**Software Project Management Plan**

**Table of Contents**

[PREFACE 4](#_Toc263068437)

[1. Overview of the Product 4](#_Toc263068438)

[Purpose Scope & Objective 4](#_Toc263068439)

[1.2 Assumptions & Constraints 4](#_Toc263068440)

[1.3 Project Deliverables 4](#_Toc263068441)

[1.4 Schedule & Budget Summary 4](#_Toc263068442)

[1.5 Evolution of the Plan 5](#_Toc263068443)

[2. References 5](#_Toc263068444)

[3. Definitions & Acronyms 5](#_Toc263068445)

[4. Project Organization 5](#_Toc263068446)

[4.1 External Interfaces 5](#_Toc263068447)

[4.2 Internal Structure 5](#_Toc263068448)

[4.3 Roles & Responsibilities 6](#_Toc263068449)

[4.4 Team Methods of Operation 6](#_Toc263068450)

[5. Managerial Process Plans 7](#_Toc263068451)

[5.1 Start-Up Plan 7](#_Toc263068452)

[Training 7](#_Toc263068453)

[5.2 Work Plan 7](#_Toc263068454)

[5.2.1 Work Activities 7](#_Toc263068455)

[5.2.2 Schedule Allocation 7](#_Toc263068456)

[5.2.3 Resource Allocation 7](#_Toc263068457)

[5.2.4 Budget Allocation 7](#_Toc263068458)

[5.3 Control Plan 8](#_Toc263068459)

[5.3.1 Requirements Management 8](#_Toc263068460)

[5.3.2 Schedule Control 8](#_Toc263068461)

[5.3.3 Resource Control 8](#_Toc263068462)

[5.3.4 Budget Control 8](#_Toc263068463)

[5.3.5 Reporting & Communication Plan 8](#_Toc263068464)

[5.3.6 Measurement Plan 8](#_Toc263068465)

[5.4 Risk Management Plan 9](#_Toc263068466)

[Risk Management Note 9](#_Toc263068467)

[5.5 Closeout Plan 10](#_Toc263068468)

[6. Technical Process Plans 10](#_Toc263068469)

[6.1 Process Model 10](#_Toc263068470)

[6.2 Methods, Tools, & Techniques 10](#_Toc263068471)

[Programming Environment 10](#_Toc263068472)

[Database Environment 10](#_Toc263068473)

[Version Control 10](#_Toc263068474)

[Documentation 10](#_Toc263068475)

[Testing 10](#_Toc263068476)

[6.3 Infrastructure Plan 10](#_Toc263068477)

[6.4 Product Acceptance Plan 10](#_Toc263068478)

[6.5 Deployment Plan 10](#_Toc263068479)

[7. Supporting Process Plans 10](#_Toc263068480)

[7.1 Configuration Management Plan 10](#_Toc263068481)

[7.2 Product Testing & Reviews Plan 11](#_Toc263068482)

[7.3 Document & Work Product Plan 11](#_Toc263068483)

[Requirements 11](#_Toc263068484)

[Product Specification 11](#_Toc263068485)

[Design Documentation 11](#_Toc263068486)

[Implementation Documentation 11](#_Toc263068487)

[Test Documentation 11](#_Toc263068488)

[7.4 Quality Assurance Plan 11](#_Toc263068489)

[7.5 Project Progress Reviews 11](#_Toc263068490)

[7.6 Issue Management 11](#_Toc263068491)

[7.7 Version Management 11](#_Toc263068492)

[7.8 Subcontract Management (Acquisition Management) Plan 12](#_Toc263068493)

[7.9 Process Improvement Plan 12](#_Toc263068494)

[Document Control 13](#_Toc263068495)

[Change History 13](#_Toc263068496)

[Document Storage 13](#_Toc263068497)

[Document Owner 13](#_Toc263068498)

[Appendices 14](#_Toc263068499)

# PREFACE

# 1. Overview of the Product

## Purpose Scope & Objective

The objective of this project is to develop a software product that simulates a 5x5x4 Tic Tac Toe game. The product will allow the client to compete against the computer on multiple levels of difficulty or another player. The result and desired effect of our work is to accomplish the task of creating a fully working application.

## 1.2 Assumptions & Constraints

* Project Deliverable deadlines must be met
* Budget must be kept to an absolute minimum
* The product must be reliable and consistent
* The product must be user friendly
* We shall assume that the user does not know how to play Tic-Tac-Toe and an accurate, informative description shall be demonstrated

## 1.3 Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Major Deliverable** | **Planned Delivery Date** | **Author** | **Delivery Mechanism** |
| Prelim. Requirements document | 9/22/2014 | Team | Blackboard |
| Prelim. Specifications document | 9/22/2014 | Team | Blackboard |
| Prelim. Software Management Plan | 9/22/2014 | Team | Blackboard |
| Initial Presentation | 9/23/2014 | Team | Oral Presentation |
|  |  |  |  |
| Requirements | 11/3/2014 |  | Blackboard |
| Product Specs | 11/3/2014 |  | Blackboard |
| SPMP | 11/3/2014 |  | Blackboard |
| Prelim. Detail Design | 11/3/2014 |  | Blackboard |
| Prelim. Test Plan | 11/3/2014 |  | Blackboard |
| Follow-up Presentation | 11/3/2014 |  | Oral Presentation |
|  |  |  |  |
| Detail Design | 12/5/2014 |  | Blackboard |
| Test Plan | 12/5/2014 |  | Blackboard |
| Source Code | 12/5/2014 |  | Blackboard |
| Testing Records | 12/5/2014 |  | Blackboard |
| Final Presentation and Demo | 12/5/2014 |  | Oral Presentation |
|  |  |  |  |
|  |  |  |  |

## 1.4 Schedule & Budget Summary

Team meetings are held once a week—Tuesday evenings. Impromptu discussions are held online using email and/or Trello.

## 1.5 Evolution of the Plan

All team members must agree to all changes to the product management plan before that change is made final. Previous versions of the document are maintained via version control. All changes shall be documented to ensure the Project Management Plan stays up to date.

# 2. References

|  |  |  |
| --- | --- | --- |
| **Resource** | **Identifier** | **Description of Use** |
| (no references made) |  |  |

# 3. Definitions & Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| (no terms to define) |  |

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| RC | Release Candidate |
| RTM | Release to Market |
| KDSI | Thousand Delivered Source Instructions |

# 4. Project Organization

## 4.1 External Interfaces

All the work will be performed by Aaron Dancer, Ernest Pallares, John Molina, Thanh Dang, and Hieu Vy.

## 4.2 Internal Structure

The development team consists of Aaron Dancer, Ernest Pallares, John Molina, Thanh Dang, and Hieu Vy.

## 4.3 Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| **Roles** | **Responsibilities** | **Person** |
| Team Leader | Develop game framework; UI/UX. | Aaron Dancer |
| SQA | Ensure product satisfies customer’s requirements. | Ernest Pallares |
| Software Secretary | Maintain document library. | John Molina |
| Programmer | Develop AI; implement design. | Thanh Dang |
| Programmer | Develop AI; implement design. | Hieu Vy |
|  |  |  |

## 4.4 Team Methods of Operation

The software development team shall meet every Tuesday to discuss progress, ideas, and changes.

# 5. Managerial Process Plans

## 5.1 Start-Up Plan

### Training

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Training** | **Method** | **Scheduled Dates** | **Cost** |
| Introduction to Trello | Face-to-face | 9/16/2014 | 0 |
| Intro. to QT and Grid Framework | Face-to-face | 10/14/2014 | 0 |
|  |  |  |  |

## 5.2 Work Plan

### 5.2.1 Work Activities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Project Member** | **Phases Involved** |
| Gather user requirements | Team | Inception |
| Analyze requirements | Team | Inception |
| Develop proof-of-concept | A. Dancer | Inception |
| Design product (specifications) | Team | Inception |
| Complete product specifications | Team | Elaboration |
| Draft technical detail design | A. Dancer, T. Dang | Elaboration |
| Draft test management plan | Team | Elaboration |
| Create proof-of-concept (alpha) | A. Dancer | Elaboration |
| Create beta version | Team | Construction |
| Complete technical detail design | Team | Construction |
| Complete test management plan | Team | Construction |
| Create RC version | Team | Transition |
| Fault-test the application | Team | Transition |
| Ensure application satisfies all requirements | Team | Transition |
| Create RTM version | Team | Transition |
|  |  |  |

### 5.2.2 Schedule Allocation

Month 1: Gather user requirements; analyze requirements; develop proof-of-concept

Month 2: Design product (specifications).

Month 3: Technical design and implementation.

Month 4: Finalize application.

### 5.2.3 Resource Allocation

N/A

### 5.2.4 Budget Allocation

N/A

## 5.3 Control Plan

### 5.3.1 Requirements Management

|  |  |  |
| --- | --- | --- |
| **Requirements Management Activities** | **Performed By Whom** | **Comments** |
| Approve major changes that affect milestones | A. Dancer |  |
| Quality assurance. | E. Pallares (and team) | Everyone tests their own code before submitting for SQA. |
| Ensure project is completed on time. | A. Dancer (and team) | Accomplished by meeting weekly with team members. |
| Major problems reported to team | Team |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### 5.3.2 Schedule Control

Team meets once-a-week. Additional meetings will be scheduled, as necessary.

### 5.3.3 Resource Control

Dancer will ensure project is on-track.

### 5.3.4 Budget Control

N/A

### 5.3.5 Reporting & Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Information** | **Frequency** | **From** | **To** | **Medium** |
| Team meetings | Every Tue | 5:45 PM | 9:45 PM | Face-to-face |
| Informal meetings | As needed | As needed | As needed |  |
|  |  |  |  |  |

### 5.3.6 Measurement Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Measure Required** | **Frequency Collected** | **Collected By Whom** | **Analyzed By Whom** | **Used By Whom** |
| KDSI | Once/week | A. Dancer | E. Pallares | Team |
|  |  |  |  |  |

## 5.4 Risk Management Plan

|  |  |  |
| --- | --- | --- |
| **Risk Management Activity** | **Performed By Whom** | **Comments** |
| Ensure product provides features not available in similar products. | Dancer |  |
| Compilation/execution faults will be reported to and addressed by responsible team member(s). | Team |  |
| In the event of short-term absence, other team members will assist, if necessary. In the event of long-term absence (e.g.: illness, class drop, transfer, recruitment, etc.), responsibilities will be reassigned. | Dancer | If Dancer has long-term absence, re-delegation will be handled by the first available, in the following order:   1. J. Molina 2. E. Pallares 3. T. Dang 4. H. Vy |
| Ensure workload is distributed based on complexity and developer availability. | Dancer |  |
| Ensure product works on the following platforms: iOS, Android, Windows, OSX | Team |  |
| Modifications to requirements will be evaluated to determine if the modifications can be addressed without adversely affecting commitments. | Team |  |
|  |  |  |
|  |  |  |

### Risk Management Note

None.

## 5.5 Closeout Plan

|  |  |  |
| --- | --- | --- |
| **Closeout Activity** | **Performed By Whom** | **Comments** |
| Submit to Instructor | A. Dancer | Submitted on behalf of team. |
| Source code will be posted publicly on GitHub.com | A. Dancer | For the benefit of open-source. |
|  |  |  |
|  |  |  |

# 6. Technical Process Plans

## 6.1 Process Model

Unified Process Model

## 6.2 Methods, Tools, & Techniques

### Programming Environment

IDE: Qt Creator

Implemented in: QML, C++, JavaScript

### Database Environment

Parse

### Version Control

Git / GitHub

### Documentation

Produced in accordance with the Unified Process

### Testing

On-going process (refer to TestPlan.docx)

## 6.3 Infrastructure Plan

The product will be developed using Qt Creator on multiple platforms.

## 6.4 Product Acceptance Plan

Acceptance of the product by our client will be achieved in accordance to the Unified Process.

## 6.5 Deployment Plan

All documentation, source code, and compiled binaries will be handed to the client. The application will be deployed on the Google Play Store and Apple App Store, and it will be available for download for desktop platforms via web site.

# 7. Supporting Process Plans

## 7.1 Configuration Management Plan

GitHub will be used throughout the project.

## 7.2 Product Testing & Reviews Plan

Testing will be performed in accordance to the Unified Process.

## 7.3 Document & Work Product Plan

### Requirements

Document will be produced using the provided template.

### Product Specification

Document will be produced using the provided template.

### Design Documentation

Document will consist of a listing of the objects per the specification document. The object member functions will be defined, technically, using pseudo code.

### Implementation Documentation

N/A

### Test Documentation

Use case scenarios, including sample input and expected output, will be outlined.

## 7.4 Quality Assurance Plan

Team members will test each other’s implementations, and E. Pallares will perform integration tests. The application will be tested for faults by all members through the life of the application.

## 7.5 Project Progress Reviews

Dancer will review the project’s progress against the deliverables’ time table.

## 7.6 Issue Management

|  |  |  |
| --- | --- | --- |
| **Issue Management Activities** | **Performed By Whom** | **Comments** |
| Log issue | Team | Posted on GitHub.com repository issue-tracking |
| Verify issue | Team | Issue will verified/reproduced by another team member |
| Fix issue | Team |  |
| Close issue | Team |  |
|  |  |  |

## 7.7 Version Management

|  |  |  |
| --- | --- | --- |
| **Change Management Activities** | **Performed By Whom** | **Comments** |
| Create a feature branch | Team | Using Git, team members will branch the master to work on features |
| Merge updates to master | J. Molina; A. Dancer | Using Git, J. Molina will merge pull-requests to the master branch. If J. Molina creates pull-request, A. Dancer will merge it. |
|  |  |  |

## 7.8 Subcontract Management (Acquisition Management) Plan

N/A

## 7.9 Process Improvement Plan

We will settle for a rough draft.

# Document Control

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description (Including Page #’s)** |
| 0.1 | 9/15/2014 | Team | Initial draft |
| 0.2 | 10/28/2014 | Team | Added sections: 4-6 |
| 1.0 | 11/3/2014 | Team | Updated sections: 5.2.1; 5.2.2; 5.3.5; 5.5  Added sections: 5.3.6; 7.3-7.9 |

## Document Storage

Documentation will be stored on the GitHub repository

## Document Owner

Documentation will be owned by the members of the team and the client.

# Appendices

|  |  |  |
| --- | --- | --- |
| **Appendix** | **Title** | **Location Or Link** |
| None |  |  |