A MANAGEMENT FRAMEWORK FOR THE SOFTWARE ENGINEERING PROFESSIONAL

A

PROJECT

Presented to the faculty of the University of Alaska Fairbanks

In Partial Fulfillment of the Requirements of

MASTER OF SOFTWARE ENGINEERING

By

Jason Weed

Fairbanks, Alaska

Spring 2012

ABSTRACT

A Management Framework for the Software Engineering Professional (MFSEP) is a conceptual management framework which can be used to assist small businesses by building on four basic building blocks, namely: Project Management, Software Engineering, Risk Management, and Architecture Development. Each of these building blocks is described in considerable detail in this report. This framework is intended to be used to help integrate best industry practices into the small business and provide a model for how the business can build their software engineering, IT, website development, or computer lab department.

This framework can be compared to two other common frameworks, the SEI's Capability Maturity Model Integration which shows businesses *what* to do, and the Information Technology Infrastructure Library which shows businesses *how* to run their business. The MFSEP framework tells both *what* and *how* things should be done. It is intended to combine the best of the CMMI and the ITIL service models.

The MFSEP was developed and used over the course of two years by the author in running a small computer lab, and based on that experience the MFSEP was found to be a reasonable solution. It could certainly be improved and possibly expanded upon by further development and testing.

The MFSEP has several distinct advantages compared to the CMMI and ITIL, which is why it is a great option for many small companies and organizations. Those advantages are summarized as follows.

- It is short and concise.
- It is straightforward and thus usable.
- It is easy to implement and thus affordable.

By implementing and customizing the MFSEP the user can gain valuable information in what to do and how to help run his business, organization, or department. In addition, this report shows that much of the information gathered *during* the process of implementing MFSEP can be used to better understand one's own role as a manager, as well as the roles of the organization and its key stakeholders.

ACKNOWLEDGEMENTS

I thank Dr. Peter J. Knoke, Dr. Orion Lawlor, and Dr. Jon Genetti, Department of Computer Science, University of Alaska Fairbanks for their oversight.

Dr. Knoke went out of his way, offering guidance and support throughout the project and without him this project would never have succeeded. His persistence and longevity as a professor here at the University is outmatched only by his love for helping students succeed.

I would also like to thank Dr. Brian Hay, Department of Computer Science, University of Alaska Fairbanks for inspiring what was the original idea behind this project. Although the original draft of lab procedures didn't make it into the final project draft for security reasons, those procedures did prove to be helpful in getting this project off the ground.

TABLE OF CONTENTS

FIGURES & TABLES

CHAPTER 1: INTRODUCTION

In my thirteen years as a web developer, database manager, and project manager I have noticed that many software engineers and others in the technical community have neglected the area of project management. Many technical experts spend their time in the details of the project, whether that is the code, database, or hardware, failing to realize the importance of the management aspect of the project.

This project's goal is to present a management framework for the software engineer to help address that neglected area. This framework is comprehensive enough that the user can apply it to his field of work and quickly see beneficial results. The examples I use in this framework are taken from specific project details that I dealt with in an IT position at the University of Alaska Fairbanks Computer Science Department. While each framework document is specific to the example project, each one is intended to be used as a customizable template to follow for any project a software engineer might manage.

This framework is divided up into four building blocks: 1) Project Management 2) Software Engineering, 3) Risk Management, and 4) Architecture Development. The reasons for choosing these four blocks are discussed in chapter 4.

CHAPTER 2: RELATED WORK

The MFSEP can be compared to the Software Engineering Institute's CMMI (Capability Maturity Model Integration) which is described as helping integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes [1]. The CMMI is a robust framework which assists many midsize and large businesses in reaching their greater potential by showing them *what* should be done rather than *how* it should be done. The MFSEP framework can also be compared to the ITIL (Information Technology Infrastructure Library) which is an approach that shifts the focus from running IT within a business to the managing of the business of IT [2]. The MFSEP framework tells both what and how things should be done. It is intended to combine the best of the CMMI (the *what*) and the ITIL (the *how*) service models.

By selecting a few concepts selected from the numerous topics covered in the CMMI and the ITIL, many smaller companies are provided with an easier and less formal framework in which their business can thrive. The MFSEP focuses on smaller companies and bridges the gap between using no formal framework in which to run the business and the implementation of the entirety of CMMI and ITIL frameworks.

The MFSEP has several distinct advantages compared to the CMMI and ITIL. That’s why it is a great option for many small companies and organizations. Those advantages are summarized as follows:

- It is short and concise.
- It is straightforward and thus usable.
- It is easy to implement and thus affordable.

CHAPTER 3: MANAGEMENT FRAMEWORK OVERVIEW

While each framework document is specific to the example project, each one is intended to be used as a customizable template to follow for any project a software engineer might manage. All the framework documents are listed below, and a brief description is given regarding the content of each one.

3.1 Project Management

- **Project Charter:** This document officially recognizes that a project exists. It is issued by senior management and gives the project manager the authority to pursue the project.
    - Project Title and Description
    - Project Manager and Authority Level
    - Goals and Objectives
    - Business Case
- **Project Scope:** This document identifies the project's deliverables and the work required to create those deliverables.
    - Scope Description
    - Deliverables
    - Exclusions
    - Constraints
    - Assumptions
- **Procurement Management:** This document gives the plan for procuring new project-related tools (i.e. printers, computers, etc.)
    - Contact Information
    - Process
- **Change Control Management:** This document describes how all aspects of the scope change management are handled, including how scope changes are requested, measuring impact of requests, and who decides if the change is approved/denied.
    - How scope changes are requested
    - How scope changes are handled
    - Measuring impact of requests
    - Approved / Denied

- **Human Resource Management:** This document helps deal with people.
    - Hiring Employees
    - Time Sheet Verification Process
    - When an Employee Doesn't Work Out
- **Process Documentation:** This document shows the processes most likely to be used while managing the project. No examples of processes from the UAF Chapman Lab have been given in this project for security reasons.
- **Communication Management:** This document summarizes the method, frequency, and schedule of the Software Engineer's communications with his supervisor.
    - Communication Approach
    - Roles
    - Meeting Guidelines

3.2 Software Engineering

- **WBS (Work Breakdown Structure):** This document identifies the tasks associated with the project.
    - Activities
    - Schedule
- **WBS Dictionary:** This document which defines the terms in the WBS.
    - Activities
    - Description

3.3 Risk Management

- **Risk Management:** This document helps to identify, analyze, and respond to risks associated with the project.
    - Methodology
    - Risk Identification
    - Qualitative Risk Analysis
    - Quantitative Risk Analysis
    - Risk Response
    - Risk Monitoring and Controlling

3.4 Architecture Development

- **Architecture Development:** This document describes the architecture of the project and shows possibilities for future development.
    - History
    - Applying the Architecture Tradeoff Analysis Method (ATAM)
    - Applying the Cost Benefit Analysis Method (CBAM)

The preceding descriptions make up the MFSEP framework and can be used as customizable templates for many projects.

CHAPTER 4: SELECTION OF BUILDING BLOCKS

Why choose the four building blocks of Project Management, Software Engineering, Risk Management, and Architecture Development? These topics are covered in the MFSEP because they offer good principles and practices which the user can capture and use in his own business environment.

More specifically, Project Management principles and topics were borrowed from the Project Management Body of Knowledge [3]. They form the initial building block of the framework because the PMBOK's comprehensive nature of all things that are project management related.

Because Software Engineering is a broad field only two topics were stressed in this project. The Work Breakdown Structure (WBS) and WBS Dictionary were added because they add a formal structure to the project schedule and provide a dictionary of project terms which can prove as useful tools while running any size project.

Risk Management was chosen because, as some have said, "software project management is really risk management." [4] This framework aims at managing risk from the beginning and throughout the life of the project.

Finally, Architecture Development was chosen because the work of Clements, Bass, and Kazman[5] has proved useful for many different companies during the past decade. The Architecture Tradeoff Analysis Method (ATAM) and the Cost Benefit Analysis Method (CBAM) are significant portions of the examples this project. They have helped to introduce new architecture ideas. They help to show which ideas are most cost beneficial by providing estimated Returns on Investment (ROIs) for each idea considered.

CHAPTER 5: EXAMPLE FRAMEWORK

5.1 Project Management

5.1.1 Project Charter

Without the Project Charter and the Project Sponsor's signature the project is not officially recognized. The Project Charter is typically issued by senior management and gives the project manager the authority to pursue the project. It also includes the following: project title and description, project manager and authority level, goals and objectives, and business case. An example project charter can be viewed in Appendix A.

5.1.2 Project Scope

The scope of the System Administrator IS Professional 3 (sysadmin) position is limited in order to focus his time and energy. The contract terms dictate that the project start on August 1 and end on the following May 31. This ten month time-frame is the block of time with which the sysadmin has to work. The sysadmin role is also limited to a part time effort (20 hours per week), and thus it requires great attention to detail as well as focus to the task at hand. .An example project scope can be viewed in Appendix B.

5.1.3 Procurement Management

The Procurement Management Plan creates the purchasing framework for this ten month project. This plan will assist the sysadmin when purchasing products throughout the life of the project. This plan should also be updated either as acquisition needs change or else at the beginning of each ten month time frame, whichever comes first.

This plan will also help to describe in detail what items should be purchased and when. Most items which must be purchased for this project cannot be created by the sysadmin or CS department and thus must be purchased from an outside vendor. Due to the ten month schedule of this project it is important for the sysadmin to understand what the procurement deadlines are at the outset of the project. An example procurement management plan can be viewed in Appendix C.

5.1.4 Change Control Management

The goal of the Change Control Management Plan is to identify and track changes to the sysadmin's projects or role. This ensures that the sysadmin, other CS staff, CS faculty, and HR are all on the same page with regard to the sysadmin's project. This plan should assist the sysadmin in large changes to the project and will allow future sysadmins to see the history of how architecture, software, hardware, etc, came into their present state within the project. Northrop Grumman has even created a Change Management Plan for the State of Montana [6]. An example change control management plan can be viewed in Appendix D.

5.1.5 Human Resource Management

The HR Procedures chapter covers a few essential procedures which the author found important while working the sysadmin position. These procedures may be documented elsewhere in a formal HR manual. However, if the procedures are included here they will be close at hand. They should be worked into the normal workflow of the management environment. This list provided here is not a comprehensive list of the procedures which should be documented for the sysadmin position, but it's a good start. An example human resource management plan can be viewed in Appendix E.

5.1.6 Process Documentation

The Lab Procedures section covers those procedures which need a more formal process defined and documented in order to prevent mistakes, memory slips, and malfunctions. However no example processes from the UAF Chapman Lab are given here for security reasons.

5.1.7 Communication Management

The purpose of the Communication Management Plan is to define communication guidelines for the sysadmin and the project team. This plan shows the roles of key stakeholders, contact information for those stakeholders, a communication matrix showing when meetings should take place, and meeting guidelines for the project.

The Communications Management Plan defines the following [7]:

- Communication requirements based on roles
- What information will be communicated
- How the information will be communicated
- When will information be distributed
- Who does the communication
- Who receives the communication

Poor communication can make a good situation bad and a bad situation catastrophic. It is the intention of this plan to mitigate poor communication by laying out a detailed approach for the sysadmin and the project team.

Because of the small size of the CS Department many of the roles mentioned overlap. This is because one person may fulfill many different roles which would be filled by separate people in a larger organization. An example communication management plan can be viewed in Appendix F.

5.2 Software Engineering

5.2.1 WBS (Work Breakdown Structure)

The WBS is a 40,000 foot view of the project. This allows us to see, from a broad perspective, where the project will head and what is accomplished in each step. The five high level steps are from the common project management framework from the Project Management Body of Knowledge (PMBOK).

There are four common views of the WBS: the outline view, the hierarchical structure, the tabular view, and the tree structure view. Each view shows the WBS elements in a slightly different format and each has its place in formal reporting. An example WBS can be viewed in Appendix G.

5.2.2 WBS Dictionary

The WBS dictionary takes the WBS and gives more detail for each element mentioned. Following these details will allow the Lab Director and sysadmin to successfully accomplish the goals set in the Charter document.

Taking the hierarchical structure view from the WBS, the WBS Dictionary expounds upon what work will need to be completed and by whom. An example WBS Dictionary can be viewed in Appendix H.

5.3 Risk Management

Risk management is the process by which an individual or team identifies and prioritizes risk. In this case the process will identify and prioritize risks for the Chapman and ASSERT labs at the University of Alaska Fairbanks. During this process we will cover the approach to risk management which we will be using. We will also identify all the associated risks to this project. We will then perform the qualitative and quantitative analysis on those identified risks. We will then give the quantified probability of meeting project objectives based on the identified risks. We will then give a response to the identified risks. Finally, we will show how we wish to monitor and control the risks throughout the life of the project. An example risk management plan can be viewed in Appendix I.

5.4 Architecture Development

The cycle of influences called the Architecture Business Cycle (ABC) travels from the business environment to the architecture and back again [5]. It is these influences which mold, critique, and ultimately make the architectures they pursue. From the Software Architecture in Practice text we see that architectures are beholden to the influences from many different areas: system stakeholders, the developing organization (including what they are not allowed to use it for [8]), the background and experience of the architects, and the technical environment [5].

Why have a lab in the first place? The reasons can be complex or simple. For one, it shows students and the occasional on-looker that the CS department knows and understands how to operate the technology that it teaches. For another, it provides a place for CS students to develop and hone their skill sets as programmers as well as software engineers. These are both important reasons for the lab to exist which are relevant to the students, the department, and the university.

We first look at the history of the Chapman Lab to see how it has evolved since its inception [9] [10] [11] [12]. Then we will look at the business goals which drive the quality attributes mentioned below. Finally we will perform the Architecture Tradeoff Analysis Method (ATAM) [13] [14] [15] and Cost

Benefit Analysis Method (CBAM) [5] on the Chapman Lab infrastructure. An example architecture development plan can be viewed in Appendix J.

CHAPTER 6: SUMMARY AND CONCLUSIONS

The management framework (MFSEP) keeps to a single principle, namely the KISS (sometimes called Keep It Simple Stupid) approach. Although simple, MFSEP allows the user to learn in depth, not only about his department, but also about his entire organization. The latter is accomplished by including other key stakeholders throughout the life of the project.

MFSEP deals with many things, such as the risks, tradeoffs, sensitivity points, and quality attributes, that the organization is likely to hold dear. It also addresses scenarios of likely happenings during a project, and a utility tree which ties quality attributes to those scenarios. And these subjects are just those from the ATAM in the Architecture Development example document! That document can also provide a better understanding on what specific architectural strategies might be best based on their predicted ROI's.

By working through this entire example framework the user can find other intangible benefits as well. The processes area can provide a great deal of structure to what are often unstructured discussions, where requirements and architectural strategies are freely mixed and where stimuli and response goals are not clearly articulated [5]. In thinking more about process, we often learn more about ourselves than anything else.

The overall conclusion is that by attempting to customize and use MFSEP, a software engineer managing a software-related project in a small organization can improve both the chances of his project success and his own set of management skills.

REFERENCES

1. **Software Engineering Institute.** CMMI Overview. *Software Engineering Institute.* [Online] 2008. [Cited: February 16, 2011.] http://www.sei.cmu.edu/cmmi/.

2. **NASCIO.** IT Management Frameworks: A Foundation for Success. *NASCIO.* [Online] August 2005. [Cited: March 15, 2012.] http://www.nascio.org/publications/documents/NASCIO-itManagementFramework.pdf.

3. **Project Management Institute.** *A Guide to the Project Management.* s.l. : Project Management Institute, 2008.

4. **Paulk, Mark C.** *Using the Software CMM® in Small Organizations.* 1998.

5. **Clements, Paul, Bass, Len and Kazman, Rick.** *Software Architecture in Practice.* 2nd. s.l. : Addison-Wesley Professional, 2003.

6. **Northrop Grumman Corporation.** Change Management Plan. *Montana's Official State Website.* [Online] November 29, 2007. [Cited: April 4, 2012.] http://interop.mt.gov/content/docs/IM_Change_Management_Plan_v3.0.pdf.

7. **PMdocs.** *Project Managment Docs.* [Online] http://www.projectmanagementdocs.com/templates/Communications%20Management%20Plan.pdf.

8. **University of Alaska.** University of Alaska: Acceptable Use of Online Resources. *University of Alaska.* [Online] August 23, 2011. [Cited: April 3, 2012.] http://www.alaska.edu/files/oit/OnlineResources.pdf.

9. **Knoke, Dr. Peter.** *History of the Chapman Lab.* [interv.] Jason Weed. October 26, 2011.

10. **Lawlor, Dr. Orion.** *History of the Chapman Lab.* [interv.] Jason. October 26, 2011.

11. **St. John, Joel.** *History of the Chapman Lab.* [interv.] Jason Weed. November 5, 2011.

12. **Smith, Christopher.** Gold mine strike leads ASUAF to cash in on new computer lab. *Sun Star.* Fairbanks, Alaska, United States of America : s.n., January 24, 1992.

13. **Software Engineering Institute Carnegie Mellon.** Software Architecturehttp://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm. *Software Engineering Institute Carnegie Mellon.* [Online] [Cited: April 4, 2012.] http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm.

14. **Kazman, Rick, Klein, Mark and Clements, Paul.** ATAM: Method for Architecture Evaluation. *http://citeseerx.ist.psu.edu.* [Online] August 2000. [Cited: April 3, 2012.] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.6014&rep=rep1&type=pdf.

15. **Witkamp, Jonathan.** Reconstructing Software Architecture Documentation for Maintainability. *http://homepages.cwi.nl.* [Online] 2006. [Cited: April 3, 2012.] http://homepages.cwi.nl/~paulk/thesesMasterSoftwareEngineering/2006/JonathanWitkamp.pdf.

APPENDICES

Appendix A: Example Project Charter

Project Title

Developing the Chapman and ASSERT Labs by the System Administrator IS Professional 3

Project Description

The UA-employee contract terms dictate that the project starts on August 1 and ends on the following May 31. This ten month time-frame is the block of time with which the sysadmin has to work. These procedures are meant to enhance the ability of the sysadmin. By compiling these planning and procedural documents into one location the goals for the sysadmin should become more easily attainable. When this project is complete there should be a dozen or so documents which help the sysadmin understand more fully their role and how to accomplish the goals set before them.

These documents may include the following: a charter document (this document), the scope document, procurement management plans, change management plans, a HR SOP, a lab SOP, a completed SRS, Work Breakdown Structure (WBS), a WBS dictionary, risk management plan, a communication management plan and a lab architecture development document.

Authority Level

Finances

The sysadmin does not have authority to authorize transactions larger than $100. Any amount above that amount needs approval of the sysadmin's PM and possibly the head of the department.

The sysadmin does have the authority to create the Project Execution Plan (PEP) and the Authorization for Expenditure (AFE), but they will need to be approved by the sysadmin's PM and possibly the head of the department.

Decisions must be made from within the financial approval of the AFE. If decisions require more finances that is approved in the AFE a new AFE will need to be reviewed and approved by the PM or department head.

Human Resources

The sysadmin does not have the authority to hire anyone without going through the hiring process which is described in detail in the HR SOP document.

Architecture

The sysadmin must get their supervisor's approval before making any architectural changes to the labs. This includes hardware, OS, and network devices for the Chapman Lab, ASSERT physical lab and ASSERT virtual lab.

Planning & Scheduling

The sysadmin has the authority to ensure the labs are functioning properly and that they are available, secure, easy to use, and performance ready within the confines of the approved architectural framework.

Goals

The goals of these documents are:

1. To define the role of the sysadmin
2. To prepare for emergency situations and disaster recovery by creating a Risk Management & Disaster Recovery Plan
3. To enhance, expand, and standardize the current plans and procedures available to the sysadmin

Business Case

Students and faculty rely on the labs for teaching assignments, homework, and general studies. It is important that the labs be available, secure, easy to use, and performance ready during their scheduled up-time.

It is not only UAF students and faculty which make use of the labs. Universities from around the country including Capitol College, Idaho State University, Colorado State University, the University of Hawaii, and other UA campuses make use of the ASSERT virtual lab.

As a matter of accreditation with the Accreditation Board for Engineering and Technology (ABET) this professional position is recommended. This ensures that students and faculty are not directly running the equipment. By making this a professional position, faculty and students can focus teaching and learning respectively.

Appendix B: Example Project Scope

Job Description Overview

This job description is taken directly from the HR document and is quoted verbatim.

> Department of Computer Science System Administrator provides day-to-day guidance, training and direction for staff in addition to other duties. Regularly assigns and reviews work. Is fluent in assigned area of responsibility. Hires, trains, evaluates performance and initiates corrective action, or effectively recommends these actions. Keeps the Computer Science Lab maintained and current for the use of students.
>
> Must possess theoretical and practical knowledge to develop (programming and system analysis) systems and adapt to rapidly changing technology. Must have extensive system administration experience working with Windows and Unix/Linux operating systems as well as virtual environments. Ability to install and configure advanced software applications such as compilers, databases, mathematical, statistical and visualization packages, emulators and clusters. Extensive knowledge of workstation tools and network management, diagnostic tools and security applications. Ability to develop automated systems management and monitoring tools using a variety of programming environments such as shell, perl, php, C++, etc. Ability to do resource capacity planning and management, and debugging various network protocols preferred.
>
> Bachelor of Science in Computer Science or equivalent in technical and theoretical training and experience required. CCNA certification is required within one year after date of hire.
>
> At least two years system administration experience with Unix/Linux and Windows server environments. This experience should include configuring and upgrading operating system and application software, applying security/maintenance patches, monitoring and tuning system performance.
>
> The ideal candidate is able to locate, read, understand and synthesize technical information from a variety of sources to solve technical problems, possesses excellent human relations skills, and has excellent written and verbal communication skills, which includes the ability to provide information clearly and concisely.

Job Description Duties

These job description duties are taken directly from the HR document and are quoted verbatim.

- Create and maintain user-oriented applications; prepare application requirements, definitions and design specifications.
- Develop, test, and implement applications according to published standards and methodologies, including application security and disaster recovery measures.

- Plan, coordinate and implement security measures to safeguard information in computer files against accidental or unauthorized modification, destruction or disclosure. Regulate access to computer data files, monitor data files use and update security files.
- Administer, monitor and/or modify automated IS software, applications and/or interfaces.
- Design, test and implement hardware platforms and operating systems networks.
- Analyze, document, install, develop and maintain operating systems software.
- Perform highly technical, responsible work in providing computer services in a lab environment: implement and maintain computer workstations, local area networks, network components, printers. Diagnose and resolve technical problems with applications and software, workstations, servers, network components, and related systems and devices.

Project Deliverables

These deliverables are more specific than the job description duties mentioned above. These are items which can be checked to ensure their accomplishment. Over the course of the project the sysadmin should be able to deliver the following items in a timely fashion:

- a working lab for the CS students (both the Chapman Lab and ASSERT Labs)
- on-time delivery of time-sheets to the Computer Science administrative assistant for processing for HR purposes
- up-to-date software in the lab
- up-to-date servers for the virtual lab
- scheduling of lab assistants during fall and spring semesters
- up-to-date wiki management of project management and standard operating procedure documentation

Exclusions

CS faculty, staff, and students should not expect the sysadmin to work on personal or work computers. Most work computers are administered by OIT only. Personal computers should never be worked on.

The Chapman and ASSERT Labs are the only labs for which the sysadmin is responsible. The math lab or other labs (including GENI) around the university are managed by other departments.

Computers in rooms 104 and 106 which are connected to the projector are not the responsibility of the sysadmin. These computers are the responsibility of the OIT department.

Events such as the CS “welcome back for the semester”, Collegiate Cyber Defense Competition (CCDC), and faculty meetings are not required.

Constraints

As mentioned above, the contract terms dictate that the project starts on August 1 and ends on the following May 31. This ten month time-frame is the block of time with which the sysadmin has to work. Within this ten month time-frame the role of the sysadmin is only part time (20 hours per week). Some weeks will require more work, typically at the beginning of each semester, and some will require fewer hours, typically in the middle of the semester.

The sysadmin is not required to work on any other equipment or software except for on the Chapman and ASSERT lab hardware and software. This allows the sysadmin to focus his time and energy in order to finish up each task in a timely fashion.

Assumptions

Because the sysadmin position is a professional position and not a student position, vacation and sick time is accrued which is reflected on each pay-stub. There will be times when vacation and sick time is taken. When this happens the sysadmins roles and responsibilities would be delegated to either the lab assistants or the sysadmin's supervisor depending on the nature of the responsibility.

If the sysadmin attends such events as the CS “welcome back for the semester”, Collegiate Cyber Defense Competition (CCDC), and faculty meetings these hours should qualify as normal working hours.

Appendix C: Example Procurement Management

Procurement Management Decision Making Process

The sysadmin will provide coordination for all procurement activities under this project. The sysadmin will need to work with their supervisor, the CS administrative assistant, and lab consultants to identify all items to be procured for the successful completion of the project. The sysadmin's supervisor will then review the procurement list prior to submitting it to the CS administrative assistant. The CS administrative assistant will review the procurement items, determine whether it is advantageous to buy locally or on the Internet, and begin the vendor selection and purchasing process.

Common Procurement Items

The following items have been identified to be essential for project completion and success. The following list of items, justification, and timeline are pending supervisor review for submission to the CS administrative assistant:

Item	Justification	Needed By	Authorized By
FTK	Need new licenses	Aug 20	
Encase	Need new licenses	Aug 20	
VMware	Need new licenses	Aug 20	
Paper	Printer is running low on paper.	Aug 25	
Printer Toner	Toner is running low for the printer.	Aug 15	
Desks	Furniture is old	Aug 15	

Table 1: Common Procurement Items

The following individuals are authorized to approve purchases for the CS Department and labs:

Name	**Role**
Brian Hay	Faculty; sysadmin supervisor
Jon Genetti	CS Department Head

Table 2: Approved Buyers

Contract Type

Most, if not all contracts are already in place and the sysadmin will not have to administer a new contract with vendors. The sysadmin and CS Administrative Assistant will work with the purchasing department to identify the items quantities, and required delivery dates.

Procurement Vendors

Name	Number	Customer Rep.	Misc.
Dell	1-800-576-6038		
VMware	Toll Free: 877-486-9273 Local: 650-475-5345		Customer# - 6775222027
FTK	801-377-5410	Haley Webb	x732
Guidance Software	615-523-5502	Alexandra Price	
Capitol Office Systems	907		

Table 3: Procurement Vendors

Performance Metrics for Procurement Activities

In order to show management why we used certain vendors it is important to keep track of which vendors we use and why. This performance metrics section assists the sysadmin in keeping track of these details to a fairly high degree. This also helps the sysadmin ensure that the products are delivered on time and within the allotted ten month time frame.

Each metric is rated on a 1-5 scale as indicated below:

- 1 – Absolutely horrible
- 2 – Not quite up to par
- 3 – Acceptable
- 4 – Really quite good
- 5 – Incredibly awesome

Vendor	**Product Quality**	**On Time Delivery**	**Documentation Quality**	**Cost per Unit**
Dell				
Vendor #2				

Table 4: Performance Metrics for Procurement Activities

Appendix D: Example Change Control Management

How Scope Changes are Requested

Scope change requests to the project can be made by anyone. The person requesting the change should begin the change control process by filling out the Change Request form which should be available at the sysadmin's office. The Change Request form should first be submitted in full to the sysadmin. The sysadmin will then send it to their supervisor for final approval. Changes which can be suggested can be related to hardware, software, procedures, an external event, an error or omission in the scope of the project, a value-added change, or a risk response.

How Scope Changes are Handled

Once the sysadmin has the Change Request form from the person submitting the potential change he will review it and discuss any issues they see with the individual. The sysadmin will also identify any additional information and / or next steps in order to complete the suggested change. Once this has been accomplished the sysadmin will forward the request onto his supervisor. The supervisor must then complete a new section in the Change Request logbook. Once the logbook has been updated the sysadmin will need to communicate with the appropriate staff or consultants to gather any final data before continuing on with the Impact Analysis.

Measuring Impact of Requests

Once all the information has been gathered and reviewed by the sysadmin they will prepare and submit the Change Request Impact Analysis form to their supervisor. The supervisor typically does not work directly on the project but rather as the organization as a whole.

Approved / Denied

Once the appropriate documentation has been submitted to the supervisor they can review the change. If the supervisor approves the Change Request form is given to the Project Sponsor for final approval. This decision lies solely with the Project Sponsor who will give their final recommendation in writing to the sysadmin and the supervisor. The Change Request logbook is updated by the supervisor with “approved” or “denied”.

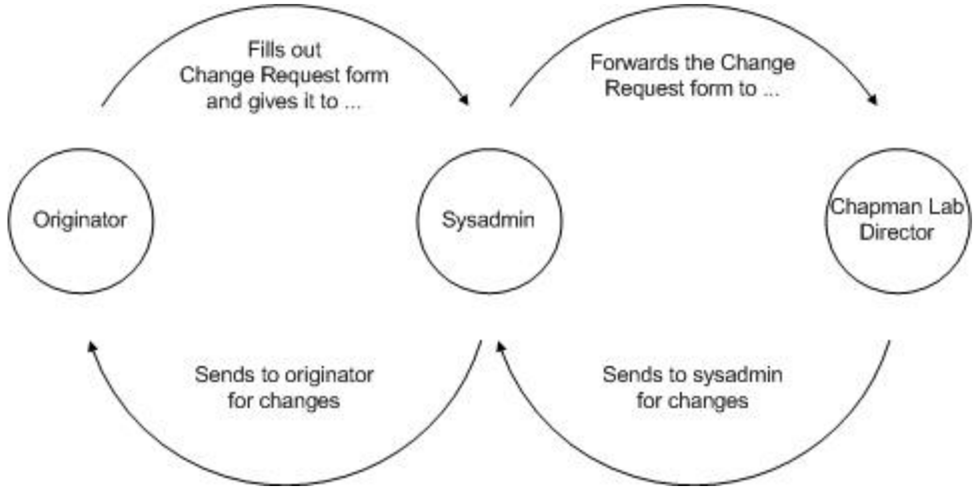


Figure 1: Change Control Workflow

Change Request Roles & Responsibilities

A change request can originate from the following: a student, the sysadmin, the project manager, and other faculty or stakeholders.

Role	**Duties**
Originator	• Identifies a possible need for a project • Notifies sysadmin of the possible need • Fills out the Change Request Form (See Appendix D.1)
Sysadmin	• Assists in providing additional research • Assists in filling out the Change Request Form • Identifies potential risks associated with the Change Request Form • Assists in identifying scope and schedule impacts • Performs initial evaluation and analysis • Assigns a Change Request Form number and updates the Change Management Register (see Appendix D.2) • Completes the Change Request Form if necessary
Program Manager	• Receives notice of an impending change • Perform technical review of the new Change Request Form • Perform financial review of the new Change Request Form
Project Sponsor	• Gives final approval

Table 5: Change Request Roles & Responsibilities

Appendix D.1: Example Change Request Form

Change Request Form [6]

Project Name:	• Chapman Lab or • ASSERT Lab or • Other
Project Phase:	• Pre-Fall Semester • Fall Semester • Christmas Break • Spring Semester • Post-Spring Semester
Program Manager:	
Sysadmin:	

Request Title:	
Request Number:	
Date Issued:	
Date Required:	

Reason for Change:
Description of Change:
Cost Estimate:
Ramifications:

Approved:	Rejected:	Pending:	Deferred:
Reason for Rejection or Deferral:			

Sysadmin	**Program Manager**	**Project Sponsor**
Name:	Name:	Name:
Signature:	Signature:	Signature:
Date Signed:	Date Signed:	Date Signed:

Table 6: Example Change Request Form

Appendix D.2: Example Management Register

CR#	Title	Description	Submitted By	Date Issued	Financial Impact	Date Required	Approved By	Status	Disposition

Table 7: Example Management Register

Appendix E: Example Human Resource Management

Lab Consultant Hiring Process

At the beginning of the project, preferably sometime in late August or early September, it is important to hire between one and three lab consultants. These lab consultants are not part of the RAs and TAs who will also assist in the lab. It typically takes between four and seven individuals to run a complete schedule for the lab. This includes the sysadmin, lab consultants, RAs, and TAs.

To begin this process we ask the Administrative Assistant to place the job listing on www.uakjobs.com. This can usually be done in a day or so. Once possible candidates send in their resumes the Administrative Assistant will give the sysadmin packets for each individual. The sysadmin should go through these packets to ensure that each person is qualified before calling them and scheduling an interview.

The interview must take place with at least three university employees; preferably the sysadmin, the sysadmin's supervisor, and one other, possibly the Administrative Assistant or other CS faculty.

The HR department provides the standard list of questions which must be used in the interview process. Other questions may be asked but the ones provided by HR must be asked during the course of the interview. The Administrative Assistant has a copy of these questions from HR.

Once the interview has been conducted we can tell the candidate that we'll be in touch. Once they leave the room a quick conversation usually takes place on deciding if they were qualified and if so if we should hire them. Once a decision has been made to hire an individual the sysadmin can call the person and have them come in and finish up any hiring paperwork which the Administrative Assistant might have.

Time Sheet Verification Process

RAs and TAs do not have their time sheets verified with the sysadmin. The Administrative Assistant hands out the time sheets to the lab consultants. Each lab assistant is responsible for filling their own time sheet out accurately and in a timely fashion so it can be verified by the sysadmin. The sysadmin is responsible for ensuring that the time sheets are accurate and are given back to the

Administrative Assistant in a timely fashion. The sysadmin must check the hours against the current schedule to be sure of the time sheet's accuracy. In addition the sysadmin should also ask the lab consultant if they covered for someone else if they have more hours on their time sheet than were scheduled.

When a Lab Consultant Doesn't Work Out

There are many reasons why a lab consultant might not work out. Some of those reasons may be: not showing up on time, not showing up at all, lying to the sysadmin, lying to CS faculty or staff, not filling out their time sheet accurately, etc.

The phrase "document, document, document" really comes into play here. If you are ever asked why you scheduled someone for fewer hours, or no hours at all, you'll want to have backup for your decision. If you've documented each infraction that the consultant committed you'll be in a much better position if someone calls you on the carpet for that decision.

Because of the difficulty of firing someone within the university system it is simply easier to stop scheduling the lab consultant who isn't working out. This may cause the CS department to have to hiring another lab consultant who can be scheduled to pick up the lost hours.

Annual Review

The annual review is an often overlooked benefit to both the employee and the organization. It is in this meeting that an employee can ask candidly about his performance. It is important to know and understand if the employee's supervisor is happy or dissatisfied with the employee's performance. Expectations may have changed, job roles may have shifted, life changes may have changed, market pressures may be different from when the employee first started, project scope may have been increased, decreased, or swapped with something else altogether. In order for the employee to better understand their role within the department and organization as a whole this is a beneficial process which should not be avoided.

This is also the meeting where potential pay raises, pay cuts, schedules and benefits packages could be addressed as well.

Appendix F: Example Communication Management

Communications Management Approach

From the project manager's perspective the communication structure can make or break a project. It is vitally important to layout the roles of the key stakeholders involved in the project and how those roles should be communicating with the rest of the project team.

Roles

Project Sponsor – CS Department Head

Also known as the Venture Leader, the project sponsor is the primary backer of this project. In this case it is the C.S. Department Head and it is this role which is responsible for the project funding. Communication with the project sponsor should be in a brief summary and should always be written from the 40,000 ft level. If the project sponsor would like more details they can certainly ask.

Program Manager – Sysadmin's Supervisor

This project has two distinct programs which run in parallel: the Chapman Lab and the ASSERT Labs (one physical, the other virtual). The Program Manager oversees both of these programs and does the major purchasing of resources, including servers, etc. Because the Program Manager is responsible for the finances for this project is important to supply them with more detailed communications than the Project Sponsor.

Stakeholders vs. Key Stakeholders – UAF, CS Dept., College of Engineering and Mines; CS Department Head; Sysadmin's Supervisor; Sysadmin

Because this document deals with the communication plan we will separate and distinguish between the stakeholders with which we communicate and those we do not. Although UAF, the C.S. Department, and the College of Engineering and Mines are stakeholders they are not stakeholders with which this project would normally communicate with. As such we will identify the key stakeholders as the C.S. Department Head, the sysadmin's supervisor, and the sysadmin.

Change Control Board – CS Department Head; Sysadmin's Supervisor; Sysadmin

The Change Control Board deals primarily with the Change Control Management Plan which accepts proposals for change, reviews the proposals, and authorizes changes within the scope of the project. Technology impact, student impact, and architecture implementation strategies are the prototypical types of communication with which this board uses to complete their task.

Customers / Clients – Students & Faculty from the University of Hawaii, UAF, Capitol College, USAFA, etc.

The customers / clients of this project are the students and faculty from a number of institutions which have included: the University of Hawaii, University of Alaska Fairbanks, Capitol College, and the United States Air Force Academy. These institutions have accounts with the UAF ASSERT Lab which allows their instructors to teach their students by using the virtual machines.

Communication with these customers usually involves account setup and deletion, scheduled maintenance downtime, emergency downtime, etc.

Project Manager – Sysadmin

The sysadmin is the primary communicator of this project. It is their responsibility to communication the overall performance and status of the Chapman and ASSERT Labs to others. The Chapman Lab is monitored through scripts running on SVAD. These scripts show daily, weekly, and monthly usage stats of the Chapman Lab.

Project Team – Sysadmin's Supervisor; Sysadmin

The Project Team is limited to those who actually do work on the project. It is this team which deals with the daily tasks of making sure the Chapman and ASSERT Labs are operations for students and other customers. Due to the nature of the work the Project Team does the type of communication they typically use is very detailed. It is also important that this team communication on a daily basis to ensure a constant awareness of what has been done for the day and what is still left to accomplish. The Project Team should also meet with the Project Manager in a weekly basis to give periodic updates as to what was accomplished the prior week, any issues left to resolve from the prior week, any work for the upcoming week and any potential risks involved with the new work.

Steering Committee – CS Dept. Head; CS Faculty; Sysadmin's Supervisor

The Steering Committee is primarily made up of management who will have a say in the direction of the department. This assuredly will involve the C.S. Department Head, potentially involve other C.S. Faculty, and most assuredly the sysadmin's supervisor. It is this committee which will drive the scope of the project for the sysadmin position. This committee should be focused on making sure the stakeholders will be satisfied with the results.

Technical Lead – Sysadmin

The sysadmin also fulfills the role of the Technical Lead. It is this role which ensures that the technical aspects of the project are brought to the forefront so that from a technical perspective the project can succeed. It is this role which is responsible for the documenting of the technical aspects of the project, such as the system architecture, etc. The Technical Lead is part of the Project Team is many situations but in some cases this can be an outside consultant.

Project Team Directory

Role	**Name**	**Email**	**Phone**
Project Sponsor			
Program Manager			
Project Manager			
Key Stakeholders			
Customers / Clients			
Project Team			
Technical Lead			

Table 8: Project Team Directory

Communication Matrix [7]

The following table identifies the communications requirements for this project.

Communication Type	Objective of Communication	Medium	Frequency	Audience	Owner	Deliverable
Kickoff Meeting	Introduce the project team and the project. Review project objectives and management approach.	Face to face	Once (in August)	• Project Sponsor • Project Team • Stakeholders	PM	Agenda meeting minutes
Project Team Meetings	Review status of the project with the team.	Face to face conference call	Weekly	• Project Team	PM	Agenda meeting minutes
Technical Design Meetings	Discuss and develop technical design solutions for the project.	Face to face	As needed	• Project Technical Staff	Technical Lead	Agenda meeting minutes
Monthly Project Status Meetings	Report on the status of the project to management.	Face to face conference call	Monthly	• Project Sponsor • Program Manager	PM	
Project Status Reports	Report the status of the project including activities, progress, costs and issues.	email	Monthly	• Project Sponsor • Project Team • Stakeholders	PM	Project status report

Table 9: Communication Matrix

Guidelines for Meetings

Meetings should take no more than 60 minutes and should always include a meeting agenda outlining the topics to be covered.

Meeting Agenda

All meetings must have an agenda. Without the agenda chaos usually ensues. The meeting agenda should be given out a minimum of 2 business days prior to the meeting. This is to help facilitate the discussion during the meeting by making every attendee aware of what will be discussed. If there is more than one person with issues on the agenda the person's name and point of discussion should be identified. If there were action items from the previous meeting then the first item on the current agenda should be to cover those items.

Meeting Minutes

The meeting minutes are taken by the Meeting Chair Person (see below) and should be distributed within 2 business days following the meeting. The meeting minutes should include the following:

- status of all items on the agenda
- new action items
- Parking Lot items

Action Items

Old Action Items are essentially moved from the Meeting Agenda to the Meeting Minutes along with their status. Once old action items have been covered new action items can be added to the list. The owner of each of the new Action Items should be identified in the meeting.

Meeting Chair Person

Every meeting needs a person who is in charge to make sure the meeting starts and ends on time as well as follows the Meeting Agenda. This person is also responsible for keeping track of the Meeting Minutes and their distribution.

Note Taker

The Note Taker's responsibility differs from the Meeting Chair Person in that they are taking notes on the following:

- Status of the Action Items
- Maintaining the Parking Lot items
- and general notes on anything else of importance

The Meeting Chair Person will use these notes to create the Meeting Minutes.

Parking Lot

The Parking Lot is a useful tool used by the Meeting Chair Person or Note Taker to record and defer items which are not on the meeting agenda. These are typically important issues which should be discussed but the current meeting is not the time. The Parking Lot should identify the issue at hand and who is in charge of following up. The Parking Lot should also be included in the Meeting Minutes.

Appendix G: Example Work Breakdown Structure (WBS)

Outline View

1. Chapman & ASSERT Labs
    1. Initiation
        1. Create Project Charter document and have stakeholders sign off
        2. Create Scope document and confirm with stakeholders
        3. Create Change Request form
        4. Create Lab Architecture Development document
    2. Planning
        1. Create Risk Management & Disaster Recovery Plan
        2. Create Work Breakdown Structure (WBS) & WBS dictionary
        3. Create the Software Requirements Specification (SRS) document
        4. Continue following Charter and Scope requirements
        5. Continue to document changes using the Change Request form
        6. Plan for procurement items
    3. Execution
        1. Create Lab SOP documentation
        2. Continue following Charter and Scope requirements
        3. Continue to document changes using the Change Request form
        4. Hire lab consultants
        5. Schedule lab hours
        6. Purchase procurement items
    4. Monitoring & Controlling
        1. Semi-annual review of Risk Management & Disaster Recovery plan
        2. Annual review of Charter and Scope documents
        3. Use Lab SOP for processes within the lab environment
        4. Weekly status updates
        5. Semester status updates
    5. Closeout
        1. Finalize changes from Change Request forms
        2. Finalize changes to Scope document
        3. Document changes to Lab SOP
        4. Document lessons learned
        5. Update files / records
        6. Archive Files / records

Hierarchical Structure

Level	WBS Code	Element Name
1	1	Chapman & ASSERT Labs
2	1.1	Initiation
3	1.1.1	Create Project Charter document and have stakeholders sign off
3	1.1.2	Create Scope document and confirm with stakeholders
3	1.1.3	Create Change Request form
3	1.1.4	Create Lab Architecture Development document
2	1.2	Planning
3	1.2.1	Create Risk Management & Disaster Recovery Plan
3	1.2.2	Create Work Breakdown Structure (WBS) & WBS dictionary
3	1.2.3	Create the Software Requirements Specification (SRS) document
3	1.2.4	Continue following Charter and Scope requirements
3	1.2.5	Continue to document changes using the Change Request form
3	1.2.6	Plan for procurement items
2	1.3	Execution
3	1.3.1	Create Lab SOP documentation
3	1.3.2	Continue following Charter and Scope requirements
3	1.3.3	Continue to document changes using the Change Request form

3	1.3.4	Hire lab consultants
3	1.3.5	Schedule lab hours
3	1.3.6	Purchase procurement items
2	1.4	Monitoring & Controlling
3	1.4.1	Semi-annual review of Risk management & Disaster Recovery plan
3	1.4.2	Annual review of Charter and Scope documents
3	1.4.3	Use Lab SOP for processes within the lab environment
3	1.4.4	Weekly status updates
3	1.4.5	Semester status updates
2	1.5	Closeout
3	1.5.1	Finalize changes from Change Request forms
3	1.5.2	Finalize changes to Scope document
3	1.5.3	Document changes to Lab SOP
3	1.5.4	Document lessons learned
3	1.5.5	Update files / records
3	1.5.6	Archive files / records

Table 10: WBS Hierarchical Structure

Tabular View

Level 1	**Level 2**	**Level 3**
1 Chapman & ASSERT Labs	1.1 Initiation	1.1.1 Create Project Charter document and have stakeholders sign off
		1.1.2 Create Scope document and confirm with stakeholders
		1.1.3 Create Change Request form
		1.1.4 Create Lab Architecture Development document
	1.2 Planning	1.2.1 Create Risk Management & Disaster Recovery Plan
		1.2.2 Create Work Breakdown Structure (WBS) & WBS dictionary
		1.2.3 Create the Software Requirements Specification (SRS) document
		1.2.4 Continue following Charter and Scope requirements
		1.2.5 Continue to document changes using the Change Request form
		1.2.6 Plan for procurement items
	1.3 Executing	1.3.1 Create Lab SOP documentation
		1.3.2 Continue following Charter and Scope requirements
		1.3.3 Continue to document changes using the Change Request form
		1.3.4 Hire lab consultants
		1.3.5 Schedule lab hours

		1.3.6 Purchase procurement items
	1.4 Monitoring & Controlling	1.4.1 Semi-annual review of Risk management & Disaster Recovery plan
		1.4.2 Annual review of Charter and Scope documents
		1.4.3 Use Lab SOP for processes within the lab environment
		1.4.4 Weekly status updates
		1.4.5 Semester status updates
	1.5 Closeout	1.5.1 Finalize changes from Change Request forms
		1.5.2 Finalize changes to Scope document
		1.5.3 Document changes to Lab SOP
		1.5.4 Document lessons learned
		1.5.5 Update files / records
		1.5.6 Archive files / records

Table 11: WBS Tabular View

Tree Structure View

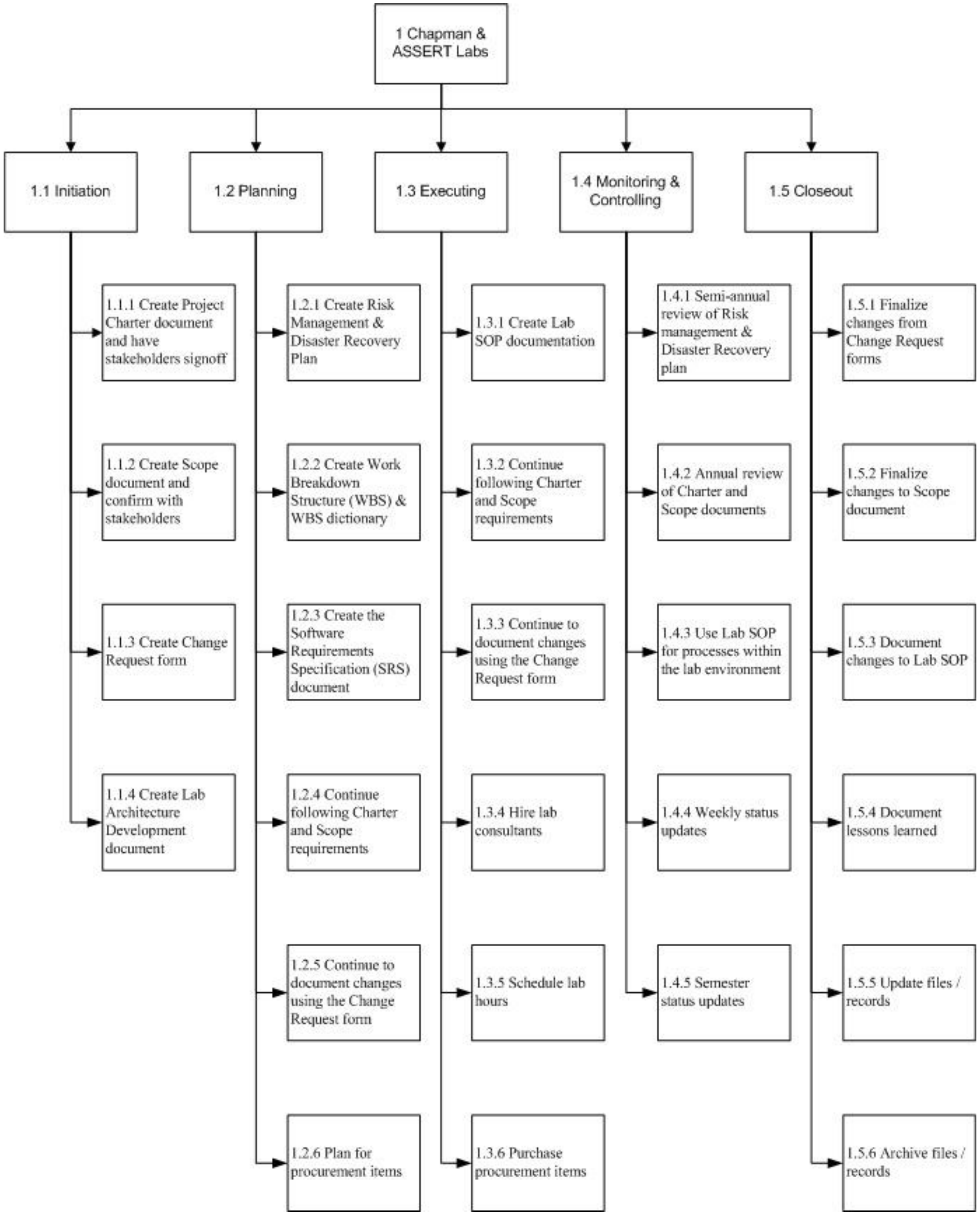

1 Chapman & ASSERT Labs
1.1 Initiation
1.2 Planning
1.3 Executing
1.4 Monitoring & Controlling
1.5 Closeout
1.1.1 Create Project Charter document and have stakeholders signoff
1.1.2 Create Scope document and confirm with stakeholders
1.1.3 Create Change Request form
1.1.4 Create Lab Architecture Development document
1.2.1 Create Risk Management & Disaster Recovery Plan
1.2.2 Create Work Breakdown Structure (WBS) & WBS dictionary
1.2.3 Create the Software Requirements Specification (SRS) document
1.2.4 Continue following Charter and Scope requirements
1.2.5 Continue to document changes using the Change Request form
1.2.6 Plan for procurement items
1.3.1 Create Lab SOP documentation
1.3.2 Continue following Charter and Scope requirements
1.3.3 Continue to document changes using the Change Request form
1.3.4 Hire lab consultants
1.3.5 Schedule lab hours
1.3.6 Purchase procurement items
1.4.1 Semi-annual review of Risk management & Disaster Recovery plan
1.4.2 Annual review of Charter and Scope documents
1.4.3 Use Lab SOP for processes within the lab environment
1.4.4 Weekly status updates
1.4.5 Semester status updates
1.5.1 Finalize changes from Change Request forms
1.5.2 Finalize changes to Scope document
1.5.3 Document changes to Lab SOP
1.5.4 Document lessons learned
1.5.5 Update files / records
1.5.6 Archive files / records

Figure 2: WBS Tree Structure View

Project Layout Matrix

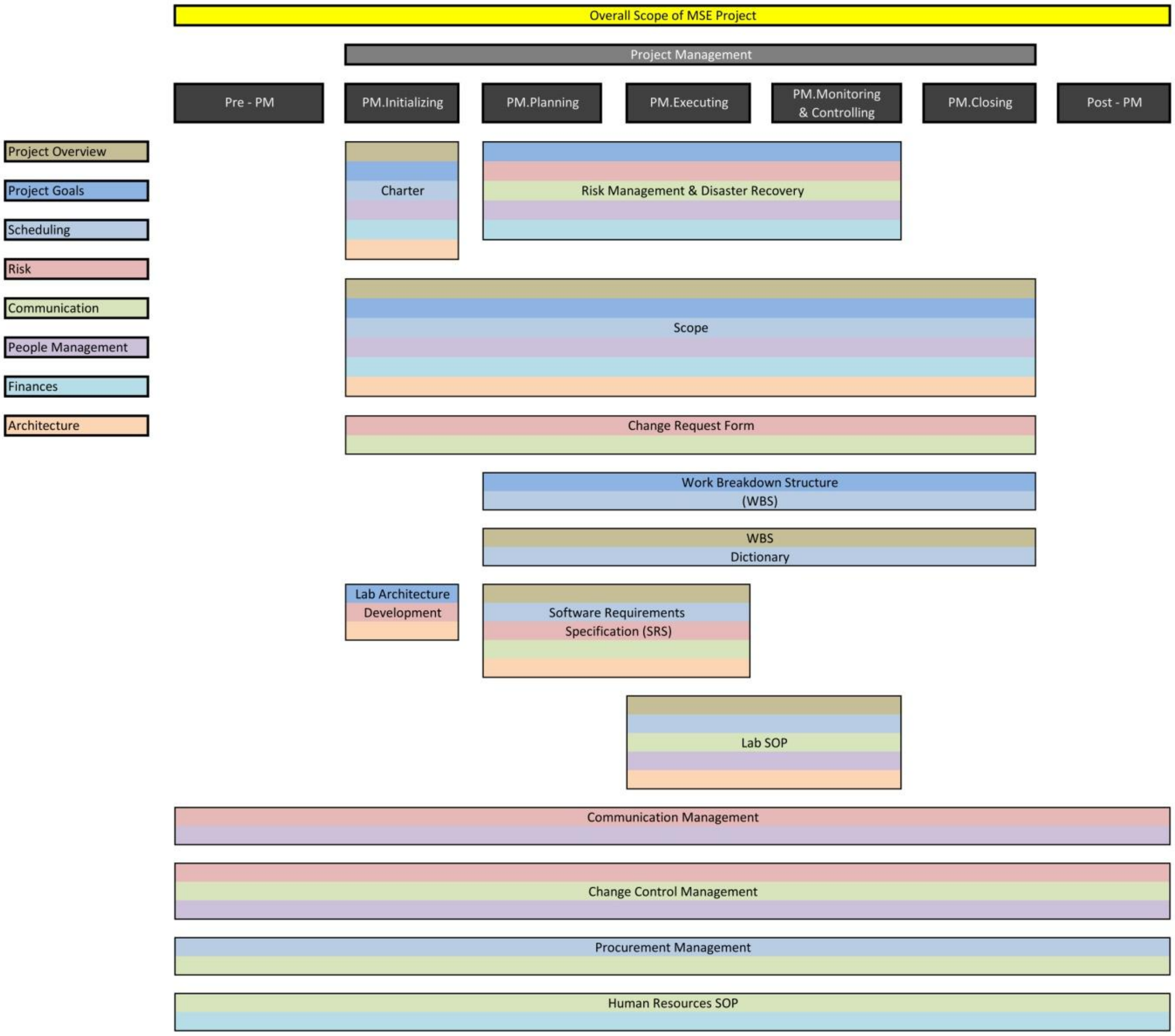


Figure 3: WBS Project Layout Matrix

Appendix H: Example WBS Dictionary

WBS Dictionary

Level	WBS Code	Element Name	Definition	Work Assigned To
1	1	Chapman & ASSERT Labs	The work to be done for both Chapman & ASSERT Labs	
2	1.1	Initiation	The work to initiate the project	
3	1.1.1	Create Project Charter document and have stakeholders sign off	The C.S. Department Head signs off on the Charter to allow the Lab Director to move to the planning stage	Lab Director
3	1.1.2	Create Scope document and confirm with stakeholders	Lab Director creates the Scope document	Lab Director
3	1.1.3	Create Change Request form	Sysadmin creates the Change Request form	Sysadmin
3	1.1.4	Create Lab Architecture Development document	Determining which architecture is the best solution for this project	Lab Director
2	1.2	Planning	All the work to be done in the planning phase	
3	1.2.1	Create Risk Management & Disaster Recovery Plan	The Lab Director and Sysadmin work together to discover the risks associated with the overall project	Lab Director or Sysadmin
3	1.2.2	Create Work Breakdown Structure (WBS) & WBS dictionary	Lab Director creates the WBS & WBS dictionary documents	Lab Director
3	1.2.3	Create the Software Requirements Specification (SRS) document	If an SRS is necessary the Lab Director and Sysadmin work together to flush out the details of the software requirements	Lab Director or Sysadmin
3	1.2.4	Continue following Charter	Follow the directions set out at	

		and Scope requirements	the beginning of the project	
3	1.2.5	Continue to document changes using the Change Request form	Make sure to use the Change Request form when making changes to the project	Sysadmin
3	1.2.6	Plan for procurement items	Plan ahead for what may need to be purchased and to acquire funding	Sysadmin
2	1.3	Execution	The work to be done in the execution phase	
3	1.3.1	Create Lab SOP documentation	The Sysadmin creates any process documentation for the labs	Sysadmin
3	1.3.2	Continue following Charter and Scope requirements	Follow the directions set out at the beginning of the project	
3	1.3.3	Continue to document changes using the Change Request form	Make sure to use the Change Request form when making changes to the project	Sysadmin
3	1.3.4	Hire lab consultants	Hire qualified students to manage the lab and help other students	Lab Director or Sysadmin
3	1.3.5	Schedule lab hours	Schedule the hours which the lab consultants will work throughout the semester	Sysadmin
3	1.3.6	Purchase procurement items	Purchase the needed items	Sysadmin
2	1.4	Monitoring & Controlling	The work to be done during the monitoring & controlling phase	
3	1.4.1	Semi-annual review of Risk Management & Disaster Recovery plan	Be sure to check for any new risks or any old risks which may not be relevant to the project any more	Sysadmin
3	1.4.2	Annual review of Charter and Scope documents	Be sure to check for any outdated material	Lab Director
3	1.4.3	Use Lab SOP for processes	Integrate Lab SOP	Sysadmin

		within the lab environment	documentation into every aspect of the Sysadmin position	
3	1.4.4	Weekly status updates	Send updates to the Lab Director	Sysadmin
3	1.4.5	Semester status updates	Send updates to the Lab Director and C.S. Department Head	Sysadmin
2	1.5	Closeout	The work to be done during the closeout phase	
3	1.5.1	Finalize changes from Change Request forms	Complete unfinished work which was approved	Sysadmin
3	1.5.2	Finalize changes to Scope document	Complete changes to Scope document	Lab Director
3	1.5.3	Document changes to Lab SOP	Ensure all lab processes are up to date	Sysadmin
3	1.5.4	Document lessons learned	Write down what worked, what did not, and possible scenarios for future development	Lab Director and Sysadmin
3	1.5.5	Update files / records	Complete the updating of files and records	Sysadmin
3	1.5.6	Archive files / records	Complete the archiving of files and records	Sysadmin

Table 12: WBS Dictionary

Appendix I: Example Risk Management

Risk Management Approach

Methodology

This document will outline both the qualitative and quantitative risk management strategies.

Roles and responsibilities

The sysadmin will be solely responsible for coming up with a risk management and disaster recovery plan. The sysadmin's supervisor should sign-off on this plan and should be given biannual updates as to the status of associated risks as part of the monitoring and controlling phase.

Budget

This portion of the sysadmin's job does not have a separate budget line and therefore will be worked on and maintained during the lifetime of the project. If this project does receive a separate budget line item it is recommended that between 3% and 5% of the overall budget be allocated to the risk management and disaster recovery planning phases.

Timing

Because of the quick time-line of this project the monitoring and controlling phases should happen once at the beginning of the project and once more over the Christmas break, before the second semester begins. This will ensure that the project is still on track with no foreseeable road blocks to completion.

Scoring and Interpretation

Defined Conditions for Impact Scales of a Risk on Major Project Objectives [3]					
	Low - 1	**Some - 2**	**Tolerable - 3**	**Serious - 4**	**Catastrophic - 5**
Cost	Insignificant cost increase	<10% cost increase	10-20% cost increase	20-40% cost increase	>40% cost increase
Time	Insignificant time increase	<5% time increase	5-10% time increase	10-20% time increase	>20% time increase
Scope	Scope decrease barely noticeable	Minor areas of scope affected	Major areas of scope affected	Scope reduction unacceptable to sponsor	Project end item is effectively useless
Quality	Quality degradation barely noticeable	Only very demanding applications are affected	Quality reduction requires sponsor approval	Quality reduction unacceptable to sponsor	Project end item is effectively useless

Table 13: Risk Management - Scoring and Interpretation

Reporting Formats

The risk management and disaster recovery plan will be distributed to the sysadmin, the sysadmin's supervisor, and the CS department head as a PDF document. Also, for maintainability and searchability this plan will be available on an internal CS wiki site. This is to encourage familiarity and consistent use throughout the life of the project.

Tracking

The identified risks will be tracked throughout the life of the project. The risk management process will be audited over Christmas break due to the short nature of the project. Identified risks can certainly change over the course of the project. This is why it is important to track the identified risks over the lifetime of the project to ensure that they are being avoided or mitigated.

Risk Identification

When identifying risks it is important to think about the different aspects of the project. Here we look at both internally and externally facing risks to the project. From there we break the risks down into the following categories: staff (which includes turnover, issues, health, and vacation/sick leave), hardware, software, size and time underestimation, organization, and requirements change. It is also important to know what the probability is of each risk, its impact on the project as a whole, as well as the history and frequency of each identified risk.

By combining the probability of a risk occurring and the impact it will have on the project we can assess the relative risk to the entire project. Our scales for probability and effect fall within 1 and 5 and are as follows.

Probability	***Impact***
1. not occur	1. low
2. doubtful	2. some
3. possible	3. tolerable
4. probable	4. serious
5. inevitable	5. catastrophic

Table 14: Risk Management - Probability and Impact

As an example, let’s say we have an identified risk which has a probability of 4 (probable) and its impact on the project is 2 (some). We can assess the relative risk associated with this particular identified risk by multiplying them together to get an 8. The lowest a risk can score is a 1 and the highest is a 25 thus our relative score of 8 is on the lower side, but we would need to calculate the relative risk for the rest of the identified risks to see where that 8 falls in comparison to all other identified risks.

All identified risks have been described and categorized in Appendix I.1.

Qualitative Risk Analysis

Qualitative Risk Response

What is considered a high risk? There are a number of ways we can declare a risk as "high", three of which are shown. One, if we take our identified risks which have a probability of 4 – probable or 5 – inevitable or impacts which are 4 – serious or 5 – catastrophic we would end up giving 27 of our 44 identified risks a rating of "high". This does not seem reasonable from a relative perspective.

Two, if we take the worst case relative risk and its average we can declare that any identified risk above the average of all identified risks can be a "high" risk, which gives us 21 "high" risks out of our 44 identified risks. This still seems high from a relative perspective.

Three, by doubling the relative risk average and declaring any relative risk worst case that falls above that number we get ten results. The average relative risk comes to about seven. We use that figure of seven and double it, giving us fourteen. So, when we take the relative risk worst case and declare any identified risks at or above fourteen we get ten results remaining as "high".

These ten identified risks will be considered the top priority for a Management Analysis and Disaster Recovery Plan. All identified high risks have been analyzed and categorized in Appendix I.2.

Non-critical Risks

Risks which do not have a description of "high" will not warrant a response in this risk management plan.

Overall Project Risk

Based on the above analysis it should be clear that there are some very real and dangerous risks to this project. If any of the above mentioned "high" risks were to happen simultaneously it would be incredibly challenging for a single staff person to manage all associated activities within their allotted time frame of 20 hours per week.

Quantitative Risk Analysis

Quantitative Risk Response

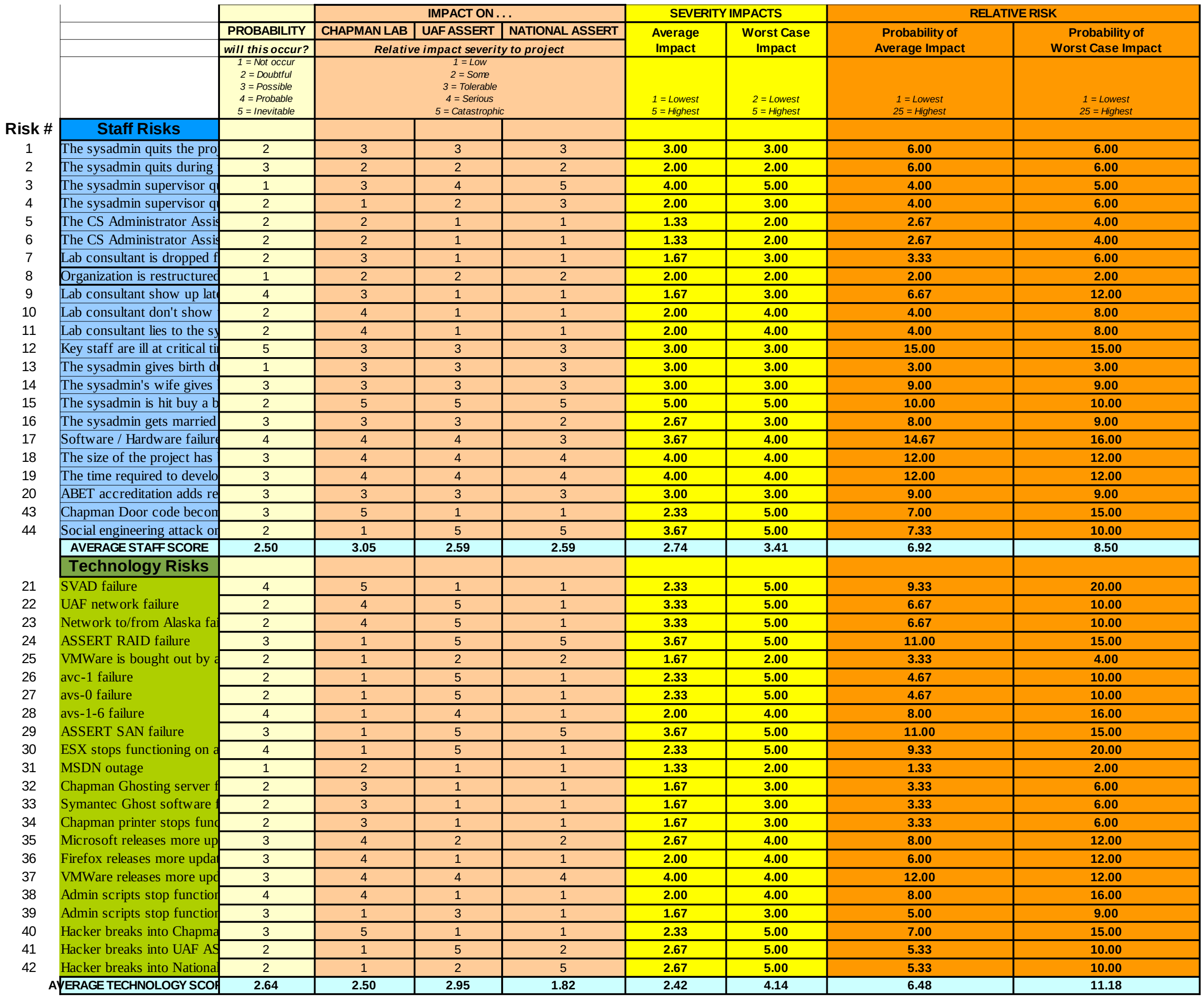
As in the qualitative risk analysis we will choose option three mentioned above to form our definition of "high" valued risks.

We begin by doubling the relative risk average and declaring any relative risk worst case that falls above that number as “high” and we get ten results. The average relative risk comes to about seven. We use that figure of seven and double it, giving us fourteen. So, when we take the relative risk worst case and declare any identified risks at or above fourteen we get ten results classified as “high”.

Only these ten identified risks will be considered the top priority for a Management Analysis and Disaster Recovery Plan.

Overall Project Risk

Below there are two matrices. The first divides the identified risks into staff and technology risks. The second divides the identified risks into internal and external risks. This will help the project team better understand where the risk is the greatest and where they should spend most of their time in coming up with a management analysis of the “high” priority risks.



			IMPACT ON . . .			SEVERITY IMPACTS		RELATIVE RISK	
		PROBABILITY	CHAPMAN LAB	UAF ASSERT	NATIONAL ASSERT	Average Impact	Worst Case Impact	Probability of Average Impact	Probability of Worst Case Impact
		will this occur?	*Relative impact severity to project*						
		1 = Not occur *2 = Doubtful* *3 = Possible* *4 = Probable* *5 = Inevitable*	*1 = Low* *2 = Some* *3 = Tolerable* *4 = Serious* *5 = Catastrophic*			*1 = Lowest* *5 = Highest*	*2 = Lowest* *5 = Highest*	*1 = Lowest* *25 = Highest*	*1 = Lowest* *25 = Highest*
Risk #	**Staff Risks**								
1	The sysadmin quits the proj	2	3	3	3	**3.00**	**3.00**	**6.00**	**6.00**
2	The sysadmin quits during	3	2	2	2	**2.00**	**2.00**	**6.00**	**6.00**
3	The sysadmin supervisor qu	1	3	4	5	**4.00**	**5.00**	**4.00**	**5.00**
4	The sysadmin supervisor qu	2	1	2	3	**2.00**	**3.00**	**4.00**	**6.00**
5	The CS Administrator Assis	2	2	1	1	**1.33**	**2.00**	**2.67**	**4.00**
6	The CS Administrator Assis	2	2	1	1	**1.33**	**2.00**	**2.67**	**4.00**
7	Lab consultant is dropped f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
8	Organization is restructured	1	2	2	2	**2.00**	**2.00**	**2.00**	**2.00**
9	Lab consultant show up late	4	3	1	1	**1.67**	**3.00**	**6.67**	**12.00**
10	Lab consultant don't show	2	4	1	1	**2.00**	**4.00**	**4.00**	**8.00**
11	Lab consultant lies to the sy	2	4	1	1	**2.00**	**4.00**	**4.00**	**8.00**
12	Key staff are ill at critical ti	5	3	3	3	**3.00**	**3.00**	**15.00**	**15.00**
13	The sysadmin gives birth du	1	3	3	3	**3.00**	**3.00**	**3.00**	**3.00**
14	The sysadmin's wife gives	3	3	3	3	**3.00**	**3.00**	**9.00**	**9.00**
15	The sysadmin is hit buy a b	2	5	5	5	**5.00**	**5.00**	**10.00**	**10.00**
16	The sysadmin gets married	3	3	3	2	**2.67**	**3.00**	**8.00**	**9.00**
17	Software / Hardware failure	4	4	4	3	**3.67**	**4.00**	**14.67**	**16.00**
18	The size of the project has	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
19	The time required to develo	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
20	ABET accreditation adds re	3	3	3	3	**3.00**	**3.00**	**9.00**	**9.00**
43	Chapman Door code becom	3	5	1	1	**2.33**	**5.00**	**7.00**	**15.00**
44	Social engineering attack or	2	1	5	5	**3.67**	**5.00**	**7.33**	**10.00**
	AVERAGE STAFF SCORE	**2.50**	**3.05**	**2.59**	**2.59**	**2.74**	**3.41**	**6.92**	**8.50**
	Technology Risks								
21	SVAD failure	4	5	1	1	**2.33**	**5.00**	**9.33**	**20.00**
22	UAF network failure	2	4	5	1	**3.33**	**5.00**	**6.67**	**10.00**
23	Network to/from Alaska fai	2	4	5	1	**3.33**	**5.00**	**6.67**	**10.00**
24	ASSERT RAID failure	3	1	5	5	**3.67**	**5.00**	**11.00**	**15.00**
25	VMWare is bought out by a	2	1	2	2	**1.67**	**2.00**	**3.33**	**4.00**
26	avc-1 failure	2	1	5	1	**2.33**	**5.00**	**4.67**	**10.00**
27	avs-0 failure	2	1	5	1	**2.33**	**5.00**	**4.67**	**10.00**
28	avs-1-6 failure	4	1	4	1	**2.00**	**4.00**	**8.00**	**16.00**
29	ASSERT SAN failure	3	1	5	5	**3.67**	**5.00**	**11.00**	**15.00**
30	ESX stops functioning on a	4	1	5	1	**2.33**	**5.00**	**9.33**	**20.00**
31	MSDN outage	1	2	1	1	**1.33**	**2.00**	**1.33**	**2.00**
32	Chapman Ghosting server f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
33	Symantec Ghost software f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
34	Chapman printer stops func	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
35	Microsoft releases more up	3	4	2	2	**2.67**	**4.00**	**8.00**	**12.00**
36	Firefox releases more updat	3	4	1	1	**2.00**	**4.00**	**6.00**	**12.00**
37	VMWare releases more upd	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
38	Admin scripts stop function	4	4	1	1	**2.00**	**4.00**	**8.00**	**16.00**
39	Admin scripts stop function	3	1	3	1	**1.67**	**3.00**	**5.00**	**9.00**
40	Hacker breaks into Chapma	3	5	1	1	**2.33**	**5.00**	**7.00**	**15.00**
41	Hacker breaks into UAF AS	2	1	5	2	**2.67**	**5.00**	**5.33**	**10.00**
42	Hacker breaks into National	2	1	2	5	**2.67**	**5.00**	**5.33**	**10.00**
	AVERAGE TECHNOLOGY SCOR	**2.64**	**2.50**	**2.95**	**1.82**	**2.42**	**4.14**	**6.48**	**11.18**

Table 15: Risk Management - Identified Risks 1

			IMPACT ON . . .			SEVERITY IMPACTS		RELATIVE RISK	
		PROBABILITY	CHAPMAN LAB	UAF ASSERT	NATIONAL ASSERT	Average Impact	Worst Case Impact	Probability of Average Impact	Probability of Worst Case Impact
		will this occur?	Relative likelihood this will occur						
		1 = Not occur 2 = Doubtful 3 = Possible 4 = Probable 5 = Inevitable	1 = Low 2 = Some 3 = Tolerable 4 = Serious 5 = Catastrophic			1 = Lowest 5 = Highest	2 = Lowest 5 = Highest	1 = Lowest 25 = Highest	1 = Lowest 25 = Highest
	Internal Risks								
1	The sysadmin quits the pro	2	3	3	3	**3.00**	**3.00**	**6.00**	**6.00**
2	The sysadmin quits during	3	2	2	2	**2.00**	**2.00**	**6.00**	**6.00**
3	The sysadmin's supervisor	1	3	4	5	**4.00**	**5.00**	**4.00**	**5.00**
4	The sysadmin's supervisor	2	1	2	3	**2.00**	**3.00**	**4.00**	**6.00**
5	The CS Administrator Assis	2	2	1	1	**1.33**	**2.00**	**2.67**	**4.00**
6	The CS Administrator Assis	2	2	1	1	**1.33**	**2.00**	**2.67**	**4.00**
7	Lab consultant is dropped f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
8	Organization is restructured	1	2	2	2	**2.00**	**2.00**	**2.00**	**2.00**
9	Lab consultant show up late	3	3	1	1	**1.67**	**3.00**	**5.00**	**9.00**
10	Lab consultant don't show	2	4	1	1	**2.00**	**4.00**	**4.00**	**8.00**
11	Lab consultant lies to the sy	2	4	1	1	**2.00**	**4.00**	**4.00**	**8.00**
12	Key staff are ill at critical ti	4	3	3	3	**3.00**	**3.00**	**12.00**	**12.00**
13	The sysadmin gives birth du	1	3	3	3	**3.00**	**3.00**	**3.00**	**3.00**
14	The sysadmin's wife gives	3	3	3	3	**3.00**	**3.00**	**9.00**	**9.00**
15	The sysadmin is hit buy a b	2	5	5	5	**5.00**	**5.00**	**10.00**	**10.00**
16	The sysadmin gets married	3	3	3	2	**2.67**	**3.00**	**8.00**	**9.00**
17	Software / Hardware failure	3	4	4	3	**3.67**	**4.00**	**11.00**	**12.00**
18	The size of the project has	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
19	The time required to develo	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
20	ABET accreditation adds re	3	3	3	3	**3.00**	**3.00**	**9.00**	**9.00**
21	SVAD failure	3	5	1	1	**2.33**	**5.00**	**7.00**	**15.00**
24	ASSERT RAID failure	3	1	5	5	**3.67**	**5.00**	**11.00**	**15.00**
26	avc-1 failure	2	1	5	1	**2.33**	**5.00**	**4.67**	**10.00**
27	avs-0 failure	2	1	5	1	**2.33**	**5.00**	**4.67**	**10.00**
28	avs-1-6 failure	3	1	4	1	**2.00**	**4.00**	**6.00**	**12.00**
29	ASSERT SAN failure	2	1	5	5	**3.67**	**5.00**	**7.33**	**10.00**
30	ESX stops functioning on a	3	1	5	1	**2.33**	**5.00**	**7.00**	**15.00**
32	Chapman Ghosting server f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
33	Symantec Ghost software f	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
34	Chapman printer stops func	2	3	1	1	**1.67**	**3.00**	**3.33**	**6.00**
38	Admin scripts stop function	4	4	1	1	**2.00**	**4.00**	**8.00**	**16.00**
39	Admin scripts stop function	3	1	3	1	**1.67**	**3.00**	**5.00**	**9.00**
43	Chapman Door code becom	3	5	1	1	**2.33**	**5.00**	**7.00**	**15.00**
44	Social engineering attack or	2	1	5	5	**3.67**	**5.00**	**7.33**	**10.00**
	AVERAGE INTERNAL SCORE	**2.44**	**2.71**	**2.76**	**2.26**	**2.58**	**3.68**	**6.31**	**9.03**
	External Risks								
22	UAF network failure	2	4	5	1	**3.33**	**5.00**	**6.67**	**10.00**
23	Network to/from Alaska fai	2	4	5	1	**3.33**	**5.00**	**6.67**	**10.00**
25	VMWare is bought out by a	2	1	2	2	**1.67**	**2.00**	**3.33**	**4.00**
31	MSDN outage	1	2	1	1	**1.33**	**2.00**	**1.33**	**2.00**
35	Microsoft releases more up	3	4	2	2	**2.67**	**4.00**	**8.00**	**12.00**
36	Firefox releases more updat	3	4	1	1	**2.00**	**4.00**	**6.00**	**12.00**
37	VMWare releases more upd	3	4	4	4	**4.00**	**4.00**	**12.00**	**12.00**
40	Hacker breaks into Chapma	3	5	1	1	**2.33**	**5.00**	**7.00**	**15.00**
41	Hacker breaks into UAF AS	2	1	5	2	**2.67**	**5.00**	**5.33**	**10.00**
42	Hacker breaks into National	2	1	2	5	**2.67**	**5.00**	**5.33**	**10.00**
	AVERAGE EXTERNAL SCORE	**2.30**	**3.00**	**2.80**	**2.00**	**2.60**	**4.10**	**6.17**	**9.70**

Table 16: Risk Management - Identified Risks 2

Quantified Probability of Meeting Project Objectives

The average score and relative risk in the above tables shows the quantified probability of meeting the overall project objectives.

Cost and Schedule Reserves

In the identified risk assessment matrix shown above we see that the relative risk (average) for staff risks is a 6.92, technology risks a 6.48, internal risks a 6.31, and the external risks a 6.17. The scale starts with one being the lowest and goes up to twenty-five being the highest. This project falls somewhere in the lower end of the risk scale with many different identified risks (not just "high" risks) which show that there is much to consider. Because of this we should plan for a 15% contingency in funding. We will not extend any contingency to the schedule planning because of the nature of this project having a hard beginning and hard finish dates.

Cost, Schedule, and Scope Targets

Our scope document lays out what the responsibilities are for the sysadmin. It is vitally important that if something is added to the scope that the sysadmin has the ability to actually do the work in the time allotted along with their other responsibilities. It is also important that the added tasks be in line with the scope document, meaning that a new item that falls outside of the scope or outside of the scope of those funding the sysadmin position be scrutinized for their addition. If scope additions (scope creep?) are added, then added funding for additional hours should also be considered.

Risk response planning

Strategies for Negative Risks

With all of the negative risks involved in this project it is important to avoid any staff turnover and to have SOPs ready when something fails. It is also important for key staff to remain healthy throughout the life of the project.

Strategies for Positive Risks

There have been no positive risks identified at this time.

Primary and Backup Strategies

The primary backup strategy for this project is to have SOPs for all software, hardware, and typical procedures completed before the semester begins. The backup strategy is to lessen the scope of work required in the scope document.

Assigning Responsibilities of Risks to People and Groups

Many risks fall squarely on the shoulders of the sysadmin. However, at some point these items are also the responsibility of the sysadmin's supervisor. This may occur if the sysadmin's supervisor has installed equipment and failed to document and give the sysadmin the SOP for the new hardware.

Risk monitoring and control

The following is a list of action items to be completed at the outset of the project as well as during Christmas break. By completing this list of action items twice during the project life-cycle we can help ensure that all risks are identified and mitigated through the management analysis.

- Track all identified risks to date
- Implement any risk responses which are necessary
- Document the occurrence of risk triggers from the semester
- Monitor "high" risks to the project
- Identify new risks to the project
- Ensure the execution of risk plans
- Evaluate the effectiveness of risk plans
- Develop new risk responses if necessary
- Collect and communicate risk status to supervisor
- Determine if scope assumptions are still valid
- Revisit low ranking or non-critical risks to see if risks responses need to be determined
- Take corrective action to adjust to the severity of actual risk events
- Look for unexpected effects or consequences of risk events
- Re-evaluate risk identification, qualification, and quantification when the project deviates from the intended scope and other planning documents
- Update risk plans

Appendix I.1: Identified Risks

Risk #1

Risk Type: staff turnover

Location: internal

Description: The sysadmin quits the project before May 31

Probability: 2 - doubtful

Chapman Lab Impact: 3 – tolerable

UAF ASSERT Impact: 3 – tolerable

National (-UAF) ASSERT Impact: 3 – tolerable

History: N/A

Risk #2

Risk Type: staff turnover

Location: internal

Description: The sysadmin quits during the off months of the project

Probability: 3 - possible

Chapman Lab Impact: 2 - some

UAF ASSERT Impact: 2 - some

National (-UAF) ASSERT Impact: 2 - some

History: Summer 2011; summer 2009

Risk #3

Risk Type: staff turnover

Location: internal

Description: The sysadmin's supervisor quits before May 31

Probability: 1 - not occur

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 4 - tolerable

National (-UAF) ASSERT Impact: 5 - tolerable

History: N/A

Risk #4

Risk Type: staff turnover

Location: internal

Description: The sysadmin's supervisor quits during the off months of the project

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 2 - Some

National (-UAF) ASSERT Impact: 3 - Tolerable

History: N/A

Risk #5

Risk Type: staff turnover

Location: internal

Description: The CS Administrator Assistant quits before May 31

Probability: 2 - doubtful

Chapman Lab Impact: 2 - some

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #6

Risk Type: staff turnover

Location: internal

Description: The CS Administrator Assistant quits during the off months of the project

Probability: 2 - doubtful

Chapman Lab Impact: 2 - some

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #7

Risk Type: staff turnover

Location: internal

Description: Lab consultant is dropped from schedule

Probability: 2 - doubtful

Chapman Lab Impact: 3 - some

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: Fall 2010

Risk #8

Risk Type: staff / management issues

Location: internal

Description: organization is restructured; different management is responsible for the project

Probability: 1 - not occur

Chapman Lab Impact: 2 - some

UAF ASSERT Impact: 2 - some

National (-UAF) ASSERT Impact: 2 - some

History: N/A

Risk #9

Risk Type: staff issues

Location: internal

Description: Lab consultant show up late for shifts

Probability: 3 - possible

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 – lo

History: Once a week

Risk #10

Risk Type: staff issues

Location: internal

Description: Lab consultant don't show up for their shifts

Probability: 2 - doubtful

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: Fall 2010

Risk #11

Risk Type: staff issues

Location: internal

Description: Lab consultant lies to the sysadmin, staff, or other faculty

Probability: 2 - doubtful

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: Fall 2010

Risk #12

Risk Type: staff health

Location: internal

Description: Key staff is ill at critical times in the project

Probability: 4 - probable

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 3 - tolerable

History: twice a semester average

Risk #13

Risk Type: staff health

Location: internal

Description: The sysadmin gives birth during the life of the project

Probability: 1 - not occur

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 3 - tolerable

History: N/A

Risk #14

Risk Type: staff health

Location: internal

Description: The sysadmin's wife gives birth during the life of the project

Probability: 3 - possible

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 3 - tolerable

History: Fall 2010

Risk #15

Risk Type: staff health

Location: internal

Description: The sysadmin is hit by a bus, is in a plane crash, or other life threatening event

Probability: 2 - doubtful

Chapman Lab Impact: 5 - catastrophic

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 5 - catastrophic

History: N/A

Risk #16

Risk Type: vacation / sick leave

Location: internal

Description: The sysadmin gets married during the life of the project and goes on a honeymoon

Probability: 3 - possible

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 2 - some

History: Fall 2011

Risk #17

Risk Type: vacation / sick leave

Location: internal

Description: Software / Hardware failure while sysadmin is on vacation or out sick

Probability: 3 - possible

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 4 - serious

National (-UAF) ASSERT Impact: 3 - tolerable

History: N/A

Risk #18

Risk Type: size underestimates

Location: internal

Description: The size of the project has been underestimated

Probability: 3 - possible

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 4 - serious

National (-UAF) ASSERT Impact: 4 - serious

History: N/A

Risk #19

Risk Type: time underestimate

Location: internal

Description: The time required to develop the project is underestimated

Probability: 3 - possible

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 4 - serious

National (-UAF) ASSERT Impact: 4 - serious

History: N/A

Risk #20

Risk Type: requirements change

Location: internal

Description: ABET accreditation adds responsibility to the sysadmin's plate

Probability: 3 - possible

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 3 - tolerable

History: Fall 2010

Risk #21

Risk Type: hardware

Location: internal

Description: SVAD failure

Probability: 3 - possible

Chapman Lab Impact: 5 - catastrophic

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: Spring 2011

Risk #22

Risk Type: network

Location: external

Description: UAF network failure

Probability: 2 - doubtful

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #23

Risk Type: hardware

Location: external

Description: Network to/from Alaska failure

Probability: 2 - doubtful

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 1 - low

History: 2008

Risk #24

Risk Type: hardware

Location: internal

Description: ASSERT RAID failure

Probability: 3 - possible

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 5 - catastrophic

History: Spring 2011

Risk #25

Risk Type: software

Location: external

Description: VMware is bought out by another company

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 2 - some

National (-UAF) ASSERT Impact: 2 - some

History: N/A

Risk #26

Risk Type: hardware

Location: internal

Description: avc-1 failure

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #27

Risk Type: hardware

Location: internal

Description: avs-0 failure

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #28

Risk Type: hardware

Location: internal

Description: avs-1-6 failure

Probability: 3 - possible

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 4 - serious

National (-UAF) ASSERT Impact: 1 - low

History: Spring 2011

Risk #29

Risk Type: hardware

Location: internal

Description: ASSERT SAN failure

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 5 - catastrophic

History: Spring 2011

Risk #30

Risk Type: software

Location: internal

Description: ESX stops functioning on avs-1-6

Probability: 3 - possible

Chapman Lab Impact: 1 – low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 1 - low

History: Spring 2011

Risk #31

Risk Type: software

Location: external

Description: MSDN outage

Probability: 1 - not occur

Chapman Lab Impact: 2 - some

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #32

Risk Type: hardware

Location: internal

Description: Chapman ghosting server failure

Probability: 2 - doubtful

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #33

Risk Type: software

Location: internal

Description: Symantec Ghost software failure

Probability: 2 - doubtful

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #34

Risk Type: hardware

Location: internal

Description: Chapman printer stops functioning

Probability: 2 - doubtful

Chapman Lab Impact: 3 - tolerable

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #35

Risk Type: software

Location: external

Description: Microsoft releases more updates than usual in a given project year

Probability: 3 - possible

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 2 - some

National (-UAF) ASSERT Impact: 2 - some

History: N/A

Risk #36

Risk Type: software

Location: external

Description: Firefox releases more updates than usual in a given project year

Probability: 3 - possible

Chapman Lab Impact: 4 - tolerable

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #37

Risk Type: software

Location: external

Description: VMware releases more updates than usual in a given project year

Probability: 3 - possible

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 4 - serious

National (-UAF) ASSERT Impact: 4 - serious

History: N/A

Risk #38

Risk Type: software

Location: internal

Description: Admin scripts stop functioning like normal on SVAD

Probability: 4 - probable

Chapman Lab Impact: 4 - serious

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: once a year

Risk #39

Risk Type: software

Location: internal

Description: Admin scripts stop functioning on AVC-1

Probability: 3 - possible

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 3 - tolerable

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #40

Risk Type: criminal

Location: external

Description: Hacker breaks into Chapman Lab computers

Probability: 3 - possible

Chapman Lab Impact: 5 - catastrophic

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #41

Risk Type: criminal

Location: external

Description: Hacker breaks into UAF ASSERT Servers

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 2 - some

History: N/A

Risk #42

Risk Type: criminal

Location: external

Description: Hacker breaks into National (-UAF) ASSERT Servers

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 2 - some

National (-UAF) ASSERT Impact: 5 - catastrophic

History: N/A

Risk #43

Risk Type: physical security

Location: internal

Description: Chapman Door code becomes known to non-staff / faculty

Probability: 3 - possible

Chapman Lab Impact: 5 - catastrophic

UAF ASSERT Impact: 1 - low

National (-UAF) ASSERT Impact: 1 - low

History: N/A

Risk #44

Risk Type: physical security

Location: internal

Description: Social engineering attack on UAF or National ASSERT labs

Probability: 2 - doubtful

Chapman Lab Impact: 1 - low

UAF ASSERT Impact: 5 - catastrophic

National (-UAF) ASSERT Impact: 5 - catastrophic

History: N/A

Risk #45

Risk Type: early / late completion

Location: internal

Description: Due to the nature of this project it cannot be completed early or late.

Probability: N/A

Chapman Lab Impact: N/A

UAF ASSERT Impact: N/A

National (-UAF) ASSERT Impact: N/A

History: N/A

Appendix I.2: Further Analysis of High Risks

Risk #12 – relative risk (worst case) – 15

Risk Type: staff health

Location: internal

Description: Key staff is ill at critical times in the project

Management Analysis: There are some things which cannot be controlled, only planned for. This is certainly the case for key staff coming down with a cold, the flu, or worse. Schedules may need to be pushed back when this occurs. However, perhaps the best plan of action is to plan for this type of risk at the outset of the project by pushing the due dates back for any tasks involved.

Risk #17 – relative risk (worst case) – 16

Risk Type: vacation / sick leave

Location: internal

Description: Software / Hardware failure while sysadmin is on vacation or out sick

Management Analysis: There are several things to keep in mind here. One, be sure to communicate with your supervisor that you'll be out of town on vacation. In this case the supervisor will be prepared in the event that something goes awry with the software / hardware. Two, be sure to have a standard operating procedure (SOP) for the employee left behind to fend for themselves. Included in the SOP should be phone numbers of potential support vendors in case of emergency. And three, be sure to include cross pollination procedures in your meetings and training so if someone is out another staff member can step in to fix things.

Risk #21 – relative risk (worst case) – 20

Risk Type: hardware

Location: internal

Description: SVAD failure

Management Analysis: If possible create backups of important data on regular intervals and check those backups after they are made to insure that they work properly when re-instituting them. It may also be possible to have a complete machine backup which can act as a hot swap item. This could be a completely different physical machine or a Virtual

Machine (VM). In either case the backup unit should be tested periodically to ensure its viability as a backup.

Risk #24 – relative risk (worst case) – 15

Risk Type: hardware

Location: internal

Description: ASSERT RAID failure

Management Analysis: In the world of RAID there should always be a hot swap backup in the rack and ready to go. As soon as any RAID device fails and the hot swap item is put into use, another hot swap unit should be ordered and placed in the rack for immediate use.

Risk #28 – relative risk (worst case) – 16

Risk Type: hardware

Location: internal

Description: avs-1-6 failure

Management Analysis: If one of these servers goes down it is possible to move the VMs from them to another host in the same cluster. This may take a little time depending on how many VMs are currently on the machine and if the new host is currently in heavy use. This could take just a few minutes to swap hosts or up to an hour or two.

Risk #29 – relative risk (worst case) – 15

Risk Type: hardware

Location: internal

Description: ASSERT SAN failure

Management Analysis: This particular SAN device is RAID'd so the same principles apply as Risk #24. In the world of RAID there should always be a hot swap backup in the rack and ready to go. As soon as any RAID device fails and the hot swap item is put into use, another hot swap unit should be ordered and placed in the rack for immediate use.

Risk #30 – relative risk (worst case) – 20

Risk Type: software

Location: internal

Description: ESX stops functioning on avs-1-6

Management Analysis: It is best to have physical access to the avs machines when working on them. An SOP should be provided for fixing possible problems such as the NIC won't keep its IP address or ESX can't find its storage arrays, etc. Emergency contact vendor numbers is also a must.

Risk #38 – relative risk (worst case) – 16

Risk Type: software

Location: internal

Description: Admin scripts stop functioning like normal on SVAD

Management Analysis: Having a good backup of SVAD is important and having a good backup of SVAD which is easily accessible and tested is even better. Also, a backup of the scripts involved is also a good idea. The first place to look for trouble would be the system log files to see if any software was installed recently which may be interfering with the scripts in question. Also, a good place to check is the user log to see if anyone was in the system recently (in relation to when the problem started).

Risk #40 – relative risk (worst case) – 15

Risk Type: criminal

Location: external

Description: Hacker breaks into Chapman Lab computers

Management Analysis: It is important to keep all software in the lab up-to-date so most holes have been patched. Also important is keeping the virus protection up-to-date for those students who like to visit questionable sites with potential malware, spyware, etc. This list may include the browsers on the system (typically Firefox, Internet Explorer, Chrome, or Opera) or any office type software (typically Microsoft or OpenOffice).

Risk #43 – relative risk (worst case) – 15

Risk Type: physical security

Location: internal

Description: Chapman Door code becomes known to non-staff / faculty

Management Analysis: It should be stressed that the door code is for faculty, staff, and lab consultants only. Students should never be given the code. This code should also be reset at the beginning of the project to ensure its secrecy.

Appendix J: Architecture Development

History

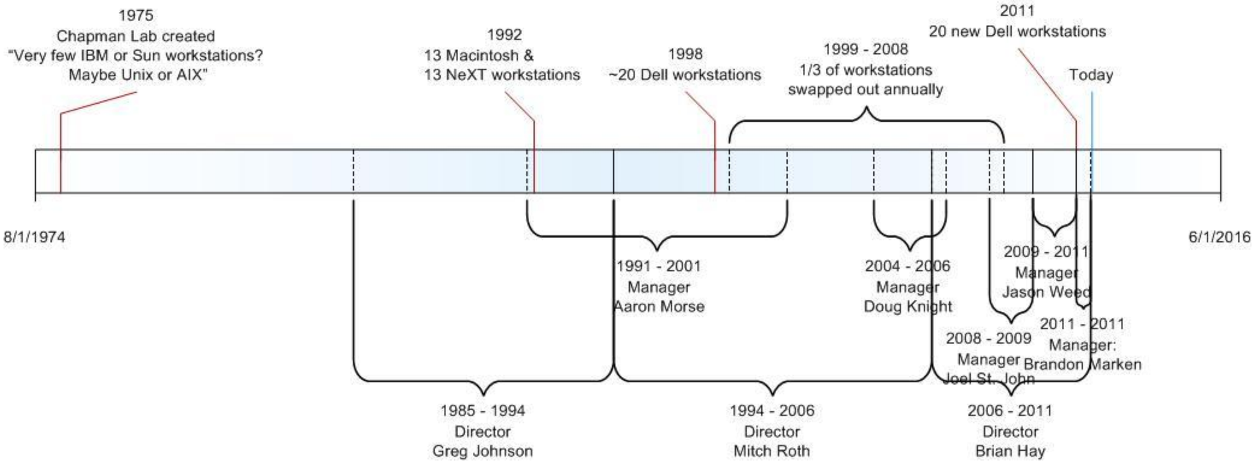


Figure 4: Chapman Lab Architecture Timeline

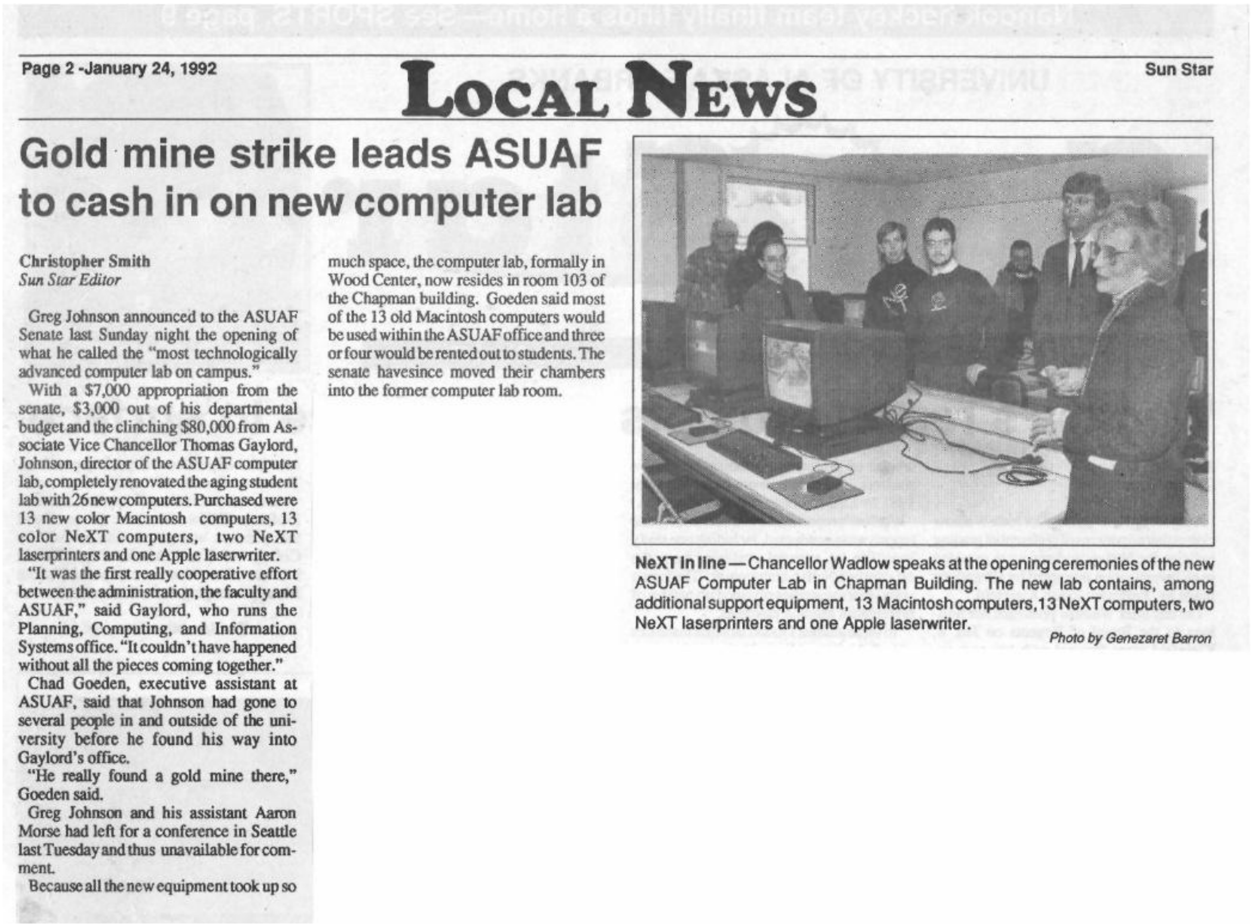
Hardware Details

- Created in mid 1970's for student homework / independent study with "Probably very few. Maybe IBM or Sun workstations? Maybe Unix or AIX?" [9]
- Complete overhaul in 1992 w/13 Macintosh and 13 NeXT workstations [12]
- Complete overhaul in early 2000's (pre-2003) with 20 new Dell workstations [11]
- Complete overhaul in 2011 with 20 new Dell workstations

Staff Details

- Directed by Greg Johnson from 19xx – 19xx [12]
- Directed by Mitch Roth from 199x – 2006 [10]
- Directed by Brian Hay from 2006 – Present

- Managed by Doug Knight from 2004 – 2006 [10]
- Managed by Joel St. John from 2008 – 2009
- Managed by Jason Weed from 2009 – 2011
- Managed by Brandon Marken from 2011
- Managed by Sam Cole from 2012 – Present



Page 2 -January 24, 1992

LOCAL NEWS

Sun Star

Gold mine strike leads ASUAF to cash in on new computer lab

Christopher Smith
Sun Star Editor

Greg Johnson announced to the ASUAF Senate last Sunday night the opening of what he called the "most technologically advanced computer lab on campus."

With a $7,000 appropriation from the senate, $3,000 out of his departmental budget and the clinching $80,000 from Associate Vice Chancellor Thomas Gaylord, Johnson, director of the ASUAF computer lab, completely renovated the aging student lab with 26 new computers. Purchased were 13 new color Macintosh computers, 13 color NeXT computers, two NeXT laserprinters and one Apple laserwriter.

"It was the first really cooperative effort between the administration, the faculty and ASUAF," said Gaylord, who runs the Planning, Computing, and Information Systems office. "It couldn't have happened without all the pieces coming together."

Chad Goeden, executive assistant at ASUAF, said that Johnson had gone to several people in and outside of the university before he found his way into Gaylord's office.

"He really found a gold mine there," Goeden said.

Greg Johnson and his assistant Aaron Morse had left for a conference in Seattle last Tuesday and thus unavailable for comment.

Because all the new equipment took up so much space, the computer lab, formally in Wood Center, now resides in room 103 of the Chapman building. Goeden said most of the 13 old Macintosh computers would be used within the ASUAF office and three or four would be rented out to students. The senate havesince moved their chambers into the former computer lab room.

NeXT in line—Chancellor Wadlow speaks at the opening ceremonies of the new ASUAF Computer Lab in Chapman Building. The new lab contains, among additional support equipment, 13 Macintosh computers,13 NeXT computers, two NeXT laserprinters and one Apple laserwriter.

Photo by Genezaret Barron

Figure 5: Sun Star article regarding Chapman Lab upgrade circa 1992

Business Goals

Declaring the business goals is the first step in achieving them. If these goals are not stated up front then when a new architecture is created it will not be known if the system has achieved what it set out to accomplish.

According to Dr. Jon Genetti, CS department head, "The main need is for hands-on labs for 103/201/202. " CS103 is a generic CS class which introduces students to low level programming such as html. CS201/202 are the core classes which teach students the C++ programming language. All three classes require students to write software in varying degrees and the Chapman Lab provides the hardware, software, and space for such work.

Some companies make use of cutting edge technologies while others choose to use older technology which is more known and stable. The CS Department has selected, for the Chapman Lab, the latter category, making use of slightly older technology which has been tested and well documented. The typical reasoning is that the older technologies have been tested thoroughly on other systems and has functioned just fine and unless there is a large benefit to switch to some newer technology there really isn't a good reason to switch. Stability plays a large role when making a decision to switch to new technology. If the software being used is mission critical and can't have much or any down time it becomes quite the precarious process to ensure that everything functions properly and that no functionality and processing time is lost in the switchover period.

Security also plays an important role. The security of a system which is older is more well known than that of a new piece of technology which may or may not have been hardened to the latest threats, both physical and network based. This is something that also needs to be considered when switching technologies.

The long term viability of a technology is also an important factor when considering a move from antiquated to newer technology. The CS department wants to be sure of the viability of a technology before it adopts it into their other systems. If that new piece of technology becomes obsolete in a few years it probably isn't going to be worth "upgrading" to because of the amount of time and dollars spent moving from that technology to something else, whether it is the older version of the technology or something even more bleeding edge.

The business goals of stability, security, and long term viability all play a role when choosing a new piece of technology. By taking this approach we can ensure that the Computer Science department has an available, reliable, easy to use, and secure student computer lab will aid in the development of the students which come through the universities doors.

The two goals mentioned in the Executive Summary should be taken to heart. They were: one, it shows to students and the occasional on-looker that the CS department knows and understands how to operate the technology that it teaches. And two, it provides a place for CS students to develop and hone their skill sets as programmers as well as software engineers. These are both important goals to achieve, for the student, the department, and the university.

Architecture Tradeoff Analysis Method (ATAM) [5]

Step 1 – What is ATAM

The Architecture Tradeoff Analysis Method (ATAM) is "a thorough and comprehensive way to evaluate a software architecture. [It] is so named because it reveals how well an architecture satisfies particular quality goals, and it provides insight into how quality goals interact—that is, how they trade off." [5] Typically an ATAM evaluation is done by a team of 3 – 5 people. In this case one (myself) will have to do.

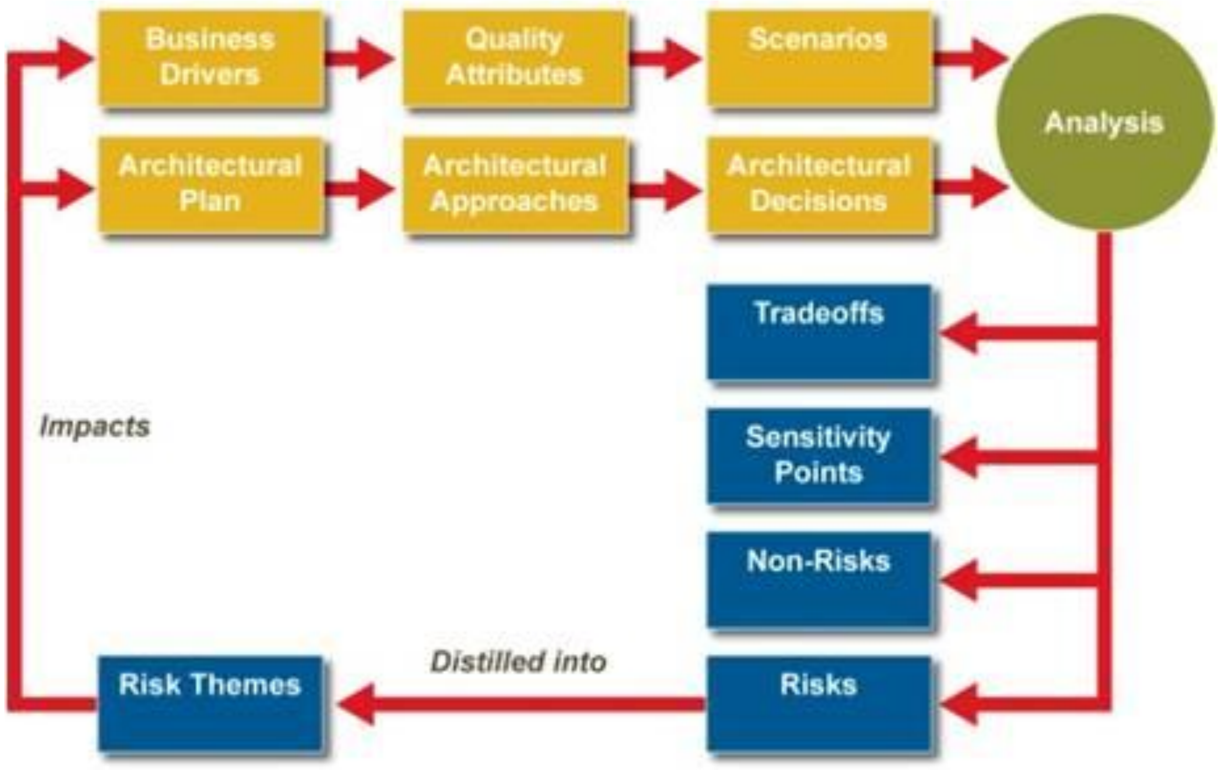


Figure 6: A Conceptual Flow of the ATAM [13]

Step 2 – Present Business Drivers

The presentation should describe the following from a business perspective:

- The system's most important functions
    - The Chapman Lab's primary function is to give a space for students to do their homework
- Any relevant managerial, economic, or political constraints
    - Managerial – The Chapman Lab has gone through many different managerial changes since its inception. The current (Fall 2011) hierarchy is the following:
        - Director – Brian Hay
        - Manager – Brandon Marken.
    - Economic – The primary economic constraint to the Chapman Lab is due to how many students sign up for CS Classes for each semester. Each student signing up for a CS class also has to pay for a lab fee. This lab fee is used to keep the Chapman Lab software and hardware current.
    - political – The last bullet point found in the Office of Information Technology online resource guide states a political constraint on the system: "Partisan political activity, e.g., sending email supporting a political party or group" [8]
- The business goals and context as they relate to the project
    - The system's business goal is providing undergraduates in CS103, CS201, and CS202 with a lab environment which they can use to write their software in. This lab provides the hardware, software, and space for students to do just that.
    - An underlying function, mentioned in the Executive Summary of this document, is also to show to students and the occasional on-looker that the C.S. department knows and understands how to operate the technology that it teaches.
- The major stakeholders
    - The University of Alaska Fairbanks
    - The CS department

    - The CS department head
    - Chapman Lab Director
    - Chapman Lab Manager
    - Computer Science and Software Engineering Students

- The architectural drivers (that is, the major quality attribute goals that shape the architecture)
    - availability
    - performance
    - security
    - usability
    - maintainability
    - scalability

Step 3 – Present Architecture

The Chapman Lab currently has 19 student workstations and 1 lab consultant workstation. These workstations are running Microsoft Windows 7 and run as part of the Chapman Lab domain. The Chapman Lab domain controller is a CentOS server called SVAD which runs Samba for the lab workstations. The lab computers also have Symantec Ghost installed for receiving new images from the Ghost server. The Ghost server is a VM which is stored on an external hard drive and can push new images to the lab workstations.

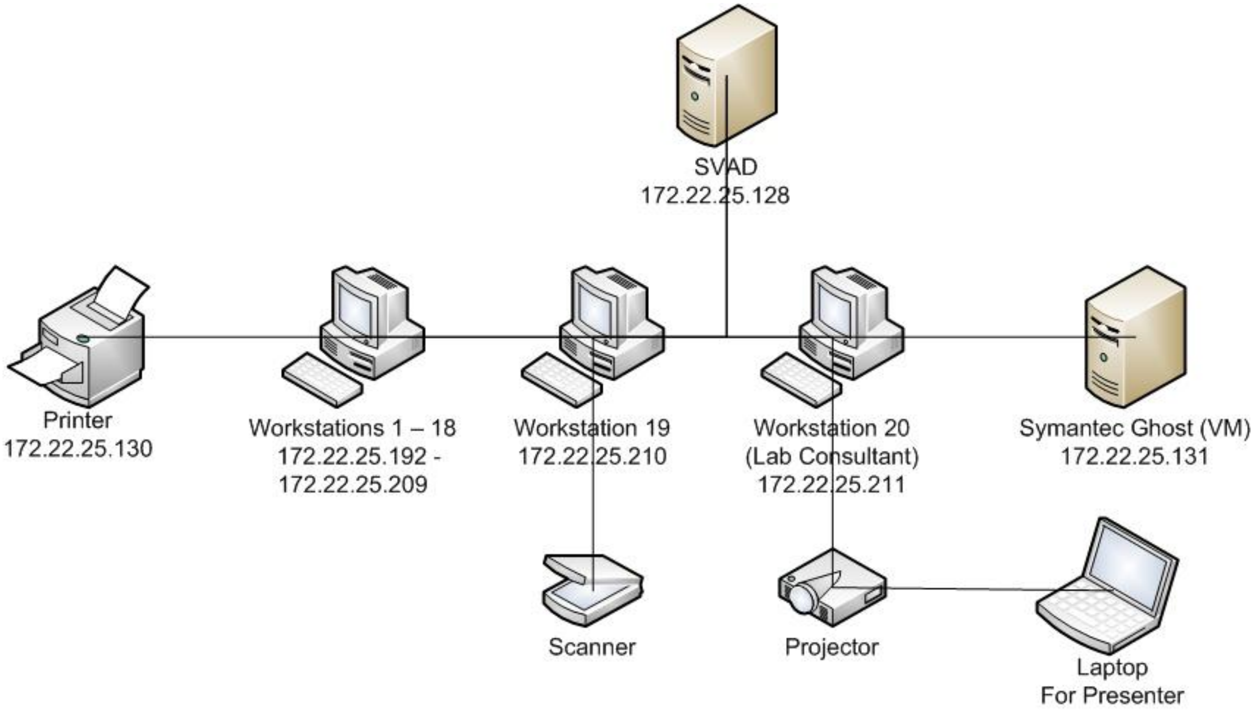


Figure 7: Chapman Lab Simplified Network Diagram

In order for a student to use the lab they must sign and date the Chapman Lab Authorization Form available at the front of the lab. Once they fill it out they give it to the lab consultant or the sysadmin who in turn creates their account. To create the student account one must login to SVAD and run a bash script (addDomainUser) to create the account. If the student needs to change their password (most likely they forgot it) one must run a separate bash script (resetDomainPass) to modify the password on the account.

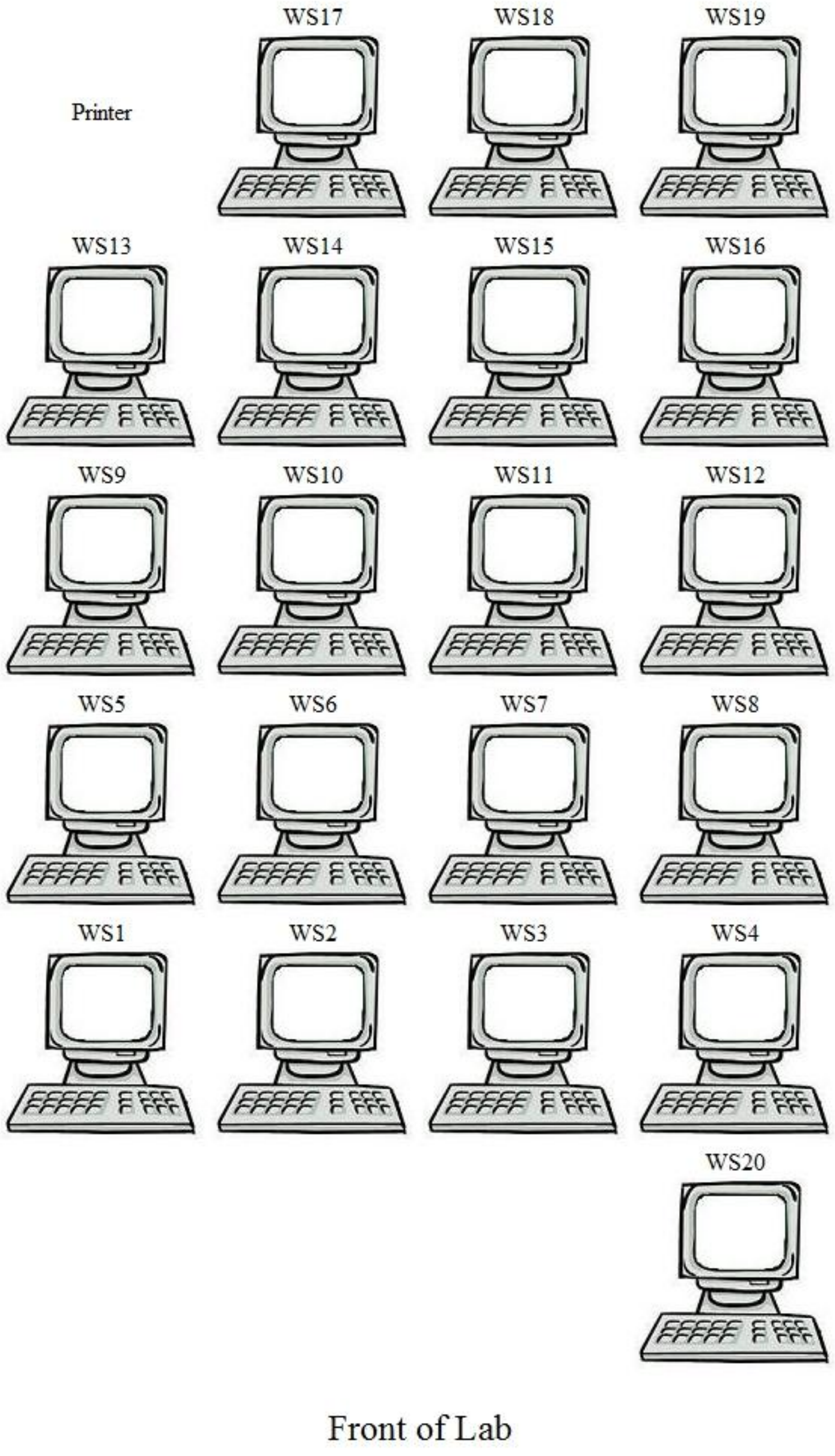


Figure 8: Chapman Lab Physical Layout

At the beginning of the academic year the sysadmin must login to SVAD to create the lab consultant accounts. This is done by hand using the command line. Currently no script, bash or otherwise, exists to automate this process.

To delete a student or lab consultant account there is currently no script to automate this process. This process is done by hand by checking the dates of last access (for student accounts) and by employment status (for lab consultant accounts).

Step 4 – Identify Architectural Approaches

We will consider six primary quality attributes which will help constrain our system. The six quality attributes we will consider are: availability, performance, security, usability, maintainability, and scalability. There are many other quality attributes which could be considered but we will limit the discussion to these six. To build our architecture approach and strategy we will use tactics [5] which will assist in the accomplishment of our quality attributes and our requirement specifications.

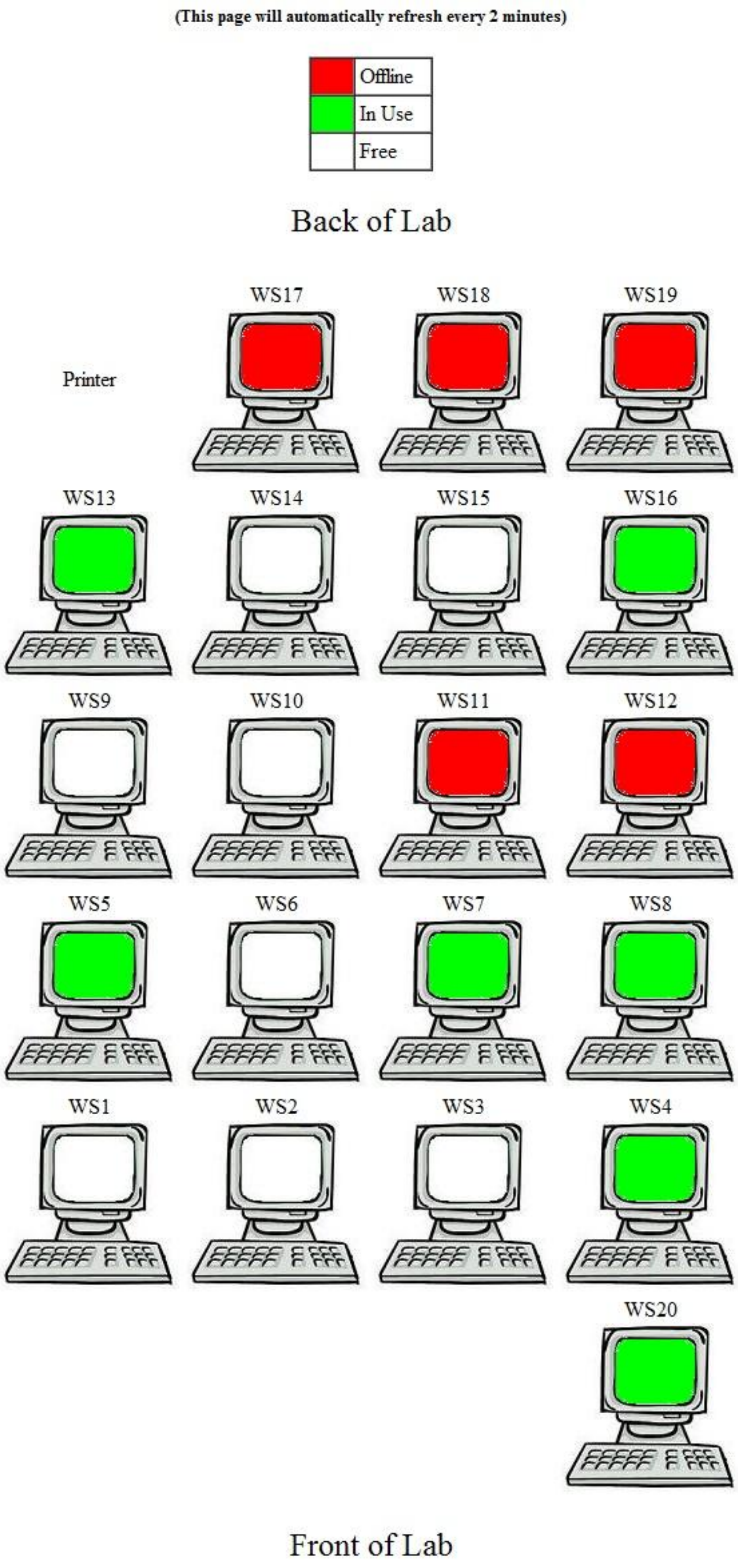
Availability Tactics

Availability has to do with the Chapman Lab being up and ready to serve its user base when a customer would like to use it, whether this is 8am or on a weekend. Because the "business" hours are fairly standard all major upgrades to the system, including Ghosting the workstations or hardware replacement, should happen prior to the semester beginning or after it has ended in order to avoid any downtime in availability. This would typically be in August, late December through early January, and in May once classes have been concluded.

Fault Detection

The Chapman Lab currently makes use of fault detection and specifically the use of the heartbeat tactics for each workstation. Each workstation has a script which is used to send a ping (heartbeat) to SVAD every 60 seconds. SVAD in turn makes note if the workstation has sent its heartbeat message and after 120 seconds it updates the web page status page shown below to reflect any changes.

In the figure below we see the physical layout of the Chapman Lab and which workstations are currently offline, which workstations in use and which workstations are free.



(This page will automatically refresh every 2 minutes)

🟥	Offline
🟩	In Use
⬜	Free

Back of Lab

Printer	WS17 🟥	WS18 🟥	WS19 🟥
WS13 🟩	WS14 ⬜	WS15 ⬜	WS16 🟩
WS9 ⬜	WS10 ⬜	WS11 🟥	WS12 🟥
WS5 🟩	WS6 ⬜	WS7 🟩	WS8 🟩
WS1 ⬜	WS2 ⬜	WS3 ⬜	WS4 🟩
			WS20 🟩

Front of Lab

Figure 9: Chapman Lab Heartbeat Results

The SVAD server which gives the workstations their capabilities does not currently make use of fault detection. It is recommended that a similar fault detection method be used for SVAD in order to minimize downtime in the lab. This could be accomplished by having SVAD send heartbeat messages to a separate server running outside of the Chapman Lab environment. If SVAD stopped sending heartbeat messages the server listening for SVAD's heartbeat could then send an email or text message to the sysadmins account or phone.

Fault Recovery

Using the active redundancy (aka hot restart) tactic for fault recovery would not be beneficial in this environment. If one of these systems goes down having a hot restart may just cycle the computer indefinitely because the issue is typically not with the workstation but rather with SVAD or the network in general. It would make good sense to have a complete backup of SVAD, meaning a separate server with the same hardware and software that could come online in the event that SVAD went down. This could be either an active or passive redundancy measure. This system would need to be thoroughly tested before it would be put into full service in order to avoid more downtime by figuring out the backup system while the primary system is also down. By making use of the active redundancy tactic we can ensure that availability remains extremely high throughout the "business" schedule.

Fault Prevention

Our fault prevention tactic would make use of the process monitor. This monitor would watch the Samba processes and look for any downtime or response delays on SVAD. Samba is the service which provides Windows services and drive mapping for each workstation. If this service stops responding users of the lab will not be able to access their networked "Z" drive. Fault prevention becomes a vital part of the availability equation when considering down time. SVAD also serves up web pages for www.cs.uaf.edu. In this event the process monitor should also be monitoring the Apache service(s) to ensure that it is also functioning properly. The process monitor should also have the ability to send email or text messages to the sysadmin in order to fix the issue quickly.

Performance Tactics

Performance has to do with the Chapman Lab running quickly and having a good response time so the user doesn't feel as if they are waiting for it to respond. Much of performance is objective. Some of the objectivity stems from the users own experience with their own system whether it be a laptop or a system at home or in their dorm room. One important aspect of performance is to keep the user informed of what the current situation is, and perhaps, how much longer it might take for completion of the task at hand. This is called managing the user's expectations.

There are two aspects of performance in the lab: the general computer speed and the network speed.

Resource Demand

The Chapman Lab workstations have many programs and services installed and running at any given time. In order to meet performance expectations in regards to general computer speed it is recommended that only essential services be run at start-up time and continue to be run in the background. The more services which run at boot-up (affecting boot-up time) and which are running concurrently with other operations, the more demand there is on the system resources and better likelihood that the user of the system will complain of performance issues.

The Chapman Lab has only one gateway to the Internet. This gateway commonly becomes a bottleneck when many users are making use of the system. This also becomes an issue when the sysadmin uses Symantec Ghost to replicate the Ghost image to all 20 workstations. Each image is approximately 30GB in size and consumes lots of bandwidth during its transfer time. If all 20 workstations are being images at once this can become a real issue. In a case such as this it is recommended that the sysadmin either; one, update the system in blocks, or two, update the system overnight or over the weekend in order to avoid downtime and availability issues.

Resource Management

As mentioned above there is only a single line in and out of the Chapman Lab for Internet connectivity and this line can become quickly saturated with dealing with more than a few users. One addition to the network architecture would be to double the gateways out of the lab and use a load balancer to keep traffic moving quickly. This will address the network speed issue but not general computer speed.

To address the general computer speed issue using the resource management tactic the lab could be upgraded with faster processors, more efficient operating systems, more and faster RAM, faster hard drives, and faster GPUs. This was done recently (May 2011) in order to meet user needs within the lab.

Resource Arbitration

The Chapman Lab currently makes use of the FIFO system or First-in/First-out system. Neither workstation, nor any type of particular work done on the computers is given priority over another. This is also the case for workstation 20, the lab consultant workstation at the front of the lab. Other systems to consider, although probably not very effective in this environment are, fixed priority scheduling, dynamic priority scheduling, and static scheduling.

Security Tactics

Security has to do with the Chapman Lab not being able to be compromised through malicious or accidental attacks. The lab must keep customer data private and out of the hands of would be criminals. Both SVAD and the 20 workstations are likely targets. SVAD is also a target because it's the primary backbone of the lab and the workstations because users generally use the system to do personal transactions along with school work. By installing a keystroke logger or other malicious software one could monitor the activities of personal transactions over the un-encrypted web traffic even though it is illegal.

Resisting Attacks

Maintaining data confidentiality should be a high priority of the lab. If word gets out that your data has been compromised your students would wonder about the integrity and knowledge base in the CS department. If your student data has been compromised your professional standing will be questioned and will not quickly return.

By ensuring that each workstation has current antivirus definitions, the latest version of Firefox, Chrome, or Opera, and helping students identify what questionable websites look like then the Chapman Lab would be more resistant to viruses, malware, spyware, and the like. It is not recommended to allow students to use Internet Explorer because of the known lag time between when bugs are discovered and when they are patched.

Detecting Attacks

This would be a great place for a graduate student to work on their project. Creating and installing their own intrusion detection software (IDS) over the Chapman Lab environment would provide great experience for the graduate student and provide the lab with a much needed intrusion detection system.

Usability Tactics

Usability has to do with the Chapman Lab being user-oriented and easy to use. This allows the user to find what they are looking for efficiently and move to the next item on their list of things to do.

Steve Krug

Steve wrote a book called "Don't Make Me Think" which is a fantastic book on website usability. His primary premise is that if a user of your website has to think too much to find what they are looking for they will give up and go elsewhere. There is so much competition on the Internet and it is so easy for a user to simply type in a competitors website address and go there instead. This is in stark contrast to the physical world where a customer may need to travel across town or even around the world to find what they are looking for. This book is a fantastic read for anyone wishing to become more acquainted with usability in general and website usability in particular.

We can take this same philosophy and apply it to the lab environment. If users have to struggle to find out how to accomplish their homework assignments in the lab they will eventually go elsewhere. By using the most common operating system in the world (Microsoft Windows) on one of the most recognized hardware vendors (Dell) the students will immediately recognize that the lab environment is an environment of which they are familiar and which they can accomplish their daily tasks with.

Jakob Nielsen

Jakob is considered the foremost guru on website usability and has been for quite some time. His book "Usability" is the Bible of website usability texts with numerous examples of past projects which he has worked on. The man has covered it all and continues to be a force to be reckoned with on his own site useit.com.

Jakob point out time and time again that if you place a user in an environment which is not standard, whether it is an operating system user interface or website, the user will get easily frustrated if the system does not keep the user informed of its progress. This idea plays into Steve's idea above, that users don't want to have to relearn a process which has already standardized on a different system.

Maintainability Tactics

Currently the Chapman Lab makes use of Symantec's Ghost software which allows a single server to push images to the workstations when there are major updates to be installed. This process works, however, it is not automatic and it does take 3 – 6 hours to complete which is not ideal.

As a potential solution to this issue it may work to use the ASSERT virtual lab infrastructure to provide virtual workstations to students. This infrastructure allows for updating of VMs in mass quantity (similar to Ghost), but it can also be automated (unlike Ghost).

Another part of maintainability is that of "understandability" or documentation [15]. This can be broken out into two categories: one, documentation of the system as a whole and two, documentation of the individual software pieces that make up the system. It is difficult to ascertain how to maintain a system when there is no documentation readily available. In that vein, during the spring of 2011, the current sysadmin (me at the time) began documenting processes which the sysadmin position regularly performed. This documenting process was not truly complete by the end of May 2011 and should be continued. If documentation is not considered a living process it quickly becomes outdated and related efforts are lost. To make this a living process the sysadmin must be diligent to piece this into their normal work flow process.

Scalability Tactics

The Chapman Lab is limited by its physicality and one room nature. By making use of the ASSERT virtual lab the CS department could scale its use of resources for students, not only for the CS department but for UA as a whole. Students from other departments and other major administrative units (MAUs) could also be included to make use of the new lab infrastructure. By converting its aim to the virtual world the university could reach a whole slew of students who, otherwise, may not have access to a lab. The ASSERT virtual lab would allow for students to have access to Microsoft, Adobe, and VMware products at the "touch of a button" as it were. If students needed additional VMs with separate options such as operating systems, etc this could also be provided through the use of the ASSERT Lab.

Step 5 – Generate Quality Attribute Utility Tree

We will stick to our previously mentioned six quality attributes when creating our utility tree. The quality attribute utility tree is listed in Appendix J.4.

Chapman Lab Scenarios

Scenarios are an important step in the development of a system architecture. Whether it's a website redesign project, a lab environment, or a shuttle mission scenarios can assist the designers of the system to see and better understand how the system may be used when it is complete.

From the business goals mentioned above we will create some system-specific scenarios. We will only mention a few scenarios for each quality attribute listed below. In reality there are an unlimited number of scenarios which could take place.

Availability Scenarios

Because of this the Chapman Lab must be available to students, faculty, and staff during the hours of 8am – 8pm Monday through Thursday, 8am – 5pm on Friday, and 10am – 5pm on Saturday and Sunday. These are the "business" hours of the Chapman Lab and the system must be operational during these hours.

Performance Scenarios

The Chapman Lab must perform reasonably well. If the system is available but takes 5 minutes to login/logout or the Internet connectivity is extremely slow users will get fed up and move on.

Security Scenarios

Only students which have paid their Computer Science Lab fee are allowed to use the Chapman Lab. Faculty and staff are welcome to use the Chapman Lab without the payment of the fee. It is also important to ensure that the workstations are free of the "wares" including but not limited to: spyware, malware, and adware. The workstations should also be free of rootkits, viruses, trojan horses, and worms. In other words the workstations should be free of malicious software which would hinder the student in any way. In some ways this is also a liability to UAF in that if one of these workstations were the originator of information being sent back to an unknown location due to malicious software being on the machine the student could sue the university if their identity or other private data was stolen.

Usability Scenarios

The Chapman Lab must be easy for the students, faculty, and staff to use. This is why the primary operating system (OS) is Microsoft Windows and not Linux based.

Maintainability Scenarios

If part of the system goes down, the network, SVAD, Symantec Ghost VM, etc, it is important to have the documentation showing how the system works as a whole and how each individual piece works on its own in support of the whole system. The sysadmin should allow time in their work flow to spend time on documentation of these aspects. The document which is created and maintained should be considered a living document and should be updated whenever the procedure, software, hardware, or personnel changes.

Scalability Scenarios

The Chapman Lab has 20 workstations for use and it is a rare exception when all of them are being used.

Prioritization

Scenarios with a priority of (low, *) or (*, low) will not be considered because either the priority is low or the architectural difficulty is considered easy. Time will only be spent on analyzing medium and higher priorities. The prioritized quality attributes are listed in Appendix J.5.

Step 6 – Analyze Architectural Approaches

In this step we will look at each prioritized scenario to continue capturing the architectural approach for each scenario. The goal is to convince ourselves that how we are solving the scenario is appropriate and will work to meet the requirements on the system as a whole [5]. The collection of sensitivities, tradeoffs, and risks can be found in Appendices J.1, J.2, and J.3.

Security Scenario 3 Priority 1	A user visits a questionable website which installs malware or spyware on the workstation without the user’s knowledge. The system has now been compromised and needs to be cleaned. Precautions also need to executed in order to prevent such vulnerabilities in the future.			
Attribute(s)	Security			
Environment	Normal Operations			
Stimulus	User visits website			
Response	Compromised website installs malware on Chapman Lab workstation			
Architectural Decisions	Sensitivity [14]	Tradeoff	Risk	
Install Anti-virus software	S9	T3	R1	
Keep OS updated	S9		R2	
Keep browsers updated	S9		R3	
Keep MS Office updated	S9		R4	
Reasoning	• Installing AV software will help to detect known malicious software. • Keeping the OS updated will help protect against malicious software attacking the underlying architecture of the system. • Keeping the browsers updated will help prevent malicious software from executing on the workstation. • Keeping MS Office updated will help prevent the workstation from becoming infected with malicious software.			

Table 17: ATAM - Architectural Approach - Priority 1

Availability Scenario 2 Priority 2	A user sits down in the Chapman Lab and tries to log-on to the network. The system receives the request and it automatically logs the user in if they are authenticated correctly.			
Attribute(s)	Availability			
Environment	Normal Operations			
Stimulus	User login			
Response	SVAD replies with "OK"			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Samba authentication against the Linux users on SVAD	S5, S6		R6	
Reasoning	• To provide "Z" drive access			

Table 18: ATAM - Architectural Approach - Priority 2

Availability Scenario 5 Priority 3	A student using the lab to do their homework tries to connect to the “Z” drive to access previous work.			
Attribute(s)	Availability			
Environment	Normal Operations			
Stimulus	User accesses “Z” drive			
Response	User sees their folders and files on the “Z” drive			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
“Z” drive stored on RAID'd setup on SVAD	S3, S4	T1	R6	
Reasoning	- RAID'd architecture for redundancy purposes			

Table 19: ATAM - Architectural Approach - Priority 3

Security Scenario 4 Priority 4	The user goes to a website which uses a security certificate from a known untrusted certificate authority (CA).			
Attribute(s)	Security			
Environment	Normal Operations			
Stimulus	User visits website			
Response	Possible infection of workstation			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Install Anti-virus software	S9	T3	R1	
Keep OS updated	S9		R2	
Keep browsers updated	S9		R3	
Keep MS Office updated	S9		R4	
Reasoning	• Installing AV software will help to detect known malicious software. • Keeping the OS updated will help protect against malicious software attacking the underlying architecture of the system. • Keeping the browsers updated will help prevent malicious software from executing on the workstation. • Keeping MS Office updated will help prevent the workstation from becoming infected with malicious software.			

Table 20: ATAM - Architectural Approach - Priority 4

Security Scenario 5 Priority 5	The user forgets to change their default password which is their student ID.			
Attribute(s)	Security			
Environment	Normal Operations			
Stimulus	User has new account created			
Response	User never changes their default password			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
None?		T2	R10	

Table 21: ATAM - Architectural Approach - Priority 5

Security Scenario 6 Priority 6	The student convinces the lab consultant that they are in a CS class and can use the lab when in fact they are not enrolled in a CS class nor have they paid their dues.			
Attribute(s)	Security			
Environment	Normal Operations			
Stimulus	User account creation process			
Response	User account is created			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
None?	S8			

Table 22: ATAM - Architectural Approach - Priority 6

<table>
<tr><td colspan="2">Maintainability Scenario 1
Priority 7</td><td colspan="4">Releases for Firefox, Internet Explorer, Chrome, or any other browser installed on the workstations needs to happen in a timely fashion, typically 72 hours from release date.</td></tr>
<tr><td>Attribute(s)</td><td colspan="5">Maintainability</td></tr>
<tr><td>Environment</td><td colspan="5">Normal Operations</td></tr>
<tr><td>Stimulus</td><td colspan="5">Browser company releases an update</td></tr>
<tr><td>Response</td><td colspan="5">Sysadmin downloads the update and installs it on the workstations or pushes it to the workstations with Symantec Ghost</td></tr>
<tr><td colspan="2">Architectural Decisions</td><td>Sensitivity</td><td>Tradeoff</td><td>Risk</td><td></td></tr>
<tr><td colspan="2">Part of the sysadmin's regular routine</td><td>S9</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Documented in the living SOP document</td><td>S9</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Keep browsers updated</td><td>S9</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Reasoning</td><td colspan="4">• By making software upgrades part of the regular maintenance program and following it the system as a whole will be more secure.
• By updating the living SOP document and following it the system as a whole will be more secure.
• Keeping the browsers updated will help prevent malicious software from executing on the workstation.</td></tr>
</table>

Table 23: ATAM - Architectural Approach - Priority 7

Maintainability Scenario 3 Priority 8	Adobe products, namely Reader, need to be updated within 72 hours from release date.			
Attribute(s)	Maintainability			
Environment	Normal Operations			
Stimulus	Software company releases an update			
Response	Sysadmin downloads the update and installs it on the workstations or pushes it to the workstations with Symantec Ghost			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Part of the sysadmin's regular routine	S9			
Documented in the living SOP document	S9			
Keep software updated	S9			
Reasoning	• By making software upgrades part of the regular maintenance program and following it the system as a whole will be more secure. • By updating the living SOP document and following it the system as a whole will be more secure. • Keeping the browsers updated will help prevent malicious software from executing on the workstation.			

Table 24: ATAM - Architectural Approach - Priority 8

Maintainability Scenario 5 Priority 9	The Windows OS should be set to automatically download the latest patches available from Microsoft. This generally occurs on the second Tuesday of the month. After patching occurs workstations should be checked for availability and performance. When pushing new images to the workstations via Symantec Ghost the new images should have the latest patches already installed to avoid the workstations having to re-download them again and thus slowing down the network.			
Attribute(s)	Maintainability			
Environment	Normal Operations			
Stimulus	Microsoft releases an update			
Response	Sysadmin downloads the update and installs it on the workstations or pushes it to the workstations with Symantec Ghost			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Part of the sysadmin's regular routine	S9			
Documented in the living SOP document	S9			
Keep OS updated	S9			
Reasoning	• By making software upgrades part of the regular maintenance program and following it the system as a whole will be more secure. • By updating the living SOP document and following it the system as a whole will be more secure. • Keeping the browsers updated will help prevent malicious software from executing on the workstation.			

Table 25: ATAM - Architectural Approach - Priority 9

<table>
<tr><td colspan="2">Maintainability Scenario 6
Priority 10</td><td colspan="4">In general, the workstations will not have their OS be upgraded until the spring semester is over. This would be done in conjunction with Symantec Ghost to avoid 20 separate installs.</td></tr>
<tr><td>Attribute(s)</td><td colspan="5">Maintainability</td></tr>
<tr><td>Environment</td><td colspan="5">Normal Operations</td></tr>
<tr><td>Stimulus</td><td colspan="5">Microsoft releases major update to their OS</td></tr>
<tr><td>Response</td><td colspan="5">Sysadmin downloads the update and installs it by pushing it to the workstations with Symantec Ghost</td></tr>
<tr><td colspan="2">Architectural Decisions</td><td>Sensitivity</td><td>Tradeoff</td><td>Risk</td><td></td></tr>
<tr><td colspan="2">Symantec Ghost to push images to workstations</td><td>S7</td><td></td><td>R5</td><td></td></tr>
<tr><td colspan="2">Reasoning</td><td colspan="4">• By using Symantec Ghost the Chapman Lab can be updated once as opposed to having each workstation done individually.</td></tr>
</table>

Table 26: ATAM - Architectural Approach - Priority 10

Maintainability Scenario 7 Priority 11	Documentation must become part of the sysadmin's work flow in order to assist with the living documentation process.			
Attribute(s)	Maintainability			
Environment	Normal Operations			
Stimulus	A change in any process or new software or hardware is changed			
Response	Sysadmin documents the new process and / or new software or hardware			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Create living document	S9			
Documentation becomes part of the sysadmin's routine	S9		R7	
Reasoning	• By updating the living SOP document and following it the system as a whole will be more secure. • By making software upgrades part of the regular maintenance program and following it the system as a whole will be more secure.			

Table 27: ATAM - Architectural Approach - Priority 11

Performance Scenario 1 Priority 12	After logging in the user is shown the desktop and is ready to do work in less than 5 seconds.			
Attribute(s)	Performance			
Environment	Normal Operations			
Stimulus	User login			
Response	Workstation logs into Chapman Lab domain and shows user the desktop			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Gbit switch for lab connectivity	S7		R8	
Reasoning	• Gbit instead of Mbit for speed purposes			

Table 28: ATAM - Architectural Approach - Priority 12

Usability Scenario 3 Priority 13	The user saves their homework to their “Z” drive on the network so they can access it from any workstation within the lab environment.			
Attribute(s)	Usability			
Environment	Normal Operations			
Stimulus	User saves their work			
Response	Workstation saves their work to the “Z” drive on SVAD			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
RAID'd drive on SVAD	S9	T2	R6	
Samba for user authentication			R6, R9	
Reasoning	• RAID'd system for redundancy purposes.			

Table 29: ATAM - Architectural Approach - Priority 13

Maintainability Scenario 4 Priority 14	VMware products need to be updated within 72 hours from release date.			
Attribute(s)	Maintainability			
Environment	Normal Operations			
Stimulus	Software company releases an update			
Response	Sysadmin downloads the update and installs it on the workstations or pushes it to the workstations with Symantec Ghost			
Architectural Decisions	Sensitivity	Tradeoff	Risk	
Part of the sysadmin's regular routine	S9			
Documented in the living SOP document	S9			
Keep software updated	S9			
Reasoning	• By making software upgrades part of the regular maintenance program and following it the system as a whole will be more secure. • By updating the living SOP document and following it the system as a whole will be more secure. • Keeping the software updated will help prevent malicious software from executing on the workstation.			

Table 30: ATAM - Architectural Approach - Priority 14

Step 7 – Brainstorm & Prioritize Scenarios

Step 7 in the ATAM can be considered a rehash of step 5. This process is typically done by 3 – 5 people and so this step will not be redone.

Step 8 – Analyze Architectural Approaches

Step 8 in the ATAM can be considered a rehash of step 6. This process is typically done by 3 – 5 people and so this step will not be redone.

Step 9 – Present Results

In the final step the following outputs have been compiled and ready to be presented:

- the architectural approaches documented
- the set of scenarios and their prioritization from the brainstorming
- the utility tree
- the risks discovered
- the sensitivity points and tradeoff points found

Cost Benefit Analysis Method (CBAM)

The CBAM picks up where the ATAM left off. That is, the CBAM takes the prioritized scenarios from the ATAM and allows the stakeholders and decision makers to find the return on investment (ROI) for each quality attribute which they deem relevant to their organizational plans. "[T]he idea behind the CBAM is that architectural strategies (a collection of architectural tactics) affect the quality attributes of the system and these in turn provide system stakeholders with some benefit." [5]

The CBAM shows the stakeholders what the ROI is for each architectural strategy they might wish to pursue and better equips them to make the best decision for their company.

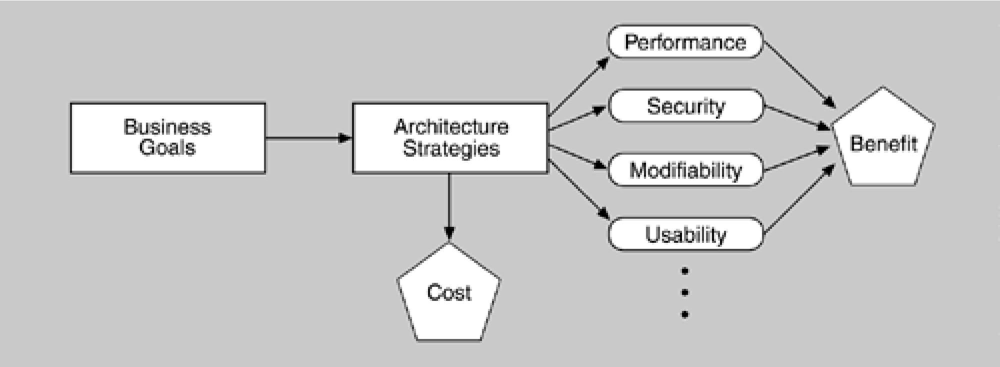


Figure 10: Context for the CBAM

The CBAM, similar to the ATAM, is broken down into 9 steps shown in the figure below.

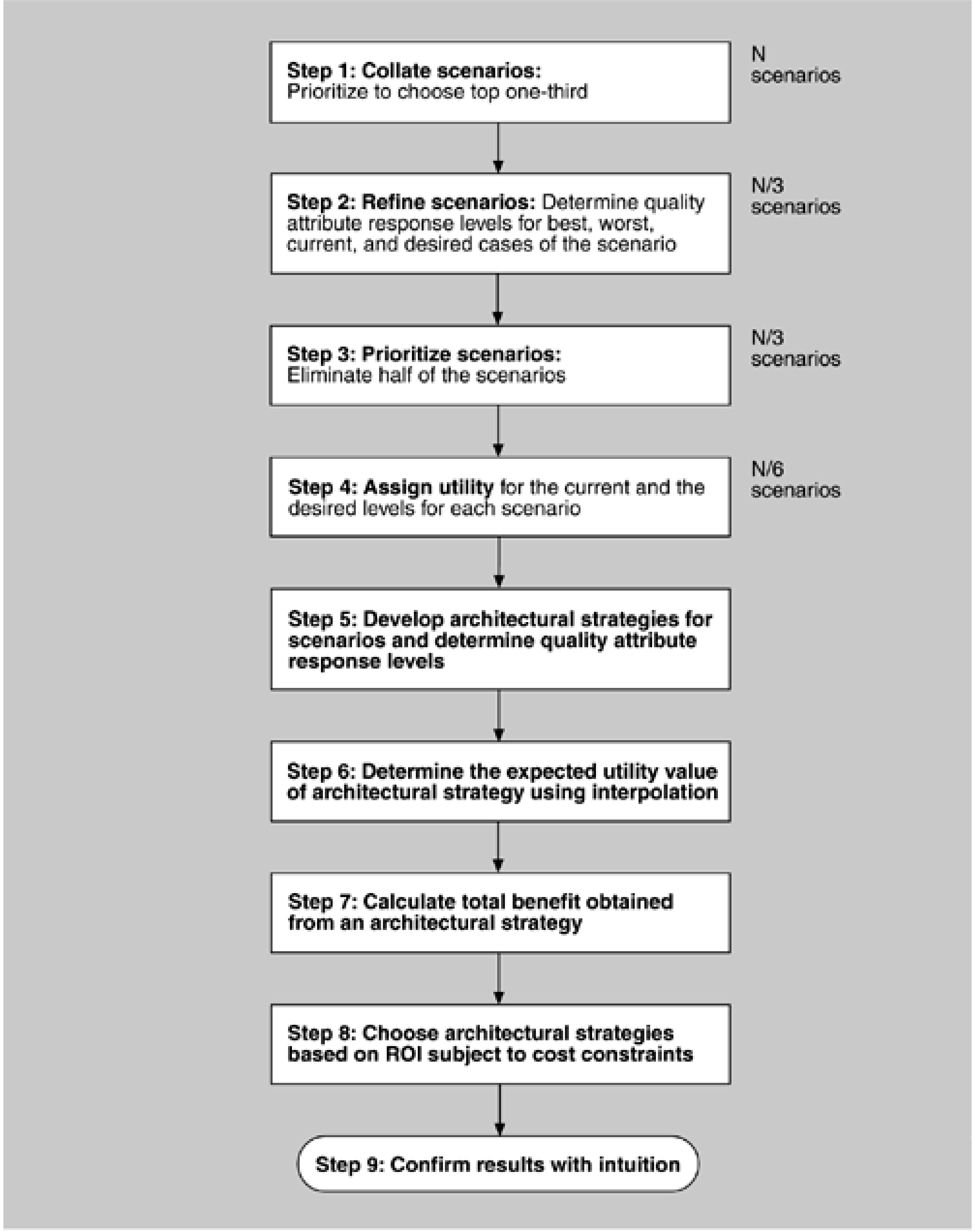

Step 1: Collate scenarios: Prioritize to choose top one-third
N scenarios
Step 2: Refine scenarios: Determine quality attribute response levels for best, worst, current, and desired cases of the scenario
N/3 scenarios
Step 3: Prioritize scenarios: Eliminate half of the scenarios
N/3 scenarios
Step 4: Assign utility for the current and the desired levels for each scenario
N/6 scenarios
Step 5: Develop architectural strategies for scenarios and determine quality attribute response levels
Step 6: Determine the expected utility value of architectural strategy using interpolation
Step 7: Calculate total benefit obtained from an architectural strategy
Step 8: Choose architectural strategies based on ROI subject to cost constraints
Step 9: Confirm results with intuition

Figure 11: Process flow diagram for the CBAM

Step 1 – Collate Scenarios

The prioritized ATAM scenarios mentioned above will also be used in the CBAM. These prioritized ATAM scenarios can be found in Appendix J.5.

In this step we find what the response goals for each scenario are. By working through the current, desired, and best response goals for each scenario we can begin to see how the system currently works and how we would like to see it perform. The response goals can be found in Appendix J.6.

Step 2 – Refine Scenarios

Step 2 in the CBAM can be considered a rehash of step 1. This process is typically done by 3 – 5 people and so this step will not be redone.

Step 3 – Prioritize Scenarios

In the third step the project stakeholders vote for which scenario they feel is the most important. These votes can be seen in Appendix J.7.

Typically voting is done by the stakeholders of the project which would probably include the C.S. department head, the director of the lab, and the sysadmin. However, in this document voting was done by one person alone. The primary quality attributes which we are concerned with here are availability which scenarios 2, 3, and 13 deal with exclusively. Scenarios 1, 4, and 6 have to do with security which is our secondary quality attribute we will need to look at.

Step 4 – Assign Utility

In step four the stakeholders determines the utility value for the worst, current, desired, and best case scenarios.

		Utility Goals			
Scenario	**Votes**	**Worst**	**Current**	**Desired**	**Best**
1	10	75	85	100	100
2	15	75	85	100	100
3	15	85	99	100	100
4	10	80	99	100	100
5	5	50	75	90	100
6	10	80	90	100	100
7	5	50	60	90	100
8	5	50	60	90	100
9	5	50	60	90	100
10	5	0	99	100	100
11	5	75	75	100	100
12	0	20	50	70	100
13	15	75	85	100	100
14	0	50	60	90	100

Table 31: CBAM - Utility Goals

Step 5 – Develop Architectural Strategies for Scenarios & Determine their Expected Quality Attribute Response Levels

Strategy	Name	Description	Scenarios Affected	Current Response	Expected Response
1	Duplicate SVAD	Availability - Make a duplicate of SVAD that can be used if a fault detection occurs	2, 3	5% hung	0% hung
2	Integrate with Registrars office	Security - Integrate scripts to pull students who are registered and paid, and therefore eligible for a Chapman Lab account	6	N/A	100% check
3	Deploy IDS	Security - Deploy an intrusion detection system (IDS) for Chapman Lab	1, 4	N/A	N/A
4	Lab virtualization	Availability / Maintainability - Use a virtualized lab running off of the ASSERT hardware for Chapman Lab	2, 3, 4, 7, 8, 9, 14	N/A	N/A
5	Integrate documentation	Maintainability - Integrate all aspects of the sysadmin's role with documenting both old and new procedures	11	50%	100%
6	Scripting	Maintainability - Write and deploy scripts which check for updates for browsers, MS Office, and VMware products	7, 8, 14	N/A	100% check
7	Automate User Accounts	Maintainability – Write a web based user account setup	5, 6	30% fail	100%

Table 32: CBAM - Architectural Strategies

Step 6 – Determine the Utility of the "Expected"

Strategy	Name	Scenarios Affected	Current Utility	Low Expected Utility	High Expected Utility
1	Duplicate SVAD	2	85	95	100
		3	99	99	100
2	Integrate with Registrars office	6	90	97	100
3	Deploy IDS	1	85	95	100
		4	99	99	100
4	Lab virtualization	2	85	95	100
		3	99	99	100
		4	99	99	100
		7	60	80	90
		8	60	80	90
		9	60	80	90
		14	60	80	90
5	Integrate documentation	11	75	90	100
6	Scripting	7	60	80	90
		8	60	80	90
		14	60	80	90
7	Automate User Accounts	5	75	85	90
		6	90	97	100

Table 33: CBAM - Expected Utility

Step 7 – Calculate the Total Benefit Obtained from an Architectural Strategy

Strategy	Scenario Affected	Scenario Weight	Low Raw Architectural Strategy Benefit	High Raw Architectural Strategy Benefit	Low Normalized Architectural Strategy Benefit	High Normalized Architectural Strategy Benefit	Low Total Architectural Strategy Benefit	High Total Architectural Strategy Benefit
1	2	15	10	15	150	225	150	240
	3	15	0	1	0	15		
2	6	10	7	10	70	100	70	100
3	1	10	10	15	100	150	100	160
	4	10	0	1	0	10		
4	2	15	10	15	150	225	450	700
	3	15	0	1	0	15		
	4	10	0	1	0	10		
	7	5	20	30	100	150		
	8	5	20	30	100	150		
	9	5	20	30	100	150		
	14	0	20	30	0	0		
5	11	5	15	25	75	125	75	125
6	7	5	20	30	100	150	200	300
	8	5	20	30	100	150		
	14	0	20	30	0	0		
7	5	5	10	15	50	125	120	225
	6	10	7	10	70	100		

Table 34: CBAM - Total Benefits

Step 8 – Choose Architectural Strategies Based on ROI Subject to Cost and Schedule Constraints

Strategy	Est. Cost	Low Total Strategy Benefit	High Total Strategy Benefit	Low Strategy ROI	High Strategy ROI	Low Strategy Rank	High Strategy Rank
1	$10,000	150	240	0.015	0.024	2	3
2	$5,000	70	100	0.014	0.020	4	5
3	$10,000	100	160	0.01	0.016	6	6
4	$50,000	450	700	0.009	0.014	7	7
5	$5,000	75	125	0.015	0.025	2	2
6	$7,500	200	300	0.027	0.040	1	1
7	$10,000	120	225	0.0120	0.0225	5	4

Table 35: CBAM - ROI

Step 9 – Confirm Results with Intuition

When we look at the results from Step 8 it becomes fairly obvious that the scripting (strategy 6) and integrate documentation (strategy 5) are going to be the best. The scripting will allow for large benefits in multiple scenarios and the document integration is easy to do with decent benefits.

The last two strategies to consider are either very expensive (strategy 4) or don't have much of a benefit compared to its cost (strategy 3).

Architecture Development Final Conclusions

By combining the Architecture Tradeoff Analysis Method and the Cost Benefit Analysis Method we have come up with a number of overall benefits to the project.

From the ATAM we gain four major things. One, we now know what the risks, tradeoffs, and sensitivity points are to the project as a whole. Two, we now know what the system quality attributes are and how they relate to the project. Three, we also have scenarios which correspond with those quality attributes. Lastly, we also have a utility tree to which ties together our quality attributes and scenarios.

From the CBAM the biggest gain is the fact that we now have an ordering of architectural strategies based on the predicted ROI.

By working through the ATAM and CBAM processes there are other intangible benefits as well. The "process[es] provides a great deal of structure to what is always largely unstructured discussions, where requirements and architectural strategies are freely mixed and where stimuli and response goals are not clearly articulated.: [5]

The #1 ROI was the strategy of scripting. This had to do with the non-functional quality attribute of maintainability. The scripting strategy would allow the sysadmin to automatically check for updates to web browsers, MS Office, and VMware products without having to remember to do the task themselves. This saves time and effort on the sysadmins part and does not take many dollars up front to complete.

The #2 ROI was for the strategy of integrating documentation with the sysadmin's role. Again, this strategy helps meet the maintainability quality attributes of the system.

Both the #1 and #2 strategies have to do with maintainability but at an even more basic level of understanding lies its primary attribute, that of reuse. In both scenarios of integrating documentation and scripting lies the heart of the reuse attribute. A procedure is something that is done over and over by numerous people and scripting for updates automatically should be a no brainer. Both have to do with reuse and taking what we have learned and making it simpler for others. This is why, I believe, these two scenarios became the #1 and #2 strategies based on their ROI. They were easy to implement and they have a large impact on the project as a whole.

Appendix J.1: ATAM Sensitivities

1. Chapman Lab Internet connectivity is sensitive to the UAF network being run properly.
2. Chapman Lab connectivity speed is sensitive to how many students are using the lab.
3. Chapman Lab "Z" drive access is sensitive to the SVAD server being operational.
4. Chapman Lab "Z" drive access is sensitive to Samba being operational.
5. User login is sensitive to the SVAD server being operational.
6. User login is sensitive to the Linux users and groups being administered correctly.
7. Pushing OS images quickly with Symantec Ghost is sensitive to the Gbit connection being overwhelmed.
8. User account creation is sensitive to users telling the truth regarding their class schedule and payment of lab fee.
9. Software and OS patches are sensitive to the sysadmin following the prescribed timeline from the living SOP document.

Appendix J.2: ATAM Tradeoffs

1. System complexity increases with the RAID'd setup.
2. Give users more challenging passwords to begin with upon user creation.
3. By installing AV software the workstation performance will be impacted by an estimated 5 – 10% depending on what type of scans the AV software is executing.

Appendix J.3: ATAM Risks

1. Anti-virus software definitions are out of date.
2. OS is not patched or upgraded.
3. Browsers are old and not patched with latest releases.
4. MS Office is not patched or upgraded.
5. Symantec Ghost fails to update workstations for an unknown reason.
6. SVAD failure
7. Sysadmin forgets to update living SOP document
8. Switch failure
9. Samba service failure
10. User account is hacked.

Appendix J.4: ATAM Quality Attribute Utility Tree

Quality Attributes	**Attribute Refinement**	**Scenarios**	**Decision Makers Priority (High, Medium, Low)**	**Architect Difficulty Priority (High, Medium, Low)**
Availability	business hours (i.e. normal operating times)	**Availability scenario 1** The doors to the lab must be open to allow students into the Chapman Lab during the following hours: M – R: 8am – 8pm F: 8am – 5pm S – S: 10am - 4pm	High	Low
	Chapman Lab network availability	**Availability scenario 2** A user sits down in the Chapman Lab and tries to log-on to the network. The system receives the request and it automatically logs the user in if they are authenticated correctly.	High	Medium
	Workstation availability	**Availability scenario 3** The Chapman Lab workstations must be available during the following hours: M – R: 8am – 8pm F: 8am – 5pm S – S: 10am - 4pm	High	Low
	Internet availability	**Availability scenario 4**	High	Low

		A student using the lab to do their homework tries to connect to a website for further instruction.		
		Availability scenario 5 A student using the lab to do their homework tries to connect to the "Z" drive to access previous work.	High	Medium
	OS stability	**Availability scenario 6** After using the workstation for awhile the system crashes because of an unknown error. The blue screen of death appears and the user must reboot to get back into the system all the while their unsaved work is lost.	High	Low
	Program stability	**Availability scenario 7** A student tries using a program for their homework (MS Word, MS Excel, OpenOffice, Visual Studio, etc)	High	Low
Performance	Login response time	**Performance scenario 1** After logging in the user is shown the desktop and is ready to do work in less than 5 seconds.	Medium	Medium
	Startup time	**Performance scenario 2** The student turns the workstation on and the login screen is available in less than 30 seconds.	Medium	Low
	Internet speed	**Performance scenario 3**	Medium	Low

		The user must download a large file from the Internet for a project. This 200MB file takes just 5 minutes to download.		
	Computing power	**Performance scenario 4** The user must compile a large block of source code. This process takes only a few seconds to complete allowing the user to use their program or find the bugs within it in a reasonable time frame.	Medium	Low
		Performance scenario 5 The user opens many programs to multitask including – MS Word, MS Excel, Firefox, and Visual Studio.	Medium	Low
	Printer response time and speed	**Performance scenario 6** The user sends a document to the printer. The printer responds by printing the document in less than 10 seconds.	Medium	Low
Security	Authentication – accepted	**Security scenario 1** A user sits down in the Chapman Lab and types their credentials into the prompt on the computer screen. The system receives the request and automatically logs the user in if they are authenticated correctly.	High	Low
	Authentication – denied	**Security scenario 2** A user sits down in the Chapman Lab	High	Low

		and types their credentials into the prompt on the computer screen. The system receives the request and informs the user that their username and password are incorrect or have been revoked.		
	Data integrity	**Security scenario 3** A user visits a questionable website which installs malware or spyware on the workstation without the users knowledge. The system has now been compromised and needs to be cleaned. Precautions also need to executed in order to prevent such vulnerabilities in the future.	High	High
		Security scenario 4 The user goes to a website which uses a security certificate from a known untrusted certificate authority (CA).	High	Medium
	Credentialing	**Security scenario 5** The user forgets to change their default password which is their student ID.	High	Medium
		Security scenario 6 The student convinces the lab consultant that they are in a CS class and can use the lab when in fact they are not enrolled in a CS class nor have they paid their dues.	High	Medium
Usability	Assistance	**Usability scenario 1**	Medium	Low

		The user logs into the system and tries to accomplish a new task such as creating drop shadows in Adobe Photoshop. The user loads the program and searches the help feature for their answer.		
	Proficiency	**Usability scenario 2** The user does their homework in Microsoft Word and Excel.	Low	Low
	Accessibility	**Usability scenario 3** The user saves their homework to their "Z" drive on the network so they can access it from any workstation within the lab environment.	Medium	Medium
Maintainability	Software Patches	**Maintainability scenario 1** Releases for Firefox, Internet Explorer, Chrome, or any other browser installed on the workstations needs to happen in a timely fashion, typically 72 hours from release date.	High	Medium
		Maintainability scenario 2 Minor updates to Microsoft Office products are a common occurrence. Workstations need to be updated within one week of release.	Low	Medium
		Maintainability scenario 3 Adobe products, namely Reader, need	High	Medium

		to be updated within 72 hours from release date.		
		Maintainability scenario 4 VMware products need to be updated within 72 hours from release date.	Medium	Medium
		Maintainability scenario 5 Minor updates to Microsoft Visual Studio happen on occasion. Workstations need to be updated within one week of release.	Low	Medium
	Operating System updates	**Maintainability scenario 5** The Windows OS should be set to automatically download the latest patches available from Microsoft. This generally occurs on the second Tuesday of the month. After patching occurs workstations should be checked for availability and performance. When pushing new images to the workstations via Symantec Ghost the new images should have the latest patches already installed to avoid the workstations having to re-download them again and thus slowing down the network.	High	Medium
	Operating System upgrades	**Maintainability scenario 6** In general, the workstations will not be upgraded in this manner until the spring semester is over. This would be	Medium	High

		done in conjunction with Symantec Ghost to avoid 20 separate installs.		
	Documentation	**Maintainability scenario 7** Documentation must become part of the sysadmin's work flow in order to assist with the living documentation process.	High	Medium
Scalability	Growing the system	**Scalability scenario 1** The Chapman Lab has 20 workstations for use and it is a rare exception when all of them are being used.	Low	Low

Table 36: ATAM - Chapman Lab Utility Tree

Appendix J.5: ATAM Prioritized Chapman Lab Utility Tree

Overall Priority	Quality Attributes	Attribute Refinement	Scenarios	Decision Makers Priority (High, Medium, Low)	Architect Difficulty Priority (High, Medium, Low)
1	Security	Data integrity	**Security scenario 3** A user visits a questionable website which installs malware or spyware on the workstation without the users knowledge. The system has now been compromised and needs to be cleaned. Precautions also need to executed in order to prevent such vulnerabilities in the future.	High	High
2	Availability	Chapman Lab network availability	**Availability scenario 2** A user sits down in the Chapman Lab and tries to log-on to the network. The system receives the request and it automatically logs the user in if they are authenticated correctly.	High	Medium
3			**Availability scenario 5** A student using the lab to do their homework tries to connect to the "Z" drive to access previous work.	High	Medium
4	Security		**Security scenario 4**	High	Medium

			The user goes to a website which uses a security certificate from a known untrusted certificate authority (CA).		
5		Credentialing	**Security scenario 5** The user forgets to change their default password which is their student ID.	High	Medium
6			**Security scenario 6** The student convinces the lab consultant that they are in a CS class and can use the lab when in fact they are not enrolled in a CS class nor have they paid their dues.	High	Medium
7	Maintainability	Software Patches	**Maintainability scenario 1** Releases for Firefox, Internet Explorer, Chrome, or any other browser installed on the workstations needs to happen in a timely fashion, typically 72 hours from release date.	High	Medium
8			**Maintainability scenario 3** Adobe products, namely Reader, need to be updated within 72 hours from release date.	High	Medium
9		Operating System updates	**Maintainability scenario 5** The Windows OS should be set to automatically download the latest patches available from Microsoft. This generally	High	Medium

			occurs on the second Tuesday of the month. After patching occurs workstations should be checked for availability and performance. When pushing new images to the workstations via Symantec Ghost the new images should have the latest patches already installed to avoid the workstations having to re-download them again and thus slowing down the network.		
10		Operating System upgrades	**Maintainability scenario 6** In general, the workstations will not be upgraded in this manner until the spring semester is over. This would be done in conjunction with Symantec Ghost to avoid 20 separate installs.	Medium	High
11		Documentation	**Maintainability scenario 7** Documentation must become part of the sysadmin's work flow in order to assist with the living documentation process.	High	Medium
12	Performance	Login response time	**Performance scenario 1** After logging in the user is shown the desktop and is ready to do work in less than 5 seconds.	Medium	Medium
13	Usability	Accessibility	**Usability scenario 3**	Medium	Medium

			The user saves their homework to their "Z" drive on the network so they can access it from any workstation within the lab environment.		
14	Maintainability	Software patches	**Maintainability scenario 4** VMware products need to be updated within 72 hours from release date.	Medium	Medium

Table 37: ATAM - Prioritized Chapman Lab Utility Tree

Appendix J.6: CBAM Response Goals

Scenario	Worst	Current	Desired	Best
1	Workstation is compromised. Student's private data is compromised. Student finds out it was compromised through this system and sues the university.	Workstations are not compromised.	Workstations are never compromised.	Workstations are never compromised.
2	SVAD or Samba service does not respond. Network is down.	Chapman Lab is sometimes unavailable.	Chapman Lab is available when SVAD and Samba service is running.	Chapman Lab domain never goes down.
3	"Z" drive crashes and the student loses all their work from the beginning of time.	The "Z" drive is accessible most of the time.	The "Z" drive is accessible to the student whenever they are logged in.	The "Z" drive is accessible to the student whenever they are logged in.
4	Workstation is compromised. Student's private data is compromised. Student finds out it was compromised through this system and sues the university.	Workstations are not compromised.	Workstations are never compromised.	Workstations are never compromised.
5	Another person learns of the user's password and uses the system for malicious purposes.	No way of knowing if passwords have been compromised.	Users always change their default password or they are given a more challenging password	Users always keep their passwords secret.

			to begin with.	
6	A nonpaying non CS student uses the lab for malicious purposes.	No way of knowing if users have used social engineering to gain access to the lab.	Reinstitute payment list from registrar's office into user creation.	A script that checks against the registrar's database automatically instead of a static list.
7	Browsers are never updated.	Browsers are updated at the beginning of every spring and fall semester.	Browsers are updated within 72 hours of release.	Browsers would automatically update so the sysadmin would not have to do it manually.
8	Adobe products are never updated.	Adobe products are updated at the beginning of every spring and fall semester.	Adobe products are updated within 72 hours of release.	Adobe products would automatically update so the sysadmin would not have to do it manually.
9	Microsoft Windows is never updated.	Microsoft Windows is updated at the beginning of every spring and fall semester.	Microsoft Windows is updated within 72 hours of release.	Microsoft Windows would automatically update so the sysadmin would not have to do it manually.
10	Microsoft Windows is never upgraded.	Microsoft Windows is upgraded when there is a major release.	Microsoft Windows is upgraded when there is a major release.	Microsoft Windows is upgraded when there is a major release.
11	Sysadmin forgets to update the living SOP document. Sysadmin puts false information in the living SOP document.	Updating the living SOP document is not currently part of the sysadmin's routine.	Updating the living SOP document becomes part of the Sysadmin's routing.	Updating the living SOP document in an accurate and complete fashion becomes part of the Sysadmin's routing.

12	The desktop is never shown once the user logs in.	The desktop is shown to the user within 30 seconds.	The desktop is shown to the user within 15 seconds.	The desktop is shown to the user within 5 seconds.
13	"Z" drive crashes and the student loses all their work from the beginning of time.	The "Z" drive is accessible most of the time.	The "Z" drive is accessible to the student whenever they are logged in.	The "Z" drive is accessible to the student whenever they are logged in.
14	VMware products are never updated.	VMware products are updated at the beginning of every spring and fall semester.	VMware products are updated within 72 hours of release.	VMware products would automatically update so the sysadmin would not have to do it manually.

Table 38: CBAM - Response Goals

Appendix J.7: CBAM Prioritized Scenarios

		Response Goals			
Scenario	Votes	Worst	Current	Desired	Best
1	10	Workstation is compromised. Student's private data is compromised. Student finds out it was compromised through this system and sues the university.	Workstations are not compromised.	Workstations are never compromised.	Workstations are never compromised.
2	15	SVAD or Samba service does not respond. Network is down.	Chapman Lab is sometimes unavailable.	Chapman Lab is available when SVAD and Samba service is running.	Chapman Lab domain never goes down.
3	15	"Z" drive crashes and the student loses all their work from the beginning of time.	The "Z" drive is accessible most of the time.	The "Z" drive is accessible to the student whenever they are logged in.	The "Z" drive is accessible to the student whenever they are logged in.
4	10	Workstation is compromised. Student's private data is compromised. Student finds out it was compromised through this system and sues the	Workstations are not compromised.	Workstations are never compromised.	Workstations are never compromised.

		university.			
5	5	Another person learns of the user's password and uses the system for malicious purposes.	No way of knowing if passwords have been compromised.	Users always change their default password or they are given a more challenging password to begin with.	Users always keep their passwords secret.
6	10	A nonpaying non CS student uses the lab for malicious purposes.	No way of knowing if users have used social engineering to gain access to the lab.	Reinstitute payment list from registrar's office into user creation.	A script that checks against the registrar's database automatically instead of a static list.
7	5	Browsers are never updated.	Browsers are updated at the beginning of every spring and fall semester.	Browsers are updated within 72 hours of release.	Browsers would automatically update so the sysadmin would not have to do it manually.
8	5	Adobe products are never updated.	Adobe products are updated at the beginning of every spring and fall semester.	Adobe products are updated within 72 hours of release.	Adobe products would automatically update so the sysadmin would not have to do it manually.
9	5	Microsoft Windows is never updated.	Microsoft Windows is updated at the beginning of every spring and fall semester.	Microsoft Windows is updated within 72 hours of release.	Microsoft Windows would automatically update so the sysadmin would

					not have to do it manually.
10	5	Microsoft Windows is never upgraded.	Microsoft Windows is upgraded when there is a major release.	Microsoft Windows is upgraded when there is a major release.	Microsoft Windows is upgraded when there is a major release.
11	5	Sysadmin forgets to update the living SOP document. Sysadmin puts false information in the living SOP document.	Updating the living SOP document is not currently part of the sysadmin's routine.	Updating the living SOP document becomes part of the Sysadmin's routing.	Updating the living SOP document in an accurate and complete fashion becomes part of the Sysadmin's routing.
12	0	The desktop is never shown once the user logs in.	The desktop is shown to the user within 30 seconds.	The desktop is shown to the user within 15 seconds.	The desktop is shown to the user within 5 seconds.
13	15	“Z” drive crashes and the student loses all their work from the beginning of time.	The “Z” drive is accessible most of the time.	The “Z” drive is accessible to the student whenever they are logged in.	The “Z” drive is accessible to the student whenever they are logged in.
14	0	VMware products are never updated.	VMware products are updated at the beginning of every spring and fall semester.	VMware products are updated within 72 hours of release.	VMware products would automatically update so the sysadmin would not have to do it manually.

Table 39: CBAM - Prioritized Scenarios