**Software Project Management Plan**

For the

**Fire and Inﬁltration Quashing System**

(FAIQ)

December 17, 2012

(14:23:54)

Project 4

Version 2.0

CSCI4830 - Introduction to Software Engineering

Fall 2012

Prepared for

***Dr. Mansour Zand***

Written by

***Bijoy Arif***

***Shaozhuo Jia***

***Ben Schaffner***

***Shawn OBrien***

Revised by

***Shawn OBrien***

**Revision/Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Name | Activity |
| 1 | 10/28/12 | Bijoy Arif | Sections 5.2 and 7 |
| 1 | 10/28/12 | Shaozhuo Jia | Sections 4 and 6 |
| 1 | 10/28/12 | Ben Schaffner | Sections 5.3-5.5 |
| 1 | 10/28/12 | Shawn OBrien | Preface. Sections 1-3 and 5.1 and 8. |
| 1 | 10/29/12 | Shawn OBrien | Merging all documents. Editing. Proofing. |
| 1 | 10/30/12 | Shawn OBrien | Finishing Touches. |
| 2 | 12/17/12 | Shawn OBrien | Final editing and additions based on software. |
| 2 | 12/17/12 | Shawn OBrien | Major rewrite of meaningless subsections. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Preface

The scope of this paper indeed encompasses the entire lifetime of our nascent piece of software. Our audience is our professor. This is a learning tool in a hypothetical context.

I include the Prologue of the SRS here, which further expounds the context:

“This project, well suited for this class, is exceedingly hypothetical. Threats must be simulated, as must be their interactions, as must be their eventual quashing (we hope.) Our professor’s monitor must simulate a two-terminal control panel. Ultimately, our professor must simulate using our system to quash a slew of, often concurrent, threats, which in themselves, may be concurrent with the professor’s quashing attempts.

Project 4 has taken the liberty of bringing to life, to paraphrase, “an automated system which monitors and controls.” We call it Smart Core ™. Doing so makes all of our work consistent, because we have chosen to take the purely hypothetical route suggested in the Project Description.

In our scenario, humans are marginalized, and Smart Core ™ automatically counters threats. Nonetheless, we shall construct an excellent interface for the user, and compile a user guide for a human user. Our system has a training system for users in the unlikely event that Smart Core ™ malfunctions. Or maybe Smart Core ™ does indeed fail, and you, professor, must save the building from multiple threats.

We may document calls that fit our design, but in reality are just empty calls, but called at the appropriate time. Similarly, we have found it too cumbersome to document our threat simulations subsystem and plan to omit it from our documents, though the code will be well documented.”

Table of Contents

[**1** **Overview** 8](#_Toc343517111)

[1.1 Project Summary 8](#_Toc343517112)

[1.1.1 Purpose, Scope and Objectives 8](#_Toc343517113)

[1.1.1.1 Purpose 8](#_Toc343517114)

[1.1.1.2 Scope 8](#_Toc343517115)

[1.1.1.3 Objective 8](#_Toc343517116)

[1.1.1.4 Exclusions 9](#_Toc343517117)

[1.1.2 Assumptions and Constraints 9](#_Toc343517118)

[1.1.3 Deliverables 9](#_Toc343517119)

[1.1.4 Schedule and Budget Summary 10](#_Toc343517120)

[1.1.5 Evolution of the SNMP 10](#_Toc343517121)

[**2** **References** 10](#_Toc343517122)

[3 Special Definitions 10](#_Toc343517123)

[4 Project Organization 10](#_Toc343517124)

[4.1 External interfaces 10](#_Toc343517125)

[4.2 Internal structure 11](#_Toc343517126)

[4.3 Roles and responsibilities 11](#_Toc343517127)

[Work Activity 11](#_Toc343517128)

[5 Managerial Process Plans 11](#_Toc343517129)

[5.1 Project Start-Up Plan 11](#_Toc343517130)

[5.1.1 Estimation Plan 11](#_Toc343517131)

[5.1.1.1 Cost 11](#_Toc343517132)

[5.1.1.2 Schedule 11](#_Toc343517133)

[5.1.1.3 Resources 11](#_Toc343517134)

[5.1.1.4 Basis 12](#_Toc343517135)

[5.1.1.4.1 M, Multiplier 12](#_Toc343517136)

[5.1.1.4.2 **PMAuto**, Auto-Generated Code Integration Effort 13](#_Toc343517137)

[5.1.1.4.3 Initial Calibration 13](#_Toc343517138)

[5.1.1.4.4 Lines of Code per Function Point for C++ 13](#_Toc343517139)

[5.1.1.4.5 GUI Screens 13](#_Toc343517140)

[5.1.1.4.6 External Interfaces 14](#_Toc343517141)

[5.1.1.4.7 Function Points 14](#_Toc343517142)

[5.1.1.4.7.1 Complexity Multiplier, C = (0.65 + 0.01 x N) 14](#_Toc343517143)

[5.1.1.4.8 B, Disproportionate Large Project Effort (1.10 to 1.24) 14](#_Toc343517144)

[5.1.1.4.9 Lines of Code Estimate 15](#_Toc343517145)

[5.1.1.4.10 Person Months 15](#_Toc343517146)

[5.1.2 Staffing Plan 15](#_Toc343517147)

[5.1.3 Resource Acquisition Plan 15](#_Toc343517148)

[5.1.4 Project Staff Training Plan 15](#_Toc343517149)

[5.2 Work Plan 16](#_Toc343517150)

[5.2.1 Work Activities 16](#_Toc343517151)

[5.2.1.1 System Requirements Specification Document Breakdown 16](#_Toc343517152)

[5.2.1.2 Software Requirements Specification Document Breakdown 17](#_Toc343517153)

[5.2.1.3 Software Project Management Plan Document Breakdown 17](#_Toc343517154)

[5.2.1.4 Software Project Design Document Breakdown 18](#_Toc343517155)

[5.2.1.5 Alternate Analysis 18](#_Toc343517156)

[5.2.1.6 Final Documentation 18](#_Toc343517157)

[5.2.1.7 GUI Prototype 19](#_Toc343517158)

[5.2.1.8 Coding / Debugging 19](#_Toc343517159)

[5.2.1.9 Program Testing / Validation 19](#_Toc343517160)

[5.2.2 Schedule Allocation 20](#_Toc343517161)

[5.2.2.1 System Requirements Specification Document 20](#_Toc343517162)

[5.2.2.2 Software Requirements Specification Document 21](#_Toc343517163)

[5.2.2.3 Software Project Management Plan 21](#_Toc343517164)

[5.2.2.4 Software Project Design Document 21](#_Toc343517165)

[5.2.2.5 Alternative Analysis 22](#_Toc343517166)

[5.2.2.6 Final Documentation 22](#_Toc343517167)

[5.2.2.7 GUI Prototype 22](#_Toc343517168)

[5.2.2.8 Coding / Debugging 23](#_Toc343517169)

[5.2.2.9 Program Testing / Validation 23](#_Toc343517170)

[5.2.3 Resource Allocation 23](#_Toc343517171)

[5.2.3.1 System Requirements Specification Document 24](#_Toc343517172)

[5.2.3.2 Software Requirements Specification Document 24](#_Toc343517173)

[5.2.3.3 Software Project Management Plan 24](#_Toc343517174)

[5.2.3.4 Software Project Design Document 24](#_Toc343517175)

[5.2.3.5 Alternative Analysis 24](#_Toc343517176)

[5.2.3.6 Final Documentation 24](#_Toc343517177)

[5.2.3.7 GUI Prototype 24](#_Toc343517178)

[5.2.3.8 Coding / Debugging 24](#_Toc343517179)

[5.2.3.9 Program Testing / Validation 24](#_Toc343517180)

[5.2.4 Budget Allocation 24](#_Toc343517181)

[5.3 Control Plan 25](#_Toc343517182)

[**5.3.1** **Requirements control plan** 25](#_Toc343517183)

[**5.3.2** **Schedule control plan** 25](#_Toc343517184)

[**5.3.3** **Budget control plan** 25](#_Toc343517185)

[**5.3.4** **Quality control plan** 25](#_Toc343517186)

[*5.3.4.1* *Quality control of documentation* 25](#_Toc343517187)

[*5.3.4.2* *Quality control of code* 25](#_Toc343517188)

[**5.3.5** **Reporting plan** 25](#_Toc343517189)

[*5.3.5.1* *Reporting for subtasks* 25](#_Toc343517190)

[*5.3.5.2* *Reporting for milestone tasks* 25](#_Toc343517191)

[**5.3.6** **Metrics collection plan** 25](#_Toc343517192)

[*5.3.6.1* *Metrics collection for documentation* 25](#_Toc343517193)

[*5.3.6.2* *Metrics collection for code* 26](#_Toc343517194)

[5.4 Risk management plan 26](#_Toc343517195)

[**5.4.1** **Determining risk factors** 26](#_Toc343517196)

[*5.4.1.1* *Assessing risk factors* 26](#_Toc343517197)

[*5.4.1.2* *Nature of the risk* 26](#_Toc343517198)

[*5.4.1.3* *Probability* 26](#_Toc343517199)

[*5.4.1.4* *Effect* 26](#_Toc343517200)

[*5.4.1.5* *Measures* 26](#_Toc343517201)

[**5.4.2** **Prioritizing risk factors** 26](#_Toc343517202)

[**5.4.3** **Monitoring risk factors** 26](#_Toc343517203)

[**5.4.4** **Risks** 26](#_Toc343517204)

[5.5 Closeout plan 26](#_Toc343517205)

[5.5.1 Staff reassignment 27](#_Toc343517206)

[**5.5.2** **Personnel debriefings** 27](#_Toc343517207)

[**5.5.3** **Final report / lessons learned** 27](#_Toc343517208)

[**5.5.4** **Archiving project materials** 27](#_Toc343517209)

[**5.5.5** **Materials to be distributed** 27](#_Toc343517210)

[3) The final report / lessons learned document. 27](#_Toc343517211)

[6 Technical process plans 27](#_Toc343517212)

[6.1 Process model 27](#_Toc343517213)

[6.1.1 Work Product Flow 27](#_Toc343517214)

[6.1.2 Work Products Delivery Timing 28](#_Toc343517215)

[6.2 Methods Tools and Techniques 28](#_Toc343517216)

[6.2.1 Methodology 29](#_Toc343517217)

[6.2.2 Tools Used 29](#_Toc343517218)

[6.3 Infrastructure Plan 29](#_Toc343517219)

[7 Supporting Process Plan 29](#_Toc343517220)

[7.1 Configuration Management Plan 29](#_Toc343517221)

[7.2 Verification and Validation Plan 29](#_Toc343517222)

[7.3 Documentation Plan 29](#_Toc343517223)

[7.4 Quality Assurance Plan 29](#_Toc343517224)

[7.4.1 Communication During Development 29](#_Toc343517225)

[7.4.2 Installation 30](#_Toc343517226)

[7.4.3 Support 30](#_Toc343517227)

[7.5 Reviews and Audits 30](#_Toc343517228)

[7.6 Problem Resolution Plan 30](#_Toc343517229)

[7.7 Subcontractor Management Plan 30](#_Toc343517230)

[7.8 Process Improvement Plan 30](#_Toc343517231)

[8 Additional Plans 30](#_Toc343517232)

[Table 1 11](#_Toc343517233)

[Table 2 15](#_Toc343517234)

[Table 3 16](#_Toc343517235)

[Table 4 16](#_Toc343517236)

[Table 5 17](#_Toc343517237)

[Table 6 17](#_Toc343517238)

[Table 7 18](#_Toc343517239)

[Table 8 18](#_Toc343517240)

[Table 9 18](#_Toc343517241)

[Table 10 19](#_Toc343517242)

[Table 11 19](#_Toc343517243)

[Table 12 19](#_Toc343517244)

[Table 13 20](#_Toc343517245)

[Table 14 20](#_Toc343517246)

[Table 15 21](#_Toc343517247)

[Table 16 21](#_Toc343517248)

[Table 17 21](#_Toc343517249)

[Table 18 22](#_Toc343517250)

[Table 19 22](#_Toc343517251)

[Table 20 22](#_Toc343517252)

[Table 21 23](#_Toc343517253)

[Table 22 23](#_Toc343517254)

[Table 23 28](#_Toc343517255)

[Table 24 29](#_Toc343517256)

[Figure 1 10](#_Toc343517259)

[Figure 3 28](file:///Q:\School\4830%20Intro%20to%20SE\hack\03-ProjectPlan\SPMP.docx#_Toc343517260)

1. **Overview**

## Project Summary

This clause of the SPMP provides an overview of the purpose, scope, and objectives of the project, the project assumptions and constraints, a list of project deliverables, a summary of the project schedule and budget, and the plan for evolution of the SPMP.

### Purpose, Scope and Objectives

#### Purpose

To learn how to properly engineer software. For experienced programmers, this is often tantamount to cleansing themselves of impure habits, such as the “code first” imperative. With build times still high, and still expensive, even the old paradigm of “code, build, and test” must be washed away.

Sadly, many middle managers, to the exclusion of all other metrics, count lines of code. Many of them do not want to see (let alone read) some thick Microsoft Word document about requirements or specifications. Maybe *their* bosses want lines of code. More than likely, they have seen so many projects break during implementation that they think if they start the implementation sooner, they will be able to absorb the inexorable time expenses. This, of course, leads to directly to things breaking sooner.

The only “advantage” this model presents is that a requirement may be reinterpreted at any time, forcing a redesign. This will likely happen right about the time that something major breaks. So everyone bathes in the righteous luxury of pointing the finger at someone else.

Hence, we must learn the proper sequence. One must have faith in proper engineering principles. If we raise the black flag and hold our ground, the Engineering gods shall reward us, for the code shall write itself.

#### Scope

Other than wearing most of the letters off a few keyboards, and providing a bounty of high-grade reading material for a certain professor, the scope of this project finds itself wrapped around one man, over eighty-two thousand slides, one textbook, and a looming final in December.

#### Objective

Ultimately, to mold ourselves into engineers, to reshape our minds and unfurl our wings so we may cruise the high skies of professional software engineering, building the next generation of software. Alternatively, one could simply state our objective as attempting to avoid working at WalMart.

#### Exclusions

Revenue, paychecks and, likely, further evolution of this project. (We may of course hope that one day our boss will demand an SPMP as we stare at her blankly. We assure her she will have it on her desk Monday. We wake in the middle of the night and dig through some archives. We find this document, feeling confident we may change the title page and submit it as such Monday. Our boss is not a reader. She likes to tick checkboxes. She is known as a *ticker*.)

### Assumptions and Constraints

We assume to finish early, precisely on December 3rd, 2012. In this way, we have a full week to prepare for the final while preparing the project legacy and compilation. This is also the primary schedule constraint.

This assumption, however, flies smack into the face of the fact that three of the four FAIQ team members now find themselves simply overwhelmed from taking too many hours, or simply cannot imagine the concept of typing a dozen pages of technical prose. Hence, the assumption to finish early is flawed. The project will be late.

We find ourselves constrained to $50 in printing costs that I have been ironically chosen to pay. I thought that strong-arming my way into the leadership position would absolve me from such cumbersome burdens.

I hope that I will be able to finish the class; otherwise, my absence will impose a resource restraint on the rest of the team.

I may cannibalize some software I wrote last year using the Qt Platform from Nokia. As this code illustrates best practices for sending messages and using threads against the Qt API, I fancy that this code shall help meet the requirements for a messaging subsystem and concurrency, as mentioned in the SRS.

The technology employed shall be C++ in a Windows 7 environment.

As stated in the SRS, there are no interfaces to other products.

### Deliverables

1. System Requirements Specification
2. Software Requirements Specification
3. Software Project Management Plan
4. Software Design Document
5. Software User’s Manual
6. Software Testing and Validation Document
7. Software Application Binary
8. Software Project Legacy
9. Activity Log
10. Technical Lessons Learned
11. Managerial Lessons Learned
12. Compilation of all Documents and Source Code

### Schedule and Budget Summary

Other than that December 3rd target date and the $50 in printing costs from section (1.1.2) we have nothing further to summarize.

### Evolution of the SNMP

Assuming our professor is something of an environmentalist, he will most certainly recycle the paper upon which it is printed.

1. **References**

We used the following sources in compiling this document:

1. *FAIQ System Requirements Specification.*
2. *FAIQ Software Requirements Specification.*
3. *Project Description.* Dr. Mansour Zand. September 2012.
4. *IEEE Standard for Software Project Management Plans*- IEEE Std 1058-1998
5. *Serberus Planning Document*. Serberus.

# Special Definitions

None.

# Project Organization

## External interfaces

The Fire and security alarm system is a software product developed by Group 4 development team. This Fire and security alarm system shall provide the real-time fire and security information to the team. Our team shall also have close relationship with hardware vendors to provide the best customizable hardware that fit our system at a lower price.

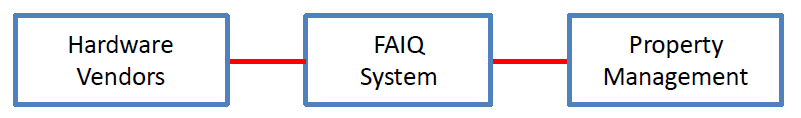


Figure 1

## Internal structure

The Group 4 development team has 4 team members, with one project manager, one lead programmer, one requirements/design/test engineer, and one editor. Surprisingly, they are all the same person! Who is this exceedingly co-dependent and altruistic doormat upon which superior people may wipe the mud off their shoes? Shawn OBrien. Yes indeed.

## Roles and responsibilities

Table 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Work Activity | Team Leader | Design/Test Engineer | Lead Programmer | Editor |
| Project Management | X |  |  |  |
| Document Contribution | X | X | X | X |
| Document Editing | X |  |  | X |
| Document Distribution |  |  |  | X |
| Requirements Validation | X |  |  | X |
| Design Validation | X | X |  |  |
| Configuration Management |  | X | X |  |
| GUI Programming |  | X | X |  |
| Interface Design |  | X |  |  |
| Component Programming | X |  | X |  |
| Testing | X | X | X | X |

# Managerial Process Plans

## Project Start-Up Plan

This subclause of the SPMP shall specify the estimation plan, staffing plan, resource acquisition plan, and training plan.

### Estimation Plan

#### Cost

One thousand dollars. This approximates tuition and fees for the four of us in the UNO College of Computer Science, Fall 2012.

#### Schedule

See section ([1.1.2](#_Assumptions_and_Constraints)) for a summary, section ([5.2.1](#_Work_Activities)) for details. For deliverable timing, see section ([6.1.2](#_Work_Products_Delivery)).

#### Resources

FAIQ requires a plethora of office-oriented resources, on the surface. Under the hood, the primary resource is my time, my effort and my hardware:

* Qt
* 1 savvy computer engineer
* 500 sheets of bright white paper
* 2 ink cartridges
* Meeting Locations (optional)
* Email and phone connectivity across team members (optional)
* PC and Internet access across team members (optional)
* Google Project Hosting (<https://faiq.googlecode.com/svn/trunk/>) (optional)
* 1 2GB USB stick (contingency.)
* Redundant backup facilities on Shawn’s PC

#### Basis

COCOMO2 Early Design Algorithmic Cost Estimation Model:

**PM = A x SizeB x M + PMAuto**

**Key*:***

**PM –** person months

**A** – initial organization-dependent calibration

**Size –** thousands of lines of code (*we estimate functionality in function points and convert*)

**B** – disproportionate large-project effort

**M** – process, product, developer, development multiplier

**PMAuto –** auto-generated code integration effort

For this calculation we use function points, expressible in lines of code (LOC). See section ([5.1.1.4.3](#_Lines_of_Code)) below for details.

The following sections provide the preliminary results to satisfy our main equation, PM, directly above.

##### M, Multiplier

What is this? A product of seven multipliers, each of which receives a number from 1 to 10. We list the multipliers below:

* + RCPX - product reliability and complexity. [1]
  + RUSE - the reuse required. [4]
  + PDIF - platform difficulty. [1]
  + PREX - personnel experience. [1]
  + PERS - personnel capability. [2]
  + SCED - required schedule. [1]
  + FCIL - the team support facilities. [0]

Our score is 10.

##### **PMAuto**, Auto-Generated Code Integration Effort

**PMAuto = (ASLOC x AT/100) / ATPROD**

**Key*:***

**ASLOC –** total lines of reused code, includes auto-generated code

**AT** – percentage of reused code that is automatically generated

**ATPROD –** productivity of integration engineers (industry average is 2400 LOC/month)

PMAuto = 500 x 0.33 / 2400 = 0.06875

##### Initial Calibration

**Let:**

**A = 2.94**

This represents the industry standard.

##### Lines of Code per Function Point for C++

**29**

We will need this later.

##### GUI Screens

**6**

We calculate 30 function points for the main window plus the five embedded tabs, of which only one is visible at a time.

##### External Interfaces

Of the average type, 1 – the user. Of the simple type, 3 – the mouse and the monitor. (I lock the keyboard, hopefully, to avoid errant keystrokes that may crash the program.)

##### Function Points

30 for six screens

7 for user

15 for simple external interfaces, directly above

32 for 8 user inputs embedded in difficult screen at average weight of 4 points each

TOTAL 84 raw function points

GRAND TOTAL FUNCTION POINTS is 84 x 0.70 (see directly below) = 58.80

###### Complexity Multiplier, C = (0.65 + 0.01 x N)

What is N? The degree of influence of external factors such as reuse, concurrency and operating system. The level of concurrency the system requires to offer the user a quality experience we estimate to be 8. Since the code reuse provides significant time savings and “training while developing,” we shall let N finally weigh in at 5. Hence, C equals 0.70.

##### B, Disproportionate Large Project Effort (1.10 to 1.24)

Technically, B may, in principle, range from 1.01 to 1.26. The raw baseline is 1.01. That is where we shall start. Then we shall add from 0.0 to 0.5 for five different categories:

Precedentness 0.3 (new, yet overall small)

Development Flexibility 0.0 (only a deadline)

Risk Analysis 0.3 (GPA at stake)

Team Cohesion 0.5 (no team, no cohesion)

Process Maturity 0.1 (I have used Qt before)

Total 0.12

Hence, we shall assign the value 1.13 to B.

##### Lines of Code Estimate

The estimate of the number of lines of code is based on the number of function points times the lines of code per function point factor:

58.80 x 29 = 1705

Size= Thousands of lines of code=1.705

##### Person Months

PM = A x SizeB x M + PMAuto

PM = 2.94 x 1.7051.13 x 10 + 0.06875 = 53.73

I *might* miss that December 3rd deadline.

### Staffing Plan

As our team is so small, the Staffing Plan equivocates to the Project Staff Training Plan, two hops below, in section ([5.1.4](#_Project_Staff_Training)).

### Resource Acquisition Plan

Freeware.

<http://qt-project.org/downloads>

### Project Staff Training Plan

Projected Skills Training for the four principals.

Table 2



***Key:***

**○ :** Needs Training

**\* :** Expert

**🗸:** Accomplished

## Work Plan

The project will involve following resources, budgets and work activities:

### Work Activities

Table 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Work Activities | Start | Finish | Duration | Aug 2012 | | | Sep 2012 | | | Oct 2012 | | | Nov 2012 | |
| 17 | 24 | 30 | 10 | 20 | 30 | 10 | 20 | 30 | 15 | 30 |
| 1 | Requirements Specification | 8/17 | 8/30 | 14 d | =========== | | | | | | | | | | |
| 2 | Software Requirements Specification | 8/30 | 9/15 | 16 d | ====== | | | | | | | | | | |
| 3 | Project Planning Document | 9/15 | 9/30 | 16 d | ======= | | | | | | | | | | |
| 4 | Project Design | 9/30 | 10/15 | 16 d | ====== | | | | | | | | | | |
| 5 | Alternative Analysis | 10/15 | 10/20 | 6 d | == | | | | | | | | | | |
| 6 | Final Documentation | 10/20 | 10/25 | 6 d | == | | | | | | | | | | |
| 7 | GUI Prototype | 8/17 | 9/30 | 44 d | ======================== | | | | | | | | | | |
| 8 | Coding/Debugging | 10/25 | 11/15 | 20 d | ======= | | | | | | | | | | |
| 9 | Program Testing/ Validation | 11/15 | 11/30 | 16 d | == | | | | | | | | | | |

#### System Requirements Specification Document Breakdown

Table 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Work Activities of  Requirements specification | Start | Finish | Duration | Aug  2012 | | | | | | | | | | | |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 28 | 29 | 30 |
| 1 | Requirements Discovery | 8/17 | 8/20 | 3 d | =============== | | | | | | | | | | | |
| 2 | Requirements Validation | 8/20 | 8/24 | 4 d | ================== | | | | | | | | | | | |
| 3 | Drafting of Requirements Document | 8/24 | 8/27 | 3 d | ======= | | | | | | | | | | | |
| 4 | Finalizing Requirements Document | 8/27 | 8/30 | 3 d | ======== | | | | | | | | | | | |

#### Software Requirements Specification Document Breakdown

Table 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Work Activities of  Software Requirements specification | Start | Finish | Duration | Sep  2012 | | | | | | | | | | | |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 11 | 12 | 13 | 15 |
| 1 | Use Cases | 8/30 | 9/5 | 6 d | =========== | | | | | | | | | | | |
| 2 | Discovery System Object | 9/5 | 9/7 | 2 d | ======= | | | | | | | | | | | |
| 3 | Drafting of software Requirements Document | 9/7 | 9/10 | 3 d | ======= | | | | | | | | | | | |
| 4 | Finalizing software Requirements Document | 9/10 | 9/15 | 6 d | ============= | | | | | | | | | | | |

#### Software Project Management Plan Document Breakdown

Table 6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Planning Document | Start | Finish | Duration | Sep  2012 | | | | | | | | | | | |
| 16 | 17 | 18 | 20 | 22 | 23 | 25 | 26 | 27 | 28 | 29 | 30 |
| 1 | Project Estimation | 9/15 | 9/20 | 5 d | =============== | | | | | | | | | | | |
| 2 | Drafting of Planning Document | 9/20 | 9/25 | 5 d | =============== | | | | | | | | | | | |
| 3 | Finalizing of Planning Document | 9/25 | 9/30 | 5 d | ================== | | | | | | | | | | | |

#### Software Project Design Document Breakdown

Table 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Project Design | Start | Finish | Duration | Oct  2012 | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 15 |
| 1 | Documentation of Methods/Classes | 9/30 | 10/3 | 3 d | ======== | | | | | | | | | | | |
| 2 | Documentation of Databases | 10/3 | 10/6 | 3 d | ========== | | | | | | | | | | | |
| 3 | Documentation of GUI | 10/6 | 10/9 | 3 d | ========== | | | | | | | | | | | |
| 4 | Drafting of Project Design | 10/9 | 10/12 | 3 d | ======== | | | | | | | | | | | |
| 5 | Finalizing of Project Design | 10/12 | 10/15 | 3 d | == | | | | | | | | | | | |

#### Alternate Analysis

Table 8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Alternative Analysis | Start | Finish | Duration | Oct  2012 | | | | | | | | | | | |
| 15 |  |  | 17 |  |  |  | 18 |  |  |  | 20 |
| 1 | Method Analysis | 10/15 | 10/17 | 2 d | ============= | | | | | | | | | | | |
| 2 | Test Case Execution | 10/17 | 10/18 | 2 d | ============= | | | | | | | | | | | |
| 3 | Documentation | 10/18 | 10/20 | 2 d | ============ | | | | | | | | | | | |

#### Final Documentation

Table 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Final Documentation | Start | Finish | Duration | Oct  2012 | | | | | | | | | | | |
| 20 |  |  | 22 |  |  |  | 23 |  |  |  | 25 |
| 1 | User Manual | 10/20 | 10/22 | 2 d | ============= | | | | | | | | | | | |
| 2 | Testing Document | 10/22 | 10/23 | 2 d | ============= | | | | | | | | | | | |
| 3 | Final Document Compilation | 10/23 | 10/25 | 2 d | ============ | | | | | | | | | | | |

#### GUI Prototype

Table 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  GUI Prototype | Start | Finish | Duration | Aug-Sep  2012 | | | | | | | | | | | |
| 17 | 20 | 23 | 26 | 30 | 5 | 10 | 15 | 20 | 25 | 27 | 30 |
| 1 | GUI Design and Specification | 8/17 | 8/30 | 14 d | =================== | | | | | | | | | | | |
| 2 | GUI Validation | 8/30 | 9/15 | 15 d | ============= | | | | | | | | | | | |
| 3 | Finalize GUI Document | 9/15 | 9/30 | 15 d | =============== | | | | | | | | | | | |

#### Coding / Debugging

Table 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Coding/Debugging | Start | Finish | Duration | Oct-Nov  2012 | | | | | | | | | | | |
| 25 | 26 | 28 | 30 | 2 | 4 | 6 | 8 | 10 | 12 | 13 | 15 |
| 1 | Method Coding | 10/25 | 10/29 | 4 d | =========== | | | | | | | | | | | |
| 2 | Individual Method Testing | 10/28 | 11/1 | 4 d | ========= | | | | | | | | | | | |
| 3 | Integration with Database | 11/1 | 11/5 | 4 d | ========== | | | | | | | | | | | |
| 4 | Database/Method Testing | 11/5 | 11/10 | 4 d | ======== | | | | | | | | | | | |
| 5 | Integration with GUI | 11/10 | 11/15 | 4 d | ====== | | | | | | | | | | | |

#### Program Testing / Validation

Table 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities of  Program Testing/ Validation | Start | Finish | Duration | Nov  2012 | | | | | | | | | | | |
| 15 | 16 | 18 | 20 | 22 | 23 | 24 | 25 | 27 | 28 | 29 | 30 |
| 1 | System Testing/ Debugging | 11/15 | 11/19 | 4 d | =========== | | | | | | | | | | | |
| 2 | Test Cases | 11/19 | 11/22 | 3 d | ========= | | | | | | | | | | | |
| 3 | Alternative Analysis | 11/22 | 11/26 | 4 d | ============= | | | | | | | | | | | |
| 4 | Finalization | 11/26 | 11/30 | 4 d | ================ | | | | | | | | | | | |

### Schedule Allocation

The following diagram describes the nine main work activities and their relationships:

Table 13

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Work Activities | Start | Finish | Duration | Aug 2012 | | | Sep  2012 | | | Oct  2012 | | | Nov  2012 | |
| 17 | 24 | 30 | 10 | 20 | 30 | 10 | 20 | 30 | 15 | 30 |
| 1 | Requirements Specification | 8/17 | 8/30 | 14 d | =========== | | | | | | | | | | |
| 2 | Software Requirements Specification | 8/30 | 9/15 | 16 d | ====== | | | | | | | | | | |
| 3 | Project Planning Document | 9/15 | 9/30 | 16 d | ======= | | | | | | | | | | |
| 4 | Project Design | 9/30 | 10/15 | 16 d | ====== | | | | | | | | | | |
| 5 | Alternative Analysis | 10/15 | 10/20 | 6 d | == | | | | | | | | | | |
| 6 | Final Documentation | 10/20 | 10/25 | 6 d | == | | | | | | | | | | |
| 7 | GUI Prototype | 8/17 | 9/30 | 44 d | ======================== | | | | | | | | | | |
| 8 | Coding/Debugging | 10/25 | 11/15 | 20 d | ======= | | | | | | | | | | |
| 9 | Program Testing/ Validation | 11/15 | 11/30 | 16 d | == | | | | | | | | | | |

#### System Requirements Specification Document

Table 14

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Written List of Requirements | Agreement by Shawn |
| Draft of System Requirements | Contain: System Definition, Diagram of System Architecture, Functional and non- Functional requirements. |
| **Deliverable** |  |
| Completed System Specification | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Software Requirements Specification Document

Table 15

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Written List of Requirements | Agreement by Shawn |
| Draft of Software Requirements Specification | Contain: Use Cases, State Diagrams, Sequence Diagrams for all major components |
| **Deliverable** |  |
| Completed Software Requirements Specification | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Software Project Management Plan

Table 16

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Determination of Schedule | Review of Deadline given by Dr. Zand and Shawn schedule |
| Draft of Planning Document | Contain: Proposed Budgets, Resources, Risk management, and COCOMO 2 estimates |
| **Deliverable** |  |
| Completed Planning Document | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Software Project Design Document

Table 17

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Written List of Classes/Methods | Agreement by Shawn |
| Draft of Project Design | Contain: Detail Methods and Classes |
| **Deliverable** |  |
| Completed Project Design | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Alternative Analysis

Table 18

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Completion of Primary Testing | All primary tests of the systems have been completed, and any bugs solved or in progress is solved. |
| Draft of Analysis and Testing Document | Contain: List of all test and outcomes. |
| **Deliverable** |  |
| Completed System Specification | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Final Documentation

Table 19

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| Draft of User Manual | All major sections are included in User Manual |
| Compilation of Previous Document | Contain: All previous Document brought together, any changed upon agreement of Shawn |
| **Deliverable** |  |
| Final Delivery of all Documents | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### GUI Prototype

Table 20

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| GUI Layout | Diagrams Describing all main screen, logon, guards and admin screen |
| GUI Prototype | Contain: Small subsets of screen showing to Shawn |
| **Deliverable** |  |
| GUI | Screen must be approved by the Shawn |

#### Coding / Debugging

Table 21

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| System Design | Diagrams Detailing all major classes and their methods |
| Major Classes Implementation | Implementation of major classes of the system. They should run without error |
| Integration with Database | Implementation of methods interacting with Database. It ensures add and delete without any error. |
| Integration with GUI | Main classes integrate all classes with GUI Screen.  Methods and screen should be paired up and GUI Should contain major functions like configuration of custom zone, adding deleting users. |
| **Deliverable** |  |
| Total System | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

#### Program Testing / Validation

Table 22

|  |  |
| --- | --- |
| **Milestones** | **Acceptability Criteria** |
| List of Test Cases | Tests have been developed majority of the existing system. |
| Completion of Test Cases | All designed test cases have been carried out. All errors have been fixed. |
| **Deliverable** |  |
| System | Meets all criteria of Draft and has been reviewed and approved by Shawn. |

### Resource Allocation

The following outlines details the resource required for all work items.

#### System Requirements Specification Document

Shawn and Ben working.

#### Software Requirements Specification Document

Shawn working.

#### Software Project Management Plan

Shawn, Ben, Bijoy, and Josh working.

#### Software Project Design Document

Shawn working.

#### Alternative Analysis

Shawn working.

#### Final Documentation

Shawn working.

#### GUI Prototype

Shawn working.

#### Coding / Debugging

Shawn working.

#### Program Testing / Validation

Shawn working.

### Budget Allocation

The anticipated budget is presented in US Dollar.

|  |  |
| --- | --- |
| Item | Estimated Cost |
| Printed Documentation | Per page:$ .15  Page: 200  Sub -total:$ 30 |
| Printed Document Binding | Per Document:$ 5  Document: 8  Sub-total: $40 |
| Personal salary | $ 100,000  .33  4  Sub-total: 132,000 |
| Total | $ 132,070 |
| Contingency Rate | %20 |
| Grand Total | $ 158,484 |

## Control Plan

The following measures will be used for determining project status and for controlling project resources.

* + 1. **Requirements control plan**

Requirements, forming the basis for project implementation, must be an integral part of all subsequent development processes. Therefore, requirements documents shall be regularly (no less than bi-weekly) reviewed by Shawn.

* + 1. **Schedule control plan**

Shawn , as project manager, shall be responsible for monitoring progress. He shall make team members aware of milestone deadlines and monitor progress of individual team members with respect to their assigned subtasks.

* + 1. **Budget control plan**

The only expected real project costs are small quantities of printing and digital media (CDs). Any expenses shall be paid by Shawn.

* + 1. **Quality control plan**
       1. *Quality control of documentation*

Shawn.

* + - 1. *Quality control of code*

Shawn and subversion source control system open to anyone who wants to look.

* + 1. **Reporting plan**

All reporting shall be conducted via email and addressed to Shawn.

* + - 1. *Reporting for subtasks*

None. Anarchy. Anyone may do as they please.

* + - 1. *Reporting for milestone tasks*

There is not team to which I might report.

* + 1. **Metrics collection plan**
       1. *Metrics collection for documentation*

Shawn will edit, collect and compile.

* + - 1. *Metrics collection for code*

Shawn, being the only team member, hence being the team, shall perform all coding and source control procedures.

* 1. Risk management plan
     1. **Determining risk factors**

I will not have time to study for the final.

* + - 1. *Assessing risk factors*

This means I may not do so great on the final.

* + - 1. *Nature of the risk*

My GPA is at stake, and my induction into the college.

* + - 1. *Probability*

It is 100% certain that I will not have time to study for the final.

* + - 1. *Effect*

Fatal.

* + - 1. *Measures*

None.

* + 1. **Prioritizing risk factors**

None.

* + 1. **Monitoring risk factors**

None.

* + 1. **Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Condition | Probability | Effect | Measures |
| Change in requirements | Team determines necessity according to 5.3.1 | 3 | 3 |  |
| Loss of personnel | Any team member leaves project | 3 | 3 |  |
| Loss of documents / code |  | 1 | 6 | Keep backups |
| Underestimation of effort for specific subtask | Indicated by report as per 5.3.5.1 | 3 | 6 | Pre: Allow for extra time.  Post: Assist team member. |

* 1. Closeout plan

The following measures shall be taken in order to ensure an orderly project closeout as well as ensuring an adequate return on time investments.

### Staff reassignment

There is nobody left to reassign.

* + 1. **Personnel debriefings**

No team equals no debriefings.

* + 1. **Final report / lessons learned**

Shawn.

* + 1. **Archiving project materials**

The project manager shall be responsible for collecting and compiling all material turned in by project members over the course of the project.

* + 1. **Materials to be distributed**

Finally, all collected material shall be compiled onto a suitable electronic memory medium and distributed to the course instructor. Material to be included:

1. All deliverables defined in the course syllabus including all revisions.
2. All source code including revisions.
3. The final report / lessons learned document.

# Technical process plans

## Process model

Shawn will use Microsoft Office, UML Tools, Qt, TortoiseSVN, Subversion, cygwin, Adobe Acrobat, Notepad++, Paint Shop Pro and Windows 7.

### Work Product Flow

Hypothetical of course.

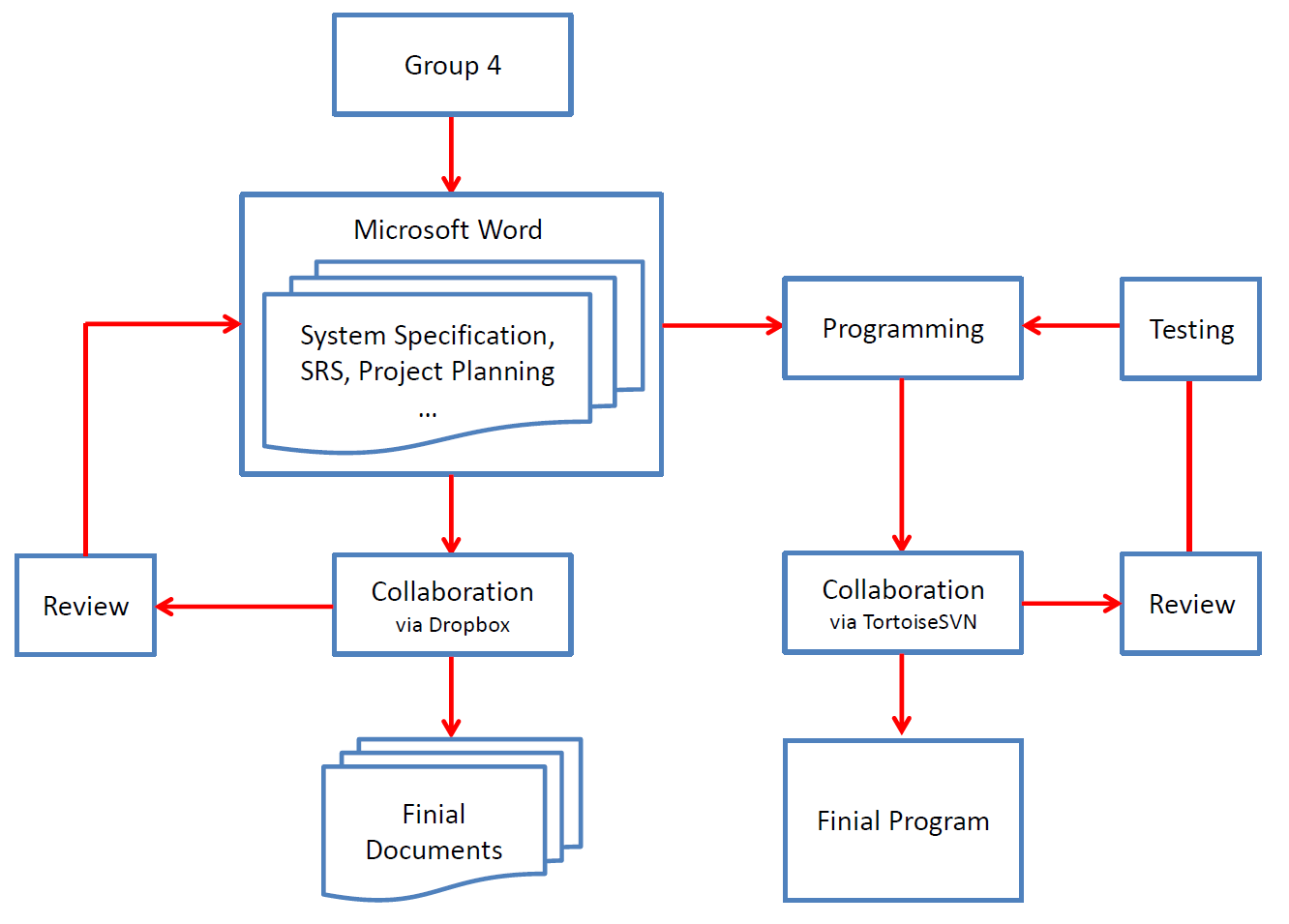


Figure 2

### Work Products Delivery Timing

Table 23

|  |  |
| --- | --- |
| **Deliverable** | **Date** |
| System Specification | 2012/09/10 |
| SRS | 2012/10/01 |
| Planning Document (SPMP) | 2012/10/29 |
| Design Document | 2012/11/18 |
| Code, Build, Test | 2012/12/03 |
| Test Document | 2012/12/03 |
| Legacy Document | 2012/12/10 |
| Compilation of all Documents | 2012/12/10 |

## Methods Tools and Techniques

See 6.1

### Methodology

The team shall use waterfall methodology for the development of this project. The team shall develop document and program steps by steps, and shall consider the feedback from last document/program to build the new document/program. After each document/program draft has been developed, all team members shall provide feedback to the initial draft to improve the quality of the document/program. Then with the combined feedback from group leader, team member and Dr. Zand, we shall develop the next document.

### Tools Used

Table 24

|  |  |
| --- | --- |
| **Work Activity** | **Tool Used** |
| Project Management | None |
| Document Development | Microsoft Word |
| Document Collaboration | None |
| Program Development | C++ |
| Programming Collaboration | TortoiseSVN |

## Infrastructure Plan

See 6.1.

# Supporting Process Plan

## Configuration Management Plan

See 6.1

## Verification and Validation Plan

The project is subject to prototyping, simulation and modeling. First, the GUI prototype will be verified and validated. The project is validated through subversion. Testing, demonstration and analysis are also be used with this tool.

## Documentation Plan

Shawn typing.

## Quality Assurance Plan

Shawn testing.

### Communication During Development

None.

### Installation

The system has passed initial test before it declares usable.

### Support

None.

## Reviews and Audits

None.

## Problem Resolution Plan

Shawn.

## Subcontractor Management Plan

Shawn.

## Process Improvement Plan

Shawn.

# Additional Plans

I plan to juxtapose design models immediately in order to decide upon the development tools I am going to use. This is not to produce implementation details or artifacts that would be premature obfuscations to the design process.

I plan to get very little sleep.