

SQUARE Workshop Guide: Client Team

**October 2009**

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.](mailto:permission@sei.cmu.edu)

**Table of Contents**

[Workshop Overview 2](#_bookmark0)

[Purpose 2](#_bookmark1)

[The Ideal Company 2](#_bookmark2)

[Possessions Organization Program (POP) 2](#_bookmark3)

[Security Goals 3](#_bookmark4)

[Case Introduction 4](#_bookmark5)

[The SQUARE Process 6](#_bookmark6)

[Step 1 – Agree on Definitions 6](#_bookmark7)

[Step 2 – Identify Assets and Security Goals 9](#_bookmark8)

[Step 3 – Develop Artifacts 11](#_bookmark9)

[Step 4 – Perform Risk Assessment 19](#_bookmark10)

[Performing Risk Elicitation Using RFRM 20](#_bookmark11)

[Step 5 – Select Elicitation Techniques 23](#_bookmark12)

[Step 6 – Elicit Security Requirements 27](#_bookmark13)

[Step 7 – Categorize Requirements 29](#_bookmark14)

[Step 8 – Prioritize Requirements 31](#_bookmark15)

[Step 9 – Inspect Requirements 32](#_bookmark16)

[Supplemental Material 34](#_bookmark17)

[Conducting Brainstorming 34](#_bookmark18)

[References 36](#_bookmark19)

CERT | SOFTWARE ENGINEERING INSTITUTE | i

# Workshop Overview

## Purpose

The intention of this workshop is to instill in you a greater understanding of the SQUARE (Security QUAlity Requirements Engineering) process. This will be done by walking you through a sample scenario based on previously carried out case studies. The group will be split into two teams:

Client team: One team (you) will act as a client. This document describes your company

and the situation for which you are using SQUARE.

 Requirements engineering team: The other team will be the requirements engineering team. They have a different document to guide them through the scenario. In several places in this document, they are referred to as the RE team.

To reduce the complexity of the evaluation, a set of canned answers has been provided for the beginning of each step. It is assumed that both teams have seen an overview about the SQUARE process and are thus at least somewhat familiar with it.

Information common to you and the RE team is shown below in regular type. Information specific to your team is shown in *italic*.

## The Ideal Company

The Ideal Company is a private company headquartered in Pennsylvania, with a staff of approximately 1,000 employees spread across multiple offices in the United States. It provides technical and management services to various public sectors and a number of diversified private companies.

## Possessions Organization Program (POP)

PLU (Possessions Listing Unlimited) Services is one of the four major subsidiaries of The Ideal Company. PLU provides a range of specialized services related to managing assets. One of its software products is the Possessions Organization Program (POP). POP is a tool for helping companies to make strategic allocations and plans for their critical IT assets. It is an Executive Asset Management Information System that provides decision support capabilities via customized views. These views are displayed in graphical forms and consist of information such as asset information, operational performance, and other user-defined metrics.

POP also integrates with many third-party software suites to provide enterprise-level services and features. BILDLODE/AM, which is used internally, is a facility infrastructure management and operation tool that supports all aspects of infrastructure management. It is also fully integrated with AutoCAD, an industry standard software application that ensures proper change management. All changes made on architectural drawings are immediately reflected in the

database. Another integrated tool is a backend Geographical Information System (GIS) used to organize information and geographic locations by site.

Overall, the POP Software Suite is a full-service support product in all aspects of infrastructure management and facility-related services.

*The company is planning on a major expansion of the system, moving it from a local implementation to a distributed one. As a company, Ideal is uncertain as to what risks this might entail. They are asking for help in identifying security requirements in part to avoid drawing attention to the potential weakness of this new expansion.*

## Security Goals

These goals are to aid The Ideal Company in a high-level assessment of its existing security standards and policies regarding POP.

*You, as a client, think that these are the security goals necessary for the project to be successful. Please review and remember these security goals when you are prioritizing the requirements with the Requirements Team.*

**Confidentiality**

The security goal of confidentiality will help to ensure that information and resources are accessed only by those who are authorized to use them. Confidentiality is closely related to access control in that access control is a component of confidentiality. The protection against unauthorized disclosures will guard against malicious coding, hackers and crackers, and accidental disclosures. Confidentiality involves policy and procedures, as well as the implementation of security controls.

**Availability**

This goal is to ensure that the project is functional and available at all times. This includes the core facility management services, Sybase databases, etc. The project client should assess how the system is used and determine disaster recovery needs. All data should be backed up to tape or disk daily.

Clients of a POP application classified as critical should determine what other disaster recovery procedures (e.g., hardware redundancy, data mirroring, remote data mirroring, remote disaster recovery facility) are necessary.

**Data Integrity**

The integrity of data is absolutely critical in the POP system. If the underlying information upon which facility managers must make their decisions is corrupted or wrong, the purpose of the system has been defeated. Therefore, data backups and checksum integrity verification are of the utmost importance.

**Monitoring**

One security goal is to preserve or enhance the ability to accurately record the activities that take place. When users interact with the system, a complete accounting of all the commands issued and the internal transactions of the system should be available. In order for this to occur, logging capabilities that are currently in place for the Asset Management System are needed.

**Code Review**

Periodic code review should be performed to ensure script and code confidentiality and integrity. This process verifies that malicious scripts or code have not been inserted into the source code of the software suite. The potential risks of not performing this periodic review are the failure of the system and the potential loss of all data.

**Access Control**

Access control helps ensure that only authorized users of the POP system have access to their specified and authorized resources. The team would like information on what authentication procedures are in place and whether the system will allow for remote connectivity. If this holds true, information regarding intact authentication controls and penetration testing ensure the access controls are working properly.

**Maintaining Mission-Critical Services**

The most important goal is the ability to deliver essential services in the face of attack, failure, or accident. This is dependent upon maintaining necessary system properties in unfavorable environments. First, identify the critical services that must be delivered and the resources that support those services. Next, the consultant team (in this case, the SQUARE team) will proceed to implement controls and defensive measures for protection.

**Disaster Recovery**

The POP client should have procedures in place to address disaster recovery plans for different levels of applications and degrees of disaster. The plans should address how and when key personnel are contacted, along with their duties and responsibilities. These plans should be tested, and the results and lessons learned should be documented in a central location for easy reference.

## Case Introduction

The Ideal Company has experienced situations in the past in which sensitive client data was accidentally transmitted to an unauthorized user or other access violations have occurred. While developing POP, Ideal realized that they had no process in place to respond to this kind of security situation and had no historical or statistical data. Therefore, they decided that they should have a formal security requirements engineering process. They had recently heard of SQUARE being effective in a similar situation with a company in their field, so they decided to try it.

*Ideal is also interested in knowing more about the SQUARE process, how the steps specifically work, so that they can employ it locally. The client team should be willing to ask questions if they do not understand the reasoning for a piece of data. Such requests might be forwarded on to the moderator, since in this case the RE team is not experienced, but the client team should treat the RE team as if they were experienced*.

# The SQUARE Process

## Step 1 – Agree on Definitions

In this step, definitions of terms that will be used have to be agreed on by the stakeholders and the requirements engineering team. Some suggested terms and definitions are shown below. You may want to identify other terms that should be defined. Participants can proceed with this in two parts. First, focus on generating the list of words, and then select the group of words that will be used. For the initial list, you can use the brainstorming technique to quickly generate a large number of words. (Guidelines for conducting a brainstorming session are included in the supplemental material on page 39.)

**Task**

You, as a client, think that these are the accurate definitions of some terms. Discuss these with the RE team. Feel free to change them if you like, but both you and the RE team must agree on and record the same definitions.

Time: 15 minutes

**Definitions**

**Access control—**An **access control system** is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system [Wikipedia 2009a]. Access control ensures that resources are granted only to those users who are entitled to them [SANS 2009].

**Antivirus software**—A program that searches hard drives and floppy disks for any known or potential viruses [SSQ 2009a].

**Asset—**A mission-critical object or entity that a company owns and wants to secure.

**Attack**—An attack is defined as an action wherein a piece of software, a chunk of data, or sequence of commands is used to take advantage of a bug, glitch, or vulnerability in order to get unintended or unanticipated behavior out of computer software, hardware, or something electronic (usually computerized) [Wikipedia 2009b]. It is conducted by an adversary, the attacker, on a potential victim, and is a set of events that an observer believes to have information assurance consequences on some entity, the target of the attack [Ellison 2003].

**Auditing—**The information gathering and analysis of assets to ensure such things as policy compliance and security from vulnerabilities [SANS 2009].

**Authentication—**The process of attempting to verify the digital identity of the sender of a communication, such as a request to log in.

The process of determining whether someone or something is, in fact, who or what it is declared to be [SSQ 2009b].

**Confidentiality—**Ensuring that information is accessible only to those authorized to have access [ISO/IEC 2005].

The property that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity) [SANS 2009].

**Firewall—**A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both [Webopedia 2009].

**Integrity—**For systems, the condition of a system wherein its mandated operational and technical parameters are within the prescribed limits. For data, the property that the data is ―whole‖ or complete [Wikipedia 2009c].

For systems, the quality that a system has when it can perform its intended function in a unimpaired manner, free from deliberate or inadvertent unauthorized manipulation. For data, the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner [Allen 1999].

**Vulnerability—**A condition or weakness in (or absence of) security procedures, technical controls, physical controls, or other controls that could be exploited by a threat [Guttman 1995].

As a point of information, the following were the suggested terms from the initial SQUARE case study [Mead 2005, Table 24].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| access control | access control list | antivirus soft- ware | artifact | asset |
| attack | auditing | authentication | availability | back door |
| breach | brute force | buffer overflow | cache cram- ming | cache poisoning |
| confidentiality | control | corruption | cracker | DoS attack |
| disaster recov- ery plan | disclosure | disgruntled em- ployee | downtime | disruption |
| encryption | espionage | essential servic- es | exposure | fabrication |
| fault line attacks | fault tolerance | firewall | hacker | honey pot |
| HTTP header manipulation | Impact | incident | incident han- dling | integrity |
| interception | interruption | intrusion | intrusion detec- tion system | intrusion pre- vention system |
| liability | luring attack | malware | man-in-the- middle attack | masquerade |
| modification | non-essential services | non-repudiation | patch | penetration |
| penetration testing | physical securi- ty | 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 assess- ment | threat model | toolkits | Trojan |
| trust | uptime | victim | virus | vulnerability |
| worm |  |  |  |  |

## Step 2 – Identify Assets and Security Goals

The purpose of this step is to agree on a set of assets to protect and prioritized security goals for the project. This will help later in identifying the priority and relevance of the security risks.

Ideally, one business goal must be identified, and then the assets and five or six security goals that map to the business goal must be identified. Remember, security goals are not requirements!

**Brainstorming Security Goals**

The brainstorming technique used in Step 1 can be repeated in Step 2 in a different context. In this step, the focus is on eliciting security goals. The problem addressed using the brainstorming technique in this case might be ―What are the important security goals of the system?‖ Conduct the brainstorm as usual and ensure a list of security goals are produced as an output.

As multiple stakeholders are involved in the process, it is likely that they will have conflicting priorities for goals. Each goal can be elicited without any judgment of its relative priority. The non-essential goals will be pushed down or eliminated using the prioritization technique described below.

**Prioritizing Security Goals Using Multivoting**

The output of the brainstorming session is usually a non-prioritized list of ideas. Often, however, there are not sufficient resources to implement every idea generated. A technique called multivoting can be used to quickly find the common important goals among the stakeholders.

***Multivoting***

The multivoting technique can be used to reduce a large number of solutions to a smaller, manageable quantity. The idea is to have multiple iterations of votes, which will reduce the number of candidates in every iteration [<http://www.ca.uky.edu/agpsd/multivot.pdf>].

These are the steps in multivoting:

1. Write down the brainstormed security goals in a numbered list.
2. Reach consensus on the number of goals that you would like to have in the end.
3. Let all members choose the top goals and write the number of goals down. The number of top solutions should be approximately one-third of the total number of goals.
4. After all members have written down their choices, count the votes for each goal.
5. Eliminate goals with the fewest votes.
6. Repeat steps 3 and 4 until the agreed upon number of goals is reached.
7. If there is no clear favorite, have the team discuss which goals should be considered or have one final vote.

***Note***

While multivoting is a relatively quick decision-making process, its main drawback is that not all participants will agree with the final outcome. The facilitator should be aware of any major conflict from voting and may need to suggest additional time to resolve it [DAU 2002].

**Task**

You have already provided the requirements engineering team with your business objectives and functional requirements, as a result of which both you and the RE team have agreed on a common business goal. However, several security goals have been identified. Your task now is to prioritize these security goals and discuss them with the RE team.

The security goals in the following list have already been prioritized in a suggested order. You may change the priorities if you like after discussing them with the SQUARE RE team.

Time: 15 minutes

**Business Goal**

*“This tool … provides the means to make informative decisions based on available sources.“*

**Security Goals**

Prioritize 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 POP’s data shall be maintained. | 2 |
| The POP system shall be available for use when needed. | 3 |
| System performance should not be significant- ly impacted by the new security measures. | 4 |
| New security measures should not hamper ease of use of the system. | 5 |

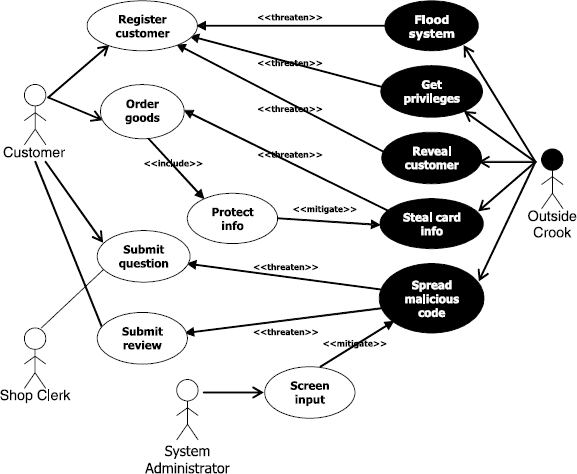
## Step 3 – Develop Artifacts

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.

**Misuse Cases**

Misuse cases are derived from the use case technique, which is a modular way to describe a way in which a complex system is employed by its users [Armour 2000].

With slight modifications, use cases can be used to aid in the identification of security requirements. Misuse cases specify behavior that is *not* wanted in the proposed system. Many security breaches can be described in a stepwise fashion that resembles ordinary use cases. Misuse cases can also be associated with a use case, indicated with a ―threaten‖ relationship; that is, the misuse case represents a threat to normal usage. An example this kind of relationship is shown below [Sindre 2000].



**Notations**

**Misuser**—―An actor that initiates misuse cases, either intentionally or inadvertently.‖

**Misuse case**—A sequence of actions and their variants that a system can perform by interacting with misusers, resulting in harm or losses to some stakeholder.

**Eliciting a Misuse Case**

Similar to a use case, a misuse case can be captured in a form with information in fields such as described below [Chen 2004]. Note that there are many misuse case templates—this is just one example.

|  |  |
| --- | --- |
| **Field** | **Description** |
| Number | Misuse case number. Each misuse case represents a single threat/vulnerability. |
| Name | The name of a particular threat or vulnerability. Threats may derive from user or system component interaction. |
| Scope | System vulnerability concern(s) |
| Priority | Relative importance of the misuse case (low, medium, or high) |
| Deployment Envi- ronment | Network environment where the affected system/component is dep- loyed  Intranet  Extranet/Internet |
| Mis-actors | Attacker type |
| Access Right Levels | The access right levels needed to perform the misuse case |
| Point of Entry | The point of entry from which the mis-actor can initiate the misuse case |
| Security Attributes Affected | What aspects of security are compromised (confidentiality, integrity, or availability) |
| Description | A detailed description of the steps to perform the misuse case |
| Sophistication | The relative complexity of performing the misuse case (low, medium, or high) |
| Pre-conditions | All conditions that must be satisfied before the misuse case can be performed |
| Assumptions | Existing system and network configuration settings (e.g., existing user access control list) |
| Post-conditions | The outcome of a successful execution of the misuse case, along with the measures desired to prevent, detect, and recover from the misuse case. There are four sections for post-conditions:  **Worst Case Threat** – The most undesired outcome from successful execution of the misuse case  **Wanted Prevention Guarantee** – The measures to put in place to try to prevent the misuse case from being successfully executed **Wanted Detection Guarantee** – The measures to put in place to detect the occurrence of the misuse case  **Wanted Recovery Guarantee** – The measures to put in place to be  able to recover if the misuse case is executed |
| Potential Mis-actor Profiles | Attacker profile (e.g., attacker characteristics) |
| Stakeholders and Threats | Stakeholders who will be affected if the misuse case is successfully executed, and the potential loss to each stakeholder |
| Related Use Cases | The use cases that this misuse case directly threatens |

**Task**

Your task is to generate as many system artifacts as possible and give them to the RE team. Since it may not be feasible to come up with artifacts in this workshop, two use cases and four misuse cases have been provided for you. You have already created the use cases and you now have to discuss and finalize them with the RE team. The RE team has, however, come up with certain misuse cases, which need to be completed after discussions with you. Be sure to document all the artifacts at the end of this step. **Incomplete sections are specified in bold.**

Please fill them. Time: 25 minutes

***Use Cases***

|  |  |
| --- | --- |
| Number | UC-01 |
| Use Case | Damage Assessment |
| Description | The Medium-Level Possessions Organization Program user wants to make changes to the floor plan to indicate damaged areas in the facility. |
| Actors | Medium-Level User, High-Level User, System Administrator |
| Assumptions | The user has proper edit privileges. The data entered is correct.  The user has proper security privileges. |
| Steps | 1. Select Operations Management. 2. Select Building. 3. Select Floor Plans. 4. Select Area Status to view the current condition. 5. Highlight the specific area for damage assessment. 6. From the drop-down menu, select the status you wish to assign to the room (Damaged, Destroyed, Inventory, Not Usable, Renovation, Construction). 7. Press Go. 8. To continue marking areas, select “Floor Plan” and choose another floor.   Repeat steps 4-7. |
| Variations | NA |
| Non-Functional | NA |
| Related Misuse Cas- es | MC-01,MC-03, MC-04 |

|  |  |
| --- | --- |
| Number | UC-02 |
| Use Case | BILDLODE Administration Adding a User and Assigning Privileges |
| Description | The BILDLODE Administrator adds a user to the users table so that the user will have the ability to use the Possessions Organization Program. The user must also assign the proper privileges associated with their user-level. |
| Actors | BILDLODE Administrator |
| Assumptions | The BILDLODE Admin has the proper security privileges. |
| Steps | Add Individual   1. Open BILDLODE. 2. Select the project (in this case, it is Possessions Organization Program but varies according to client). 3. Navigate to System Management. 4. Select Security. 5. Click the Secure Padlock. 6. Select Users. 7. Open a new record. 8. Enter the username (must match the login name). 9. Select the user-level (Review, Edit…). 10. Assign groups.   Add Group  Open BILDLODE.   1. Select Security Groups. 2. Add new record. 3. Add group name. 4. Add description. |
| Variations | Go directly to the data through BILDLODE |
| Non-Functional | No user password |
| Related Misuse Cas- es | MC-01, MC-02,MC-03, MC-04 |

***Misuse Cases***

|  |  |  |
| --- | --- | --- |
| Number | MC-01 | |
| Name | Unauthorized logon on the server | |
| Scope | User Authorization Concerns | |
| **Priority** | Low Medium High | |
| **Deployment Envi- ronment** | Intranet  Extranet/Internet | |
| Mis-actors | Unauthorized users | |
| Access Right Levels | \_X\_ Low-Level System Users  \_X\_ Medium-Level System Users  \_X\_ High-Level System Users  \_X\_ Sys Admin  \_X\_ Other Network User | |
| Point of Entry | Network \_X\_ Host Application | |
| **Security Attributes Affected** | Confidentiality  Integrity  Availability | |
| Description | An unauthorized user attempts to log on to the server and succeeds. | |
| Sophistication | \_X\_ Low  Medium  High | |
| **Pre-conditions** | Access control lists (ACL) are configured properly in a domain based network.  The unauthorized user has unintended logon rights to the server.  The server resides on an (internet/intranet?) network. | |
| Assumptions | The user does not have permission to log on to the server. | |
| Post-conditions | Worst Case Threat | The unauthorized user logs onto the server ma- chine. His/her actions are never caught. |
| Wanted Preven- tion Guarantee | Enforce machine ACL security policy (role-based user authentication). |
| Wanted Detec- tion Guarantee | Logon attempts are logged and viewed by system administrators. |
| Wanted Recovery Guarantee | Remove users’ unauthorized logon rights on the server. |
| Potential Mis-actor Profiles | Medium to highly skilled, potentially host administrators with medium criminal intent | |
| Stakeholders and Threats | POP Client Company: loss of data integrity and/or confidentiality Ideal Co.: loss of reputation, loss of current and potential clients | |
| Related Use Cases | UC-01, UC-02 | |

|  |  |  |
| --- | --- | --- |
| Number | MC-02 | |
| Name | Sys admin gains access to system data | |
| Scope | User Authorization Concerns | |
| **Priority** | Low Medium High | |
| **Deployment Environment** | Intranet  Extranet/Internet | |
| Mis-actors | Sys Admin | |
| Access Right Levels | Low-Level System Users  Medium-Level System Users  High-Level System Users  \_X\_ Sys Admin  Other Network User | |
| Point of Entry | Network Host \_X\_ Application | |
| **Security Attributes Affected** | Confidentiality  Integrity  Availability | |
| Description | A sys admin attempts to gain access data on the Windows 2003 server and succeeds. | |
| Sophistication | Low  \_X\_ Medium  High | |
| Pre-conditions | The sys admin has logon rights to the server or he/she has the creden- tials to access the database. | |
| Assumptions | The sys admin does not have permission to access data on the server. | |
| Post-conditions | Worst Case Threat | The sys admin sees and/or tampers with the sys- tem data. His/her actions are never caught. |
| Wanted Preven- tion Guarantee | Enforce machine ACL security policy. Separate credentials for system administration and applica- tion access (role-based user authentication). |
| Wanted Detec- tion Guarantee | Logon attempts are logged, application usage is logged, and database accesses are logged. Audit information is cross reviewed by a group of sys  admins and managers. |
| Wanted Recovery Guarantee | Restore data from backup if data is tampered with. |
| Potential Mis-actor Profiles | Highly skilled system administrators who understand how the system works and know about backdoors (if any exist) | |
| Stakeholders and Threats | POP Client Company: loss of data integrity and/or confidentiality Ideal Co.: loss of reputation, loss of current and potential clients | |
| Related Use Cases | UC-01, UC-02 | |

|  |  |  |
| --- | --- | --- |
| Number | MC-03 | |
| Name | Malicious users install malicious programs that can tap into Excel’s memory to steal exported data. | |
| Scope | Data confidentiality concerns | |
| **Priority** | Low Medium High | |
| **Deployment Environment** | Intranet  Extranet/Internet | |
| Mis-actors | Users, sys admins | |
| Access Right Levels | \_X\_ Low-Level System Users  \_X\_ Medium-Level System Users  \_X\_ High-Level System Users  \_X\_ Sys Admin  \_X\_ Other Network User | |
| Point of Entry | Network Host \_X\_ Application | |
| **Security Attributes Affected** | Confidentiality  Integrity  Availability | |
| Description | Malicious users install malicious programs that can tap into Excel’s memory to steal exported data. Because Excel and Microsoft Office overall use shared memory, the shared memory can be tapped by  programs or malicious scripts. | |
| Sophistication | Low  Medium  \_X\_ High | |
| Pre-conditions | The malicious user gained access to the victim’s machine at some point and installed malicious programs. | |
| Assumptions | NA | |
| Post-conditions | Worst Case Threat | The malicious user steals information in the ex- ported Excel memory. His/her actions are never caught. |
| Wanted Preven- tion Guarantee | Enforce security policies for workstation system access. |
| Wanted Detec- tion Guarantee | Monitor system resource usage for unusual pro- grams. |
| Wanted Recov- ery Guarantee | Limit user privileges regarding installation of any programs. |
| Potential Mis-actor Profiles | Highly skilled  users with high criminal intent. |  |
| Stakeholders and Threats | POP Client Company: loss of data integrity and/or confidentiality Ideal Co.: loss of reputation, loss of current and potential clients | |
| Related Use Cases | UC-01, UC-02 | |

|  |  |  |
| --- | --- | --- |
| Number | MC-04 | |
| Name | Input Validation Attack | |
| Scope | System integrity concerns | |
| **Priority** | Low Medium High | |
| **Deployment Envi- ronment** | Intranet  Extranet/Internet | |
| Mis-actors | Users, sys admins | |
| Access Right Levels | \_X\_ Low-Level System Users  \_X\_ Medium-Level System Users  \_X\_ High-Level System Users  \_X\_ Sys Admin  \_X\_ Other Network User | |
| Point of Entry | Network Host \_X\_ Application | |
| **Security Attributes Affected** | Confidentiality  Integrity  Availability | |
| Description | A user uses buffer overflow attacks or SQL injection attacks to gain unauthorized access to the system. | |
| Sophistication | Low  Medium  \_X\_ High | |
| Pre-conditions | The mis-actor has network access to the Possessions Organization Program. | |
| Assumptions | NA | |
| Post-conditions | Worst Case Threat | User gains unauthorized access to sensitive sys- tem data. His/her actions are never caught. |
| Wanted Preven- tion Guarantee | Perform thorough input validation. Hide HTML source code. |
| Wanted Detec- tion Guarantee | Audit information must be reviewed routinely (monthly). |
| Wanted Recovery Guarantee | NA |
| Potential Mis-actor Profiles | Highly skilled users or sys administrators with high criminal intent | |
| Stakeholders and Threats | POP Client Company: loss of data integrity and/or confidentiality Ideal Co.: loss of reputation, loss of current and potential clients | |
| Related Use Cases | UC-01, UC-02 | |

## 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.*

An abbreviated version of this step is used for this seminar. First, a process for selecting a risk assessment method is explained, and then a method is given to you. Next, you will perform some (but not all) of the steps of the risk assessment method.

Time : 20 minutes

**Selecting Risk Assessment Techniques**

As seen in the overview, 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]

An actual selection of risk assessment techniques is outside of the scope of this seminar, but multi-criteria decision analysis is briefly described as an example of a tool to help in choosing a technique.

***Evaluating Risk Assessment Techniques Using MCDA***

Multi-criteria decision analysis (MCDA) is a process for supporting decision making when there are many evaluation criteria [Wikipedia 2009d]. The overall score is computed by a summation of scores in all criteria.

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

These are the steps of the MCDA 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 [Linkov 2004].

The following criteria were used to evaluate the techniques in the case study in [Mead 2005].

**1 corresponds to *most suitable*; 4 corresponds to *least suitable*.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Methodologies** |  | **Suitable for small compa- nies** | **Feasible to complete this seme- ster** | **Does not require additional data collection** | **Suitable for requirements** | **Aver- age Score** |
| 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 can be seen above, the NIST and Haimes RFRM models suited the case best.

## Performing Risk Elicitation Using RFRM

Risk Filtering, Ranking and Management (RFRM) is a methodology for total risk management, which includes identifying, prioritizing, and managing risks. There are two main structural components [Haimes 2004]:

**Head topics**—Major visions, concepts, and criteria of success

**Subtopics**—Detailed classification of requirements for success scenarios and sources of risks for risk scenarios

Steps 2 and 3 of the RFRM model involve threat and vulnerability identification. *Break for 10 to 15 minutes to discuss known threats and vulnerabilities with the RE team, both information shown in the existing misuse case data and the anecdotal incidents listed below.* Please do not improvise anything here. Keep to the material shown below and the information from the listed misuse cases.

Currently, The Ideal Company does not have any historical or statistical data regarding security events. While individual employees remember there being an incident or two in the past with other pieces of software, they were dealt with as they came up and were then largely forgotten.

***Incident 1 (Accounting Wiz software)***

A disgruntled employee, about to be fired for incompetency, set up a database query that switched around numbers in records in the system. Fortunately, another employee happened to overhear him talking about it to his wife and the changes were caught before they became a permanent part of the backups. The disgruntled employee did not directly circumvent security measures; as a developer, he had full access.

***Incident 2 (Asset Data Storage software)***

A localized power outage occurred. During the time when the power went down, a systems operator noticed that when the emergency backup generators kicked in, the authentication server powered up ten minutes after the main database of the ADS system. During that time it was possible for an attacker to access data without needing authentication. The system operator noted this vulnerability in his weekly report as an oddity, but no formal action was taken.

***General concerns***

 Insider attacks—The people of Ideal are concerned about the ability of employees to change data, possibly in malicious ways.

 Social engineering—Recently, there's been a large turnover in administrative staff. While the new staff is undergoing training, they are still relatively inexperienced, and there are fears that an attacker might gain access through these people, such as pretending to be a user who has lost their password.

 Hardware damage or subversion—The technical staff are aware that they are dependent on the hardware systems to provide authentication. There are fears about what might happen if hardware is damaged by natural disasters or sabotaged by attackers. Furthermore, the hard- ware currently being used is common on the market, so there is a fear that an attacker could purchase near copies of the hardware and practice the breaking of it in private.

*As the client, you are aware of some security holes, but you have only a partial understanding of how these holes could be exploited. You know that there is a danger in the failure of the authentication server, but you're not entirely sure how it could be exploited and how bad it could be.*

***Risk Evaluation and Prioritization***

*Now that you know some of the threats and vulnerabilities of the company, you will work with the other team at evaluating likelihood, impact, and thereby priority of the risks. When assessing impact, it is important to consider cost and performance.*

*You will be concerned about cost and performance. You do not have unlimited funds for security and you need the system to be accessible enough for remote employees to be able to do their*

*work. For any given risk, you will want to know the severity and probability of the risk, as well as the cost and likelihood of success of the mitigation.*

## Step 5 – Select Elicitation Techniques

For the purpose of this seminar, you will not actually go into the details of determining an elicitation technique. However, in determining an elicitation technique, you will need to consider the suitability of the technique for both the client and the RE team.

**Criteria to Consider**

You can use MCDA to select an elicitation technique in the same manner as you selected a risk assessment method in Step 4. The following is a non-exhaustive but representative list of criteria to consider when evaluating elicitation techniques:

### Adaptability to security requirements

The ability of the technique to produce accurate requirements in diverse environments. For example, does the technique apply only to functional requirements?

 **CASE tool**

Does the technique have a software tool to complement the process?

### Client acceptance

The likelihood that the client would agree to the elicitation technique in analyzing their re- quirements. Is the process too invasive in a business environment?

 **Complexity**

The degree of difficulty in understanding and properly executing the elicitation technique. Can the requirements engineers and stakeholders easily perform the technique correctly once they learn the process?

### Graphical output

The ability of the elicitation technique to produce readily understandable visual artifacts that appeal to the stakeholders.

### Implementation duration

The length of time the requirements engineers and clients need to fully execute the elicitation technique.

### Learning curve

The speed with which the requirements engineers and clients can fully comprehend the elici- tation technique.

 **Maturity**

The time, exposure, and analysis the elicitation technique has experienced in its vetting by the requirements engineering community.

 **Scalability**

The ability of the elicitation technique to address the requirements of enterprise-level sys- tems, in addition to smaller applications.

**Elicitation Techniques to Consider**

Elicitation techniques that could be considered in this step are described below.

### 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].

**Controlled Requirements Expression (CORE)**

CORE is a requirements analysis and specification method that clarifies the user’s view of the services to be supplied by the proposed system. In CORE, the requirements specification is created by both the user and the developer—not solely one or the other. The problem to be analyzed is defined and broken down into user and developer viewpoints. Information about the combined set of viewpoints is then analyzed. The last step of CORE deals with constraints

analysis, such as the limitations imposed by the system’s operational environment, in conjunction with some degree of performance and reliability investigation.

**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 terms *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 [Wood 1995] is specifically designed for the development of large computer systems. 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 session is then used to validate this information 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 their implementation.

The FODA method was founded on two modeling concepts: abstraction and refinement [Kean 1997]. Abstraction is used to create domain models from the specific applications in the domain. Specific applications in the domain are then developed as refinements of the domain models. The example domain used in the initial report on FODA [Kang 1990] is window management systems. The window management examples of that time are no longer in use, but include VMS, Sun, and Macintosh, among others.

**Critical Discourse Analysis (CDA)**

CDA uses sociolinguistic methods to analyze verbal and written discourse [Schiffrin 1994]. In particular, this technique can be used to analyze requirements elicitation interviews and to understand the narratives and ―stories‖ that emerge during those interviews.

**Accelerated Requirements Method (ARM)**

The ARM process [Hubbard 2000] is a facilitated requirements elicitation and description activity. 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, which 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.

**MCDA Results from Case Study**

Many of these evaluation criteria are in common with the evaluation criteria for the risk assessment techniques listed earlier. In the case study on which this seminar is based, the following assessment was made:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Misuse Cases** | **SSM** | **QFD** | **CORE** | **IBIS** | **JAD** | **FODA** | **CDA** | **ARM** |
| Adaptability | 3 | 1 | 3 | 2 | 2 | 3 | 2 | 1 | 2 |
| CASE Tool | 1 | 2 | 1 | 1 | 3 | 2 | 1 | 1 | 1 |
| Client Accep- tance | 2 | 2 | 2 | 2 | 3 | 2 | 1 | 3 | 3 |
| Complexity | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 2 |
| Graphical Out- put | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 3 |
| Implementation Duration | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 3 |
| Learning Curve | 3 | 1 | 2 | 1 | 3 | 2 | 1 | 1 | 1 |
| Maturity | 2 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Misuse Cases** | **SSM** | **QFD** | **CORE** | **IBIS** | **JAD** | **FODA** | **CDA** | **ARM** |
| Scalability | 1 | 3 | 3 | 3 | 2 | 3 | 2 | 1 | 2 |

Scale: 3 = very good, 2 = fair, 1 = poor

Ultimately, the ARM technique was chosen for evaluation and will be used in the next step, elicitation of requirements.

## Step 6 – Elicit Security Requirements

This step is the heart of the SQUARE process. In this step, the requirements engineering team executes the method that was chosen in the last step.

**Using ARM to Elicit Requirements**

In this step, ARM’s brainstorming technique is used to generate a large quantity of security requirements.

The brainstorm session can be conducted in a similar manner as described in Step 1. The focus question for this brainstorm session would be to fill in the blank of the following sentence: ―An important security requirement of the project is .‖ The facilitator allows the participants seven minutes to write down their answers. The answers are then collected and consolidated for any duplicates or invalid requirements, and participants share their ideas about the security requirements [Hubbard 2000, Mead 2005].

The Accelerated Requirements Method (ARM) has been chosen for this workshop.

*Although most of the elicitation work will be done by the requirements engineering team, it is important that you follow their instructions. It is also imperative that you extend your full cooperation during the elicitation process so that verifiable and quantifiable requirements are generated.*

**ARM:** ARM is a technique that has been designed to elicit, categorize, and prioritize security requirements. Please read the following description, as it will help you up until Step 8.

ARM is centered around ―Brainstorm, Organize, and Name (BON).‖ The requirements engineering team and the client (you) meet to develop the initial security requirements. To start with, the RE team will ask you for a focus question. For this workshop, you can keep it as the following:

―An important security requirement of the project is ‖

**Task**

Listed below are requirements that are relevant to this case. Discuss these with the requirements engineering team and finalize them in the space provided.

Time : 30 minutes

|  |  |
| --- | --- |
| **#** | **Requirement** |
| R01 | The system is required to have strong authentication measures in place at all system gate- ways/entrance points. |

|  |  |
| --- | --- |
| **#** | **Requirement** |
| R02 | The system is required to have sufficient process-centric and logical means to govern which sys- tem elements (data, functionality, etc.) users can view, modify, and/or interact with. |
| R03 | It is required that the designated security personnel be able to audit the status and usage of sys- tem resources (including security devices). |
| R04 | It is required that the system’s network communications be protected from unauthorized infor- mation 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. The categorization could be either essential and non-essential requirements or software level and system level requirements. Doing this classification makes prioritization much easier.

**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 customer 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. An example of this could be that the company is partnered to use technology from other specific technology. Even when comparable technology is available to be used from other vendors, the architect must use the one provided by the business partner.

**Technical constraints** are put in because of technical limitations that the company has. This usually involves existing systems that an architect must use in the construction of a new system [Lattanze 2008].

*This step is also a joint effort by the requirements engineering team and the client. However, the client (you) is expected to categorize these requirements. The RE team might give you some preconceived classifications.*

**Task**

Categorize the requirements based on whether they are system or software level requirements. If you and the RE team deem that a particular requirement is actually an architectural constraint, it should be eliminated.

Time: 20 minutes

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

## Step 8 – Prioritize Requirements

In a real organization, there may be many more security requirements than have been listed here. Hence, prioritization of requirements might be essential, because implementing all of them might be infeasible.

**Defining a Prioritization Scale**

Before using a prioritization scale, ensure that all stakeholders reach consensus about the definition of each word in the scale. The table below shows some possible definitions for a scale of high, medium, and low.

Use the following scale for prioritization:

|  |  |
| --- | --- |
| **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**

You as the client must prioritize the requirements for the RE team. The prioritization could be an unstructured, informal discussion or a structured one. Structured prioritization techniques like AHP might take long and require a cost/benefit analysis for prioritization. For this workshop, restrict the prioritization to an informal discussion. Consider the results of Step 4 (risk assessment) and Step 7 (categorization of requirements) while doing the prioritization.

Time: 20 minutes

The RE team might try to influence your decision based on their internal analysis. You must take their analysis into account while prioritizing the requirements. R01–R04 are suggested to be important for Ideal. You are free to come up with a different prioritization.

H – High Priority M – Medium Priority L – Low Priority

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| R01 |  |
| R02 |  |
| R03 |  |
| R04 |  |
| R05 |  |
| R06 |  |

## Step 9 – Inspect Requirements

This is the last step of the SQUARE process. In Step 6, it was mentioned that requirements should be accurate, quantifiable, and verifiable. In this step, all the requirements are carefully reviewed and changed if need be. Note that if there is any drastic change in the requirements, the categorization and prioritization might change.

**Conducting Formal Inspection**

Formal inspection, such as Fagan inspection, 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

**Participant—**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

1. Prepare for inspection

The facilitator prepares for an inspection.



Every member in the meeting reviews the requirements by themselves and generates an initial list of questions they have and/or defects they find.

1. Conduct inspection meeting
   1. The reader reads one requirement at a time.
   2. Participants state the defects they find or ask questions.
   3. The reader reviews each checklist item against the current requirement, and the participants agree or disagree about whether the checklist item is passed.
   4. The writer writes down any defects found by the team.
2. Revise the requirements

After the meeting, the author revises the requirements based on 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**

Use the given checklist to review the requirements with the RE team. It is important to document the review comments. The RE team will do that. You may want to write them down too for your reference.

**Review Log**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Description of Defect** | **Severity** | **Suggested Changes** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Supplemental Material

## Conducting Brainstorming

Brainstorming is a technique that can be used to generate a large number of ideas to arrive at a solution for a problem. It was popularized by Alex Osborn in his book *Applied Imagination* [Osborn 1979]. The main idea is to quickly generate as many ideas as possible and then select the best few from them.

1. Assign roles

Before conducting the brainstorming meeting, the team needs to agree on the roles that each participant will play. There are two administrative roles; the rest of the team will be participants who help to generate ideas.

**Facilitator—**Leads the session and enforces the brainstorming ground rules

**Idea Collector—**Writes summaries of ideas on a white board during the brainstorming session

**Participants**—Generate ideas

1. Prepare for the meeting
   1. Set the problem. In this case, the problem is the terms that would be useful in the security requirements elicitation specific to the project.
   2. Provide problem background. Provide information about the brainstorming ses- sion and the problem to the participants before the meeting.
   3. Prepare leading questions. If the idea generating starts to slow down in the brainstorming session, the facilitator should stimulate the group by asking lead- ing questions, such as to break the problem down or look at it from a different perspective. These questions should be prepared before the session.
2. Conduct the brainstorming session
   1. Before starting the meeting, the facilitator briefly reminds the participants about the ground rules.
   2. The facilitator presents the objective of the meeting (―to generate a list of defini- tions‖) and then answers any questions the team might have about the problem.
   3. When all questions about the objective are answered, the facilitator proceeds to ask the participants to give the words that they are concerned with.
      1. If the group is stuck, ask a prepared leading question.
      2. The idea collector records ideas proposed by the participants.
   4. When the time is up, the facilitator organizes the words into groups.
   5. The facilitator then removes duplicate or synonymous words from the list.

<http://en.wikipedia.org/wiki/Brainstorming> <http://www.brainstorming.co.uk/tutorials/historyofbrainstorming.html>

**Guidelines for Brainstorming**

Focus on collecting words and welcome all ideas without judgment. Avoid discussion during the brainstorming activity.



Allow new ideas to be built from existing ideas.

Write all ideas on the whiteboard where every participant can see them. Set a strict time limit for the brainstorming session.

Number the ideas generated for a quick reference during the session.

When faced with overwhelming ideas, the one with highest association should take priority. This is to encourage building on each other’s ideas.

 Managers may be excluded from attending because their presence may inhibit the generation of ideas, especially for unusual ideas.

<http://en.wikipedia.org/wiki/Brainstorming> <http://www.mcli.dist.maricopa.edu/authoring/studio/guidebook/brain.html>

# References

[Allen 1999]

Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., & Stoner, E. *State of the Practice of Intrusion Detection Technologies* (CMU/SEI-99-TR-028, ADA375846). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html>

[Armour 2000]

Armour, F. & Miller, G. *Advanced Use Case Modeling: Software Systems*. Addison-Wesley Professional, 2000 (ISBN 0-20161-592-4).

[Butler 2002]

Butler, S. A. & Fischbeck, P. ―Multi-Attribute Risk Assessment.‖ *SREIS 2002, Second Sympo- sium on Requirements Engineering for Information Security*. Raleigh, NC, October 16, 2002. La- fayette, IN: CERIAS, Purdue University, 2002.

[CERT/CC 2002]

CERT Coordination Center. *Survivable Systems Analysis Method*, 2002. <http://www.cert.org/archive/html/analysis-method.html>

[Checkland 1990]

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

[Chen 2004]

Chen, P., Dean, M., Ojoko-Adams, D., Osman, H., Lopez, L., & Xie, N. *Systems Quality Re- quirements Engineering (SQUARE) Methodology: Case Study on Asset Management System* (CMU/SEI-2004-SR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon Uni- versity, 2004.

<http://www.sei.cmu.edu/library/abstracts/reports/04sr015.cfm>

[Cornford 2004]

Cornford, Steven L., Feather, Martin S., & Hicks, Kenneth A. *DDP – A Tool for Life-Cycle Risk Management*, 2004. <http://ddptool.jpl.nasa.gov/docs/f344d-slc.pdf>

[DAU 2002]

Defense Acquisition University. ―What is the multi-voting method?‖ https://acc.dau.mil/CommunityBrowser.aspx?id=50588 (2002).

[Ellison 2003]

Ellison, R. J. & Moore, A. P. *Trustworthy Refinement Through Intrusion-Aware Design* (CMU/SEI-2003-TR-002, ADA414865). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/library/abstracts/reports/03tr002.cfm>

[Fagan 1976]

Fagan, M. E. ―Design and Code Inspections to Reduce Errors in Program Development,‖ *IBM Systems Journal 15*, 3 (1976): 182-211.

[Guttman 1995]

Guttman, B. & Roback, E. *An Introduction to Computer Security*. Gaithersburg, MD: U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1995. <http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf>

[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.

[ISO/IEC 2005]

*Information technology – Security techniques – Code of practice for information security management* (ISO/IEC 17799:2005). Geneva, Switzerland: International Organization for Standardization, 2005.

[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>

[Kean 1997]

Kean, E. ―Feature-Oriented Domain Analysis,‖ 185-189. *C4 Software Technology Reference Guide - A Prototype* (CMU/SEI-HB-97-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. <http://www.sei.cmu.edu/library/abstracts/reports/97hb001.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-](http://www-/) iurd.ced.berkeley.edu/pub/WP-131.pdf

[Lattanze 2008]

Lattanze, A. J. *Architecting Software Intensive Systems*. Auerbach Publications, 2008 (ISBN 1- 42004-569-5).

[Linkov 2004]

Linkov, I., Varghese, A., Jamil, S., Seager, T.P., Kiker, G., & Bridges, T. ―Multi-Criteria Decision Analysis: A Framework for Structuring Remedial Decisions at Contaminated Sites,‖ 15-

1. *Comparative Risk Assessment and Environmental Decision Making*. Edited by I. Linkov and

A. Ramadan. Kluwer, 2004. <http://www.environmentalfutures.org/Images/ArmyPaper_Oct31.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*, 2004. <http://www.iatrp.com/iam.cfm>

[Osborn 1979]

Osborn, A. *Applied Imagination*. Scribner, 1979 (ISBN 0-02389-520-9).

[QFD 2005]

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

[SANS 2009]

The SANS Institute. *SANS Glossary of Terms Used in Security and Intrusion Detection*. <http://www.sans.org/resources/glossary.php>

[Schiffrin 1994]

Schiffrin, D. *Approaches to Discourse*. Oxford, England: Blackwell Publishers Ltd, 1994.

[Sindre 2000]

Sindre, G. & Opdahl, A. L. ―Eliciting Security Requirements by Misuse Cases,‖ 120-131. *Proceedings of the 37th International Conference on Technology of Object-Oriented Languages (Tools 37-Pacific 2000)*. Sydney, Australia, November 20-23, 2000. Los Alamitos, CA: IEEE Computer Society, 2000.

[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>

[SSQ 2009a]

―access control list.‖ SearchSoftwareQuality.com Definitions, 2009. [http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92\_gci213757,00.html](http://searchsoftwarequality.techtarget.com/sDefinition/0%2C%2Csid92_gci213757%2C00.html)

[SSQ 2009b]

―authentication.‖ SearchSoftwareQuality.com Definitions, 2009. [http://searchsecurity.techtarget.com/sDefinition/0,,sid14\_gci211621,00.html](http://searchsecurity.techtarget.com/sDefinition/0%2C%2Csid14_gci211621%2C00.html)

[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.

[Webopedia 2009]

―firewall.‖ Webopedia.com, 2009. <http://www.webopedia.com/TERM/F/firewall.html>

[Wikipedia 2009a]

Wikipedia. ―Access control.‖ <http://en.wikipedia.org/wiki/Access_control#Computer_security> (2009).

[Wikipedia 2009b]

Wikipedia. ―Computer insecurity.‖ <http://en.wikipedia.org/wiki/Computer_insecurity>(2009).

[Wikipedia 2009c]

Wikipedia. ―Data integrity.‖ <http://en.wikipedia.org/wiki/Data_integrity>(2009).

[Wikipedia 2009d]

Wikipedia. ―Multi-criteria decision analysis.‖ <http://en.wikipedia.org/wiki/Multi-> criteria\_decision\_analysis (2009).

[Wood 1995]

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