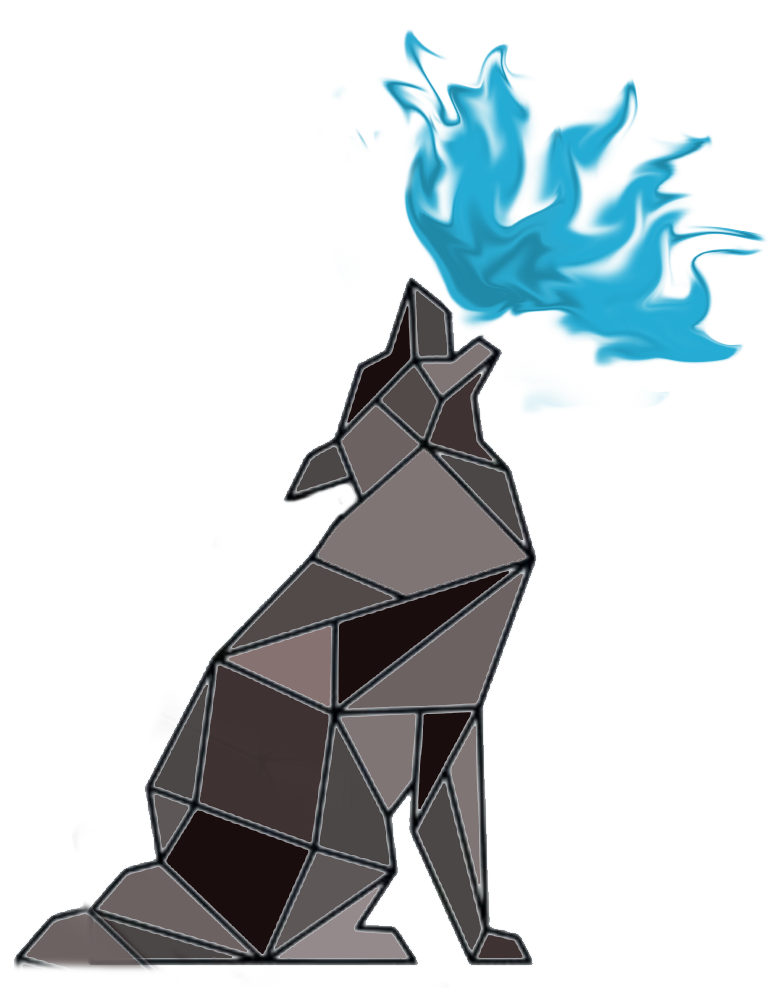
**Software Project Management Plan**

**Breath of the Coyote**

****

**Spring 2020**

**Cameron Maclean – Project Manager**

**Stephen Anderson – Assistant Project Manager**

**Valentino Fernandez – Software Engineer**

**Jugen Fornoles – Software Engineer**

**Christian Morales – Software Engineer**

**Brandon Edmonds – Software Engineer**

Management Plan

1. **Overview**……………………………………………………………………………………..4

1.1 Project summary……………………………………………………………………4

1.1.1 Purpose, scope, and objectives………………………………………...4

1.1.2 Assumptions and constraints…………………………………………....4

1.1.3 Project deliverables……………………………………………………....5

1.1.4 Schedule and budget summary…………………………………………5

1.2 Evolution of the plan………………………………………………………………..5

2. **References**…………………………………………………………………………………...5

3. **Definitions**…………………………………………………………………………………....6

4. **Project organization**………………………………………………………………………..7

4.1 External interfaces………………………………………………………………….7

4.2 Internal structure…………………………………………………………………....8

4.3 Roles and responsibilities………………………………………………………….8

5. **Managerial process plans**………………………………………………………………...9

5.1 Start-up plan………………………………………………………………………...9

5.1.1 Estimation plan…………………………………………………………...9

5.1.2 Staffing plan……………………………………………………………….9

5.1.3 Resource acquisition plan……………………………………………….9

5.1.4 Project staff training plan………………………………………………...9

5.2 Work plan…………………………………………………………………………..10

5.2.1 Work activities…………………………………………………………...10

5.2.2 Schedule allocation……………………………………………………..11

5.2.3 Resource allocation……………………………………………………..11

5.2.4 Budget allocation………………………………………………………..11

5.3 Control plan………………………………………………………………………...11

5.3.1 Requirements control plan……………………………………………..11

5.3.2 Schedule control plan……………………………………………….11-12

5.3.3 Budget control plan……………………………………………………..12

5.3.4 Quality control plan………………………………..……………………12

5.3.5 Reporting plan……………………………………………...…..……….12

5.3.6 Metrics collection plan………………………………………………….12

5.4 Risk management plan…………………………………………………………...13

5.5. Closeout plan……………………………………………………………………..13

6. **Technical process plans**………………………………………………………………....14

6.1 Process model……………………………………………………………………..14

6.2 Methods, tools, and techniques………………………………………………….14

6.3 Infrastructure plan………………………………………………………………....15

6.4 Product acceptance plan………………………………………………………....15

7. **Supporting process plans**……………………………………………………………….15

7.1 Configuration management plan………………………………………………...15

7.2 Verification and validation plan…………………………………………………..15

7.3 Documentation plan……………………………………………………………….15

7.4 Quality assurance plan…………………………………………………………...15

7.5 Reviews and audits……………………………………………………………….15

7.6 Problem resolution plan…………………………………………………………..16

7.7 Subcontractor management plan………………………………………………..16

7.8 Process improvement plan……………………………………………………….16

**1. Overview**

**1.1 Project Summary**

This Software Project Management Plan will lay out the details of the management plan to be followed during the development of the Breath Of The Coyote mobile game. It includes the development cycle, organization, specific roles, projected timelines, and testing protocols.

**1.1.1 Purpose, Scope, and Objective**

This Software Project Management Plan (SPMP) will relay all the details regarding the development plan and the development cycle. It will assign each member roles and duties regarding the development, as well as specify what methods will be used in order to finish assigned tasks. It will also assign deadlines to ensure that the project will be developed on time. It will follow the specifications declared in the most current version of the SRS that was signed by the client. The first prototype will be a functioning mobile build of the previous game with the controls and UI properly built and functional. The second prototype will have all bug fixes that have been discussed with the client, a new level, and the introduction of a new battle style.

This game was originally developed for Fall 2019 CSE 440 Game Design course and was created by a team composed of Daniel Meyer (the Client), Cameron Maclean, Benjamin Alexander, Jose Perez, and Chris Magnuson. It was designed to be an open world 3rd person role playing game (RPG) where you play as a young man trying to prove himself worthy of becoming the new chief of his village. It had a mix of puzzle solving, platforming, and fighting mechanics. Ultimately, it won the Game Design Competition at the end of the quarter for the class. Now, we are planning on making a mobile port for the game with the guidance of our client, Daniel Meyer.

**1.1.2 Assumptions and Constraints**

The list of all Assumptions and Constraints:

* Team members will attend all meetings
* Team members will meet all deadlines
* Team members will follow the requirements specified in SRS
* Team members will communicate effectively with the rest of the team
* Team members will be using Unity version 2019.2.3f1 and will work outside of class hours to meet deadlines

**1.1.3 Project Deliverables**

The project will deliver the following items:

* Working game on Google Play store
* SRS, SPMP, Documented Source Code, Design, Test Plan

**1.1.4 Schedule and Budget Summaries**

There is no current budget for this game. Prototype one will be delivered by the end of Week 7, and the second prototype will be delivered during the final week of class.

**1.2 Evolution of the Plan**

The whole team met with the client on April 13th, 2020 and discussed the expectations for the project and the current situation with the game. After all requirements were listed out, roles were assigned depending upon interest and experience in their respective fields. Deadlines were given for the first set of goals, and we have weekly meetings to keep everyone up to date on the software’s progress.

**2. References**

[1] IEEE Software Engineering Standard Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] Unity user Manual<https://docs.unity3d.com/Manual/index.html>

[3] Original Game Design Document for Breath of the Coyote, CSE 440

[4] UMVELT Software Program Management Plan <https://mobileappdev.academic.csusb.edu/wp-content/uploads/2019/04/Software-Project-Management-Plan.pdf>

[5] CoyoteQuest Software Program Management Plan <https://mobileappdev.academic.csusb.edu/wp-content/uploads/2020/04/CoyoteQuest-SPMP-Revision-5-6-19.pdf>

**3. Definitions**

**Android** – Mobile Operating System running on the user’s device

**Unity Engine** – Video Game Engine used for developing games on many platforms, including Android

**User** – Someone who uses the application or software specified in this Software Requirement Specification

**SRS** – The Software Requirement Specification, the technical document that outlines the structure and organization for the software and its requirements

**Android SDK** – A Software Development Kit (SDK), a set of development tools used to develop applications for the android platform

**Client** – Daniel Meyer, the person for whom the app is for

**App** – shortened for mobile application

**Launch Page** – The first page of the app that the user can interact with

**Main Menu** – The launch page for the app

**Game Screen** – The page where the app is played, accessible from Main Menu

**UI** – User Interface, the different elements the user can see and interact with

**UML** – Unified Modeling Language (UML), a standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of a software system

**Use Case Diagram** – A representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved

**C#** - A General purpose object-oriented programming language

**Project Manager** - Cameron Maclean, interfaces with the client and communicates the client’s desires to the rest of the team, organizes the team’s structure and assignments

**Assistant Project Manager** - Stephen Anderson, assists with software development and monitors the team’s progress

**Software Engineering Team** - Software engineers who are a part of the Breath of the Coyote team and are responsible for creating or fixing software elements

**QA** - Quality Assurance, test the apps functionality and check for software bugs

**QA Team** - Member of the classes QA team who are tasked with testing the game

**Documented Source Code** -All the code written for the project properly commented and explained clearly

**RPG -** Role Playing Game

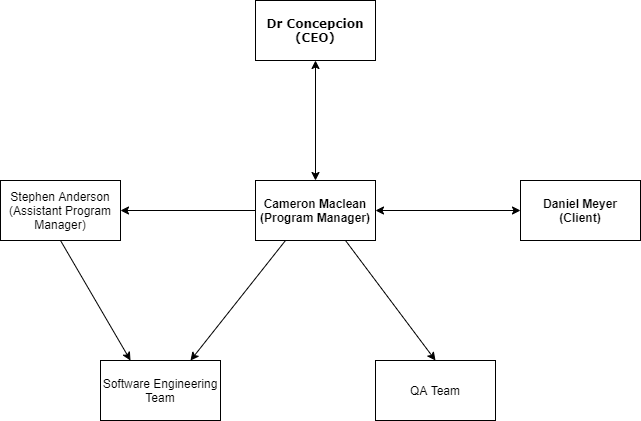
**SPMP -** This document, the Software Project Management Plan

**LOC** - lines of code

**KLOC -** Thousand lines of code

**4. Project Organization**

**4.1 External Interfaces**



**Dr. Concepcion (CEO)** - Monitors performance of the development team and provides guidance to the project managers

**Cameron Maclean (Project Manager)** - Fosters communication between the client and the team. Uses communication resources such as Discord, slack, and github project boards to communicate between the different members of the development team.

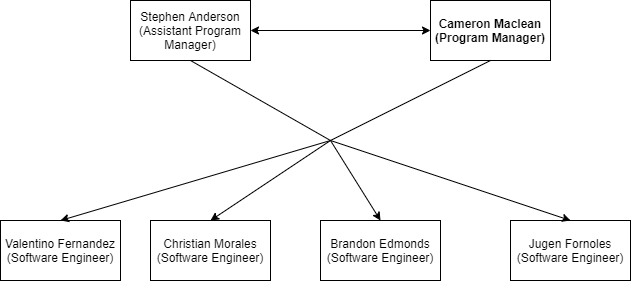
**Stephen Anderson (Assistant Project Manager)** - Monitors team progress in accomplishing development goals

**Daniel Meyer (Client)** - Communicates to Project Manager the requirements of the mobile app and gives feedback on presented prototypes

**Software Engineering Team** - Codes the required features of the app

**QA Team** - Evaluates prototypes as they are made available and tests for worst case scenarios

**4.2 Internal Structure**



**4.3 Roles and Responsibilities**

**Cameron Maclean (Project Manager)** - Communicates to the team what the client desires in the mobile game. Developing additional levels for the game.

**Stephen Anderson (Assistant Project Manager)** - Works with the Project Manager to complete prototype development. Work on fixing multiple issues with the player Animator, and experiment on blending existing animations

**Valentino Fernandez (Software Engineer)** - Bug fixes already existing in the project such as the ground slam ability not activating on input, the player sometimes failing to get an end state in the case of death

**Christian Morales (Software Engineer)** - Touch Screen input and UI development and functionality for mobile port

**Brandon Edmonds (Software Engineer)** - Fix AI by attaching wirespheres for the enemy to detect the player instead of fighting only when the player is in a certain area.

**Jugen Fornoles (Software Engineer)** - Improve the combat system by changing the combo system in the game and making it more impactful.

**5. Managerial Process Plans**

**5.1 Start-Up Plan**

* Discussed with the client on April 8th on what expectations he had for the upcoming project and what we both wanted to see done
* Met with team on April 13th with what goals were expected of us
* Each member chose what specific job they wanted to focus on
* Once roles were chosen, we moved onto when each individual job should be finished

**5.1.1 Estimation Plan**

* Discuss with the client all the specifications for the mobile game
* Study and research the existing code and work to integrate the new features into the game
* Figure out how much could be accomplished in the given time
* Create guidelines for working prototypes 1 and 2
* Create estimated deadlines and task completions deadlines to create these prototypes on time

**5.1.2 Staffing Plan**

The project staff was selected by the professor and the TA’s by looking at the survey that each member had filled out in the first week of the class.

**5.1.3 Resource Acquisition Plan**

All the software for this project is free. Every member has the means of working on the project from home.

**5.1.4 Project Staff Training Plan**

All the group members will complete the tutorials given the first three weeks of the class. After finishing those labs each member of the group will study the code on Unity Engine and learn how the technology works in order to complete the project. Tutorials given to us were the Android development tutorial and Asynchronous tutorial given on Android Studio. The entire group already has experience in Unity, so tutorials for that development tool were not necessary.

**5.2 Work Plan**

**5.2.1 Work Activities**

**Prototype 1**:

* UI/UX - All mobile game UI will be present, or at the least have placeholder assets. Current UI is inadequate for a mobile release. We are going to need to add 4 buttons for the abilities and 2 joysticks for camera and player movement. By prototype 1, the buttons and joysticks will be in position. This will be done by Christian Morales and Valentino Fernandez.
* Gameplay - The game will play very similar to how it has already played before in CSE 440. The only new fix that will be in the game will be the Groundslam ability and it will begin to work as intended. This will be done by Valentino Fernandez and Jugen Fornoles.
* All other fixes mentioned during prototype 2 will be worked on during the prototype 1 phase but are not planned to be completed by then.

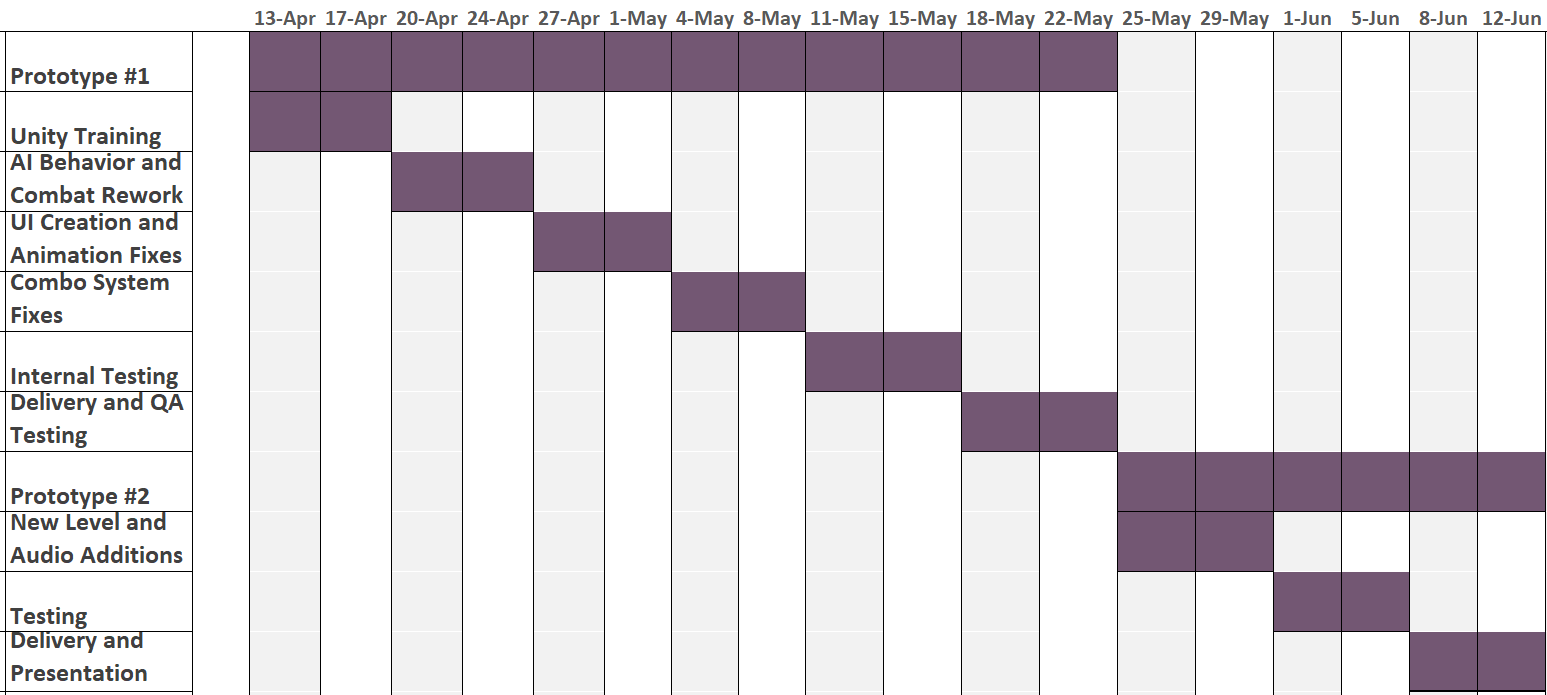
**Prototype 2:**

* Gameplay - There will be several additional fixes added to the game, including the player not being able to stop attacking, and some other issues that are caused by the Combo system currently present in the game. This will be done by Jugen Fornoles and Valentino Fernandez.
* AI - AI will be revamped so that instead of enemies being spawned into the map and being given the player target they will be already on the map and be able to detect the player when he gets closer. This will be done by Brandon Edmonds.
* Audio - There will be additional thematic music added to the game. This will be done by Christian Morales and Stephen Anderson.
* Animations - Many issues with the gameplay originates from improper calls by the player animator. These will be fixed in the final product and will be done by Stephen Anderson and Jugen Fornoles.
* Level - an additional level will be added into the game. This level will attempt to emphasize the new combat and AI systems created by creating more combat encounters without sacrificing the other key systems in the game. This will be done by Cameron Maclean.
* Game will be ready to be put on the Google Play Store

**Documentation:**

* All members will make sure that all the code they write will be properly documented

**5.2.2 Schedule Allocation**



**5.2.3 Resource Allocation**

Each member has access to the same resources, which include the Unity Engine and all the documentation that has come with the project.

**5.2.4 Budget Allocation**

No budget has been allocated for this project.

**5.3 Control Plan**

**5.3.1 Requirements Control Plan**

Each member of the group is required to attend the meetings in lab time. Each member is also required to document their code, follow the guidelines decided in the SRS and meet each deadline as well. Any unexpected issues, technical difficulties or requests by the clients will be assessed by the Managing team and decided upon.

**5.3.2 Schedule Control Plan**

Aside from the regular online class meeting time, the team might be required to meet outside the class as well in order to finish and deliver the product on time. These meetings will be held online as well. Managers will make sure that each member is completing their tasks and on time. The managing team will keep a constant eye on the progress of the project and make sure that everything is completed by the deadline. The Project Manager and the Assistant Project Manager will keep each other up to date on the progress of the project.

**5.3.3 Budget Control Plan**

There is no budget allocated to the Breath of the Coyote team. Everything the team is using is already free and in the project itself.

**5.3.4 Quality Control Plan**

The managing and the developing team will constantly perform a quality check on the software at least once a week to make sure that the project meets all the expectations. Also the client will be present during the development of the application and therefore will be informed of the quality of the application.

**5.3.5 Reporting Plan**

The managing team will notify Dr Concepcion of any problems, as well as the attendance of labs. The project manager will send him the attendance of the group members and updates on the progress of the project.

**5.3.6 Metrics Collection Plan**

The managing team will make sure that each week every developer completes their tasks so that the project is on track and progressing as expected. The managing team will make sure that the code is efficient, and meets all the standards. The project manager will calculate the LOC/hr productivity and # of faults/KLOC and report it to Dr. Concepcion to keep him up to date on the progress of the team.

**5.4 Risk Management Plan**

**Development**:

* The team will meet regularly to make sure that the production is not stopped at one particular point
* There will be scheduled deadlines that everyone will be following to make sure the project is not delayed
* Each team member will be kept up to date, and inform the managing team of any changes or difficulties that might affect their ability to complete the task they are given on time.

**Project Failure**

* If the technology does not exist or is not viable the managing team will discuss with the client and make sure that there is another route that will result in a satisfactory outcome for the client.

**Server Failure**

* There is no server involved with this project

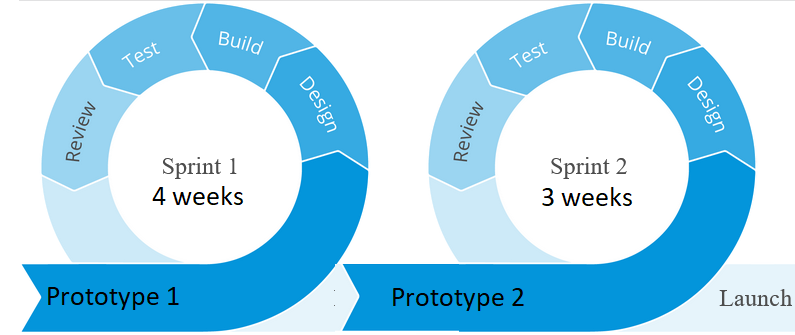
**5.5 Close-out Plan**

The team will submit all the deliverables on to the bitbucket repository, along with a maintenance manual. The team will also present their application on the finals day. Everything in the bitbucket will be as follows:

* Documented Source Code
* SRS
* SPMP
* SQAP
* SAD
* Built Project
* Maintenance Manual

**6. Technical Process Plan**

**6.1 Process Model**

****

The Breath of the Coyote team will be using a version of the Agile Development Model. The Agile Development model has several phases: Planning, Development, Review/Testing, and Launching. While the Software Engineering Team is getting themselves familiar with the code in the project and understanding what each program does, the management team will be planning roles and deadlines for everything needed in the project. During the Development process, each member of the team will be working hard on their jobs to get its respectives deadlines finished. On top of their own duties, the management team will make sure they are available if any member is in need of assistance during this process. The review phase will be the most important phase when we comb through everything we have worked on to try and find and fix any bugs before the final launch. We must repeat this cycle twice, spending the first 4 weeks on prototype 1, getting whatever info we can back from the QA team and repeating the previous cycle again over a 3 week spread to complete prototype 2. If anything truly serious is found by the QA team, the current plan can be easily changed to accommodate these issues, if any exist.

**6.2 Methods, Tools, and Techniques**

Method: Incremental Development Model

Tools: Unity Engine, Github, Bitbucket, Discord

Techniques: Regular meetings with the client.

**6.3 Infrastructure Plan**

There is no need for an infrastructure plan at the moment, since there will be no server used by the application for now.

**6.4 Product Acceptance Plan**

The client will be kept in the loop throughout the development. After which the Q&A team will test the application and make sure that the prototypes are working and acceptable.

**7. Supporting Process Plan**

**7.1 Configuration Management Plan**

We are using bitbucket for the configuration. It will contain all the changes made to the code and will allow for a smoother configuration.

**7.2 Verification and Validation Plan**

Verification and validation is done through periodic testing of the mobile game. Each team member will perform unit testing on the units they work on, as well as integration testing with fellow teammates. The entire team will participate in system testing before giving any prototypes to the QA team. Any bugs or errors that are found are documented and reported online.

**7.3 Documentation Plan**

The managing team will prepare the SRS and SPMP. The development team will write the documentation for design and architecture.

**7.4 Quality Assurance Plan**

The Q&A team will make sure of the quality of the application.

**7.5 Reviews and Audits**

During development and testing phases, every member of the development team will test and report any deficiencies in the mobile app. Design flaws or bugs will be reported and documented for immediate fix or future review.

**7.6 Problem Resolution Plan**

Each member of the development team will keep the managing team members up to date on any issues that they might encounter. Afterwards the managing team will decide on how to handle those issues and make sure that the project is completed. They will also make any changes necessary to make sure that the project runs as smoothly and as efficiently as possible.

**7.7 Subcontractor Management Plan**

We have no subcontractors.

**7.8 Process Improvement Plan**

In order to improve the process and development of the software, it is crucial to write good documentation of the source code that will be provided. There are features that cannot be implemented in the given time at the moment, and therefore have been pushed back to the next stage of development. Writing quality code now, documenting it and creating a good maintenance manual will help the future development team to make all the changes that they think are necessary. Also, all features and bugs will be logged and will be available to the whole team for testing and quality checking.