniques.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| R05 | The designated personnel must audit the status of the system resources and their usage on a regular basis.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| R06 | It is required that the software components be designed using software security best practices.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Step 7 – Categorize Requirements

The purpose of this step is to categorize the security requirements. For example, the categorization could be either essential and non-essential requirements or software-level and system-level requirements. Doing this classification makes prioritization much easier.

The template in Table 7 can be used to help the acquisition organization come up with the classification of the security requirements.

Table 7: Template for Classifying Security Requirements

|  |  |  |  |
| --- | --- | --- | --- |
|  | **System Level** | **Software Level** | **Architectural Constraint** |
| Essential |  |  |  |
| Non Essential |  |  |  |

System- and Software-Level Requirements

Distinguishing between system and software requirements is important for ensuring that the requirements are assigned to the appropriate team. The development team usually is the best for handling software requirements, while the system team will handle hardware and deployment aspects of the system requirements.

Architectural Constraints

An architectural constraint is a limitation set by the acquisition organization of what needs to be used in the system at an architecture level. There are two major types of constraints: business and technical.

**Business constraints** involve strategic decisions made from a business perspective about components to be used in the architecture.

**Technical constraints** define the company’s technical limitations. This usually involves existing systems that an architect must use in the construction of a new system.

|  |
| --- |
| **Task**  This is a task for the acquisition organization. Classify the requirements based on whether they are system- or software-level requirements using the template in Table 7. Then, categorize the requirements using the template in Table 8. The exit criteria for this task are the requirements classified into the given categories and identification of architectural constraints.  Time: 15 minutes |

Table 8: Template for Categorizing Security Requirements

| **Requirement #** | **Classification** |
| --- | --- |
| R01 |  |
| R02 |  |
| R03 |  |
| R04 |  |
| R05 |  |
| R06 |  |

## Step 8 – Prioritize Requirements

In a real-world organization, given the large number of security requirements present in an organization, it may be difficult to implement them all given the changing dynamics of the project, the scarcity of the resources, and the requirement to adhere to the schedule. Hence, prioritization of requirements might be essential, because implementing all of them might be infeasible.

Several structured prioritization techniques such as the Analytical Hierarchy Process, Triage, and Win-Win are quite effective and can be used to prioritize.

Defining a Prioritization Scale

Table 9 shows some possible definitions for a scale of high, medium, and low, which can be used as rankings for the prioritization.

Table 9: Prioritization Definitions

|  |  |
| --- | --- |
| **Priority** | **Definition** |
| High | This requirement is crucial to the success of the project. The project would be considered a failure if this requirement is not implemented. |
| Medium | This requirement is important but could be dropped for development of a high priority one. The project would be considered a total success if these requirements are implemented. |
| Low | This requirement is optional and should be implemented only if there is time left at the end of the project. |

|  |
| --- |
| **Task**  This task is for the acquisition organization to prioritize the requirements. The prioritization could be an unstructured, informal discussion or a structured one. The exit criterion is a set of prioritized security requirements.  Time: 15 minutes |

Table 10 classifies sample requirements according to their priority (high, medium, or low).

Table 10: Sample Security Requirements Classified by Priority

| **Requirement #** | **Description** | **Priority** |
| --- | --- | --- |
| R01 | The system is required to have strong authentication measures in place at all system gateways/entrance points. | High |
| R02 | The system must prevent unauthorized access to viewing or tampering with any employee-specific sensitive data. | High |
| R04 | It is required that the system’s network communications be protected from unauthorized information gathering and/or eavesdropping by encryption and other reasonable techniques. | Medium |
| R03 | It is required that the designated security personnel be able to audit the status and usage of system resources (including security devices). | Medium |
| R06 | It is required that the software components be designed using software security best practices. | Medium |
| R05 | The designated personnel must audit the status of the system resources and their usage on a regular basis. | Low |

## Step 9 – Inspect Requirements

This is the last step of the SQUARE process. In this step, all the requirements are carefully reviewed and changed if necessary. This is done to ensure that the requirements are accurate, verifiable, and in agreement with the security and business goals of the organization.

Conducting Formal Inspection

Formal inspection, such as the Fagan inspection [Fagan 1976], is a systematic way to find defects in documents with verification from other entities besides the author. As non-executable artifacts, quality requirements can be effectively controlled by a process of formal inspection. The inspection process presented is a simplified version of Fagan inspection, focused towards requirements inspection.

1. Assign roles.

* facilitator —leads the inspection and moderates the meeting
* reader—reads the requirements for all participants
* participants—follow the material that the reader reads and find defects in requirements
* recorder—records defects found during inspection
* author—answers any questions that participants may have

2. Prepare for inspection.

* The facilitator prepares for an inspection.
* Every member who will attend the inspection meeting reviews the requirements by themselves and generates an initial list of questions they have and/or defects they find.

3. Conduct the inspection meeting.

* The reader reads one requirement at a time.
* Participants state the defects they find or ask questions.
* The reader reviews each checklist item against the current requirement, and the participants agree or disagree about whether the checklist item is passed.
* The writer documents any defects found by the team.

4. Revise the requirements.

After the meeting, the author revises the requirements based on the list of defects generated during the inspection meeting.

Requirements Inspection Checklist

This is the inspection checklist from the Fagan inspection [Fagan 1976]:

1. Are the requirements written correctly and concisely?
2. Are all the requirements written at a consistent and appropriate level of detail?
3. Do the requirements address the concern of the client?
4. Is any necessary information missing?
5. Does any requirement duplicate or conflict with any other requirements?
6. Are all the requirements really requirements and not implementation details? (Remember that requirements concern what, not how.)

|  |
| --- |
| **Task**  This is a task for the acquisition organization. Use the Requirements Inspection Checklist to review the requirements. It is important to document the review comments using the sample review log in Table 11.  Time: 10 minutes |

Table 11: Sample Review Log

| **Step Number** | **Description of**  **Defect** | **Severity** | **Suggested Changes** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Step 10 – Review of Requirements by Acquisition and Contractor Organizations

In this step, the contractor organization comes back into the picture. After the acquisition team develops the initial set of requirements by using the SQUARE process (elaborated through Steps 1 to 9), the contractor organization does a traditional review of the requirements to ensure they are consistent with the security goals and map with the overall business goal that was identified in Step 1. Also, the contractor organization needs to ensure that the requirements are not too low-level and that there is some flexibility that the contracting team can exercise in order to further decompose the requirements.

|  |
| --- |
| **Task**  The acquisition organization and the contractor organization together need to review the requirements. The exit criterion for this task is a final set of reviewed and complete requirements.  Time: 15 minutes |

# References

[Butler 2002]   
Butler, S. A. & Fischbeck, P. “Multi-Attribute Risk Assessment.” *SREIS 2002, Second Symposium on Requirements Engineering for Information Security*. Raleigh, NC, October 16, 2002. Lafayette, IN: CERIAS, Purdue University, 2002.

[CERT/CC 2002]   
CERT Coordination Center. *Survivable Systems Analysis Method*. http://www.cert.org/archive/html/analysis-method.html (2002).

[Checkland 1990]   
Checkland, P. *Soft System Methodology in Action*. Toronto, Ontario, Canada: John Wiley & Sons, 1990.

[Cornford 2004]   
Cornford, Steven L.; Feather, Martin S.; & Hicks, Kenneth A. *DDP—A Tool for Life-Cycle Risk Management*. http://ddptool.jpl.nasa.gov/docs/f344d-slc.pdf (2004).

[Haimes 2004]   
Haimes, Yacov Y. *Risk Modeling, Assessment, and Management,* 2nd ed. Hoboken, NJ: John Wiley and Sons, Inc., 2004.

[Hubbard 2000]   
Hubbard, R.; Mead, N.; & Schroeder, C. “An Assessment of the Relative Efficiency of a Facilitator-Driven Requirements Collection Process with Respect to the Conventional Interview Method.” *International Conference on Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, June 2000.

[IEEE 2005]  
IEEE. *Guide to the Software Engineering Body of Knowledge*. http://www.swebok.org/ (2005).

[IEEE 1990]  
IEEE. *IEEE Standard Glossary of Software Engineering Terminology*

(IEEE Std 610.12-1990). Los Alamitos, CA: IEEE Computer Society Press, 1990.

[Kang 1990]   
Kang, K. C.; Cohen, S. G.; Hess, J. A.; Novack, W. E.; & Peterson, A.S. *Feature-Oriented Domain Analysis Feasibility Study* (CMU/SEI-90-TR-021, ADA235785). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990. http://www.sei.cmu.edu/library/abstracts/reports/90tr021.cfm

[Kunz 1970]   
Kunz, W. & Rittel, H. *Issues as Elements of Information Systems, Working Paper 131*. Berkeley: Institute of Urban & Regional Development, University of California. http://www-iurd.ced.berkeley.edu/pub/WP-131.pdf

[Lipson 2001]   
Lipson, Howard F.; Mead, Nancy R.; & Moore, Andrew P. *A Risk-Management Approach to the Design of Survivable COTS-Based Systems*, 2001. http://www.cert.org/research/isw/isw2001/papers/Lipson-29-08-a.pdf

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

[NSA 2004]   
National Security Agency. *INFOSEC Assessment Methodology*. http://www.iatrp.com/iam.cfm (2004).

[QFD 2005]   
QFD Institute. *Frequently Asked Questions About QFD* (2005). http://www.qfdi.org/what\_is\_qfd/faqs\_about\_qfd.htm

[Stoneburner 2002]   
Stoneburner, Gary; Goguen, Alice; & Feringa, Alexis. *Risk Management Guide for Information Technology Systems* (Special Publication 800-30). Gaithersburg, MD: National Institute of Standards and Technology, 2002. http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf

[USGA 1999]   
U.S. General Accounting Office. “Information Security Risk Assessment: Practices of Leading Organizations, A Supplement to GAO’s May 1998 Executive Guide on Information Security Management.” Washington, D.C.: U.S. General Accounting Office, 1999.

[Wikipedia 2010]

Wikipedia: The Free Encyclopedia. http://www.wikipedia.org/ (2010).

[Wood 1995]   
Wood, J. & Silver, D. *Joint Application Development, 2nd ed.*, New York: Wiley, 1995