**Software Project Management Plan**

**Team: Code Slayer**

**Elvis Jimenez – Team lead and SQA Representative**

**Akshay Patel – Co-Team lead and Co-Tester**

**Luis Oropeza – Tester and Co-Secretary**

**Nhan Nguyen – SQA Representative and Co-Tester**

**Stephanie Reyes – Secretary and Tester**

**Table of Contents**

[PREFACE](#_gjdgxs)

[1. Overview of the Product](#_30j0zll)

[Purpose Scope & Objective](#_1fob9te)

[1.2 Assumptions & Constraints](#_3znysh7)

[1.3 Project Deliverables](#_2et92p0)

[1.4 Schedule & Budget Summary](#_tyjcwt)

[1.5 Evolution of the Plan](#_3dy6vkm)

[2. References](#_1t3h5sf)

[3. Definitions & Acronyms](#_4d34og8)

[4. Project Organization](#_2s8eyo1)

[4.1 External Interfaces](#_17dp8vu)

[4.2 Internal Structure](#_3rdcrjn)

[4.3 Roles & Responsibilities](#_26in1rg)

[4.4 Team Methods of Operation](#_lnxbz9)

[5. Managerial Process Plans](#_35nkun2)

[5.1 Start-Up Plan](#_1ksv4uv)

[Training](#_44sinio)

[5.2 Work Plan](#_2jxsxqh)

[5.2.1 Work Activities](#_z337ya)

[5.2.2 Schedule Allocation](#_3j2qqm3)

[5.2.3 Resource Allocation](#_1y810tw)

[5.2.4 Budget Allocation](#_4i7ojhp)

[5.3 Control Plan](#_2xcytpi)

[5.3.1 Requirements Management](#_1ci93xb)

[5.3.2 Schedule Control](#_3whwml4)

[5.3.3 Resource Control](#_2bn6wsx)

[5.3.4 Budget Control](#_qsh70q)

[5.3.5 Reporting & Communication Plan](#_3as4poj)

[5.3.6 Measurement Plan](#_1pxezwc)

[5.4 Risk Management Plan](#_49x2ik5)

[Risk Management Note](#_2p2csry)

[5.5 Closeout Plan](#_147n2zr)

[6. Technical Process Plans](#_3o7alnk)

[6.1 Process Model](#_23ckvvd)

[6.2 Methods, Tools, & Techniques](#_ihv636)

[Programming Environment](#_32hioqz)

[Database Environment](#_1hmsyys)

[Version Control](#_41mghml)

[Documentation](#_2grqrue)

[Testing](#_vx1227)

[6.3 Infrastructure Plan](#_3fwokq0)

[6.4 Product Acceptance Plan](#_1v1yuxt)

[6.5 Deployment Plan](#_4f1mdlm)

[7. Supporting Process Plans](#_2u6wntf)

[7.1 Configuration Management Plan](#_19c6y18)

[7.2 Product Testing & Reviews Plan](#_3tbugp1)

[7.3 Document & Work Product Plan](#_28h4qwu)

[Requirements](#_nmf14n)

[Product Specification](#_37m2jsg)

[Design Documentation](#_1mrcu09)

[Implementation Documentation](#_46r0co2)

[Test Documentation](#_2lwamvv)

[7.4 Quality Assurance Plan](#_111kx3o)

[7.5 Project Progress Reviews](#_3l18frh)

[7.6 Issue Management](#_206ipza)

[7.7 Version Management](#_4k668n3)

[7.8 Subcontract Management (Acquisition Management) Plan](#_2zbgiuw)

[7.9 Process Improvement Plan](#_1egqt2p)

[Document Control](#_3ygebqi)

[Change History](#_2dlolyb)

[Document Storage](#_sqyw64)

[Document Owner](#_3cqmetx)

[Appendices](#_1rvwp1q)

# PREFACE

# 1. Overview of the Product

## Purpose Scope & Objective

The purpose of the product is to design a computer game called Tic-Tac-Toe on a 5x5 grid. The software will offer a single and multiplayer game feature, where the winner is objective is to meet the client’s requirements and launch a demonstration of the product on November 29, 2016.

## 1.2 Assumptions & Constraints

The assumptions are:

* The concept of tic-tac-toe is not new so we will be using an existing idea.
* The team members will dedicate between 8 - 10hrs per week on this project.
* All team members are knowledgeable in C++ and Visual Studio IDE.

Our team will face constraints which include:

* The completion of the product must be executable by Nov 29, 2016.
* The product must be error free and user friendly.
* The documentation must be considered for logging purposes during all the phases of the project.

## 1.3 Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Major Deliverable** | **Planned Delivery Date** | **Author** | **Delivery Mechanism** |
| Tic-tac-toe 5x5x4 Rough Draft | Oct 2, 2016 | Team Code Slayer | Documentation |
| Tic-tac-toe 5x5x4 2nd Draft | Nov 6, 2016 | Team Code Slayer | Documentation and Preliminary Design. |
| Tic-tac-toe 5x5x4 GUI | Nov 29, 2016 | Team Code Slayer | Complete paperwork, and Demo. |
| Tic-tac-toe 5x5x4 Complete | Dec 6, 2016 | Team Code Slayer | Complete paperwork, and Complete Program. |

## 1.4 Schedule & Budget Summary

Project schedule summary:

8/26/2016 Team’s Week 1 meeting

9/9/2016 Team’s Week 2 meeting

9/16/2016 Team’s Week 3 meeting

9/23/2016 Team’s Week 4 meeting

9/30/2016 Team’s Week 5 meeting

10/2/2016 Submission of baseline plan to client

10/7/2016 Team’s Week 6 meeting

10/14/2016 Team’s Week 7 meeting

10/21/2016 Team’s Week 8 meeting

10/28/2016 Team’s Week 9 meeting

11/1/2016 First Product Presentation

11/4/2016 Team’s Week 10 meeting

11/6/2016 Submission of improvement plan to client

11/11/2016 Team’s Week 11 meeting

11/18/2016 Team’s Week 12 meeting

11/25/2016 Team’s Week 13 meeting

11/29/2016 Demo Presentation

12/2/2016 Team’s Week 14 meeting

12/4/2016 Final Documentation submission to client

12/6/2016 Product Competition

Project budget summary:

1. Project cost is $0.00
2. Product maintenance cost is $0.00

## 1.5 Evolution of the Plan

Project plans will be update on a weekly basis every Monday and Friday. All notification will be done by email, groupme, github, and slack.

# 2. References

|  |  |  |
| --- | --- | --- |
| **Resource** | **Identifier** | **Description of Use** |
| Cplusplus.com | Website | Gather information how the program should act. |
| Youtube.com | Website | Learn to implement a GUI interface |
| Object-Oriented Classical Software Engineering | Book | Gather information on how to implement a product as a team |

# 3. Definitions & Acronyms

## Definitions

|  |  |
| --- | --- |
| Mark | Figures assigned to each player, which are used to make a line. |
| Lines | Four marks in a row from any angle rewards a point to a player. |
| Points | Number in which a player has connected a line of four in a row. |
| Score | Total of how many points each player has accumulated. |
| Player | Actor taking part in competing to win within the rules of the game. |
| Versus | One party opposing a different party in the game. |
| Start | Initiating an action. Launching the game. Beginning a game session. |
| Index | Box selected to set a stone |
|  |  |
|  |  |

## Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| TTT | Tic-Tac-Toe |
| AI | Artificial Intelligence |
| GUI | Graphical User Interface |
| SQA | Software Quality Assurance |

# 4. Project Organization

## 4.1 External Interfaces

Our client is Dr. Shengli Yuan and the team lead meets with him on Tuesdays at 12:45 for clarification of product requirements.

## 4.2 Internal Structure

Our Team is named Code Slayer and the team members are Elvis Jimenez, Luis Oropeza, Nah Nguyen, Akshay Patel, Stephanie Reyes

## 4.3 Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| **Roles** | **Responsibilities** | **Person** |
| Team Lead and SQA Representative | Reporting of project management and control of risks and team cohesiveness. | Elvis Jimenez |
| Tester and Co-Secretary | Oversees standards and procedures are correctly implemented and consolidate document. Insuring that code is working. | Luis Oropeza |
| SQA Representative and Co-Tester | Oversees standards and procedures are correctly implemented and insuring the code is working. | Nah Nguyen |
| Co-Team Lead and Co-tester | Back up of project management and control of risks and team cohesiveness and insuring the code is working | Akshay Patel |
| Secretary and Tester | Consolidate document communication with application  and insuring the code is working | Stephanie Reyes |

## 4.4 Team Methods of Operation

Team member will meet every Friday to work on project. Documentation and code will be updated accordingly. Team members will also have access to work remotely and update progress through GitHub. If there an issue it will be reported to the team lead to find a solution. Any concerns that the team has involving the program operation will lead to setting time aside to talk to the client.

# 5. Managerial Process Plans

## 5.1 Start-Up Plan

### Training

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Training** | **Method** | **Scheduled Dates** | **Cost** |
| C++ programming | online instructional | September 2, 2016 | No cost |
| Win32 API interface | online instructions | October 7, 2016 | No cost |
|  |  |  |  |

## 5.2 Work Plan

### 5.2.1 Work Activities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Project Member** | **Phases Involved** |
| Form Team | All team members | Form a team, decide on roles, and establish method of communication |
| Planning | All team members | Set dates to meet up and work on the project. |
| Documentation | All team members | Start the documentation process. |
| Design | All team members | Plan the components of the GUI |
| Implementation | All team members | Begin coding |
| Testing | All team members | Testing iterations |
| Rehearsals | All team members | Demo and presentation rehearsals |

### 5.2.2 Schedule Allocation

Refer to section 1.4 for schedule. The contingency plan is that the team lead will guide the group on work to be conducted outside of the planned schedule.

### 5.2.3 Resource Allocation

Refer to section 4.3 for the team’s resource. The contingency plan is if the team loses a member from the project everyone responsibility is enhanced and the work load requires more time.

### 5.2.4 Budget Allocation

The projected costs are set to $0.00 and it should not exceed that limit.

## 5.3 Control Plan

Changes made to any document or code have to be approved by all team members. The team will have to ensure that the changes are feasible and permissible within the time and budget constraints of the project.

### 5.3.1 Requirements Management

|  |  |  |
| --- | --- | --- |
| **Requirements Management Activities** | **Performed By Whom** | **Comments** |
| Team communication | Stephanie Reyes | Use Group Me, Slack, and Email applications |
| Documentation workspace | Stephanie Reyes | Use GitHub |
| Application workspace | Elvis Jimenez | Use visual studio & GitHub |
| Training workspace | Elvis Jimenez | Online resources |

### 5.3.2 Schedule Control

Our schedule is structured so that the team meets every Friday for two hours since August 24th, 2016. If the time allotted is not enough, the team will be working online via GitHub.

### 5.3.3 Resource Control

All members are responsible for the task assigned to him/her. If a task falls behind schedule, the team member will need to spend extra time to make up for the time loss.

### 5.3.4 Budget Control

All members are required not to exceed the allocated budget.

### 5.3.5 Reporting & Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Information** | **Frequency** | **From** | **To** | **Medium** |
| Documentation changes | Weekly | Any member | All members | GroupMe and Sack apps and GitHub |
| Code updates | Weekly | Any member | All members | GroupMe app and Visual Studios, and GitHub |
| Code errors | Weekly | Any member | All members | GroupMe app, Visual Studios, and GitHub |
| Issues with attending meetings | Prior to meeting, weekly | Any member | Team Lead | GroupMe app |
| Issues with understanding a coding concept | Weekly | Any member | All members | GroupMe app |
| Backup or loss of project data | ASAP | Any member | Team Lead | GroupMe app |

### 5.3.6 Measurement Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Measure Required** | **Frequency Collected** | **Collected By Whom** | **Analyzed By Whom** | **Used By Whom** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 

## 5.4 Risk Management Plan

|  |  |  |
| --- | --- | --- |
| **Risk Management Activity** | **Performed By Whom** | **Comments** |
| Time constraints risk | All team members | If a member start to fall being the team lead have to pick up the slack to make up the time. |
| Data loss risk | All team members | All team members have a back of the data for this project. GitHub has a version of our data as well. |
| Members leaving risk | All team members | If a team member leave the roles will need to be divided among the rest of the team members. |
| Equipment malfunction risk | All team members | All members have a computer or a way to get a computer to perform project presentation needed. |
| Not enough logged of the codding | All team members | Any and all team members who have the knowledge of C++, would instruct the team member/s in question. |
| Poor communication risk | All team members | The group has set up a slack, group me, and email to pass along any message. |

### Risk Management Note

The risks are considerably important; therefore, the best method to mitigate the risks is by constant communication and consistent teamwork.

## 5.5 Closeout Plan

|  |  |  |
| --- | --- | --- |
| **Closeout Activity** | **Performed By Whom** | **Comments** |
| Documentation copies to client | Secretaries | Electronic and hard copy |
| Conduct product post maintenance | SQA representatives |  |
| Archive complete project | Team Lead |  |
|  |  |  |

# 6. Technical Process Plans

## 6.1 Process Model

This project will be implemented and executed using the Code-and-Fix Life-Cycle Model.

## 6.2 Methods, Tools, & Techniques

### Programming Environment

This project will be coded in C++ language.

### Database Environment

The project will be using Visual Studio with MFC applications as the GUI interface.

### Version Control

The project documentation will use in GitHub to store all versions and allow for changes.

### Documentation

There are 4 documents covering the entire project:

1. Requirements
2. Product Specification
3. Software Project Management Plan
4. Test Plan

### Testing

Testing will begin after the first prototype has been completed. Any changes and improvements to the code will be tested as added. Testing will be conducted by all members. Testing will continue post submission for maintenance of product.

## 6.3 Infrastructure Plan

All team member have to have the ability to access a computer that has the Visual Studio software. Keeping in mind that the minimum requirements are 2gb of ram, 10GB of hard drive space, i3 processor, and at oldest, a windows 7 operating system. Computer access can be at university computer labs or personal computers.

## 6.4 Product Acceptance Plan

The client and the team lead will set a time before Nov 29, 2016 to confirm approval of the final product.

## 6.5 Deployment Plan

Goal for project to be completed by Nov 8, 2016, to include client requirements review.

# 7. Supporting Process Plans

## 7.1 Configuration Management Plan

All preliminary versions that are submitted to the client for review will be labeled by the file name and version number.

## 7.2 Product Testing & Reviews Plan

Product testing and reviews will be done by the Software Quality Assurance team members following the Validation Plan. Testing is conducted for every change and improvement. Documentation of errors and improvement is recorded.

## 7.3 Document & Work Product Plan

### Requirements

The requirement document is completed by Nah Nguyen and Akshay Patel. All other team members will review and edit. Team consensus is required before submission.

### Product Specification

The Specification document is completed by Luis Oropeza and Elvis Jimenez. All other team members will review and edit. Team consensus is required before submission.

### Design Documentation

All team members will be submitting input in the design documentation. Team consensus is required before submission.

### Implementation Documentation

Stephanie Reyes, Luis Oropeza, and Elvis Jimenez will be completing the implementation documentation. All other team members will review and edit. Team consensus is required before submission.

### Test Documentation

The testing documentation will be completed by Nah Nguyen, Luis Oropeza, Akshay Patel. Team consensus is required before submission.

## 7.4 Quality Assurance Plan

Stephanie Reyes and Elvis Jimenez will manage the project.

Stephanie Reyes, Luis Oropeza, and Elvis Jimenez will work on the project development.

Nah Nguyen, and Akshay Patel will work on the project deployment.

## 7.5 Project Progress Reviews

The team lead will check on the progress of the project. Every weekly team meeting will be used to work on new tasks. Any work conducted remotely by any team member will be notified to the entire group via GroupMe communication.

## 7.6 Issue Management

|  |  |  |
| --- | --- | --- |
| **Issue Management Activities** | **Performed By Whom** | **Comments** |
| Training management | SQA representatives | if not enough training, provide online resources |
| Equipment management | Team Lead | if personal laptops give technical difficulties, use school computers |
| Communication management | Secretaries | use alternate methods, not just GroupMe app |
| Any new additional issues | Team Lead |  |

## 7.7 Version Management

|  |  |  |
| --- | --- | --- |
| **Change Management Activities** | **Performed By Whom** | **Comments** |
| Requirements version 1 | Akshay Patel, Nah Nguyen | GitHub for remote |
| Specifications version 1 | Luis Oropeza | GitHub for remote |
| Software Management Plan version 1 | Elvis Jimenez, Stephanie Reyes | GitHub for remote |

## 7.8 Subcontract Management (Acquisition Management) Plan

## 7.9 Process Improvement Plan

The product satisfaction will be in accord to the external interfaces requirements. The improvement of the capability maturity model is targeted to advance from level 2 to level 3 within 2 years.

# Document Control

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description (Including Page #’s)** |
| 1.0 | 10/1/2016 | Code-Slayer | Whole document |
| 1.1 |  | Code-Slayer | Pages 9, 8, and 12 |
|  | 10/31/2016 |  |  |
| 2.1 | 11/6/2016 | Code Slayer | Updated all documents |

## Document Storage

Our documentation will be accessed and stored through GitHub via https://github.com/EJJG/Code-Slayer. Only accessible to members with folder permissions to edit and view. GitHub email is required to login.

## Document Owner

All documentation is accessible to members for remote changes. Final submissions will be done by Team Lead.

# Appendices

|  |  |  |
| --- | --- | --- |
| **Appendix** | **Title** | **Location Or Link** |
| N/A |  |  |