Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/03/2014 | 1.0 | The initial version | Eren Sezener |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

Revision History 1

1 Identification 3

1.1 Document overview 3

1.2 Abbreviations 3

1.2.1 Abbreviations 3

1.3 References 3

1.3.1 Project References 3

2 Software Development Activities 3

2.1 Software development process 3

2.1.1 Overview of process phases 3

2.1.2 Technical documentation 3

2.1.3 Deliverables: 3

2.2 Software development tools 4

2.2.1 Workstation 4

2.2.2 Requirements management and documentation 4

2.2.3 Software Design 4

2.2.4 Coding and automated tests 4

2.2.5 Configuration management 4

2.3 Software development rules and standards 4

3 Responsibilities 4

3.1 Activities and responsibilities 4

4 Risk Assessment 5

4.1 Risk Analysis 5

4.2 Risk Planning 5

# Identification

## Document overview

This document contains the software development plan of the TicTacToe game.

## Abbreviations

### Abbreviations

The TicTacToe game software project: “the game” or “the software”

The TicTacToe game software project’s GitHub repository page: “the repo”

## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| - | - | - |

# Software Development Activities

This section lists and describes the software development activities of the TicTacToe game development project.

## Software development process

The software development process chosen for the project is the Waterfall programming model.

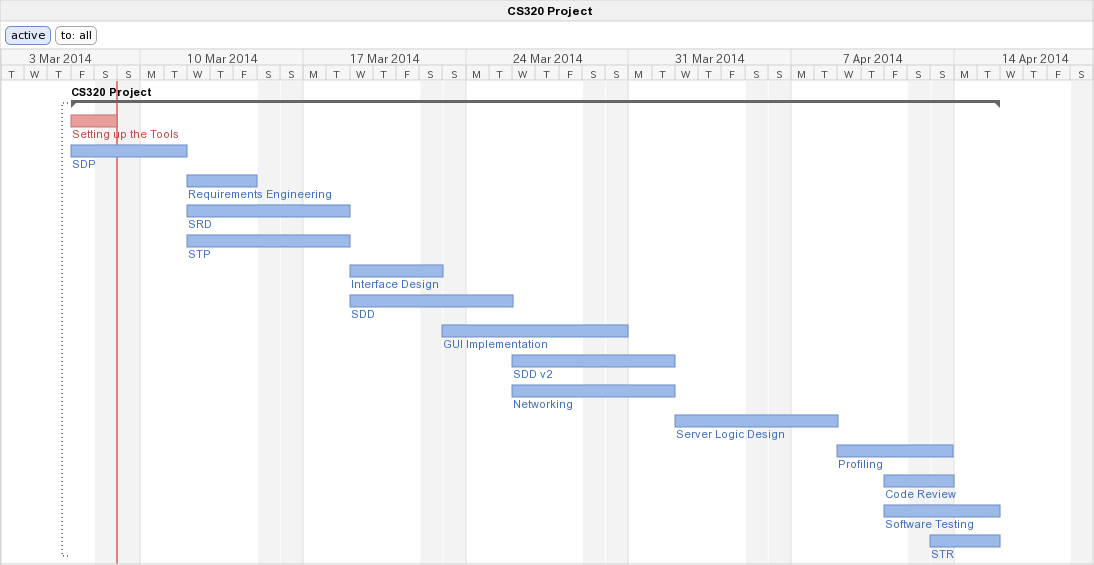
The Waterfall model was chosen for the reasons below:

* Software project is modular,
* The team size is big,
* The project must be carefully planned.

### Overview of process phases

The lifecycle of the software development project is composed of:

* **Software specification,**
* **Software detailed design,**
* **Software coding and unit tests,**
* **Software integration**
* **Software verification tests**



### Technical documentation

The following documentation is produced during the design phases:

* Software specification: SRS, IRS, STP,
* Software detailed conception: updated SRS, SDD, IDD, updated STP, STD
* Coding and unit tests: STR of unit tests
* Software tests phases: STR, VDD.

### Deliverables:

The following items are delivered at the end of the process:

* Technical documentation,
* User documentation: user guide, administration procedures and installation procedure,
* Software and its configuration files.

## Software development tools

### Workstation

Development will be done on:

1. x64 Windows 8 machines, with Eclipse as the preferred IDE,
2. Macs, with Eclipse as the preferred IDE,
3. Linux machines (Fedora & Ubuntu), with Eclipse as the preferred IDE.

### Requirements management and documentation

The following tools will be used to write and manage requirements:

* The repo (to synchronize the documents),
* Microsoft Office,
* Notepad++,
* SublimeText.

### Software Design

The following tools will be used to design and manage the software:

* The repo (to synchronize the documents),
* Omnigraffle Professional,
* TextMate,
* SublimeText,
* Eclipse,
* Emacs.

### Coding and automated tests

Coding and testing will be made with the following software:

* Eclipse with Java development tools,
* JUnit unit testing framework,
* FindBugs (Eclipse plug-in),
* Notepad++,
* SublimeText,
* Emacs.

### Configuration management

Configuration management, version control and bug management will be maintained with the following tools:

* Git version control system, via GitHub
  + The GitHub repository page of the project is located in: <http://goo.gl/Bxx3IR> (The repository contains the necessary software documentation, the game’s Eclipse project files, which include the source code of the game.)
* GitHub repo Issues page, for keeping a record of the status of bugs and issues
* GitHub repo Wiki page, for official information regarding the software.

## Software development rules and standards

Describe here the standards and rules used for software development, like modelling (UML), data modeling, coding rules, etc.

# Responsibilities

## Activities and responsibilities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Responsibility** | **Comment** |
| Project management | Eren Sezener | In charge of managing the progress and the deadlines. |
| Configuration tools management | A. Emre Ünal | In charge of the repository |
| Setting up the Development tools | A. Emre Ünal | In charge of setting up the Eclipse project copies |
| Software specifications | Erdi Gültekin | In charge of writing specifications for the game |
| Interface Design | Erdi Gültekin | In charge of designing the TicTacToe GUI |
| GUI Implementation | Erdi Gültekin | In charge of polling client messages and binding them with the GUI |
| Networking | Deniz Sökmen | In charge of designing sockets and implementation of Client-Server pattern |
| Software Testing | A. Emre Ünal | In charge of unit testing |
| Server Logic Design | Deniz Sökmen  Eren Sezener | In charge of implementing the game on server-side |
| Code Review | A. Emre Ünal | In charge of reviewing code cleanness, quality and reusability |
| Requirements Engineering | Eren Sezener | In charge of meeting the requirements. |
| Profiling | Deniz Sökmen | In charge of testing of network, memory and CPU use. |

# Risk Assessment

## Risk Analysis

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Effects** |
| The time required to develop the software is underestimated. | Moderate | High |
| Designing the game with bad user experience | Moderate | Tolerable |
| A project member is not able to contribute to the project due to potential internal or external problems. | Moderate | Tolerable |
| The size of the software is underestimated | High | Tolerable |
| The project team is not able to keep documentation up-to-date | Moderate | High |

## Risk Planning

|  |  |
| --- | --- |
| **Risk** | **Strategy** |
| The time required to develop the software is underestimated. | Increase the workload of other project members. |
| Designing the game with bad user experience | If there is enough time, redesign the game and reassign responsibilities. |
| A project member is not able to contribute to the project due to potential internal or external problems. | Increase the workload of other project members. |
| The size of the software is underestimated | Remove unnecessary features from development plan. |
| The project team is not able to keep documentation up-to-date | Reassign the document keeping responsibilities. |