System Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II

Dan Gordon
Ted Stehney
Neha Wattas
Eugene Yu

Nancy R. Mead, Advisor

May 2005

SPECIAL REPORT
CMU/SEI-2005-SR-005

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

System Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II

CMU/SEI-2005-SR-005

Dan Gordon
Ted Stehney
Neha Wattas
Eugene Yu

Nancy R. Mead, Advisor

May 2005

Networked Systems Survivability Program

This report was prepared for the

SEI Joint Program Office
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

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

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK

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

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

About This Report

This report reflects the work of four graduate students at Carnegie Mellon University working to fulfill their security synthesis project requirements. Two other students, who were involved in an earlier iteration of the work, also lent help and guidance. Together, this group of students was tasked with testing an implementation of the System Quality Requirements Engineering (SQUARE) Methodology [Firesmith 04] on a client system under the guidance of Dr. Nancy Mead.

Previous work in the matter was conducted at Carnegie Mellon University over the summer of 2004, which produced *System Quality Requirements Engineering (SQUARE) Methodology: Case Study on Asset Management System* [Chen 04]. The work conducted by the new team during the fall of 2004 builds on this previous work, with a specific focus on certain areas of the nine-step SQUARE Methodology.

This report will deal mainly with the new areas of focus concerning the SQUARE Methodology. It will present the methodology followed for each step, will briefly discuss findings for the client, and will make recommendations for the SQUARE Methodology. This report is intended to serve individuals involved in future iterations of the SQUARE Methodology, as well as the faculty members at Carnegie Mellon® who continue to refine the structure of the SQUARE Methodology. The work from this group has produced a final deliverable for the client.

Acknowledgements

The project team would like to thank Dr. Nancy Mead for overseeing and mentoring our work on the subject matter. We have all gained a great deal of insight into the security consulting industry, and thank Dr. Mead for providing the opportunity. We would also like to thank the Acme Corporation for working closely with us to provide a system for us to analyze. Finally, we would like to thank Peter Chen for providing guidance throughout the project, especially in his efforts to help us find direction as we made our transition from the work provided by the previous SQUARE team.

Abstract

This report describes the second phase of an application of the System Quality Requirements Engineering (SQUARE) Methodology developed by the Software Engineering Institute's Networked Systems Survivability Program on an asset management system. An overview of the SQUARE process and the vendor is presented, followed by a description of the system under study. The research completed on Steps 4 through 9 of this nine-step process is then explained and feedback on its implementation is provided. The report concludes with a summary of findings and gives recommendations for future considerations of SQUARE testing.

This report is one of a series of reports resulting from research conducted by the SQUARE team as part of an independent research and development project of the Software Engineering Institute.

1 Introduction

1.1 Overview

The following section gives some background information on the SQUARE Methodology, a description of the client (the Acme Corporation), and an explanation of the Asset Management System (AMS) and its applications [Chen 04].

1.2 Square Methodology

The System Quality Requirements Engineering (SQUARE) Methodology is a nine-step process developed by Professor Nancy Mead as a part of a research project with Professors Donald Firesmith and Carol Woody to ensure the safety and survivability of IT systems and applications. Although the SQUARE Methodology is still under review by the SEI's Networked Systems Survivability (NSS) Program, it demonstrates great potential for industry-wide adoption for developing secure applications and systems. The methodology was applied on the Acme Corporation's Asset Management System for evaluation, where it assisted in identifying potential threats and vulnerabilities. Its application also resulted in recommendations for improvements to ensure normal system operation in the event of a security breach [Chen 04].

1.3 Acme Corporation

The Acme Corporation is a private company headquartered in Pittsburgh with a staff of approximately 1,000 across multiple offices in the United States. It provides technical and management services to various government sectors and a number of diversified private companies.

1.4 Asset Management System

ABC Services is one of four major subsidiaries of the Acme Corporation. ABC provides a range of specialized services for asset management. With over 15 years of experience in this arena, ABC developed the Asset Management System (AMS). AMS is an Executive Information System for asset management 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.

AMS also integrates with many third-party software suites to provide enterprise-level services and features. Archibus/FM, 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), which is used to organize information and geographic locations by sites.

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

1.5 SQUARE Team, Phase II

The original research in the application of SQUARE against the Acme Corporation's Asset Management System was conducted by a group of students from Carnegie Mellon University over the summer of 2004. The work described in this report outlines the second phase of SQUARE research that builds from the work of the previous iteration.

This first phase provided a number of important inputs to our continued work, including artifacts (attack trees, use cases, misuse cases, etc.), a client deliverable, and a process document entitled *System Quality Requirements Engineering (SQUARE) Methodology: Case Study on Asset Management System* [Chen 04].

In the project proposal outlined for this phase II team, a number of tasks were flagged by faculty as needing more in-depth study. Our first task as a team was to assign group roles, which we agreed to as follows:

- Dan Gordon, process manager
- Ted Stehney, project manger
- Neha Wattas, process analyst
- Eugene Yu, financial manager

Next, we were tasked with creating a project plan outlining which of the tasks we planned to accomplish. We followed the plan closely with very little deviation along the way. In our work, we completed a number of stepping stone deliverables for Dr. Mead, some of which are partially incorporated either in the body of this document or as appendices. The following is a list of important deliverables we completed, with the final one being this report:

- project plan
- use case deliverable and diagrams
- attack trees deliverable and diagrams
- Survivable Systems Analysis Step 2 deliverable
- risk assessment literature review
- risk assessment document

- initial security requirements document
- final security requirements document
- *System Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II*

We divided the work between group members, and kept regular meetings with both the team and with Dr. Mead to ensure that our work was progressing properly. We kept in contact with the client through phone and email, and held three client meetings at the client site throughout our work.

After setting up the initial logistics of our team environment, we began analyzing the plethora of information provided by this summer’s first application phase. Though some of the SQUARE steps were worked on in parallel, we present our work and findings in line with the steps outlined in our research proposal.

2 SQUARE Step 4: Artifact Development

2.1 Overview

The work provided from the previous SQUARE iteration contained a set of artifacts that needed more attention. For instance, an elementary set of attack trees was provided, as well as a set of use cases that was not satisfactorily inclusive of major system functionality. Our first task was to create a more detailed, comprehensive set of attack trees and use cases to serve as artifacts for analysis. We were then to trace the attack trees to both the client mission and to the use cases and identify any areas that might be determined to be insufficient in quality or level of abstraction. Along the way, we also discovered the need to determine essential services and assets per Step 2 of Survivable Systems Analysis.

2.2 Methodology

2.2.1 Attack Trees

Attack tree work began with an analysis of the previous group's attack tree deliverable. Faculty comments suggested that the original attack trees were not fully inclusive and that a full set of attack trees should be compared to the misuse cases to determine whether the two techniques provided similar results. As a result, a new set of attack trees was created (Appendix A).

One of the goals of the attack trees was to ensure that we had a complete set of misuse cases. Table 1 shows a mapping between the attack trees and the misuse cases. While the attack trees give a general picture of the nature of potential attacks on the system, the misuse cases drill down to the details of the interactions between system components in the event of an attack.

Table 1: Mapping of Misuse Cases to Attack Trees

Misuse Case Name	**Attack Tree**
Unauthorized logon on the Windows 2003 server	AT-01-04
Sys admin gains access to system data	AT-01-02
User gains sys admin rights on the Windows 2003 server (elevation of privilege)	AT-01-04
Sys admin deletes critical system configurations on the Windows 2003 server	AT-01-02
Sys admin creates holes in the system configurations on the Windows 2003 server	AT-01-02
User deletes critical data from the AMS system	AT-01-03

Table 1: Mapping of Misuse Cases to Attack Trees, cont.

Misuse Case Name	Attack Tree
User falsifies system data	AT-01-03
System data accessed through developmental machines	AT-01-01,02
System data accessed directly to/from database	AT-01-01,02
User credential information stolen through developmental machines	AT-01-01,02
User sees data that he or she should not see from workstation	AT-01-01,02,03
Malicious user uses replay attack in the same browser to assume the identity of another user	AT-01-05
Malicious user taps communications channel between workstations and servers	AT-01-05
Malicious user gains access to sensitive data via saved Excel export files on victim's machine	AT-01-05
Malicious user installs malicious programs that can tap into Excel's memory to steal exported data	AT-01-05
Input validation attack	AT-01-05
Infect Windows 2003 server with virus/worms	AT-01-05
User gains access to the system using spoofed identities	AT-01-04
Information gathering/network eavesdropping	AT-01-05
Brute force attacks: password cracking/credential theft	AT-01-03
Denial of service	AT-02-01
Execute malicious code	AT-01-05

2.2.2 Use Cases

Use case work provided from the previous group included eight usage scenarios:

1. UC-01 View Floor Plans
2. UC-02 Damage Assessment
3. UC-03 Mark Up/Create Floor Plans
4. UC-04 Find Specialized Employees
5. UC-05 Journal Entry
6. UC-06 Install the Asset Management System
7. UC-07 Create Links
8. UC-08 Archibus Administration Adding a User and Assigning Privileges

Our group worked to discover other important usage scenarios and to revise old use case work as necessary. To begin, our group obtained and utilized a working online prototype of AMS. Rigorous testing was conducted using the online demo. Our group even discovered new functionality (and created a use case around it) that had to be stricken from the record, as the Acme Corporation had accidentally included functionality involving overseas support that was not intended to be part of this analysis. In conjunction with online testing, we researched important functions outlined in the AMS user's guide provided by the client.

We discovered new and important use cases and have moved forward with a total of 11. We met with the client to ensure accuracy in our use case analysis. Per the client's comments, some small details have been updated in many of the previous use cases. Of the 11 use cases listed below, those in bold are either old use cases that have undergone somewhat important facelifts or are new use cases entirely:

- UC-01 View Floor Plans
- UC-02 Damage Assessment
- **UC-03 Add/Delete/Edit Post-it Note**
- UC-04 Find Specialized Employees
- UC-05 Journal Entry
- UC-06 Install the Asset Management System
- **UC-07 Create Links to the Documents**
- UC-08 Archibus Administration Adding a User and Assigning Privileges
- **UC-09 View Contact Information for Maintenance Tasks**
- **UC-10 Create Open Space Report**
- **UC-11 View Incident Command**

The expanded set of use cases can be seen in Appendix B.

In conjunction with textual guidelines of use case analysis, our group conducted research regarding visual use case analysis as it relates to the system architecture. That is, we traced the use cases through the system architecture and outlined components and connectors that were necessary for a usage scenario. Through working with the client, we were able to make revisions from previous use case diagrams, creating more accurate artifacts. In certain cases, original use cases were inaccurate. The new set of use cases correctly defines the components and connectors associated with each use case trace. The new set of use case diagrams can be seen in Appendix C.

2.2.3 Essential Services and Assets

The Survivable Systems Analysis (SSA) Method, developed at the CERT® Coordination Center, is a white team exercise aimed at providing survivable recommendations for a system [CERT/CC 02]. From the work completed by the previous SQUARE team, we noticed that there had been some overlap of work that would be completed in a Survivable Systems Analysis. We noticed that one important element of the SSA Method—Step 2, defining essential service scenarios and components—had not been fully realized. Further, its importance in understanding the vital characteristics of a system has not yet been built into the SQUARE model. We found it necessary to determine essential services and assets.

To begin understanding the essential elements, we first looked back to the business mission. According to Acme, AMS is designed to provide the ability to make important decisions based on current and available information. We have analyzed the major usage scenarios of the Asset Management System and have made a determination as to which services, assets, and components are essential to making informative decisions regarding emergency scenarios. We have also considered the security goals as outlined by the previous team's work.

Essential Services

We have analyzed the importance of each of the major system services, outlined in the 11 use cases shown in Table 2, and made a determination on its essentiality:

Table 2: Essential Services

Use Case	**Service**	**Status**
UC-1	View Floor Plans	***Essential***
UC-2	Enter Damage Assessment	***Essential***
UC-3	Add/Delete/Edit Post-it Notes	Non-Essential
UC-4	Find Specialized Employees	***Important***
UC-5	Create Journal Entry	Non-Essential
UC-6	Install the Asset Management System	Non-Essential
UC-7	Create Links to Documents	Non-Essential
UC-8	Archibus Admin- Add User and Assign Privileges	Non-Essential
UC-9	View Contact Information for Maintenance Tasks	***Important***
UC-10	Create Open Space Report	***Essential***
UC-11	View Incident Command	***Essential***

® CERT and CERT Coordination Center are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

The major business goal of the Asset Management System is to allow decisions to be made both before an event takes place (i.e., in the planning phase), as well as during and after an event. The most critical services needed to assist decision making are those that directly affect viewing and altering event-specific information. Thus, viewing floor plans, entering damage assessments, creating open space reports, and viewing incident commands would be of top interest. Should an emergency or an attack occur, we would want at a minimum to preserve these system functions. Though probably not critical, two services have been flagged as important but not essential: view contact information for maintenance tasks and find specialized employees.

The other major functions have all been deemed non-essential. Though important to the functioning and upkeep of the system, the ability to add this information can be recovered after an attack. Many of the other functions deal with configuring the actual system or its user profiles, as well as creating enhancements to the system for future decision making, including linking documents to assets. Others involve making handy but non-critical posts in the form of journal entries or post-it notes. Still other services support the viewing of non-critical data: overseas contact information. While all of this functionality is important to the long-term usability of the system, an attack on these services does not threaten the ability of the Asset Management System to allow decision making while under attack. If compromised, the information and services would need to be repaired before the system would become fully usable and functional again. That is, these services would need to be repaired to allow information in the Asset Management System to be brought up to date. However, if the information in the Asset Management System is kept current, and an attack occurs, the ability to add new assets, documents, etc. is secondary to viewing the current state of assets.

Essential Assets

There are two major assets in this system. The first is the Windows Server Computer, which houses the majority of the production system's intellectual assets (i.e., the code that runs the system). This computer acts as a server that allows remote users to access the Asset Management System. Next, the information inside the Windows Server Computer, specifically the files stored in the Microsoft Internet Information Server (IIS), as well as the information stored in the Sybase Database and in the MapGuide Database, is critical for making informed decisions. If this information is lost or compromised, the ability to make accurate decisions is lost.

The AMS User Workstation is not considered essential, and neither is the AMS Development Workstation. No important files or intellectual assets critical to the Asset Management System's mission are housed on these machines. Should they go down, a spare machine could easily fill in as a replacement, provided the proper software is available. This is not the case with the Windows Server or the information that it contains. An attack on its ability to function, or on its ability to deliver accurate information, will tremendously impact survivability. The system will most likely fail to achieve its mission under such circumstances.

Essential Components

As we followed the user traces of the essential services through the system architecture, we determined which components are essential to the survivability of the system. That is, the usage scenarios of the essential services touch the following components, creating an essential dependence on them (for a visual representation, see Appendix D):

Component	Architectural Location
Windows Server CPU	Various network locations
MapGuide Files	Windows Server CPU
MapGuide Client	Windows Server CPU
Sybase Central Server	Windows Server CPU
Microsoft IIS Server	Windows Server CPU
HTML/web Files	Windows Server CPU
Internet Explorer	AMS User Workstation
Ethernet Fiber	Various network locations
Ethernet Connectors	Various network locations

2.3 Client Feedback

Throughout our work, we communicated with the client to make sure that our artifacts were falling in line with the ways they viewed the system. The client helped in providing feedback that was considered before delivering our final results. After viewing all of our final work in attack trees, use cases, and essential services and assets, the client has determined that all work satisfactorily characterizes the AMS in its intended form. The client found attack tree deliverables especially persuasive when defending the need for SQUARE analysis to colleges not directly concerned with security.

2.4 Recommendations

All artifacts—including use cases, misuse cases, attack trees, and essential services and assets—were important to our SQUARE research. Misuse cases and attack trees were used specifically as inputs to the risk assessment phase. Essential services and asset analysis was used to write initial safety and security requirements. Essential services and asset identification and attack trees should be formally included as part of Step 4, Develop Artifacts to Support Elicitation Technique.

It would be clearer to name Step 4 Artifact Development and to list attack trees, use cases, and essential assets and services analysis as outputs for this step.

3 SQUARE Steps 5, 6: Elicit and Categorize Safety and Security Requirements

3.1 Overview

The work from the previous SQUARE team produced a security requirements document as an output. It was conveyed by faculty, and confirmed by this SQUARE team, that the previous work was insufficient as a security requirements document. The task of our team, then, was to reengineer this document so that security requirements could be easily mapped back to the client's security and business goals, as well as to a lower level of implementation detail, and serve as a meaningful, final output to the client.

3.2 Methodology

Our team began by analyzing the architectural recommendations (ARs) and policy recommendations (PRs) posed by the previous group's security requirements document. There were many problems with the initial document. First, a major shortcoming of the previous work was that the requirements were more like recommendations, or fixes. Even though these may be beneficial to the client, recommendations are not the final output from SQUARE, requirements are. Second, there were redundancies and inconsistencies in the requirements provided by the first SQUARE team. Last, there wasn't much clarity or cohesion in the presentation of these recommendations, especially in understanding how they mapped back to business and security goals or how they related to the misuse cases. Our initial task was to make sense of the various ARs and PRs provided to us by the previous team's work. From there, we focused on creating a hierarchical structure (shown in Figure 1) into which we could fit the ARs, PRs, and other outputs from the previous group's work, a structure that would allow us to start from Acme's high-level business goals for the Asset Management System and drill down into security goals, security requirements, and lastly, the specific architectural (technological) fixes their system required.

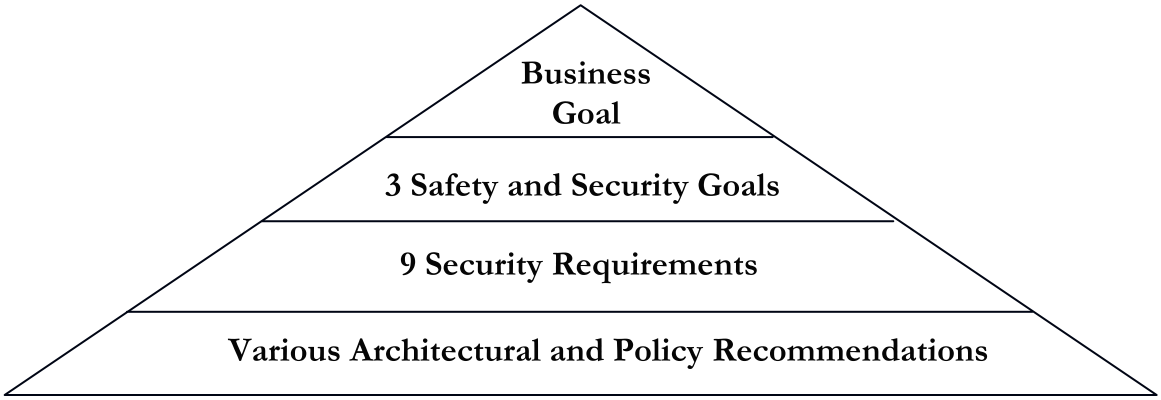

Business Goal
3 Safety and Security Goals
9 Security Requirements
Various Architectural and Policy Recommendations

Figure 1: Goal and Requirements Hierarchy

Our initial analysis resulted in the framework shown in Figure 2, including what was then a draft of nine security requirements.

	Acme's Business Goal for the Asset Management System (AMS):	
	"This tool … provides the means to make informative decisions based on available sources. "	
#	**Asset Management System (AMS) Security Goals:**	**Goal #**
1	Management shall exercise effective control over the system's configuration and usage.	G-01
2	The confidentiality, accuracy and integrity of the AMS's data shall be maintained.	G-02
3	The AMS system shall be available for use when needed.	G-03
#	**Asset Management System (AMS) Security Requirements:**	**Refers to Goal #**
R-01	The system is required to have strong authentication measures in place at all system gateways/entrance points.	G-01,02
R-02	The system is required to have sufficient process-centric and logical means to govern which system elements (data, functionality, etc.) users can view, modify and/or interact with.	G-01,02
R-03	It is required that a continuity of operations plan (COOP) be in place to ensure appropriate system availability.	G-03
R-04	It is required that the AMS's designated security personnel be able to audit the status and usage of system resources (including security devices).	G-01
R-05	The AMS's designated personnel are required to audit the status of system resources and their usage on a regular basis.	G-01
R-06	It is required that the system's network communications be protected from unauthorized information gathering and/or evesdropping by encryption and other reasonable techniques.	G-01,02
R-07	It is a requirement that both process-centric and logical means be in place to prevent the installation of any software or device without prior authorization.	G-01
R-08	It is required that the AMS's physical devices be protected against destruction, damage, theft, tampering or surreptitious replacement (including but not limited to damage due to vandalism, sabotage, terrorism or acts of God/Nature).	G-03
R-09	It is required that the AMS's software components be designed utilizing software security best practices.	G-02,03

Figure 2: Summary of Security Requirements

With this framework in place, we were then able to map the existing ARs and PRs to each security requirement, as appropriate. Table 3 provides an example of associated ARs and PRs for the first general recommendation, R01, "The system is required to have strong authentication measures in place at all system gateways/entrance points."

Table 3: Recommended Control Measures, R01

Architectural / Logical Controls:	
AR-01	All shared drives on the network should enforce authentication policies.
AR-04	Block all unnecessary ports at the firewall and host.
AR-06	Configure routers to restrict foot printing requests.
AR-08	Developmental machines should have strong access control mechanisms.
AR-09	Disable non-critical services and protocols.
AR-10	Display generic information on log-in screen.
AR-15	Harden weak default configuration setting.
AR-17	Implement account lock-out policies.
AR-18	Implement hierarchical authorization levels.
AR-19	Implement role-based authentication.
AR-20	Install software-based firewalls on all systems in the network.
AR-26	Set up firewalls with filtering rules between servers and workstations.
AR-27	Set up an intrusion detection system.
AR-28	Set up IIS to prompt for user credential every time.
AR-29	Shorten the timeout for session kept-alive.
AR-33	Use least privileged account to access the database.
Process-Centric Controls:	
PR-02	Applications and operating systems must be patched routinely. (Bi-Monthly)
PR-06	Do not set up shared files/folders/drives on the AMS network server or workstation.
PR-07	Enforce strong password policies.
PR-08	Firewalls and intrusion detection systems (IDS) must be patched routinely. (Monthly)
PR-09	Follow the principle of least privilege and use least privileged service accounts to run processes and access resources.
PR-17	Routers must be patched routinely. (Monthly)
PR-19	Set clear and defined user access controls for all users. (Low, Medium, High, System Administrators).
PR-22	Users should have an automated log out of the AMS system after a certain time of idle activity.
PR-24	Users should not reveal their account names and passwords in any situation.

With this step accomplished, we had a framework for breaking requirements down into differing levels of abstraction. Because the previous team was able to assign cost values to the ARs and PRs, we were able to tie cost data to the nine security requirements, providing a rough estimate of how much it would cost to fulfill each requirement. Our efforts produced an initial security requirements document. This document was not a finished effort, but it served as a starting point in completing the final document.

3.3 Client Feedback

We reviewed our initial results with the client. The client agreed to our definition of the business goal and to the nine higher level requirements. We asked the client to review the financial figures provided by the previous group, and the client responded that they were comfortable with the estimates that had been made.

3.4 Recommendations

Our artifacts, including attack trees, use cases, misuse cases, and essential services and asset analysis, all played a role in helping us create the original security requirements document. Without this input, we could not have performed this work. We therefore recommend that these artifacts be formally listed as inputs to this step.

In addition, we do not see a need to distinguish between the tasks outlined in Steps 5 and 6. SQUARE Step 5, Elicit Safety and Security Requirements, is so closely coupled with Step 6, Categorize Requirements as to level (system, software, etc.) and whether they are requirements or other kinds of constraints, that it is confusing to list them as separate steps. Ranking and categorizing the threats is a natural outgrowth of eliciting them, so our recommendation is to rename Step 5 Develop Initial Security Requirements, and to list its activities as Elicit Safety and Security Requirements and Categorize Requirements.

4 SQUARE Step 7: Perform Risk Assessment

4.1 Overview

The previous SQUARE team did not provide us with any material for risk assessment. This portion of our research is the first material that is based solely on our own work. Our first task was to perform a literature review of the various risk assessment techniques available for field testing. After conducting a literature review, we selected two techniques and independently field tested the two methods. The purpose of risk assessment is to use the results in prioritizing and sanity checking security requirements.

4.2 Methodology

4.2.1 Literature Review

We began our literature review by brainstorming a list of techniques that seemed applicable to our research. Ideas came from faculty, course work completed by team members at Carnegie Mellon, and Internet and library searches. We narrowed down the applicable list of techniques to eight:

1. General Accounting Office's (GAO's) models [USGA 99]
2. National Institute of Standards and Technology's (NIST's) models [Stoneburner 02]
3. National Security Agency's INFOSEC Assessment Methodology (IAM) [NSA 04]
4. Shawn Butler's Security Attribute Evaluation Method (SAEM) [Butler 02]
5. CERT/CC's Vendor Risk Assessment & Threat Evaluation (V-RATE) [Lipson 01]
6. Yacov Haimes' Risk Filtering, Ranking, and Management (RFRM) Framework [Haimes 04]
7. CERT/CC's Survivable Systems Analysis (SSA) Method [CERT/CC 02]
8. Martin Feather's Defect Detection and Prevention (DDP) Process [Cornford 04]

From here, we completed a brief analysis to determine which models would provide a fit for testing in our case study. The results of our analysis are as follows.

We found that attempts to quantify risks on the basis of dollar value per attack are either too complicated or too involved for our limited time on this project, and we therefore rejected these methods. We believe we will add more value to Acme and this research project by relying on methodologies based on qualitative methods.

Table 4 shows which criteria we used to evaluate these methodologies and how we scored them (we used a scale of 1-4, with ‘1’ being the highest mark, and ‘4’ being the lowest). Here is a brief explanation of each rating:

1. Very suitable for the requirement
2. Well suited for the requirement
3. Somewhat unsuitable for the requirement
4. Very unsuitable for the requirement

Table 4: Evaluation of Risk Assessment Methodologies

		Suitable for small companies	Feasible to complete this semester	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

Based on the chart above, the two methodologies we chose were NIST’s SP 800-30 and Yacov Haimes’ RFRM.

4.2.2 Field Test

Now that two methodologies were chosen, we went forward with an independent field test of each. This work detailed the independent methodologies we used in testing the two models. Provided here is a brief description of our work.

RFRM contains eight phases, some of which are out of scope for risk assessment. We have identified and tested two phases that we feel are strong candidates for inclusion in the SQUARE processes’ seventh step of Performing Risk Assessment. The applicable phases are

- Phase III Bicriteria Filtering and Ranking
- Phase IV Multicriteria Filtering and Ranking.

Section 3 of the *Risk Management Guide for Information Technology Systems* published by NIST is broken into nine steps, each with a definable output that serves as the input to the next step in the process. Steps 8 and 9 were omitted from the test. Step 8 deals with control recommendations; our recommendations for the AMS are handled in a separate document. Step 9—the documentation phase—was omitted because we combined these results with the

RFRM results. Step 1 was already completed by the previous SQUARE team. Thus, the steps that we tested were

- Step 2: Threat Identification
- Step 3: Vulnerability Identification
- Step 4: Control Analysis
- Step 5: Likelihood Determination
- Step 6: Impact Analysis
- Step 7: Risk Determination

Each model's specific recommendations were useful in our continued work in our analysis of Acme's Asset Management System. The result of the different approaches in these two models produced a list of security risks that are on somewhat different levels of abstraction, and thus the two disparate sets of filtered and ranked risk scenarios cannot always be easily compared. In some cases, the results of the two models were in conflict. In others cases, the models produced similar evaluations of risk scenarios. Table 5 summarizes a three-tier view of each model's risk assessment results:

Table 5: Risk Assessment Results

	NIST	RFRM
Tier 1	Insider or terrorist alters or disables key architecture components Insider or terrorist discloses proprietary information Terrorist gains unauthorized use of system resources	Intruder executes malicious code to gain unauthorized access High-level user is recruited for help System administrator is recruited for help High-level user abuses rights System administrator abuses rights
Tier 2	Insider installs malicious software (viruses, Trojans, key loggers, etc.) Insider or natural forces physically destroys AMS components Insider steals AMS components	Insider sniffs password Hardware is damaged by natural disaster or environment Intruder socially engineers password
Tier 3	Terrorist steals AMS components Terrorist installs malicious software (viruses, Trojans, key loggers, etc.) Terrorist physically destroys AMS components Insider or terrorist alters or corrupts data	Intruder uses abandoned, authenticated browser Hardware fails Intruder guesses or cracks password

We analyzed the combined results and were able to make the following conclusions:

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

Insider threat is involved in five of NIST's top six threats and is involved in four of RFRM's top five threats. Five of RFRM's 11 risk scenarios deal with directly defeating authentication, and four more relate to circumventing authentication by way of insider help. Both models are concerned with hardware failure or destruction, but rank the importance differently. Hardware damage is a Tier 2 risk for both models, but NIST's output considers deliberate destruction by an insider or terrorist a Tier I risk. Many remaining risk scenarios from each model do not map directly to one another. NIST's output focuses more on an attacker's motives once inside the system (destroying and corrupting data, disclosing proprietary information, etc.), whereas RFRM's output deals more with the ability of an attacker to break the front-line defenses of the system (e.g., cracking or sniffing passwords).

4.3 Client Feedback

Client feedback was not a factor in this part of the SQUARE process. While the client did have extensive knowledge to share during the earlier stages of the SQUARE process (describing the typical Asset Management System architecture and how it was used by their clients), it did not possess this same level of knowledge in understanding the threat environment its systems face. This is understandable, since this security perspective is a large part of our "value-added" to the client in this project.

Additionally, the client was unable to produce any historical or statistical information about cyber security events on fielded Asset Management Systems. The risks described in our documentation are therefore perceived risks, derived from our observation of system design weaknesses, general trends in cyber security, and the output of the two risk assessment methodologies, as described above. Future SQUARE project teams may face an entirely different level of client involvement during this phase of the process. Each firm will have a different level of technical sophistication and security awareness, and with that, differing amounts of historical information about past cyber security events. In performing risk assessments during future SQUARE projects, teams should be prepared to conduct a wide array of tasks (in addition to the ones performed here) during this phase of the process, ranging from reviewing system logs to conducting staff interviews. Before developing a comprehensive project plan, SQUARE project teams should first make sure they understand the client's ability and willingness to participate in this step, as it could dramatically alter the length of time required to complete the work.

4.4 Recommendations

We value the work from both the RFRM and NIST models. We would recommend the risk assessment portions of each model for future iterations of the SQUARE process.

More specifically, we viewed the first two steps of RFRM, Phase I, Scenario Identification, and Phase II, Scenario Filtering, to be of possible value to SQUARE. The sheer magnitude of these steps, coupled with the redundancy of the artifact stage, led us to rule this step as out of scope for our work here. In a larger research project, it would be interesting to see if Haimes' suggested Hierarchical Holographic Modeling (HHM) technique could be used in the artifact development stages of SQUARE. Phase III, Bicriteria Filtering and Ranking, and Phase IV, Multicriteria Filtering and Ranking, provide good ways to analyze risk, and are both recommended as good tools for risk assessment. Phase V, Quantitative Ranking, is too difficult of a task unless the client has a good understanding of its own security statistics. This task was out of scope for our purposes, but it could be considered for a larger, more mature firm. The remaining steps were out of scope, as they dealt with risk management and life-cycle analysis. All of the NIST Model Steps 2-7 are of value, and are recommended for future consideration. Step 1 is completed in earlier SQUARE steps, and NIST Steps 8 and 9 are also determined as being out of scope because of their risk management nature.

A few words about risk management: all of the risk assessment methodologies we considered also contained strategies for managing the risks they help discover. We did not conduct the risk management portions of either methodology, but we feel that our risk analysis results would have flowed nicely into either. We view risk management as a fundamentally different activity from risk analysis/risk assessment. Though it is possible to preemptively manage risks in a design environment, RFRM and NIST's models both viewed risk management as a follow-on task to be completed, once the initial analysis was completed on a fielded system. This is a very different task than the one SQUARE was designed to accomplish.

Nonetheless, we provide the beginnings of a risk management exercise as part of our output from Step 8, Prioritize Requirements. (This is because much of our case work this semester actually falls outside of the designed usage of the SQUARE methodology. We were working with an existing system instead of designing requirements for a to-be system, which is what SQUARE would typically call for. To add some value for the client as incentive and thanks for their cooperation, we added a series of risk management features.) Specifically, we correlate the cost data developed by the previous SQUARE team with the security requirements we developed, and we provide a clear prioritization scheme for our security requirements. We do, however, leave the bulk of the work of managing risk—and responding to security requirements—as a task for the client.

In our view, it is worth considering whether risk management should be the tenth step for the SQUARE process, a step invoked if the methodology is applied on an existing system. A lesson learned from this experience was that the ARs and PRs generated by the previous group wound up functioning as a de facto framework for a risk management plan. After all, if an existing system is analyzed, weaknesses are discovered, risks analyzed, and security requirements developed, isn't it incumbent on the organization to develop a plan for managing those problems?

We did not use categorized requirements as an input to risk assessment, nor did either model call for requirements to be an input to risk assessment. Instead, we viewed the artifacts as the major driver for risk assessment. Risk assessment can be done in parallel—or even before—developing and categorizing requirements. This finding contradicts the original research plan, which calls for categorized requirements to be completed first so as to serve as an input to risk assessment.

5 SQUARE Step 8: Prioritize Requirements

5.1 Overview

The results from risk assessment allowed us to prioritize the requirements developed in Steps 5 and 6. Once the risks to the system have come to fruition, each requirement can be designated a criticality level to reflect its relevance. In this manner, the final list of requirements will be better tailored to meet the environment of the system at hand. Also, an input from the previous group had outlined the costs associated with each architectural and policy recommendation, allowing us to assign a cost value to each requirement. The final prioritized output from this step shows the amount of funds necessary to fully meet a requirement, aiding in the decision-making process.

5.2 Methodology

From the results of risk assessment, we were able to determine a criticality rating for the requirements developed as an output of Steps 5 and 6. We developed a three-tiered ranking scheme that will help Acme make implementation choices in a resource-constrained environment:

- **Essential** implies that the product will not be acceptable unless these requirements are provided [IEEE 98].
- **Conditional** implies that these are requirements that would enhance the product, but would not make it unacceptable if they are absent [IEEE 98].
- **Optional** implies a class of functions that may or may not be completely necessary [IEEE 98].

Each requirement was ranked based on the above criteria. The essential requirements all map directly back to the highest level of threats listed in the risk assessment results shown in Table 5; the requirements that follow all support and strengthen the essential requirements.

Essential Requirements

In our analysis, five requirements were deemed essential:

- **R01** – Strong authentication: The system is required to have strong authentication measures in place at all system gateways/entrance points/interfaces.

- **R02** – Access control: The system is required to have sufficient process-centric and logical means to govern which system elements (data, functionality, etc.) users can view, modify, and/or interact with.
- **R06** – Protect network communication: It is required that the system's network communications be protected from unauthorized information gathering and/or eavesdropping by encryption and other reasonable techniques.
- **R07** – Configuration management: It is a requirement that both process-centric and logical means be in place to prevent the installation of any software or device without prior authorization.
- **R08** – System tampering: It is required that the AMS's physical devices be protected against destruction, damage, theft, tampering, or surreptitious replacement (including but not limited to damage due to vandalism, sabotage, terrorism, or acts of God/nature).

These five requirements directly address the security concerns of authentication, access control, confidentiality, and physical threat. We feel that these security requirements are essential in defending the AMS from insider threat and access control-related threats that were identified as the top risks from our risk assessment results. Without meeting the above requirements, there can be little to no assurance that the AMS is protected against the top threats facing the system. At a bare minimum, Acme should consider adopting these requirements.

Conditional Requirements

In our analysis, three requirements were deemed conditional. Meeting these requirements will add to the security posture of Acme but are not necessarily essential in defending the AMS from the top threats:

- **R04** – It is required that the AMS's designated security personnel be able to audit the status and usage of system resources (including security devices).
- **R05** – The AMS's designated personnel are required to audit the status of system resources and their usage on a regular basis.
- **R09** – It is required that the AMS's software components be designed utilizing software security best practices.

Having the physical and human capital resources in place to audit the AMS will allow Acme to gain a better understanding of the actual threats and attacks facing the AMS. This information will allow Acme to really know the enemy. Acme can then tailor security decisions to meet the real threats as opposed to the theoretical, perceived threats outlined in this report. Requirements 04 and 05 will aid security, but the absence of their adoption will not render an inadequate system.

It should be a goal for all software users to work with software designed under a best practices environment, per Requirement 09. Though no software can be considered vulnerability-free, a best practices approach that relies on ideas adopted as industry standards does limit exposure to vulnerabilities, especially when contrasted with ad hoc software development. A move toward best practices software development will be a step in a more secure direction, but it is not necessarily essential in defending the top threats facing the AMS.

Optional Requirements

In our analysis, one requirement was deemed optional:

> **R03** – It is required that a continuity of operations plan (COOP) be in place to ensure appropriate system availability.

This requirement will certainly add to the security of the Asset Management System, as it will serve the survivability needs in the event of a disaster or attack. We determined this to be an optional requirement, as it may not be necessary if other appropriate control measures are in place. For example, developing a system for backups may be enough to ensure availability of the AMS. A COOP can augment the ability of a firm to respond during a crisis, but the absence of such a plan will not necessarily lead to failure in the face of an emergency. Though this requirement is a good idea, it falls last when rated against the eight requirements listed above.

Cost Figures

The data provided from the previous SQUARE team broke down the costs associated with each architectural and policy recommendation. This information was elicited from the client and took into account hardware and software costs, as well as labor costs. Because we associated the ARs and PRs with the nine requirements, we were able to determine the total cost of implementing each requirement:

• R01	$9,363.17
• R02	$14,522.17
• R04	$7,724.00
• R05	$3,528.00
• R06	$3,546.00
• R07	$8,668.17
• R08	$16,693.00
• R09	$5,222.00

These cost figures, coupled with a criticality ranking, will help the client determine which scarce resources should be applied in adopting various security requirements.

5.3 Client Feedback

We spoke with the client about the cost figures associated with the architectural and policy recommendations to ensure their accuracy. The client took a second look at the figures provided by the previous SQUARE team and agreed that they remained an accurate analysis of the perceived costs. We feel comfortable with the costs we have assigned for fully implementing each requirement.

We did not seek client feedback regarding the criticality ranking assigned to each requirement. We feel that the results of risk assessment, and our application of these results in ranking the importance of each requirement, is a value-added task to be performed by our security consulting team.

5.4 Recommendations

We believe that prioritizing the requirements based on criticality is an important step to understanding how requirements defend against risks to a system. Further, prioritizing requirements gives the target user of the security requirements a framework for understanding which requirements are of pressing need and which are merely optional given the appropriate time and budget. We are comfortable in our three-tier ranking and recommend the step of prioritizing requirements for future iterations of SQUARE.

Developing cost figures for the requirements was a good value-adding exercise for our team but is not a necessary step for SQUARE. In many cases, it may be difficult to elicit the cost figures, especially if a system is not yet produced, or because the client is not in a position to make such predictions. Though we do value the work performed in assessing costs, we do not see this as a necessary step for SQUARE in every iteration. Should the client and working environment allow this work, it is recommended. However, the process of eliciting and developing safety and security requirements will not fail with the absence of this work.

We recommend assigning cost data where applicable but do not recommend mandating this process as a formal part of SQUARE.

6 SQUARE Step 9: Conduct Requirements Inspection

6.1 Overview

The following section describes the inspection methodology used by our team. A method is used to verify requirements of the project, which involves the examination of documentation against predefined criteria. A process of examining and evaluating this report and our Security Requirement Document is provided.

6.2 Methodology

The methodology used by our team is based on the idea that each inspector has assigned responsibilities and develops an inspection log that ranks problems according to their severity. The methods helped the team not only to find problems and solutions, but also to discover the cause of the problems, which helped our team prevent similar problems.

The inspection method we used is called *peer review log*, which is a spreadsheet that inspectors use to identify and rank problems found in documents. The peer review log provides serial number, date, origin, defect type, defect description, defect severity, owner, reviewer, and status of inspection. It could be used to track defects in the document [Le Vie 00].

Peer Review Log

Table 6 is a description of the elements that make up the peer log.

Table 6: Peer Log Elements

Element	Description
SNO	The inspection report begins with this serial number of the inspection log. The format used for serial numbers is ‘SNO-xx’.
Date	We specify date of inspection and ensure the document is up to date. The format used for dates is ‘mm/dd/yyyy’.

Table 6: Peer Log Elements, cont.

Element	**Description**
Origin	The origin of the defect. It specifies sections being reviewed. The format used for Origin is 'Doc-xx, Page xx'. • *Doc-01*: System Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II (this report) • *Doc-02:* Security Requirements Document
Defect Type	We classify types of the defect identified by the inspector in this section. • missing content • unclear, ambiguous • lack of understanding of requirements • oversight • repeated occurrence of an error • undefined acronyms and abbreviations
Description	We describe details of the defect identified by the inspector in this section.
Severity	We define and classify defects according to their severity. • ***High (1):*** Could jeopardize the project success. • ***Moderate (2):*** Problem that requires correction before proceeding. • ***Low (3):*** Cosmetic or style problem.
Owner	The person identified as the facilitator
Reviewer	The person identified as the inspector
Status	Status of the inspection process • ***Open (1):*** Peer review is being processed. • ***Closed (0)*** Peer review is finished.

The following is a blank snapshot of the log:

Peer review log

SNO	Date	Origin	Defect Type	Description	Severity	Owner	Reviewer	Status

6.3 Client Feedback

We did not view client feedback as a necessary factor for this part of the SQUARE process. We viewed this step as an approach to find defects in an early stage. We did not need client help or approval in reviewing our own work.

6.4 Recommendations

We value the inspection methodology used in our peer review work. The inspection method was an excellent tool for helping identify problems and defects in the documents we produced. We could receive useful and consistent feedback on our documents and identify better solutions and improve processes of the project. We would recommend this requirements inspection step for future iterations of the SQUARE process.

7 General Recommendations for SQUARE

SQUARE provides a number of meaningful and important deliverables to the client, including a set of safety and security definitions, use and misuse artifacts, and a security requirements document. The security requirements document is the most important deliverable. It should not only reflect, but should also include, all of the previously created deliverables and artifacts in a single, succinct document. The goal of the security requirements document is to outline requirements in a way that traces backwards to the business goals of the client and to the safety and security goals of the system. Clearly, the outputs from SQUARE will aid an organization in creating safe, secure systems and will provide tools for the organization to assess safety and security health along the production life cycle. We have identified a number of recommendations for SQUARE, both in terms of its useful environment of where it should be applied and also in terms of realigning the current nine-step process to create a more logical and manageable flow.

1. **SQUARE is better geared for a to-be system, not a production system.**

   Conceptually, security requirements should be written before a system is produced so the end product can be measured against them. However, it is understandable that a firm might forego a security review such as SQUARE, due to time or budgetary constraints, or if the system is not deemed mission critical.

   The problem with applying SQUARE to an existing system is that the natural inclination would be to try to “bandage” the current configuration. The earlier on in the process the requirements are created and executed, the more “ideal” the resultant solutions will be. The old axiom in the security business applies here: You can’t ever make an insecure system secure once it’s deployed.

   In addition, there are a number of psychological forces that could hamper the work of SQUARE analysis of a completed system. Often, companies complete various types of assessments and audits simply as a matter of going through the motions. Bureaucratic processes may dictate the need for a business function to be performed even though its value is not intended to be realized in the company. Drafting a security requirements document after a system has been produced runs this same risk. A company that wishes to complete this work must be ready and willing to accept that the current system is not meeting requirements and be committed to making the potentially disruptive and unpopular changes needed to fix the problem. Further, the analysis should be performed by people who do not have a stake in its outcome, thereby removing political concerns from the execution of the plan.

2. **Beware of confusing requirements with recommendations.**

   SQUARE's main client-centric output, the security requirements document, benefits a company in a number of ways. Security requirements generation can expose architectural deficiencies, software vulnerabilities, and various other security risks. Writing a security requirements document can help an organization understand how to begin defending its resources. Security requirements also give system owners a way to provide a baseline set of expectations that can be used to measure performance and can be the standards to which appropriate personnel are held responsible. However, it is also important to understand what security requirements are not. Specifically, security requirements are not a series of specific control recommendations. Security recommendations would be a follow-on activity that requires additional steps and information gathering that is not necessarily built into the SQUARE model, such as financial forecasting and financial health issues, legal requirements, etc., and which might fall into the "tenth step" in the process, as eluded to above. We felt the pressure to begin making recommendations throughout the process, but have done our best to steer clear of that course. We offer this warning as a potential misuse of SQUARE and will discuss this in more detail below.

3. **Step 6, Risk Assessment, and Steps 4 and 5, Eliciting and Categorizing Security Requirements, can be completed in parallel.**

   In the current outline for SQUARE, the initial security requirements are elicited and written in Steps 5 and 6. Then, risk assessment is conducted in Step 7, before reprioritizing and writing the final draft of the security requirements document in Step 8. Our team separated the tasks of writing the initial requirements and conducting risk assessment into two distinct tasks. It is important to write security requirements and then separately conduct an analysis of the risks to the system. Again, we did not use initial requirements as an input to risk assessment. We feel that the order of writing the initial security requirements document and conducting risk assessment is not important. At a minimum, SQUARE should be revised to note that these two tasks are independent, but must both be completed as inputs for Step 8. This leads to our next point: that there lies a potential gain of clarity in revising the order of a few SQUARE steps.

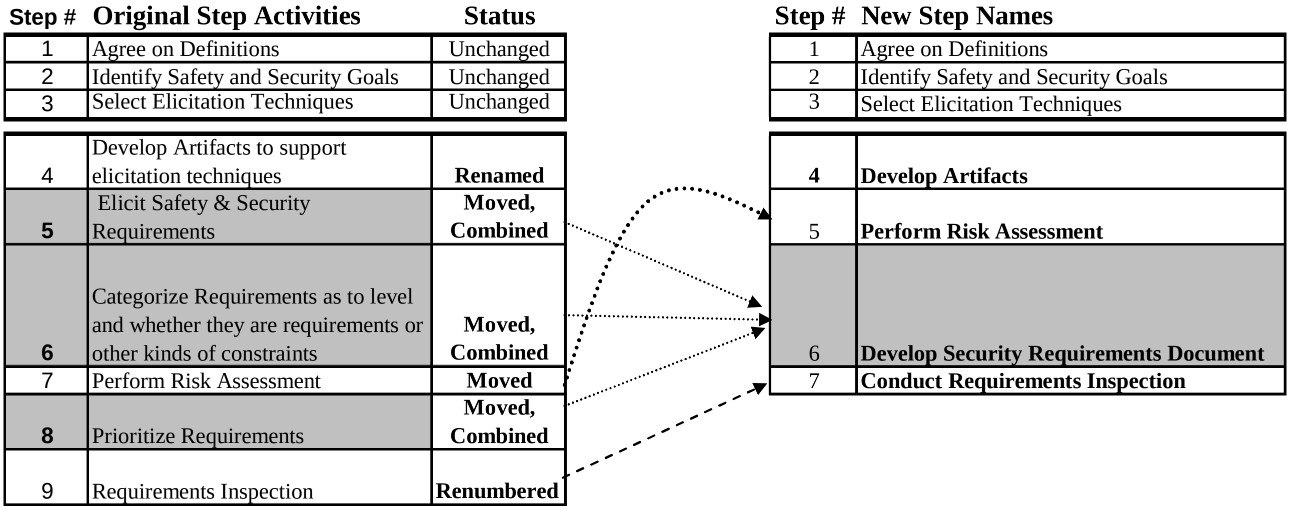
4. **SQUARE's original nine-step process can be reorganized into a more logical seven-step process.**

   Nine steps are a bit overwhelming for someone new to SQUARE. Initially, it was unclear to us what the most meaningful parts of the process were or how the steps worked together to produce a single product. Also, there was no step where the actual security requirements document was stated by name. After working with SQUARE, we have come to understand that many steps are logically similar. For clarity, some of the steps can be combined and/or renamed.

   As previously noted, Steps 5 and 6 are similar and can be combined. Step 7, Perform Risk Assessment, which we have argued can come before writing and prioritizing re-

> quirements, can be moved to an earlier place in the methodology, placing the writing of security requirement (Steps 5 and 6) and the reprioritization of those requirements (Step 8) in chronological order. Because these steps are so similar, they can be combined into one step, Develop Security Requirements Document, which would encompass the sub-activities of eliciting, categorizing, and prioritizing security requirements.

We cannot comment on making alterations to the first three steps, as we did not work directly with them. However, we do suggest changes beyond Step 3. In Figure 3, we outline a new suggested format for SQUARE:



Step #	Original Step Activities	Status
1	Agree on Definitions	Unchanged
2	Identify Safety and Security Goals	Unchanged
3	Select Elicitation Techniques	Unchanged
4	Develop Artifacts to support elicitation techniques	**Renamed**
5	Elicit Safety & Security Requirements	**Moved, Combined**
6	Categorize Requirements as to level and whether they are requirements or other kinds of constraints	**Moved, Combined**
7	Perform Risk Assessment	**Moved**
8	Prioritize Requirements	**Moved, Combined**
9	Requirements Inspection	**Renumbered**

Step #	New Step Names
1	Agree on Definitions
2	Identify Safety and Security Goals
3	Select Elicitation Techniques
4	**Develop Artifacts**
5	**Perform Risk Assessment**
6	**Develop Security Requirements Document**
7	**Conduct Requirements Inspection**

Figure 3: Suggested Modifications to SQUARE

Note that the step "Activities," as originally named in the research project guidelines, is instead listed as "Names." This helps state the goal of each step more clearly, allowing a newcomer to more easily understand the specific task and expected output just by reading the name. Also note that the original Steps 4-9 have been renamed, renumbered, combined, or moved. A more detailed description of our newly suggested structure can be seen in Appendix E.

8 General Comments About Client Selection

The Acme Corporation’s Asset Management System has been a fruitful target system in helping us provide feedback for future iterations of the SQUARE process. However, we often found that our analysis of this operational system tended to push us more into a recommendation paradigm instead of a requirements paradigm. Moreover, the security requirements document provided by the summer 2004 SQUARE team is more of a step-by-step guide toward patching the target system than it is a security requirements document. We had to constantly stay focused on the goal of SQUARE to refrain from recommending fixes for Acme and instead provide a set of requirements for the system. At times, the results of the work seemed of little usefulness, because the system is already produced. However, we feel Acme can use this work to make future decisions and to monitor and assess how it is meeting its own business and security goals.

In a future iteration, it would be useful to understand how SQUARE works in a pre-production arena, which seems to be the primary environment for SQUARE. It is our hope that our input and recommendations on the process prove valuable; we nevertheless did not use the process as it was intended or designed. Though we are confident in our results, the process will work differently and yield different results if used under a different set of conditions. It would be interesting to note whether the results from a pre-production analysis are in line with the findings of this report. For instance, the risk assessment conducted for this project benefited from our knowledge of the architecture and users of the system. Often in development, these details go through several iterations, and so do not serve risk assessment with the same degree of clarity or trustworthiness. Field testing the SQUARE Methodology in a pre-production environment may produce important, unrealized findings that are not capable of coming to fruition with an operational system.

Finally, it is worth commenting that the most difficult part of the project was getting up to speed, faced with the overwhelming amount of documentation provided by faculty and the previous team. This method of training wasn’t effective, and even though we overcame this initial obstacle, we found that the learning curve imposed by this workload did impact our efficiency early on. We do not recommend continuing this project under these types of circumstances. Instead, we recommend taking some time to consolidate the findings of the last two project teams, revising the process as needed, and then starting clean with a new group.

9 Conclusion

The ability to elicit and produce a succinct security requirements document is important for building security into the development life cycle. We have applied SQUARE against a client system and find the methodology to be an important tool for developing safety and security requirements that match the business and security goals of the client and the threat environment of the system at hand. Our research complemented and augmented work in artifact development begun by a previous team and forged new ground in analyzing risk assessment techniques and their application in terms of their appropriateness and usefulness to SQUARE. Further, we followed through on this work to develop and prioritize a final deliverable—a security requirements document—that should serve as a meaningful product to the client. Last, we experimented with a peer review technique that we found useful as we divided work—and regrouped to collaborate—to produce documents that reflected the input from all group members.

In our work, we have noted a few areas where SQUARE might be improved and how a future iteration of SQUARE research might build off of this research. We have outlined a logical reorganization of the nine SQUARE steps that condenses the methodology into a more manageable seven-step process. We have recommended the formal inclusion of various outputs that are necessary along the path toward requirements development (including the need to formally include attack trees and essential asset and service analysis as outputs of artifact development). We have analyzed a list of risk assessment techniques and were able to field test and recommend two for continued consideration of SQUARE. We have also recommended other risk assessment techniques (which we were unable to test due to time and scope constraints) that may be beneficial in a future iteration. For the first time, we have developed a final output from SQUARE containing safety and security requirements that can be mapped both upward to business and security goals as well as downward to implementation details—a hierarchical methodology that can be considered for future iterations.

We have experimented with a peer review technique and recommended our methodology as a useful tool for inclusion in SQUARE. Finally, we have made comments about the usefulness of SQUARE when applied in different environments, focusing on suitability constraints when applying the methodology to a production system.

SQUARE continues to be reviewed by the SEI’s Networked Systems Survivability Program. We find the methodology to be important and useful, and note that attempts to simplify the process may help the industry to adopt SQUARE. We are confident in our results and feel that the outputs from our work will be beneficial for our client, and hope that our suggestions for SQUARE will aid the NSS Program in their continued review of the methodology.

Appendix A New Attack Tree Diagrams

Overview

Attack trees provide a formal, hierarchical way of describing the security of the system based on the types of attacks that could happen. These diagrams represent systems in a tree structure, with the goal as the root node and tree leaves representing different ways to achieve that goal.

Contents

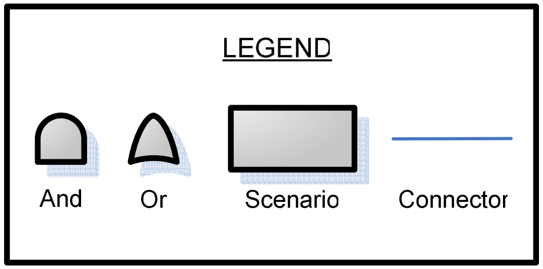


Figure 4: Diagram Legend

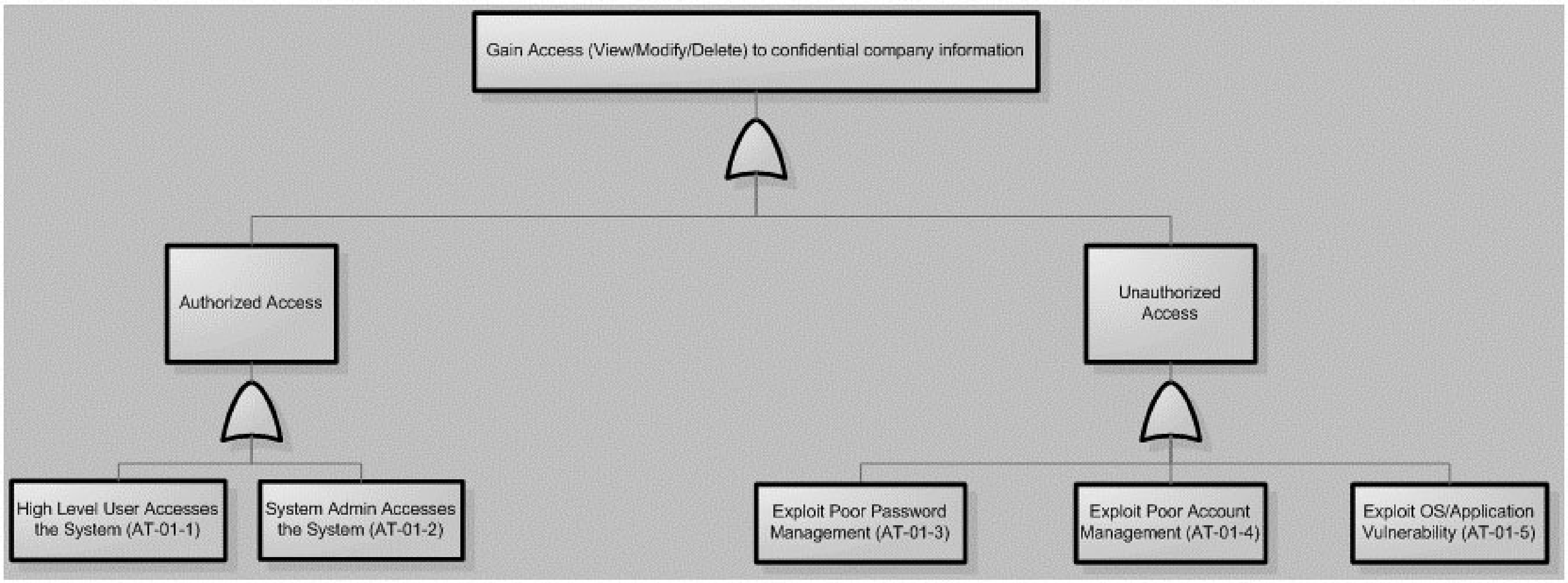

Gain Access (View/Modify/Delete) to confidential company information
Authorized Access
Unauthorized Access
High Level User Accesses the System (AT-01-1)
System Admin Accesses the System (AT-01-2)
Exploit Poor Password Management (AT-01-3)
Exploit Poor Account Management (AT-01-4)
Exploit OS/Application Vulnerability (AT-01-5)

Figure 5: Tree

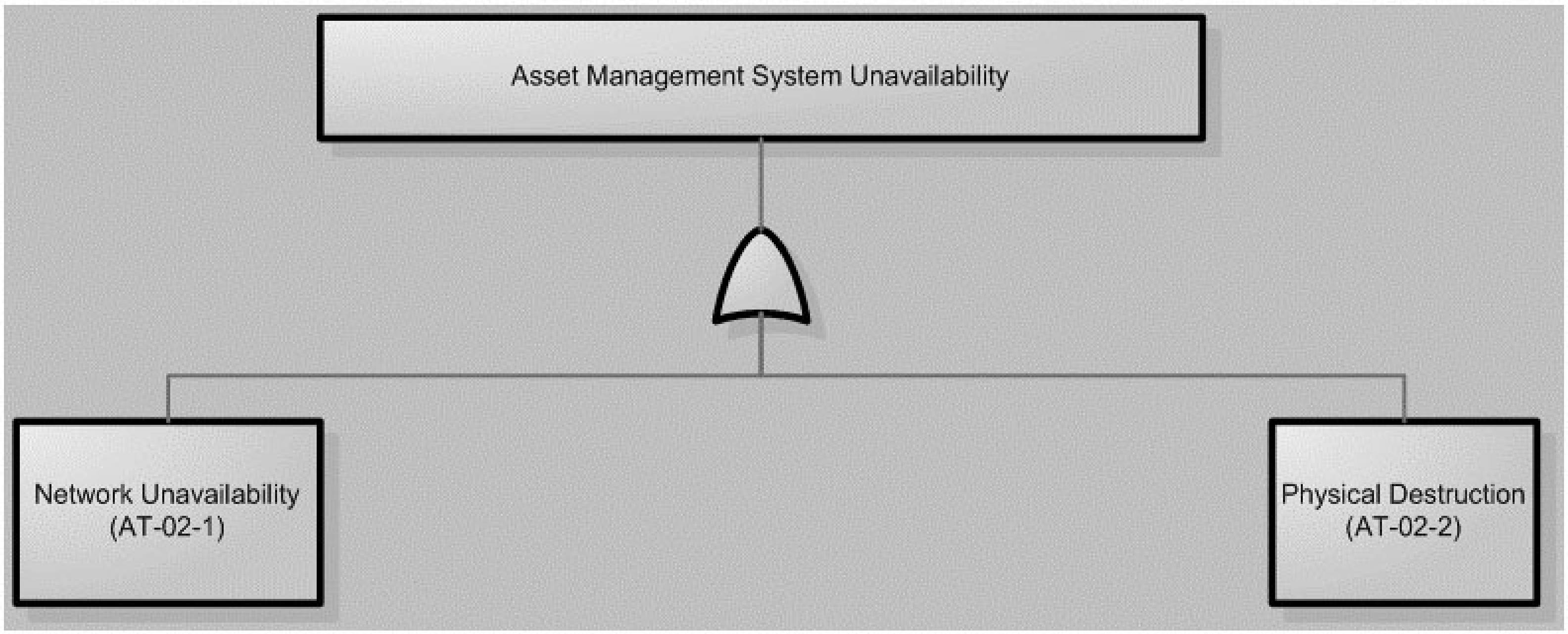

Asset Management System Unavailability
Network Unavailability (AT-02-1)
Physical Destruction (AT-02-2)

Figure 6: Attack Tree 01 – Gain Access (View/Modify/Delete) to Confidential Company Information (Higher Level View)

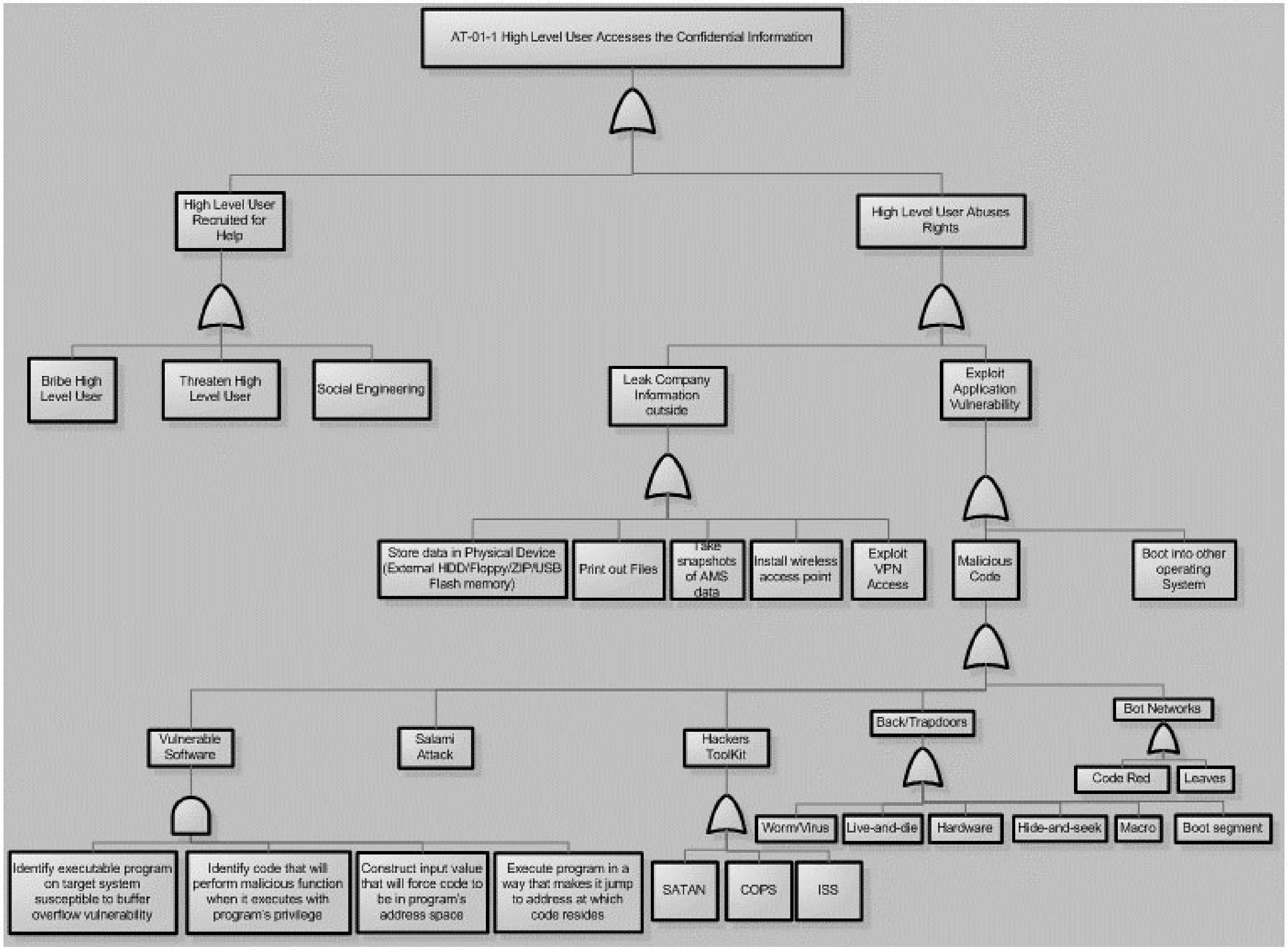

AT-01-1 High Level User Accesses the Confidential Information
High Level User Recruited for Help
High Level User Abuses Rights
Bribe High Level User
Threaten High Level User
Social Engineering
Leak Company Information outside
Exploit Application Vulnerability
Store data in Physical Device (External HDD/Floppy/ZIP/USB Flash memory)
Print out Files
Take snapshots of AMS data
Install wireless access point
Exploit VPN Access
Malicious Code
Boot into other operating System
Vulnerable Software
Salami Attack
Hackers ToolKit
Back/Trapdoors
Bot Networks
Code Red
Leaves
Worm/Virus
Live-and-die
Hardware
Hide-and-seek
Macro
Boot segment
Identify executable program on target system susceptible to buffer overflow vulnerability
Identify code that will perform malicious function when it executes with program's privilege
Construct input value that will force code to be in program's address space
Execute program in a way that makes it jump to address at which code resides
SATAN
COPS
ISS

Figure 7: Attack Tree 02 – Asset Management System Unavailability (High-Level View)

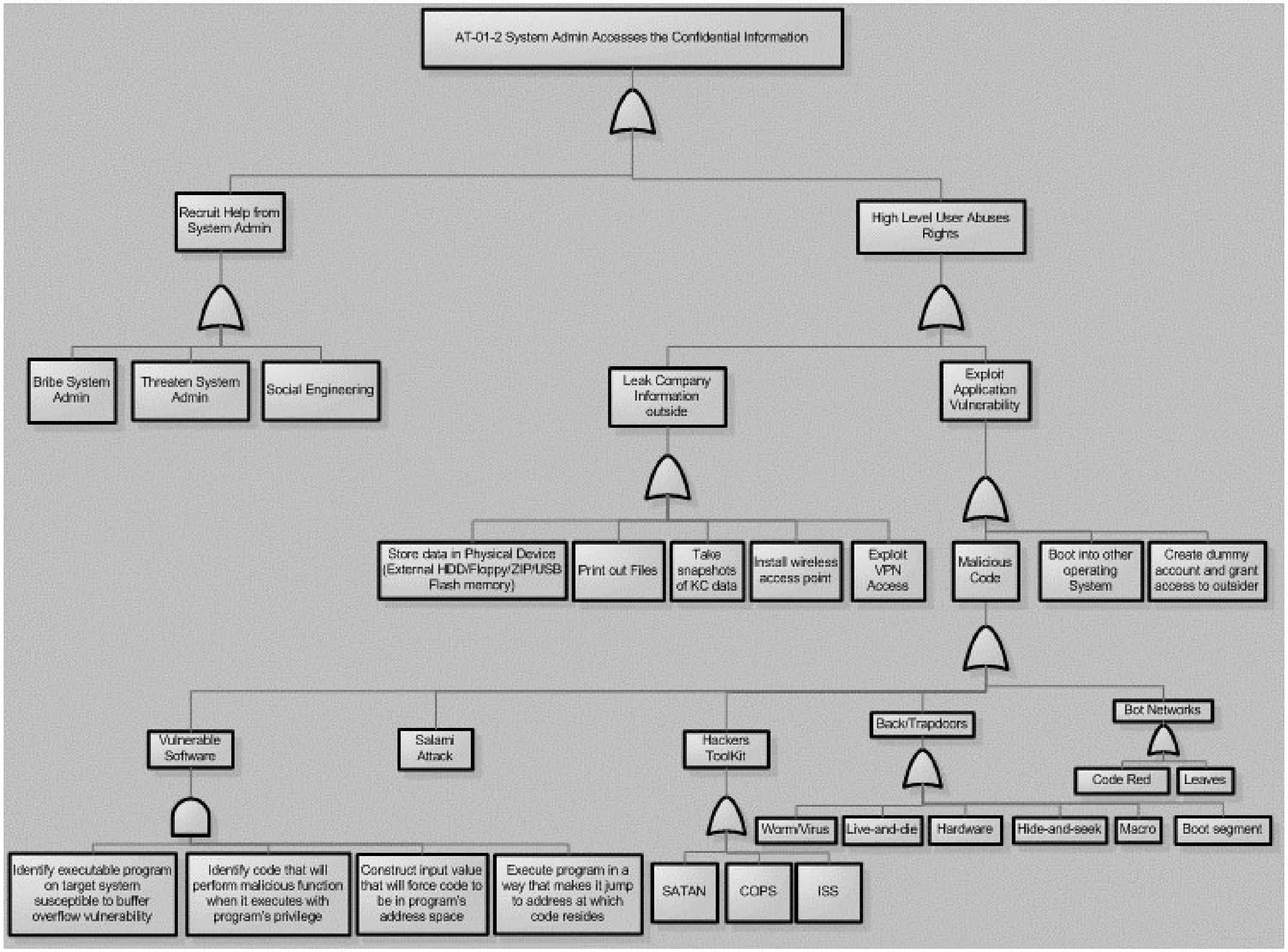

AT-01-2 System Admin Accesses the Confidential Information
Recruit Help from System Admin
High Level User Abuses Rights
Bribe System Admin
Threaten System Admin
Social Engineering
Leak Company Information outside
Exploit Application Vulnerability
Store data in Physical Device (External HDD/Floppy/ZIP/USB Flash memory)
Print out Files
Take snapshots of KC data
Install wireless access point
Exploit VPN Access
Malicious Code
Boot into other operating System
Create dummy account and grant access to outsider
Vulnerable Software
Salami Attack
Hackers ToolKit
Back/Trapdoors
Bot Networks
Code Red
Leaves
Worm/Virus
Live-and-die
Hardware
Hide-and-seek
Macro
Boot segment
Identify executable program on target system susceptible to buffer overflow vulnerability
Identify code that will perform malicious function when it executes with program's privilege
Construct input value that will force code to be in program's address space
Execute program in a way that makes it jump to address at which code resides
SATAN
COPS
ISS

Figure 8: AT-01-2 – System Admin Accesses Confidential Information

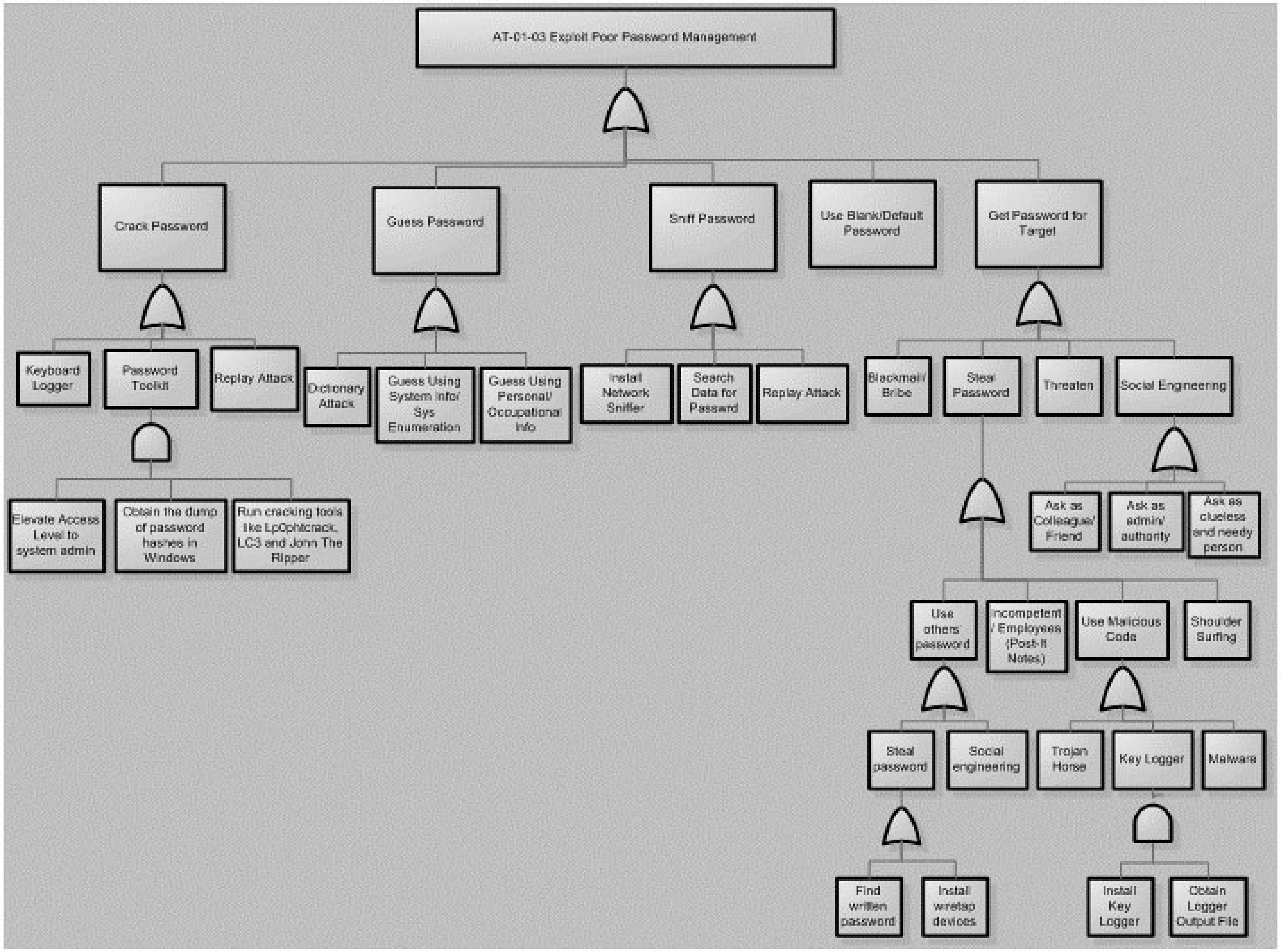


Figure 9: AT-01-03 – Exploit Poor Password Management

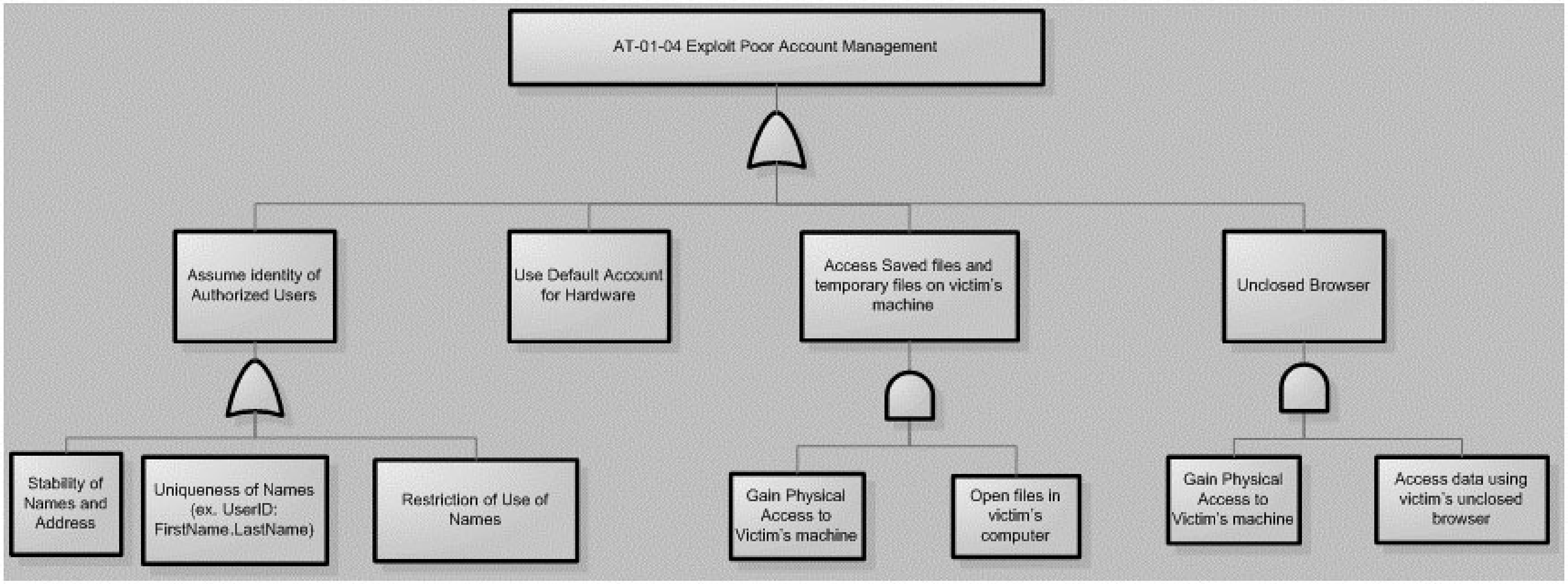

AT-01-04 Exploit Poor Account Management
Assume identity of Authorized Users
Use Default Account for Hardware
Access Saved files and temporary files on victim's machine
Unclosed Browser
Stability of Names and Address
Uniqueness of Names (ex. UserID: FirstName.LastName)
Restriction of Use of Names
Gain Physical Access to Victim's machine
Open files in victim's computer
Gain Physical Access to Victim's machine
Access data using victim's unclosed browser

Figure 10: AT-01-04 – Exploit Poor Account Management

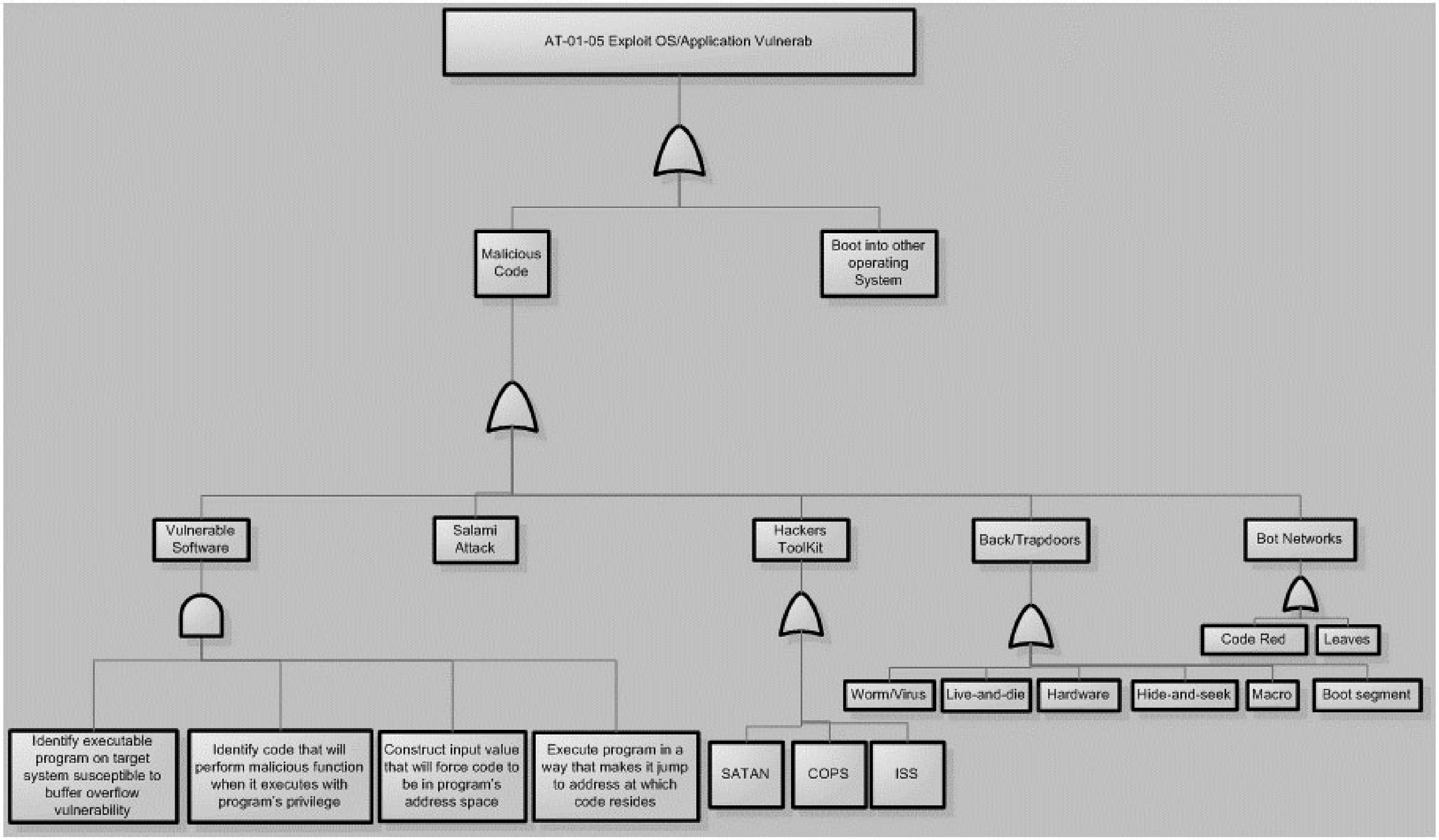

AT-01-05 Exploit OS/Application Vulnerab
Malicious Code
Boot into other operating System
Vulnerable Software
Salami Attack
Hackers ToolKit
Back/Trapdoors
Bot Networks
Identify executable program on target system susceptible to buffer overflow vulnerability
Identify code that will perform malicious function when it executes with program's privilege
Construct input value that will force code to be in program's address space
Execute program in a way that makes it jump to address at which code resides
SATAN
COPS
ISS
Worm/Virus
Live-and-die
Hardware
Hide-and-seek
Macro
Boot segment
Code Red
Leaves

Figure 11: AT-01-05 – Exploit OS/Application Vulnerability

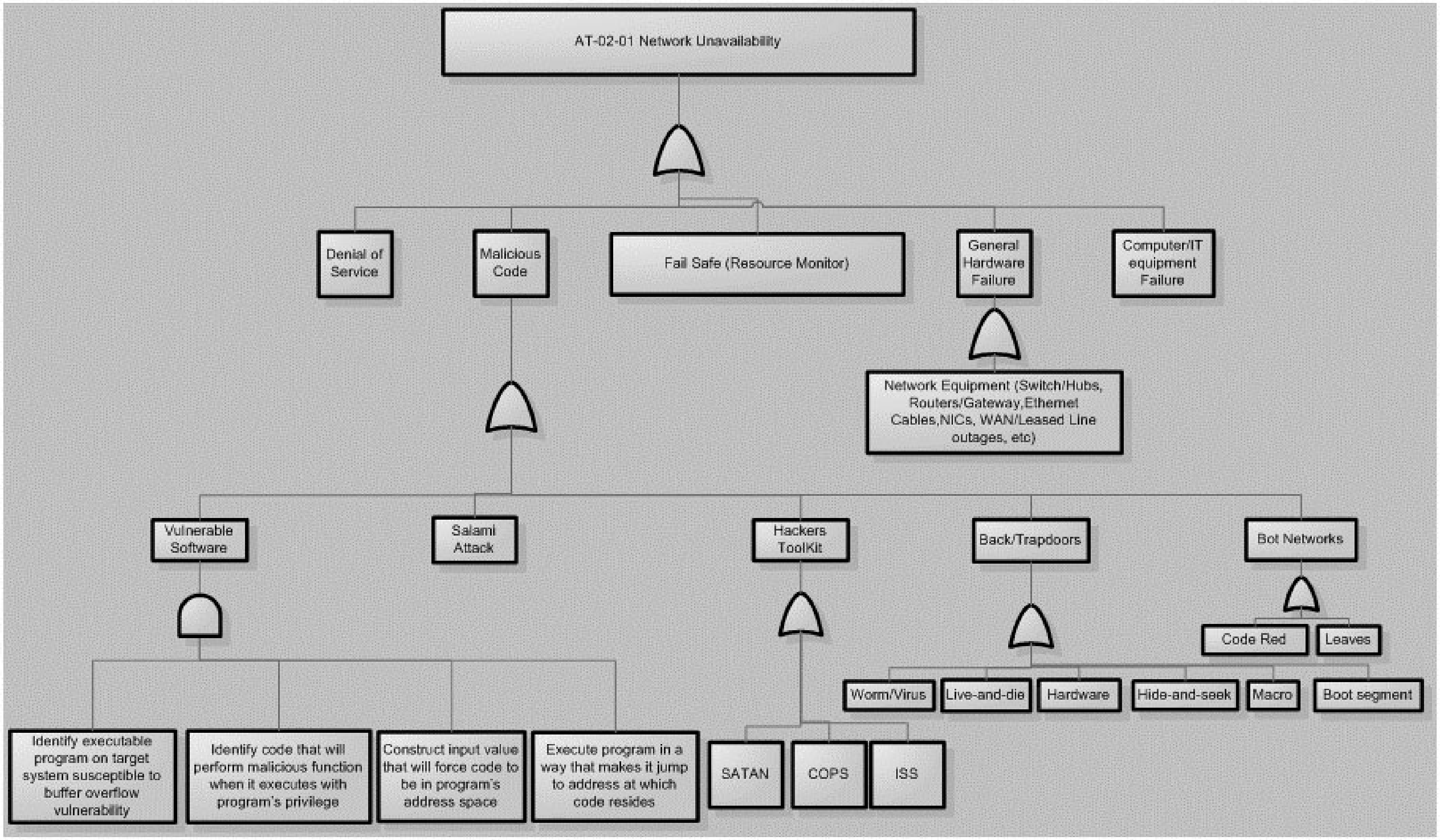

AT-02-01 Network Unavailability
Denial of Service
Malicious Code
Fail Safe (Resource Monitor)
General Hardware Failure
Computer/IT equipment Failure
Network Equipment (Switch/Hubs, Routers/Gateway,Ethernet Cables,NICs, WAN/Leased Line outages, etc)
Vulnerable Software
Salami Attack
Hackers ToolKit
Back/Trapdoors
Bot Networks
Code Red
Leaves
Worm/Virus
Live-and-die
Hardware
Hide-and-seek
Macro
Boot segment
Identify executable program on target system susceptible to buffer overflow vulnerability
Identify code that will perform malicious function when it executes with program's privilege
Construct input value that will force code to be in program's address space
Execute program in a way that makes it jump to address at which code resides
SATAN
COPS
ISS

Figure 12: AT-02-01– Network Unavailability

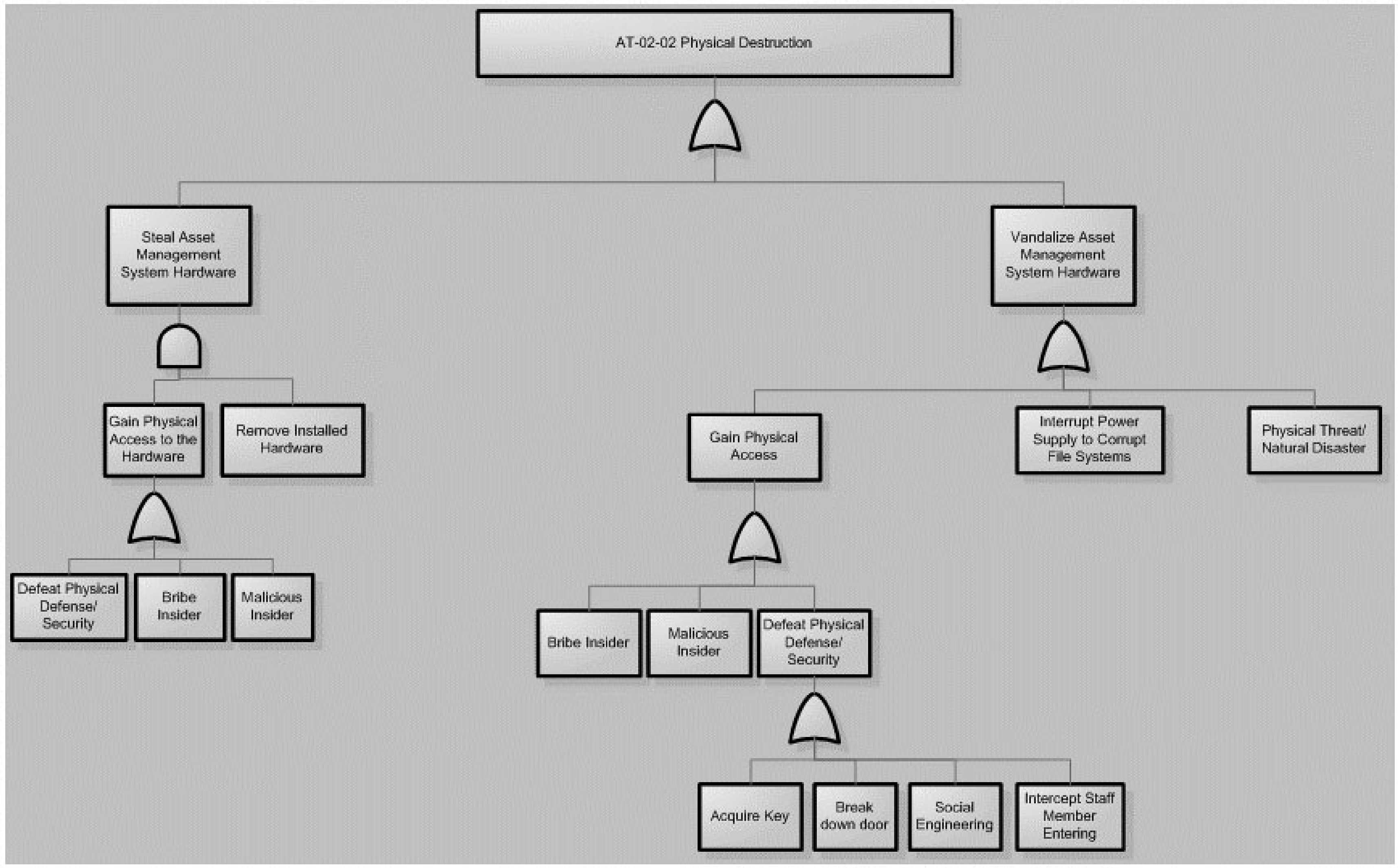

AT-02-02 Physical Destruction
Steal Asset Management System Hardware
Vandalize Asset Management System Hardware
Gain Physical Access to the Hardware
Remove Installed Hardware
Defeat Physical Defense/ Security
Bribe Insider
Malicious Insider
Gain Physical Access
Interrupt Power Supply to Corrupt File Systems
Physical Threat/ Natural Disaster
Bribe Insider
Malicious Insider
Defeat Physical Defense/ Security
Acquire Key
Break down door
Social Engineering
Intercept Staff Member Entering

Figure 13: AT-02-02 – Physical Destruction

Appendix B Use Case Artifacts

Overview

Use cases provide an outline of the system's functionality from a user's perspective, with classification of user level privileges by access control lists. They provide detailed steps for the various ways the Asset Management System can be accessed [Chen 04].

Diagram **Page**

Description of User Rights

User	Description of Rights
Low-Level User	View only
Medium-Level User	General AMS user with edit privileges (journal entries, mark-up floor plans for room status)
High-Level User	Archibus administrator at client site (edit database to add users to afm_users table, create links to EP procedures/docs, etc.)
System Administrator	IIS configuration, access controls, user accounts, etc.

Preconditions for all Asset Management System-based use cases:

Login:

- OS-based
- Unknown users are not permitted access to the Asset Management System website.

Figure 14: Use Case Legend

Number	UC-01
Use Case	View Floor Plans
Description	All level of users able to access the Asset Management System will have the ability to view authorized system information per the access control list such as floor plans, damaged areas, employee locator, etc.
Actors	Low-Level User, Medium-Level User, High-Level User, or System Administrator
Assumptions	1. System Admin has added viewing privileges to the access control list. 2. System is available. 3. Data entered is correct.
Steps	From here, the user will navigate to Operations/ Maintenance. Choose appropriate facility and then floor plans.
Variations	Once logged in, the user can also click on the floor plans tab on the right-hand side of the Asset Management System main page.
Non-Functional	They will not have edit privileges; view-only privileges will be assigned. If the user attempts to access unauthorized information, the system will display a pop up window stating that the user is not authorized to access this information.
Related Misuse Cases	MC-01, MC-08, MC-11, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21,MC-22

Figure 15: Use Case 1 – View Floor Plans

Number	UC-02
Use Case	Damage Assessment
Description	The medium-level Asset Management System 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	1. The user has proper edit privileges 2. The data entered is correct 3. The user has proper security privileges
Steps	1. Go to Floor Plans (Ref. UC-01). 2. Select Area Status to view the current condition. 3. Highlight the specific area for damage assessment. 4. From the drop-down menu, select the status you wish to assign to the room (Damaged, Destroyed, Inventory, Not Usable, Renovation, Construction). 5. Press Go. 6. To continue marking areas, select “Floor Plan” and choose another floor. Repeat Steps 4-5.
Variations	N/A
Non-Functional	N/A
Related Misuse Cases	MC-01, MC-06, MC-07, MC-08, MC-11, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21, MC-22

Figure 16: Use Case 2 – Damage Assessment

Number	UC-03
Use Case	Add/Delete/Edit Post-it Note
Description	Low-level users and higher will have the ability to add post-it notes to maps and floor plans.
Actors	Low-Level User, Medium-Level User, High-Level User, or System Administrator
Assumptions	The floor plan or map is available.
Steps	**Add** 1. Go to Floor Plans (Ref. UC-01). 2. On the Left Menu, click the **Add Post-it-Note** button. 3. Select a location on the map to place the post-it-note and click. 4. Enter the title and text of the note on the Post-it Notes window. 5. Click the Save button to persist the note. **Edit** 1. On the floor plan legend click the **Notes** checkbox to display the notes. 2. Double-click on the note to display the Post-it-Notes window. 3. Change the text of the note. OR 4. By clicking **Link to Existing Notes** on the Post-it-Notes window, link the new note to some existing notes. OR 5. Delete an existing link on the note by clicking **Delete link to this note**. **Delete** 1. Double-click the note to display the Post-it-Notes window. 2. Then click on the **Delete this Note** link.
Variations	
Non-Functional	
Related Misuse Cases	

Figure 17: Use Case 3 – Add/Delete/Edit Post-It Note

Number	UC- 04
Use Case	Find Specialized Employees
Description	The low-level user and higher users want to search for employees with a certain criterion.
Actors	Low-Level User, Medium-Level User, or High-Level User
Assumptions	The user has proper security privileges.
Steps	***Version 1*** 1. Select **Facility**. 2. Go to **Homepage**. 3. Select **Personnel Re-Call List**.
Variations	***Version 2*** 1. Select **Facility**. 2. Go to **Homepage**. 3. Under the **Business Continuity** heading, select **Personnel Call List**. ***Version 3*** 1. Select **Facility**. 2. Go to **Homepage**. 3. Select **Ad-Hoc Event Management**. 4. Select **Employee Locator**. 5. Select **Set Restriction**. 6. Add in filtering Information for query.
Non-Functional	N/A
Related Misuse Cases	MC-01, MC-08, MC-11, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21,MC-22

Figure 18: Use Case 4 – Find Specialized Employees

Number	UC-05
Use Case	Journal Entry
Description	Medium-level users and higher will have the ability to access the Asset Management System and journal entry privileges.
Actors	Medium-Level User, High-Level User, or System Administrator
Assumptions	This assumes that - System Admin has added viewing privileges to the access control list. - System is available. - Data entered is correct.
Steps	**Adding Entry** 1. Select **Daily Log**. 2. Select **Add Activity**. 3. Select the building through the drop-down menu. 4. Enter the **Activity Type**. 5. Add the **Respondent**. 6. Enter the **Description**. 7. Enter the **Comments**. 8. Save.
Variations	**Editing Entry** 1. Select **Daily Log**. 2. Select previous journal entry. 3. Click **Edit**. 4. Enter changes to entry. 5. Save.
Non-Functional	If the user attempts to access unauthorized information, the system will display a pop-up window stating that the user is not authorized to access this information.
Related Misuse Cases	MC-01, MC-06, MC-07, MC-08, MC-11, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21, MC-22

Figure 19: Use Case 5 – Journal Entry

Number	UC- 06
Use Case	Install the Asset Management System
Description	System Administrator wants to install Asset Management System on the network.
Actors	System Administrator
Assumptions	The System Admin has control over the network.
Steps	Steps for Pre-Determined Windows Server(s) 1. Install/confirm IIS. 2. Install/confirm MapGuide with MapGuide Author option. 3. Install/confirm database engine (Sybase, Microsoft SQL, Oracle). 4. Copy client database file to server (assuming that client database file was previously created and configured). 5. Configure ODBC System DSN and confirm connectivity to database. 6. Confirm that line in vbdefs.asp references the configured ODBC System DSN name. 7. Configure website in IIS: - Assign website name (ex. Asset Management System). - Associate with IP address assigned to server. - Do not allow anonymous access. - Specify Integrated Windows authentication. - Specify home directory path. - Specify default content page. - Allow access to asp in server extensions. - Add MapGuide server extension and allow access. - Create necessary virtual directories in IIS making sure that pathing matches code references. 8. Allow access to EP document repository folder to designated High-Level user. 9. Copy files to IIS server website and virtual directories. 10. Register Asset Management System website name in local DNS server(s) using IP address(es) assigned in IIS. Steps for Developmental Workstation(s) 1. Install/confirm Archibus-FM on Asset Management System developmental workstation. 2. Create project in Archibus-FM pointing to database installed on server. 3. Confirm connectivity between Archibus and database. 4. Confirm access to Archibus database according to the security level assigned in the afm_users table. 5. Install/confirm AutoCAD and configure with Archibus Overlay on Asset Management System developmental workstation. 6. Confirm connectivity between AutoCAD and Archibus project. 7. Install pre-configured SDF Loader program. 8. Confirm connectivity to the IIS server (ex. Ping server name). 9. Confirm connectivity to the Asset Management System website (ex. Ping website name). 10. Configure Internet Explorer settings for Intrasite security and Advanced security and settings. 11. Confirm access to the Asset Management System website using Internet Explorer browser.

Figure 20: Use Case 6 – Install the Asset Management System

	Steps for Asset Management System User Workstation(s) 1. Confirm connectivity to the IIS server (ex. Ping server name). 2. Confirm connectivity to the Asset Management System website (ex. Ping website name). 3. Configure Internet Explorer settings for Intrasite security and Advanced security and settings. 4. Confirm access to the Asset Management System website using Internet Explorer browser.
Variations	
Non-Functional	
Related Misuse Cases	MC-01, MC-02, MC-03, MC-04, MC-05, MC-08, MC-09, MC-10, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21, MC-22

Figure 20: Use Case 6 – Install the Asset Management System, cont.

Number	UC-07
Use Case	Create Links to the Documents
Description	High-Level users will have the ability to access the Asset Management System and create links to EP procedures/docs, etc.
Actors	High-Level User or System Administrator
Assumptions	This assumes that - System Admin has added write privileges to the access control list of the document repository folder. - System is available. - Data entered is correct.
Steps	1. User logs into developmental workstation with assigned network username and password. 2. The system authorizes and authenticates the user and then allows user into the system. 3. User enters data into Archibus-FM tables ‘ep_procedures’ and ‘ep_bl_doc_link’ to denote document path, document name, and related building. 4. User copies documents to IIS virtual directory designated as document repository whose path agrees with that entered in the above step. 5. User confirms that Asset Management System website function displays document listing and document correctly.
Variations	
Non-Functional	If the user attempts to access unauthorized information, the system will display a pop-up window stating that the user is not authorized to access this information. If the user attempts to access an unauthorized network folder, the user will be notified of insufficient privileges.
Related Misuse Cases	MC-01, MC-02, MC-03, MC-04, MC-05, MC-08, MC-09, MC-10, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21, MC-22

Figure 21: Use Case 7 – Create Links to the Documents

Number	UC- 08
Use Case	Archibus Administrator Adding a User and Assigning Privileges
Description	The Archibus Administrator adds a user to the afm_users table so that the user will have the ability to use the Asset Management System. The user must also assign the proper privileges associated with his or her user level.
Actors	Archibus Administrator
Assumptions	The Archibus Admin has the proper security privileges.
Steps	**Add Individual** 1. Open Archibus. 2. Select the project (in this case, it is Asset Management System 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** 1. Open Archibus. 2. Select Security Groups. 3. Add new record. 4. Add group name. 5. Add Description.
Variations	Go directly to the data through Archibus.
Non-Functional	No user password
Related Misuse Cases	MC-01, MC-02, MC-03, MC-04, MC-05, MC-08, MC-09, MC-10, MC-12, MC-13, MC-14, MC-15, MC-16, MC-17, MC-18, MC-19, MC-20, MC-21, MC-22

Figure 22: Use Case 8 – Archibus Administrator Adding a User and Assigning Privileges

Number	UC- 09
Use Case	View Contact Information for Maintenance Tasks
Description	The low-level user and higher users want to search for contact information for maintenance tasks.
Actors	Low-Level User, Medium-Level User, or High-Level User
Assumptions	The user has proper security privileges.
Steps	1. Go to the **Asset Management System Homepage**. 2. Go to **Maintenance Responsibility**. 3. Click on **Matrix**. 4. A new window titled **Maintenance Activity & POC Matrix** will appear. The **Maintenance Task List** is located on the left side of the matrix and the **Building List** is on top of the matrix. 5. Click on a building abbreviation with the **Building List** to open the **Facility Homepage**.
Variations	
Non-Functional	N/A
Related Misuse Cases	

Figure 23: Use Case 9 – View Contact Information for Maintenance Tasks

Number	UC- 10
Use Case	Create Open Space Report
Description	The low-level user and higher users want to create a report of the available open space in the facility.
Actors	Low-Level User, Medium-Level User, or High-Level User
Assumptions	The user has proper security privileges.
Steps	1. Go to the **Asset Management System Homepage**. 2. Under **Business Continuity**, click on the **Unoccupied Space Report** link. 3. The **Open Space Report** is displayed on the screen.
Variations	
Non-Functional	N/A
Related Misuse Cases	

Figure 24: Use Case 10 – Create Open Space Report

Number	UC-11
Use Case	View Incident Command
Description	The incident command lists the responsibilities of various employees in case of an emergency.
Actors	Low-Level User, Medium-Level User, High-Level User, or System Administrator
Assumptions	
Steps	1. Go to the **Continuity/Recovery** menu. 2. Click on **Planning**. 3. Further click on **Organization**. 4. Click on **Incident Command**.
Variations	1. Go to the **Continuity/Recovery** menu. 2. Click on **Response**. 3. Further click on **Organization**. 4. Click on **Incident Command**.
Non-Functional	
Related Misuse Cases	

Figure 25: Use Case 11 – View Incident Command

Appendix C   Use Case Diagram Artifacts

Overview

Use case diagrams provide a visual outline of systems architecture traces of use case scenarios. Important components and connectors that are accessed in each trace are highlighted to show which architectural elements are important and necessary for proper system functioning.

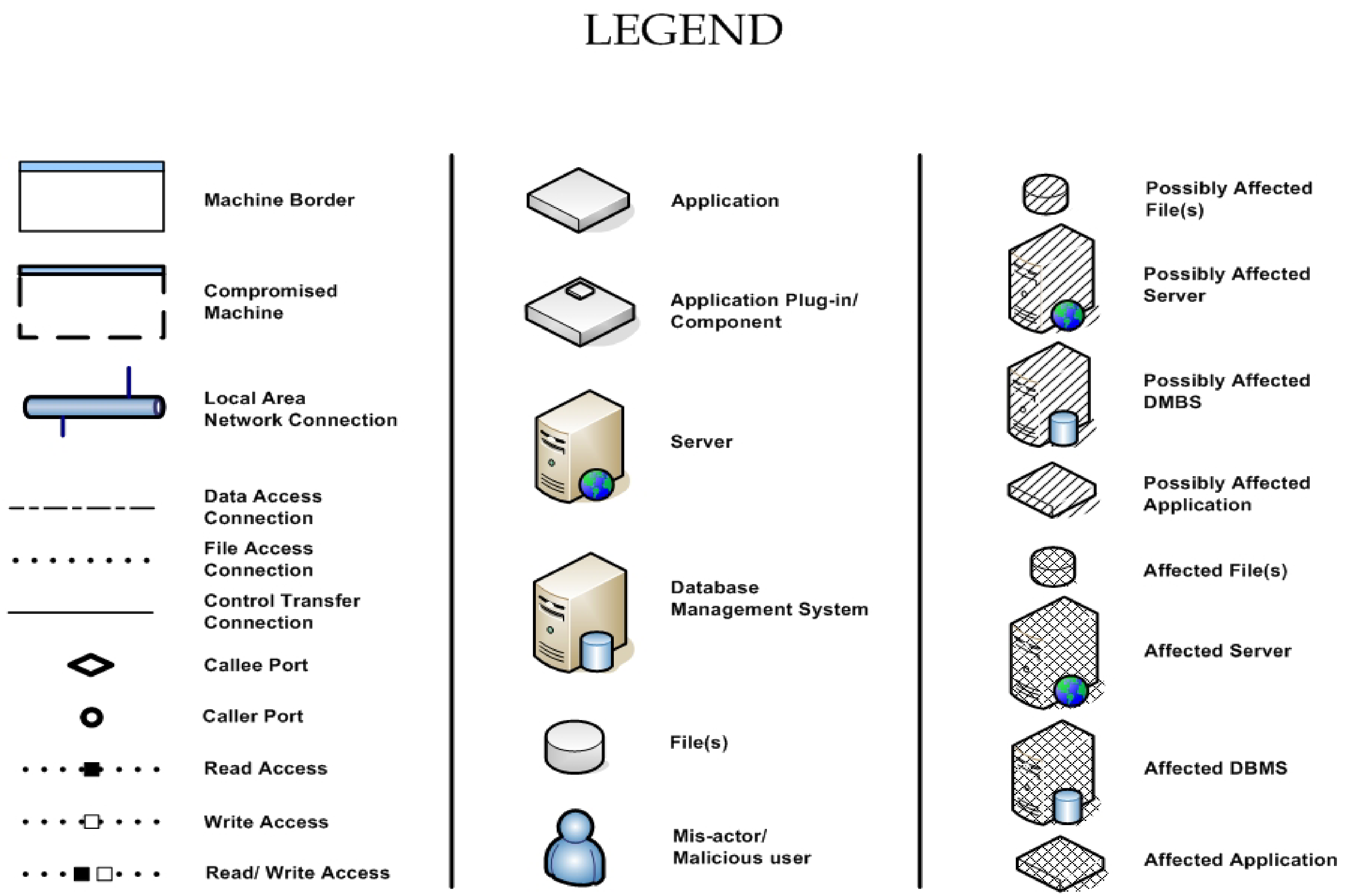


Figure 26: Use Case Diagram Legend

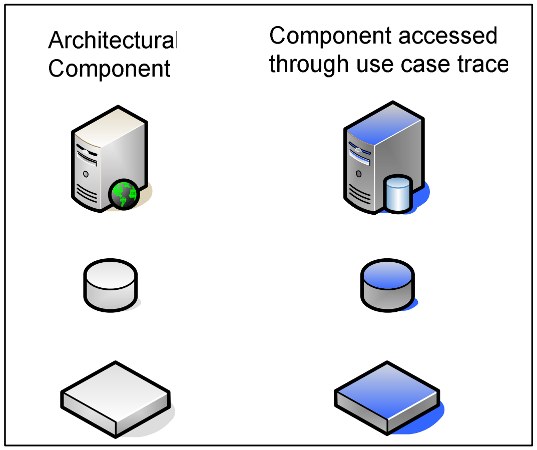


Figure 26, cont.

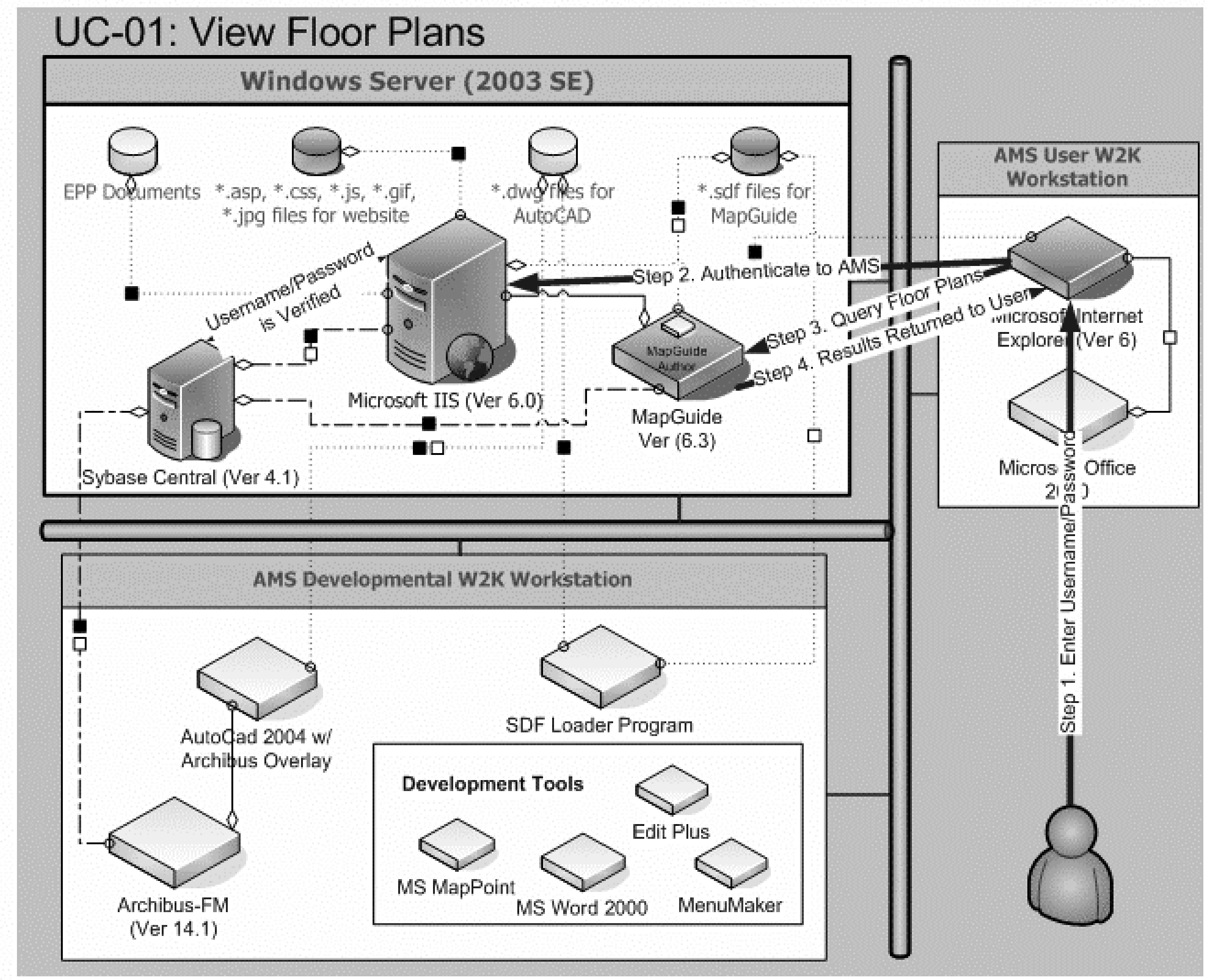


Figure 27: Use Case 1 – View Floor Plans

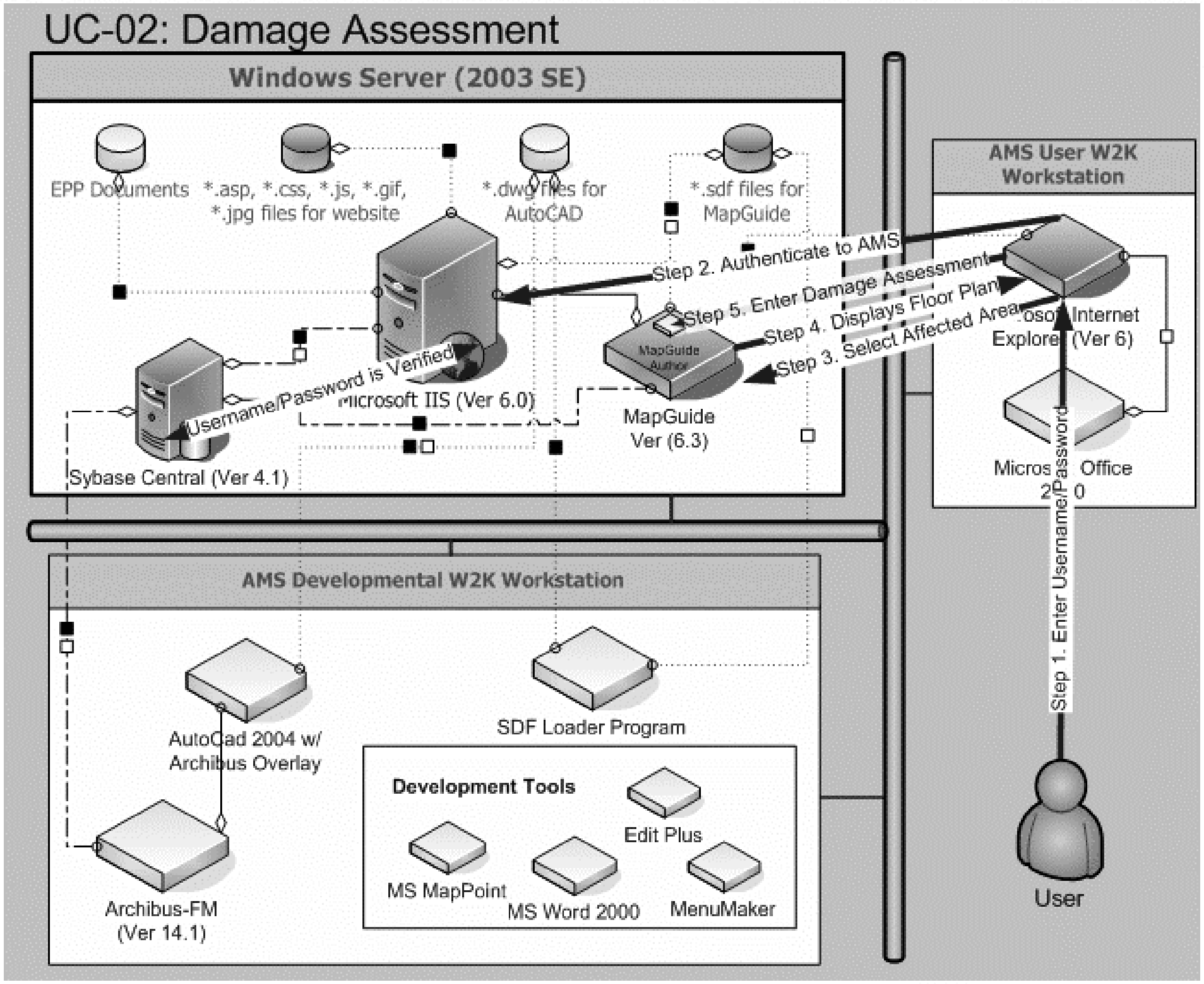


Figure 28: Use Case 2 – Damage Assessment

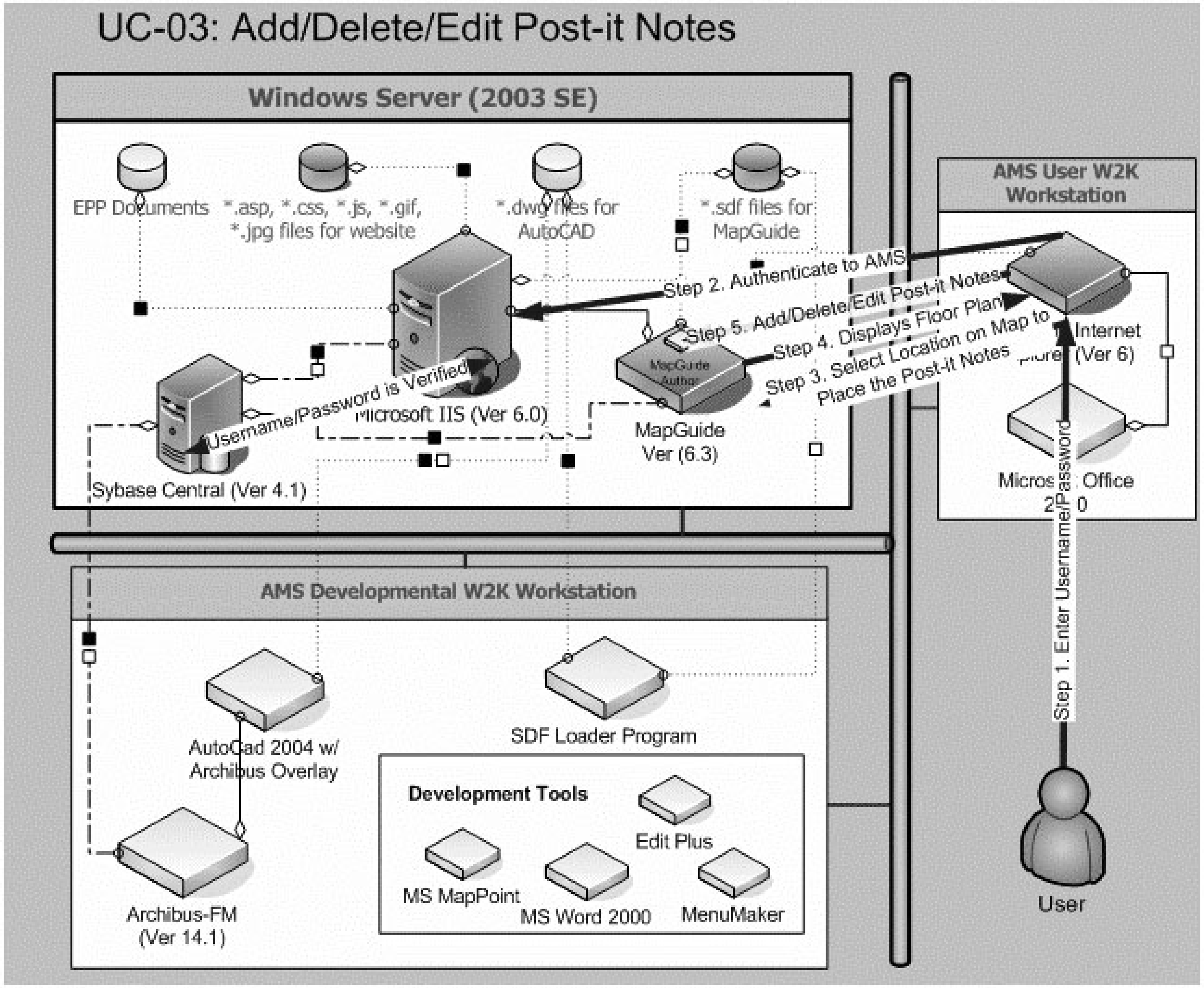

UC-03: Add/Delete/Edit Post-it Notes
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Microsoft IIS (Ver 6.0)
Sybase Central (Ver 4.1)
MapGuide Author
MapGuide Ver (6.3)
Step 1. Enter Username/Password
Step 2. Authenticate to AMS
Username/Password is Verified
Step 3. Select Location on Map to Place the Post-it Notes
Step 4. Displays Floor Plan
Step 5. Add/Delete/Edit Post-it Notes
AMS User W2K Workstation
Internet Explorer (Ver 6)
Microsoft Office 2000
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
Archibus-FM (Ver 14.1)
SDF Loader Program
Development Tools
MS MapPoint
MS Word 2000
Edit Plus
MenuMaker

Figure 29: Use Case 3 – Add/Delete/Edit Post-It Note

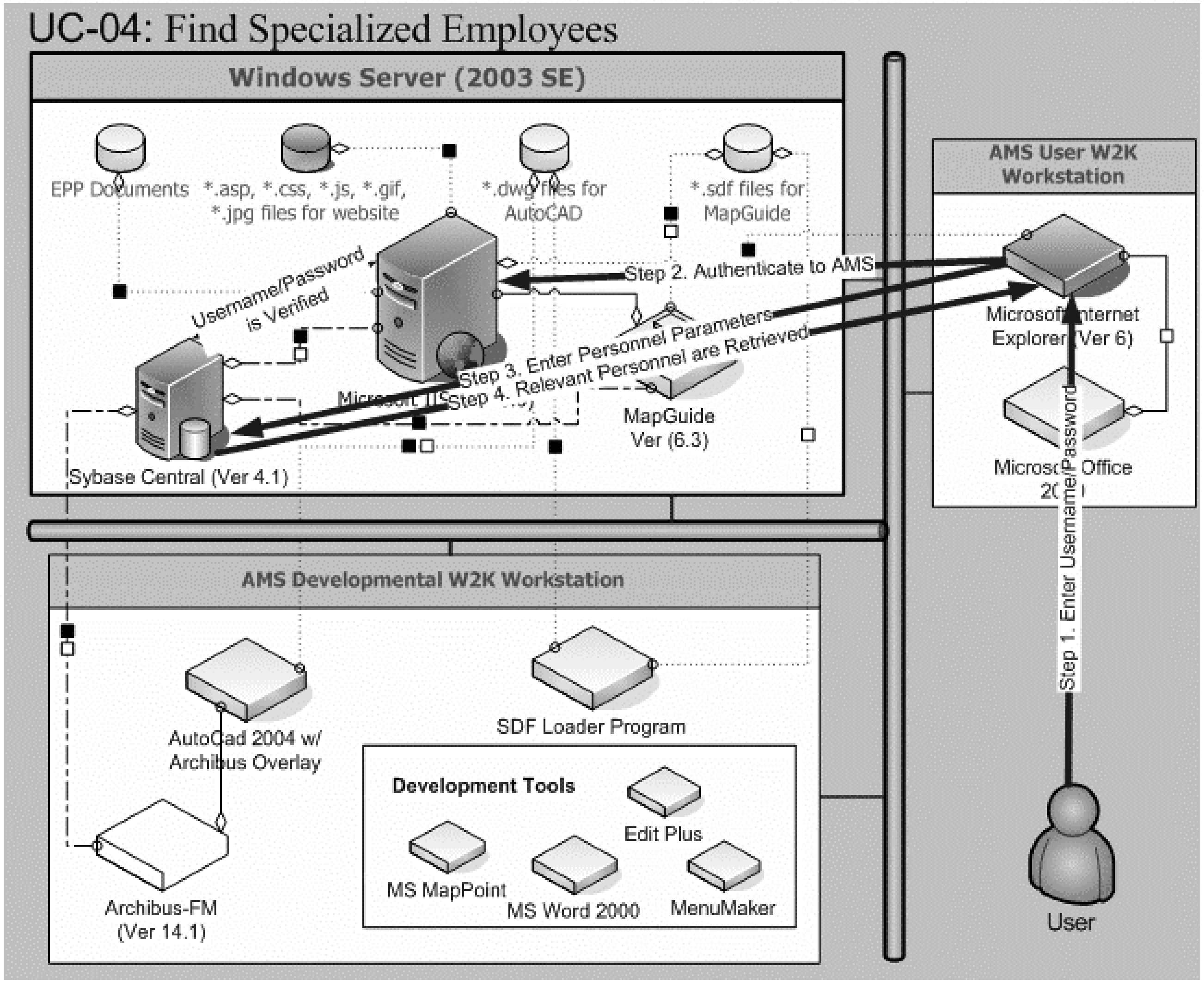

UC-04: Find Specialized Employees
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Username/Password is Verified
Microsoft IIS
Sybase Central (Ver 4.1)
MapGuide Ver (6.3)
Step 2. Authenticate to AMS
Step 3. Enter Personnel Parameters
Step 4. Relevant Personnel are Retrieved
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
Step 1. Enter Username/Password
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Archibus-FM (Ver 14.1)
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker

Figure 30: Use Case 4 – Find Specialized Employees

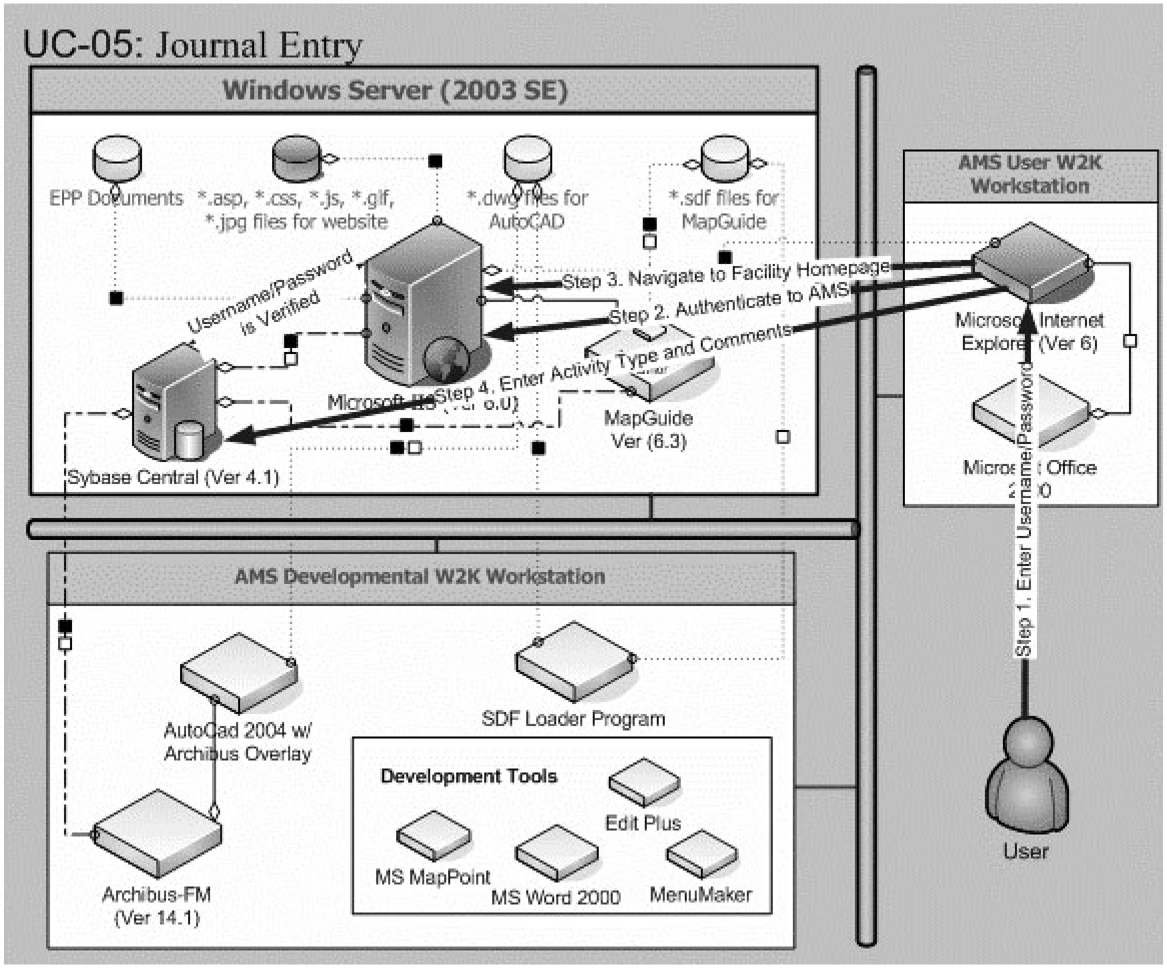
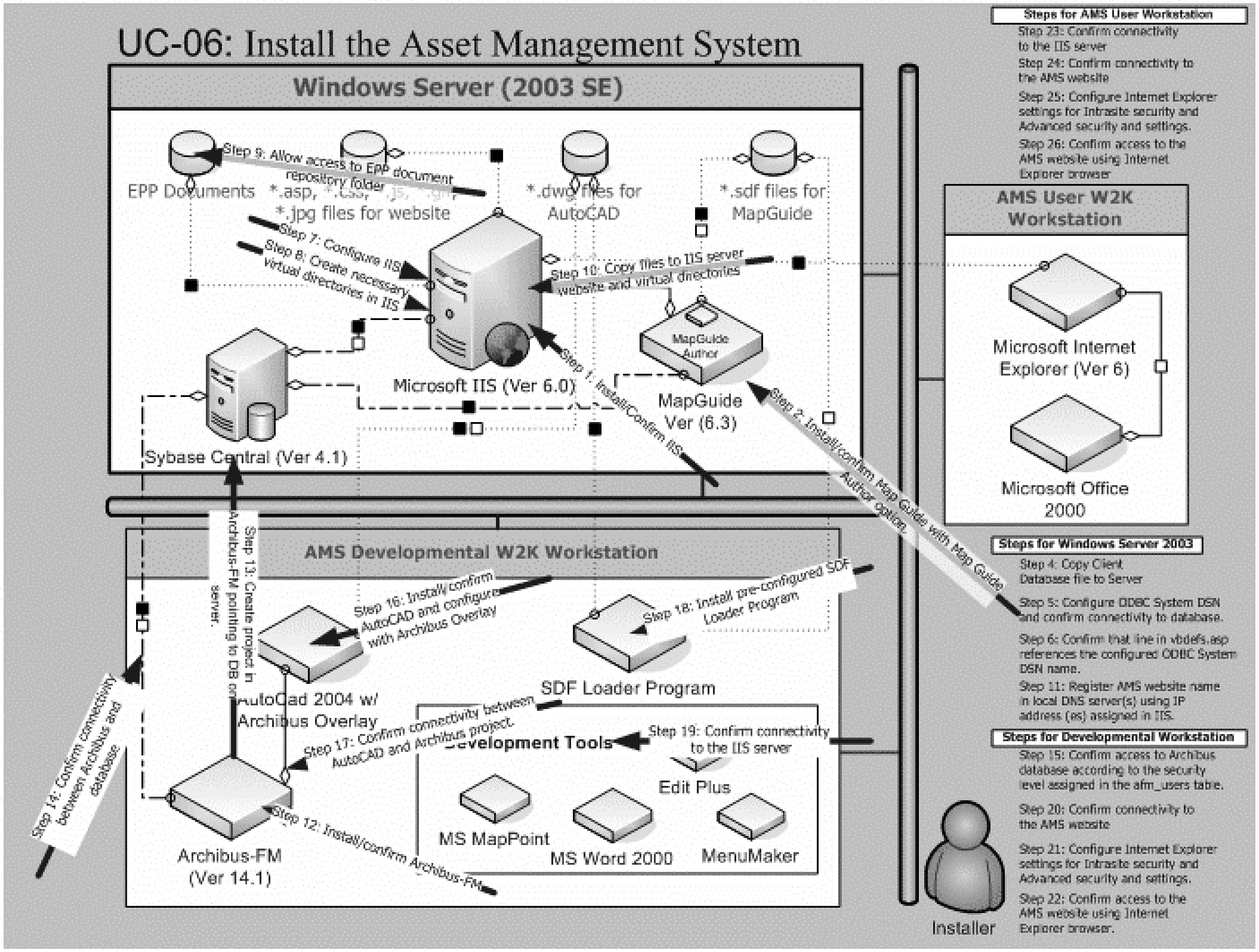

UC-05: Journal Entry
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Username/Password is Verified
Microsoft IIS
Sybase Central (Ver 4.1)
MapGuide Ver (6.3)
Step 3. Navigate to Facility Homepage
Step 2. Authenticate to AMS
Step 4. Enter Activity Type and Comments
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
Step 1. Enter Username/Password
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Archibus-FM (Ver 14.1)
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker

Figure 31: Use Case 5 – Journal Entry


UC-06: Install the Asset Management System
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Microsoft IIS (Ver 6.0)
MapGuide Author
MapGuide Ver (6.3)
Sybase Central (Ver 4.1)
Step 9: Allow access to EPP document repository folder
Step 7: Configure IIS
Step 8: Create necessary virtual directories in IIS
Step 10: Copy files to IIS server website and virtual directories
Step 1: Install/Confirm IIS
Step 2: Install/confirm Map Guide with Map Guide Author option.
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
AMS Developmental W2K Workstation
Step 13: Create project in Archibus-FM pointing to DB on server.
Step 16: Install/confirm AutoCAD and configure with Archibus Overlay
Step 18: Install pre-configured SDF Loader Program
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Step 17: Confirm connectivity between AutoCAD and Archibus project.
Development Tools
Step 19: Confirm connectivity to the IIS server
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker
Archibus-FM (Ver 14.1)
Step 12: Install/confirm Archibus-FM
Step 14: Confirm connectivity between Archibus and database
Installer

Steps for AMS User Workstation

Step 23: Confirm connectivity to the IIS server

Step 24: Confirm connectivity to the AMS website

Step 25: Configure Internet Explorer settings for Intrasite security and Advanced security and settings.

Step 26: Confirm access to the AMS website using Internet Explorer browser

Steps for Windows Server 2003

Step 4: Copy Client Database file to Server

Step 5: Configure ODBC System DSN and confirm connectivity to database.

Step 6: Confirm that line in vbdefs.asp references the configured ODBC System DSN name.

Step 11: Register AMS website name in local DNS server(s) using IP address (es) assigned in IIS.

Steps for Developmental Workstation

Step 15: Confirm access to Archibus database according to the security level assigned in the afm_users table.

Step 20: Confirm connectivity to the AMS website

Step 21: Configure Internet Explorer settings for Intrasite security and Advanced security and settings.

Step 22: Confirm access to the AMS website using Internet Explorer browser.

Figure 32: Use Case 6 – Install the Asset Management System

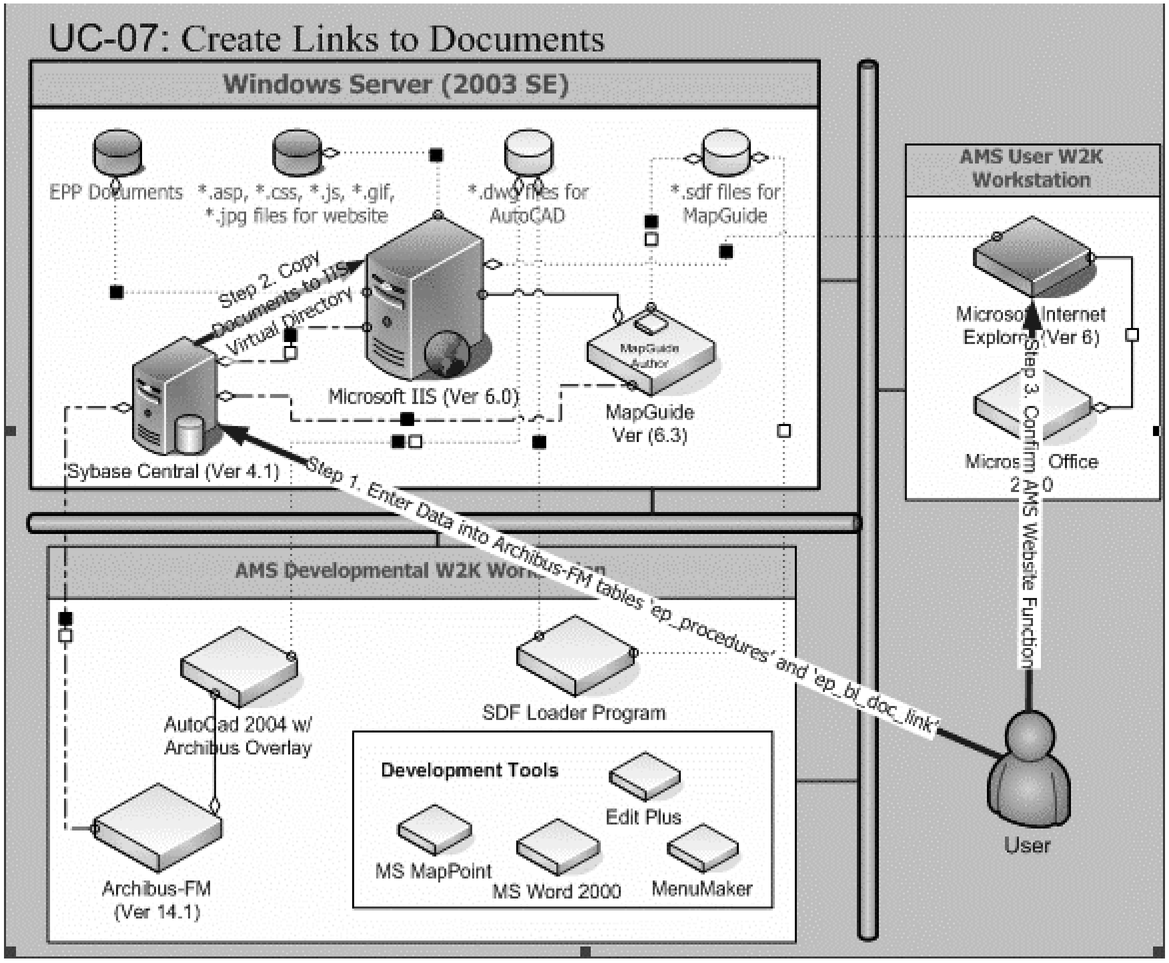

UC-07: Create Links to Documents
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Step 2. Copy Documents to IIS Virtual Directory
Microsoft IIS (Ver 6.0)
MapGuide Author
MapGuide Ver (6.3)
Sybase Central (Ver 4.1)
Step 1. Enter Data into Archibus-FM tables 'ep_procedures' and 'ep_bl_doc_link'
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker
Archibus-FM (Ver 14.1)
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Step 3. Confirm AMS Website Function
Microsoft Office 2000
User

Figure 33: Use Case 7 – Create Links to the Documents

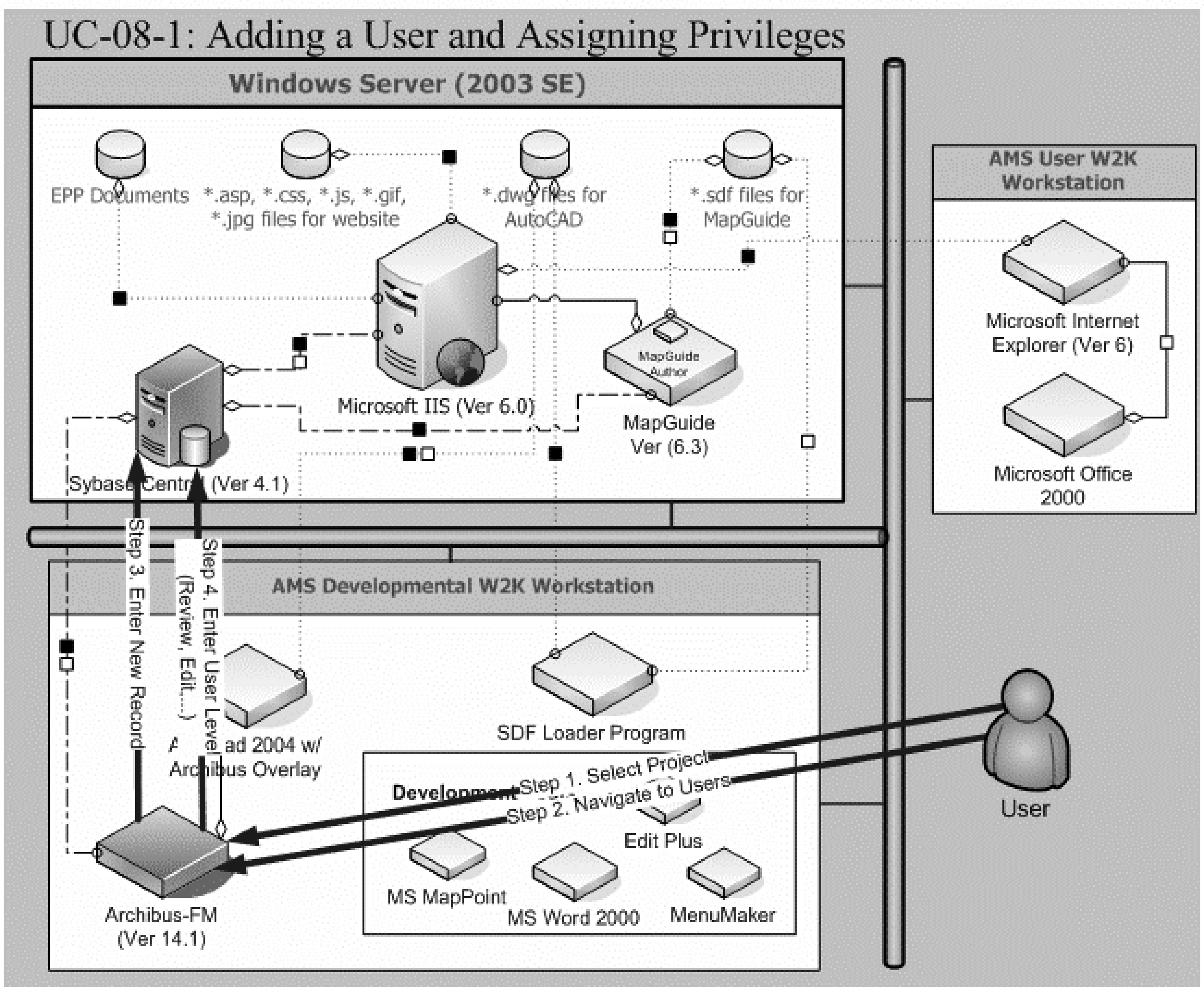


Figure 34: Use Case 8-1 – Adding a User and Assigning Privileges

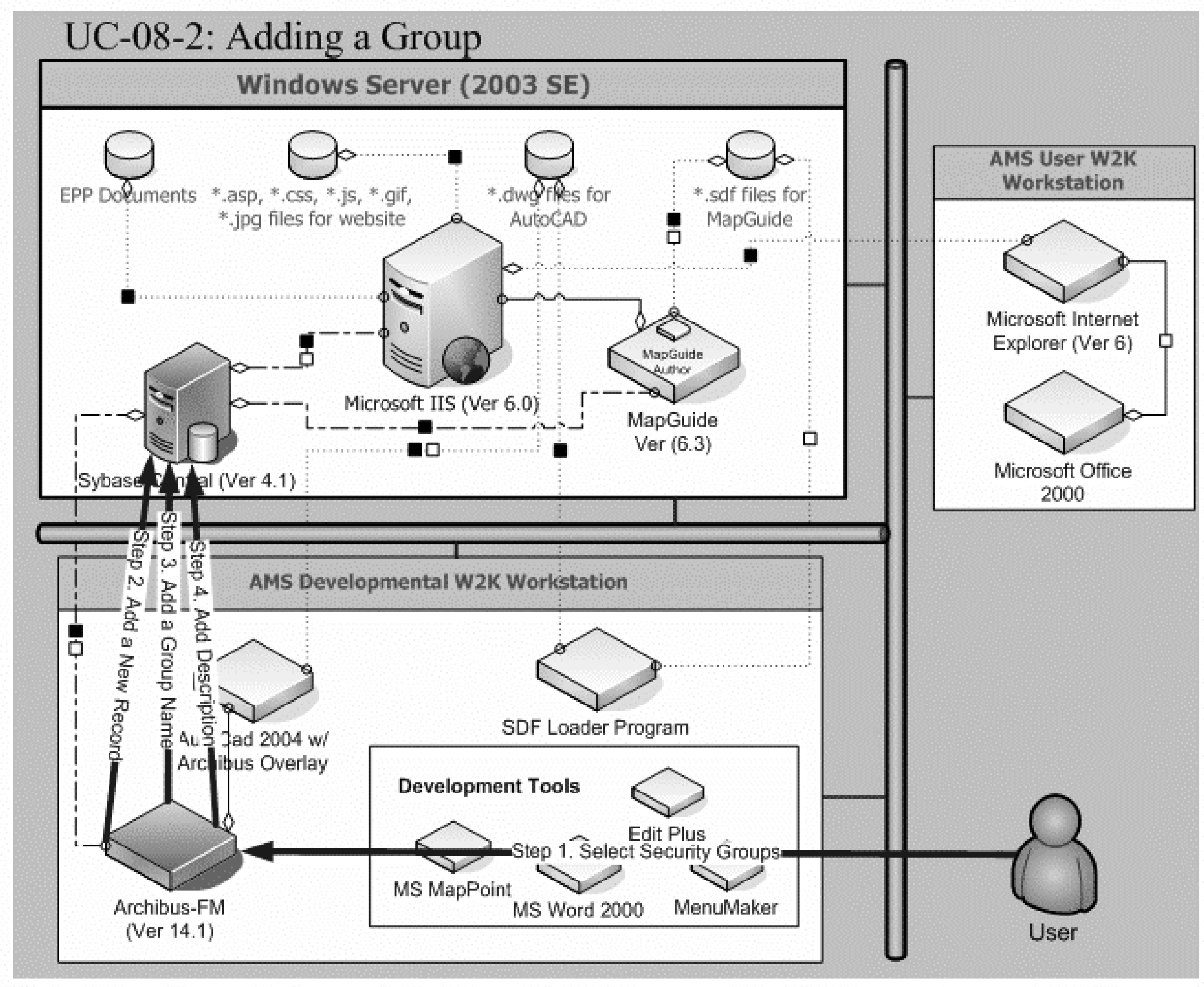


Figure 35: Use Case 8-2 – Adding a Group

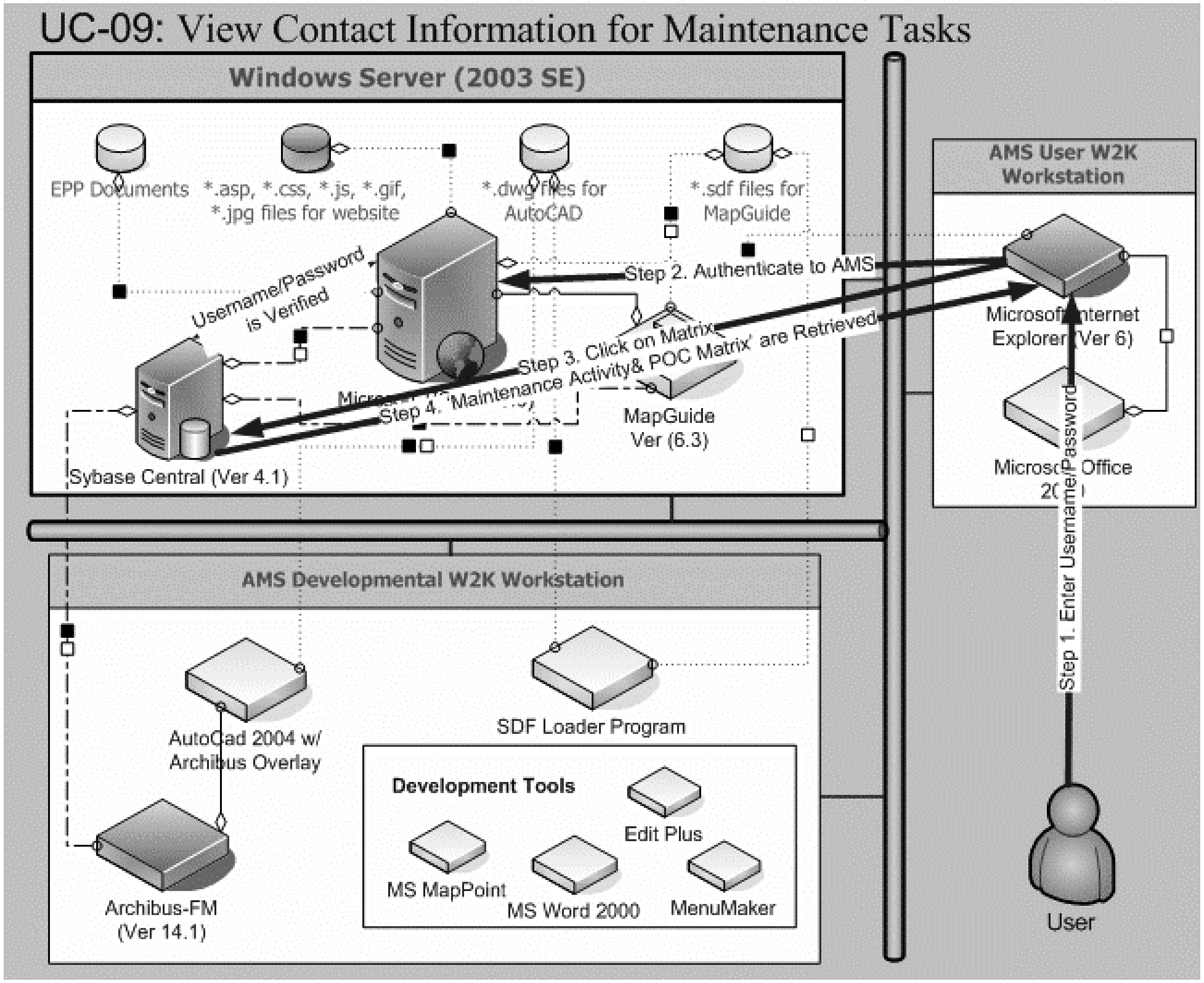

UC-09: View Contact Information for Maintenance Tasks
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Username/Password is Verified
Step 2. Authenticate to AMS
Step 3. Click on Matrix
Step 4. 'Maintenance Activity& POC Matrix' are Retrieved
MapGuide Ver (6.3)
Sybase Central (Ver 4.1)
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
Step 1. Enter Username/Password
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Archibus-FM (Ver 14.1)
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker

Figure 36: Use Case 9 – View Contact Information for Maintenance Tasks

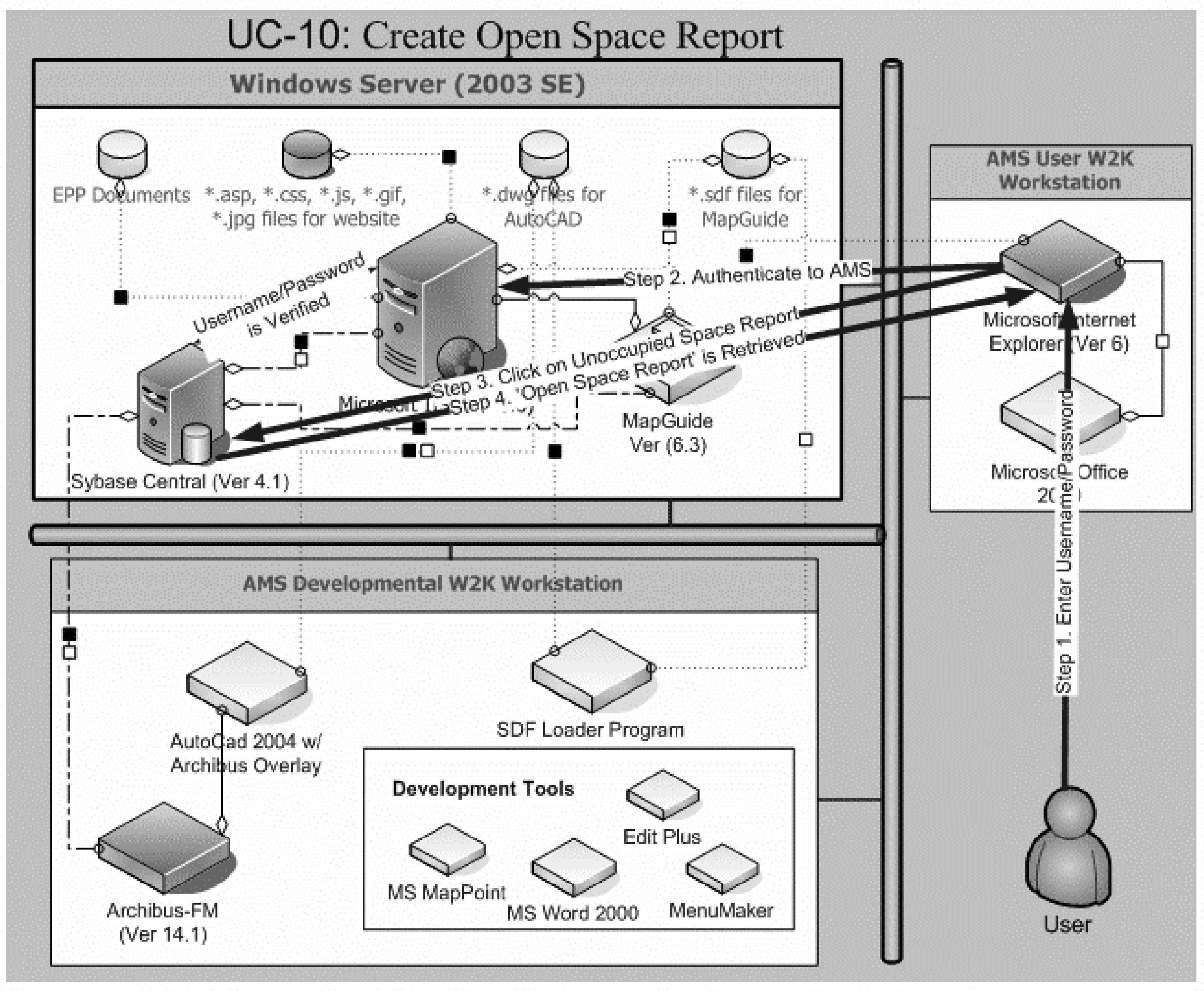

UC-10: Create Open Space Report
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Username/Password is Verified
Step 2. Authenticate to AMS
Step 3. Click on Unoccupied Space Report
Step 4. 'Open Space Report' is Retrieved
Microsoft
MapGuide Ver (6.3)
Sybase Central (Ver 4.1)
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
Step 1. Enter Username/Password
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker
Archibus-FM (Ver 14.1)

Figure 37: Use Case 10 – Create Open Space Report

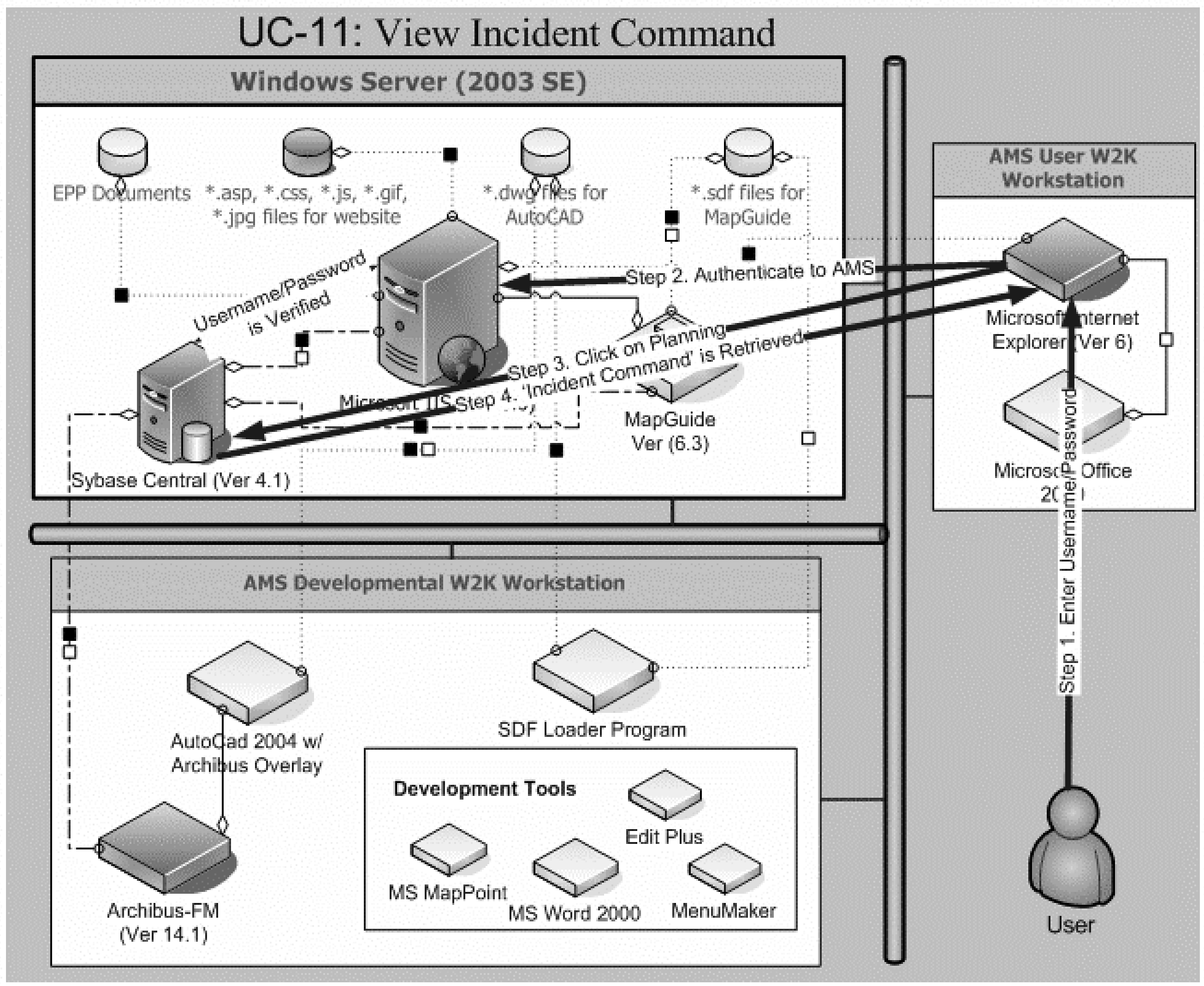

UC-11: View Incident Command
Windows Server (2003 SE)
EPP Documents
*.asp, *.css, *.js, *.gif, *.jpg files for website
*.dwg files for AutoCAD
*.sdf files for MapGuide
Username/Password is Verified
Step 2. Authenticate to AMS
Step 3. Click on Planning
Step 4. 'Incident Command' is Retrieved
Microsoft IIS
Sybase Central (Ver 4.1)
MapGuide Ver (6.3)
AMS User W2K Workstation
Microsoft Internet Explorer (Ver 6)
Microsoft Office 2000
Step 1. Enter Username/Password
User
AMS Developmental W2K Workstation
AutoCad 2004 w/ Archibus Overlay
SDF Loader Program
Development Tools
Edit Plus
MS MapPoint
MS Word 2000
MenuMaker
Archibus-FM (Ver 14.1)

Figure 38: Use Case 11 – View Incident Command

Appendix D Essential Components Diagram

Overview

The essential components of a system are those that support the essential services and assets of a system. By following the use case diagram traces that are marked essential, we can determine the total list of essential components. These are highlighted in the architectural drawing.

Diagram **Page**

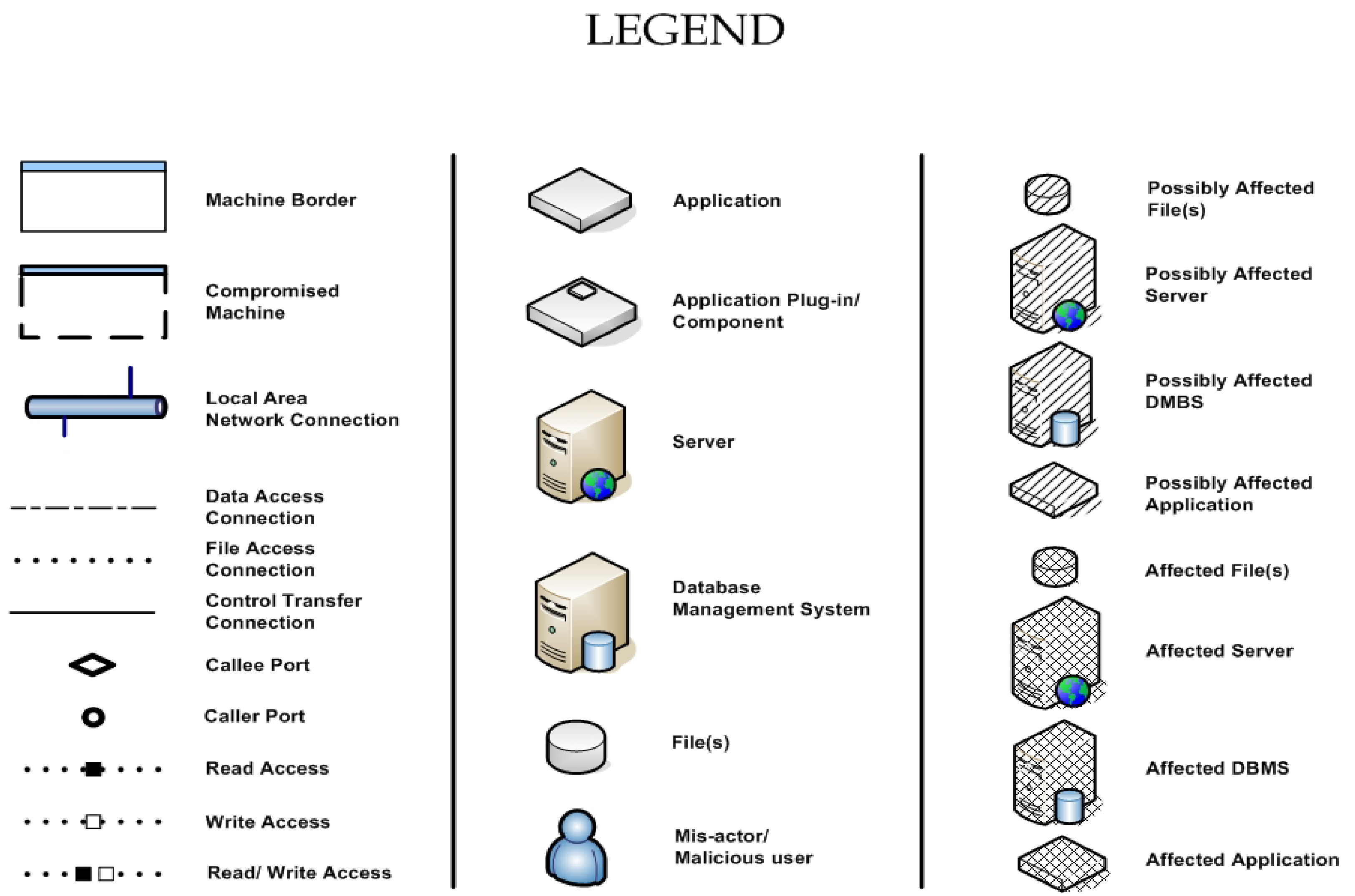


Figure 39: Essential Components Diagram Legend

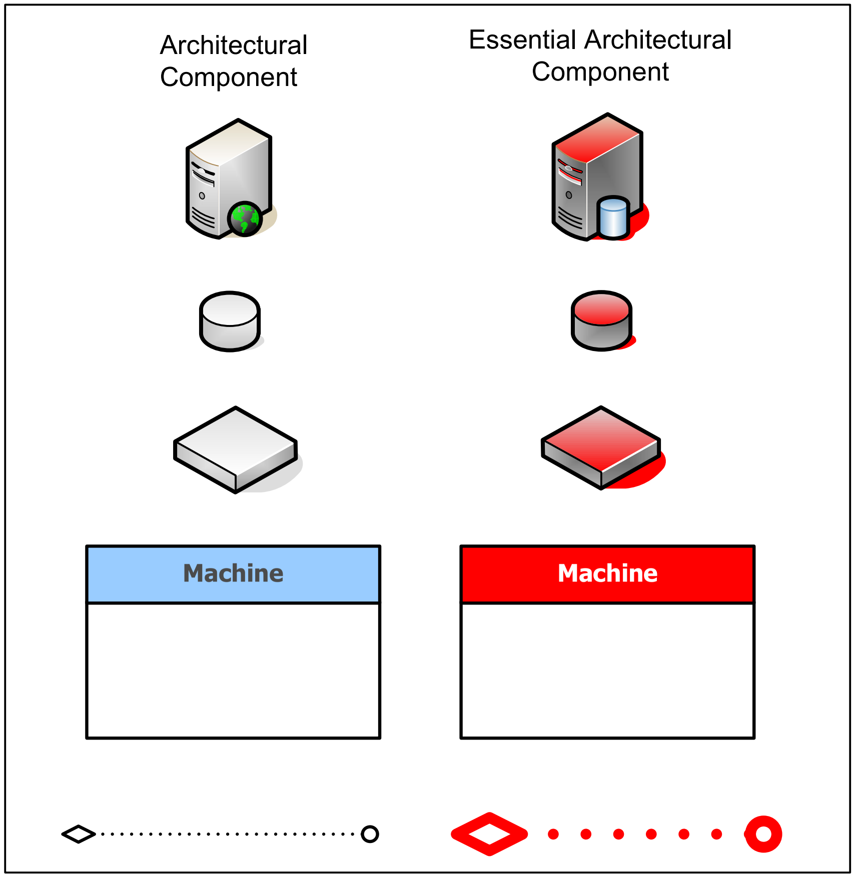


Figure 40: Essential Components Diagram

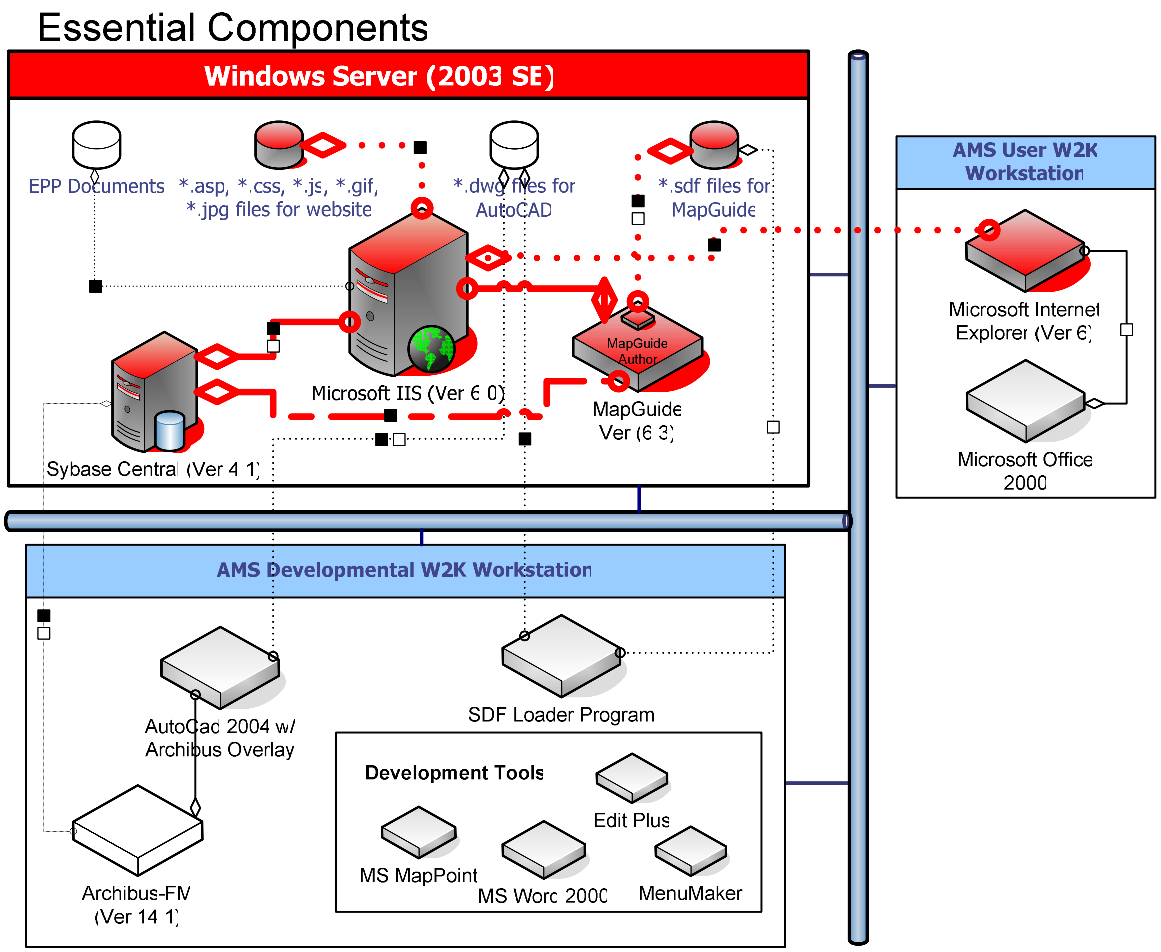


Figure 41, cont.

Appendix E SQUARE: Suggested Seven-Step Process

Overview

The SQUARE research project was delivered in a nine-step process. In investigating better methods for developing safety and security requirements, we have reorganized the process into a seven-step process. The first three steps remain completely unchanged, as this SQUARE team did not work in those areas. The four remaining steps are reorganized for clarity, contain new names, and include our new suggestions for inputs, methods, and outputs. Changes from the original nine-step process are marked in bold font.

Step #	Name	Input	Methods	Participants	Output
1	Agree on Definitions	Candidate definitions from the IEEE and other standards	Structured interviews, focus group	Stakeholders, requirements engineer	Agreed to definitions
2	Identify Safety and Security Goals	Definitions, candidate goals, business drivers, policies & procedures, examples	Facilitated work session, surveys, interviews	Stakeholders, requirements engineer	Goals
3	Select Elicitation Techniques	Goals, definitions, candidate techniques	Work session, considering expertise of stakeholders, organizational style, culture, level of safety & security needed, cost benefit analysis, etc.	Requirements engineer	Selected elicitation techniques
4	**Develop Artifacts**	Selected Techniques	Work session, **TO BE CONSIDERED: Yacov Haimes RFRM Model - Phases I and II, V-RATE**	Requirements engineer	Needed artifacts: **Architectural Diagrams**, Use Case Scenarios and Traces, Misuse Case Scenarios and Traces, **Attack Trees**, **Essential Services and Asset Identification**, templates, forms
5	**Perform Risk Assessment**	**Artifacts, specifically Attack Trees, Misuse Cases, and Essential Services and Asset Identification,** target operational environment **(if available)**	Risk assessment method, analysis of anticipated risk against organizational risk tolerance: **Yacov Haimes's RFRM Phases III and IV, NIST's (SP) 800-30 Risk Management Methodology - Steps 2 thru 7, SAEM (only when good quantitative data is available)**	Requirements engineer, risk expert, stakeholders	Risk Assessment Results, ~~**added**~~ mitigation requirements to bring exposure into acceptable level
6	**Develop Security Requirements Document**	**Goals, Artifacts, selected techniques, risk assessment results**	JAD, Interviews, Surveys, Work Session, Prioritization methods such as Triage, Win-Win	Stakeholders facilitated by requirements engineer	Prioritized **Security** Requirements **Document**
7	**Conduct Requirements Inspection**	Prioritized **Security** Requirements **Document**, formal inspection technique	Inspection method, such as Fagan, peer reviews, etc.	Inspection team	Initial requirements, documentation of decision making process and rationale

References

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

[Butler 02] Butler, Shawn. "Security Attribute Evaluation Method: A Cost-Benefit Approach," 232-240. *Proceedings of the 24th International Conference on Software Engineering*. Orlando, FL, May 19-25, 2002. New York NY: ACM Press, 2002.

[CERT/CC 02] CERT Coordination Center. *Survivable Systems Analysis Method*. http://www.cert.org/archive/html/analysis-method.html (2002).

[Chen 04] Chen, P.; Dean, M.; Ojoko-Adams, D.; Osman, H.; Lopez, L.; & Xie, N. *Systems Quality Requirements Engineering (SQUARE) Methodology: Case Study on Asset Management System* (CMU/SEI-2004-SR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/publications/documents/04.reports/04sr015.html.

[Cornford 04] 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).

[Firesmith 04] Firesmith, Donald; Mead, Nancy R.; & Woody, Carol. *System Quality Requirements Engineering (SQUARE) Project*. http://www.cert.org/sse/square.html (2004).

[Haimes 04] Haimes, Yacov Y. *Risk Modeling, Assessment, and Management,* 2nd ed. Hoboken, NJ: John Wiley and Sons, Inc., 2004.

[IEEE 98] IEEE. *IEEE Guide for Developing System Requirements Specifications*. New York, NY: Institute of Electrical and Electronics Engineers, Inc., 1998.

[Le Vie 00] Le Vie, Donn Jr. *The Inspection Method: An Approach to Planning and Managing a Successful Team Document Review*. http://www.techwr-l.com/techwhirl/magazine/writing/inspectionmethod.html (2000).

[Lipson 01] Lipson, Howard F.; Mead, Nancy R.; & Moore, Andrew P. *A Risk-Management Approach to the Design of Survivable COTS-Based Systems*. http://www.cert.org/research/isw/isw2001/papers/Lipson-29-08-a.pdf (2001).

[NSA 04] National Security Agency. *INFOSEC Assessment Methodology*. http://www.iatrp.com/iam.cfm (2004).

[Stoneburner 02] 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 99] 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.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
May 2005

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
System Quality Requirements Engineering (SQUARE):
Case Study on Asset Management System, Phase II

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Dan Gordon; Ted Stehney; Neha Wattas; & Eugene Yu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2005-SR-005

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

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

This report describes the second phase of an application of the System Quality Requirements Engineering (SQUARE) Methodology developed by the Software Engineering Institute's Networked Systems Survivability Program on an asset management system. An overview of the SQUARE process and the vendor is presented, followed by a description of the system under study. The research completed on Steps 4 through 9 of this nine-step process is then explained and feedback on its implementation is provided. The report concludes with a summary of findings and gives recommendations for future considerations of SQUARE testing.

This report is one of a series of reports resulting from research conducted by the SQUARE team as part of an independent research and development project of the Software Engineering Institute.

14. SUBJECT TERMS
information security improvement, information security costs, misuse cases, requirements engineering, system survivability

15. NUMBER OF PAGES
101

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL