CIS 4911 - Senior Project (U01)

Web Dashboard for Addigy

Javier Carmona

Francisco Marcano

September 8, 2014

Jason Dettbarn

**Feasibility Study and Project Plan**

# Copyright and trademark notices

WIP

# 

# 

# 

# 

# 

# 

# Abstract

WIP

# Table of Contents

|  |  |
| --- | --- |
| **Chapter** | **Page** |
| **1. Introduction** | **4** |
| 1.1. Problem Definition | 4 |
| 1.2. Background | 5 |
| 1.3. Definitions | 5 |
| 1.4. Overview of document | 5 |
| **2. Feasibility Study** | **6** |
| 2.1. Description of the Current System | 6 |
| 2.2. Purpose of New System | 6 |
| 2.3. High-Level Definition of User Requirements | 6 |
| 2.4. Alternative Solutions | 7 |
| 2.4.1. Description of Alternatives | 7 |
| 2.4.2. Selection Criteria | 7 |
| 2.4.3. Analysis of Alternatives | 8 |
| 2.5. Recommendations | 9 |
| **3. Project Plan** | **10** |
| 3.1. Project Organization | 10 |
| 3.1.1. Project Personnel Organization | 10 |
| 3.1.2. Hardware and Software Resources | 11 |
| 3.2. Identification of tasks, Milestones, and Deliverables | 11 |
| **4. Appendix** | **12** |
| 4.1. Appendix A - Project Schedule | 12 |
| 4.2. Appendix B - Feasibility Matrix | 13 |
| 4.3. Appendix C - Cost Matrix | 13 |
| 4.4. Appendix D - Diary of Meetings | 14 |
| **5. References** | **15** |

# 1. Introduction

This Feasibility Study and Project Plan aims to evaluate the technical and financial feasibility of implementing the Addigy Web Dashboard, which promises to bring high level network tools for professionals involved with Macintosh products.

This study is based on the key points defined in section 1.1, which aims to bring these tools to professionals in an environment filled with similar tools mostly targeted towards competing operating systems.

This Feasibility Study comprises an analysis of the various factors that affect the technical and financial feasibility of the project. The study is intended to be an information resource to assist decision makers in determining the viability of a

Web Dashboard targeted towards Mac OSx clusters.

This Feasibility Study is comprised of two parts. The first is the Feasibility Study. The second is the project plan, the project organization, and the project breakdown.

## 1.1. Problem Definition

Addigy is focused on building tools for IT departments to provision and manage the growing sector of Mac computers within corporate environments. IT departments have a huge portfolio of products to choose from for managing PCs, but little to no tools exist for the Mac assets that they are forced to manage. This causes a great deal of tension in ensuring security of the Mac assets on corporate networks and with corporate data.

Forrester Research completed a survey of Enterprises to find that 41% of enterprises won’t let Apple machines anywhere near their computing services. Enterprises have debated moving off windows for years, but the cost and lack of tools for managing even free/open-source system and the technical talent necessary, was too high.”

In Feb 2013, Gartner predicted that in 2014 “Apple will be as accepted by the enterprise IT as Microsoft is today”. This prediction has largely not come to fruition, and Addigy expects much of this is due to the continued lack of commercial tools.

Gartner recently estimated that “...at most, there might be a 5% installed base of Macs, which IT managers traditionally regarded as an additional management burden over the Windows PCs them must carefully tend for updates and patch management”. A Gartner survey found that 60% of enterprise still ‘limit’ Macs, but more and more are ‘embracing’ them. And 64% said they will likely allow more Macs into the enterprise over the next few years.

One of the major solutions to this problem is by providing an advanced real-time dashboard, that provides the Enterprise an ability to fully understand the Mac assets they are managing and more specifically if they are meeting the standards for their security, HR, compliance, and inevitably great tools for their employees.

## 1.2. Background

The individual auditing mechanisms required on Macs are quite accessible in the open-source community. Addigy leverages Facter (by PuppetLabs) to audit and further extend the auditing being performed on the machine. When Facter runs it creates very simple key pairs, that provide CPU, memory, user, uptime, etc. Facter can execute ruby code that can then store these ‘Facts’ to be provided as additional audit data. For example, Facter already provides audit results for the Mac’s Serial Number. Ruby code can do a web lookup to Apple to get back the Warranty Expiration Date of the device. Aggregating all of these values together, a Dashboard can easily display how many Mac assets are still under warranty, approaching out-of-warranty, or warranties already expired.

Addigy is able to orchestrate the operations of Facter on all the Mac assets and centrally store them in a database. A REST API will be provided JSON data for the dashboard client.

## 1.3. Definitions

Addigy - A Miami based software startup that builds IT Management Software for Mac Computers

## 1.4. Overview of document

This document will describe the Addigy Web Dashboard and its different components. It will discuss the purpose of the Dashboard. Then it will list what is required for the system to be complete. The document will explore other systems that have the same purpose as the Addigy Dashboard.

Afterwards, this document will show how the personnel will be organized and the different resources available. It will identify tasks, milestones, and deliverables, and break them down.

# 2. Feasibility Study

In this chapter, this document will study how feasible is this new system by describing what already exists, what it is meant to become, its purpose, and what is required from it. Then it will explain similar solutions that already exist in the market that seek to solve the same problem. The different existing system will be compared amongst themselves and to the new system being proposed in different terms of feasibility. Finally, recommendations will be done for the project.

## 2.1. Description of the Current System

The current system does NOT have an Audit Dashboard incorporated. Addigy currently provides other IT Management tools, but has not developed an Audit Dashboard for customers to understand what assets and configuration they have. The ability for customers to see the state of their assets is arguably the most important feature, and necessary for being able to decide what changes/policies need to be applied on the assets.

## 2.2. Purpose of New System

This new system seeks to meaningfully display information collected by the computers throughout an organization. It seeks to provide a convenient way of accessing, visualizing, understanding, and interacting with the data being collected.

## 2.3. High-Level Definition of User Requirements

Requirements involve building a high-end web client based dashboard. The web client will have to poll the Addigy web service for updated data, and render the changes on the dashboard. Some of the top dashboard items that are important to our customers:

* Where are the computers located geographically. What status are they in, in looking at them on the map (green/healthy or red/issue)
* How many machines have Filevault disk encryption on vs unencrypted
* How many machines require authentication on the domain vs locally
* What is the make-up of the Mac assets - macbook air / macbook pro / mac mini, etc.
* Uptime of the machines, that have been online in the last hour / day / week
* Apple warranty data still under warranty, warranty expiring within 30 days, out of warranty
* OSX updates are pending

## 2.4. Alternative Solutions

There exist different solutions out there that attempt to solve the same issue that the Addigy Web Dashboard attempts to solve. In the following section, this document will explore some of those options and compare different aspects of each of them to see which are the most feasible and how they stand against each other.

## 2.4.1. Description of Alternatives

Alternative to the Addigy Web Dashboard, there are similar dashboards in the market that seek to solve the same problem of managing different computers in the network and displaying meaningful information about them.

Such a dashboard is provided by the software called IpMonitor. This software allows to track machines in the network of an organization and makes visual representation of the current state of these machines ranging from network usage to cpu usage. IpMonitor has two issues against it, which are the fact that is not free to use as are other alternatives, and that it is only able to monitor Windows systems.

Another dashboard is provided by SpiceWorks. Like IpMonitor, SpiceWorks allows you to keep track of the different computers on the network and also shows different statistics about these computers. Unlike IpMonitor, SpiceWorks is free to use since they are paid by advertisers to bring IT professionals IT related products through their software. SpiceWorks only falls short on it only being able to collect information for Windows Systems.

## 2.4.2. Selection Criteria

As selection criteria we focused on four different aspects of the software: Operational Feasibility, Technical Feasibility, Economic Feasibility, and Schedule Feasibility.

The point of interest for the Operational Feasibility is the ability to support a number of operating systems with focus on Macs. It is preferred to be able to collect information on Windows, Macs, and Linux systems.

In Technical Feasibility, the technologies available are analyzed in order to conclude whether the software proposed are in fact possible to develop or use. The more technology is out there in order to use the software the better.

Being able to afford any given technology is a great concern and must be carefully analized. The most attractive technologies are those that provide the most functionality for the least amount of money. Free technologies would be ideal for this category.

Lastly, it is important to know whether any of the solutions can be completed on a reasonable amount of time. The more complete the solution in question is, the less time it would take to put into use and therefore the most appealing it is.

## 2.4.3. Analysis of Alternatives

When it came to Operational Feasibility we found out that there is great demand software to track and manage different aspects of the systems under the network of an organization. The different software that address this problem in the market today seem to be focused on collecting information and managing windows system. That is the reason why the Addigy Web Dashboard is concerned with working on all major Operating Systems such as Mac, Linux, and also Windows. On this category, points are awarded for the number of system the software supports with interest in Windows,Mac, and Linux, making it a total of 30 points. IpMonitor with a score of 10, SpiceWorks with a score of 10, and the Addigy Web Dashboard with a score of 30.

For the technical aspects we know that the technology exist since there is a number of different existing software that are able to collect information within each individual system. The Dashboards then becomes a matter of collecting this information in the different systems of the network through existing infrastructure such as Web Services and HTTP/HTTPS protocols, then organizing it on a meaningful way. All of this technology is fully available through the internet. This category also has a maximum score of 30. Given that IpMonitor and SpiceWorks are technologies that are already developed, they receive the maximum score. Since the Addigy Web Dashboard is yet to be completed, it receives 25 points because the underlying technology that makes it possible is already in place.

Looking at the economic feasibility of the different products, we have free products such as SpiceWorks, which has revenue through advertising, and paid products such as IpMonitor, which can have prices of $1995.00, for very similar services. Given that both products provide the services needed, the free software is more appealing. This category has a weight of 30 points in which IpMonitor receives a score of 10 because it requires payments but it could be afforded, SpiceWorks receives a score of 30 since it is free, which is ideal, and the Addigy Web Dashboard receives a score of 30 since it is also free.

Lastly it is worth noting that both of the alternative options are fully developed already and would be able to be deployed without any delay other than the deployment process. Both of the options provide Wizard Oriented installations which should make them easy to deploy. Since the timing is not a critical factor, it is awarded a maximum score of 10. IpMonitor and SpiceWorks are both awarded the maximum amount of points because they are both fully developed. The Addigy Web Dashboard is awarded 5 points as it still needs to be developed but it should not take more than a few weeks.

## 2.5. Recommendations

Based on the analysis made on the different alternatives available today, the Addigy Web Dashboard would be the best possible solution for the current project. There are a couple of reasons why this is the case.

The main factor is that the project maintains its own name; no third-party application are used to realize the project’s goal.

Second factor is that the project covers a wider audience and so it is a better alternative than the other products described.

Third factor is that a lot of the groundwork is already done on the project and the data is already available, allowing for a Web Interface to be quickly developed without the need to wait on that much, if any, backend implementation.

Fourth and last factor, is that this allows for the project to be developed in any style and with any technology that allows for rapid and agile development. This greatly increases productivity and maintainability for the project. By choosing an alternative product, the project loses its flexibility and direction.

# 3. Project Plan

This project is to be organized and developed using a variety of different tools gears towards an agile development approach. By using an agile development approach, it ensures that the project is constantly being updated and available to deploy at any time during its life cycle.

## 3.1. Project Organization

The project is to be split up and organized based, again, on an agile development approach. The following series of steps will be taken in order to make sure that the development lifecycle is productive with very little downtime.

1. **Scope**
   1. Meat with client and discuss all of the functional requirements, non-functional requirements, and end goals for the project
   2. Define any additional resources needed for the completion of the project
   3. Obtain all of the resources necessary to realize the project
2. **Analysis**
   1. Define any software and hardware specifications needed in order to develop the product taking into account the functional requirements that need to be realized.
   2. Meet with development team to discuss the possibility of the requirements
   3. Discuss the financial feasibility for the project with the client
   4. Meet with the development team to organize and distribute the requirements needed based on individual strengths and weaknesses.
   5. Develop the project’s delivery timeline
   6. Obtain approval to proceed with the project
   7. Provide the tools necessary to tackle the project using agile development principles.
3. **Design**
   1. Review any software needed to complete the application.
   2. Develop functional specifications and prototypes based on functional specifications
   3. Review prototypes with client and record feedback
   4. Incorporate any feedback while trying to avoid any scope creep
   5. Obtain approval for completion

## 3.1.1. Project Personnel Organization

The project will have a total of 3 members working on various aspects of the application. Each member will be responsible for following the project timeline while promptly delivering any tasks assigned to them. The team members are:

1. **Jason Drettbarn**
   1. **Title**
      1. Project Manager
   2. **Duties**
      1. Communicate with the client on a regular basis
      2. Provide the team with the software and hardware needed to complete the tasks at hand
      3. Provide the team with the functional requirements needed.
2. **Javier Carmona**
   1. **Title**
      1. Developer
   2. **Duties**
      1. Communicate with the project manager and other team members with important project information.
      2. Promptly delivering the tasks assigned while following the project timetable.
      3. Provide immediate feedback of any issues that are slowing down the project development cycle.
      4. Using the provided software in order to keep the project organized and visible to other team members.

## 3.1.2. Hardware and Software Resources

Addigy will provide AWS EC2 instances for running a hosted webserver capable of driving the AngularJS webclient. Developer can run local instances during the development process and/or can leverage AWS resources at the same time.

## 3.2. Identification of tasks, Milestones, and Deliverables

1. Dev and Product servers are set up
2. First Iteration: Scenario-Based Implementation - Unit Testing
3. Second Iteration: Scenario-Based Implementation - Unit Testing
4. Third Iteration: Scenario-Based Implementation - Unit Testing
5. Fourth Iteration: Scenario-Based Implementation - System Testing
6. Fifth Iteration: Scenario-Based Implementation - System Testing
7. Fifth Iteration: Scenario-Based Implementation - System Testing
8. Sixth Iteration: Scenario-Based Implementation - System Testing
9. Final deliverable
10. Final posters
11. Showcase

# 4. Appendix

## 4.1. Appendix A - Project Schedule

|  |  |
| --- | --- |
| Task Name | Deadline |
| Dev and Product servers are set up | 9/8/2014 |
| First Iteration: Scenario-Based Implementation - Unit Testing | 9/8/2014 |
| Second Iteration: Scenario-Based Implementation - Unit Testing | 9/22/2014 |
| Third Iteration: Scenario-Based Implementation - Unit Testing | 10/6/2014 |
| Fourth Iteration: Scenario-Based Implementation - System Testing | 10/20/2014 |
| Fifth Iteration: Scenario-Based Implementation - System Testing | 11/3/2014 |
| Sixth Iteration: Scenario-Based Implementation - System Testing | 11/17/2014 |
| Final deliverable | 12/11/2014 |
| Final posters | 12/5/2014 |
| Showcase | 12/12/2014 |

## 

## 

## 

## 

## 4.2. Appendix B - Feasibility Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feasibility Criteria** | **Wt.** | **IpMonitor** | **SpiceWorks** | **Addigy Dashboard** |
| **Operational Feasibility** | 30 | Available only on Windows systems.  score: 10 | Available only on Windows systems.  score: 10 | Available for Mac, Linux, and Windows.  score: 30 |
| **Technical Feasibility** | 30 | score: 30 | score: 30 | score: 25 |
| **Economic Feasibility** | 30 | $1995.00  score: 10 | FREE  score: 30 | FREE  score: 30 |
| **Schedule Feasibility** | 10 | Already built.  score: 10 | Already built.  score: 10 | Needs a few weeks of work.  score: 5 |
| **Ranking** | **100** | **60** | **80** | **90** |

## 

## 4.3. Appendix C - Cost Matrix

|  |  |
| --- | --- |
| **System** | **Cost** |
| IpMonitor | $1995.00 |
| SpiceWorks | FREE |
| Addigy Dashboard | FREE |

## 4.4. Appendix D - Diary of Meetings

|  |  |
| --- | --- |
| **Meeting** | **Date** |
| Meeting 1 | 9/4/2014 |
| Meeting 2 | 9/9/2014 |
| Meeting 3 | 9/16/2014 |
| Meeting 4 | 9/23/2014 |
| Meeting 5 | 9/30/2014 |
| Meeting 6 | 10/7/2014 |
| Meeting 7 | 10/14/2014 |
| Meeting 8 | 10/21/2014 |
| Meeting 9 | 10/28/2014 |
| Meeting 10 | 11/4/2014 |
| Meeting 11 | 11/11/2014 |
| Meeting 12 | 11/18/2014 |
| Meeting 13 | 11/25/2014 |
| Meeting 14 | 12/2/2014 |
| Meeting 15 | 12/9/2014 |

# 5. References

What You Need to Know. (n.d.). Retrieved September 7, 2014, from <http://www.solarwinds.com/documentation/ipmonitor/ipmondoc.aspx>

Manage everything IT… for free! (n.d.). Retrieved September 7, 2014, from <http://www.spiceworks.com/app/>