Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 5/03/2014 | 1.0 | Initial Version(First Version of SDP) | Erdi Koç, Nazlı Karalar, Gamze Küçükçolak, İsmetcan Hergünşen, Mehmet Kağan Kayaalp |
| 6/03/2014 | 1.1 | Software Development Process | Erdi Koç, Nazlı Karalar, Mehmet Kağan Kayaalp |
| 7/03/2014 | 1.2 | Software Development Tools | Nazlı Karalar, Mehmet Kağan Kayaalp |
| 8/03/2014 | 1.3 | Software Development Rules and Standards | Erdi Koç |
| 9/03/2014 | 1.4 | Responsibilities | Mehmet Kağan Kayaalp İsmetcan Hergünşen |
| 10/03/2014 | 1.5 | Risk Assessments | Gamze Küçükçolak, İsmetcan Hergünşen |

**TABLE OF CONTENTS**

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 4

2.1.1 Overview of process phases *4*

2.1.2 Technical documentation 6

2.1.3 Deliverables: *6*

2.2 Software development tools 6

2.2.1 Workstation *6*

2.2.2 Requirements management and documentation *7*

2.2.3 Software Design *7*

2.2.4 Coding and automated tests *7*

2.2.5 Configuration management 7

2.3 Software development rules and standards 7

3 Responsibilities *8*

3.1 Activities and responsibilities 8

4 Risk Assessment *8*

4.1 Risk Analysis 8

4.2 Risk Planning 9

# Identification

## Document overview

This document contains the software development plan of the software Crazy Copter Battle Project Game. It describes the CCB project, and the approach taken by the Group Irregular to deliver CCB v1.0. This project is prepared as JDA.

The document begins with an introduction of the project, abbreviations, and project references.

In *“Software development activities”,* the report describes software development process, its tools, rules and standards.

*“Responsibilities*” section includes activities and responsibilities of each member.

Final part, *“Risk Assessment”,* indicates risk analysis table and its planning.

## Abbreviations

### Abbreviations

|  |  |
| --- | --- |
| **Term** | **Description** |
| CCB Project | Crazy Copter Battle Project Game |
| DOC #v.1.0.x | Document version 1.0.x |
| JDA | Java Desktop Application |
| JDK | Java Development Kit |
| UI | User Interface |
| GUI | Graphical User Interface |
| SDD | Software Design Document |
| SDP | Software Development Plan |
| SRS | Software Requirements Document |
| UML | Unified Modelling Language |
| STP | Software Test Plan |
| STR | Software Test Report |

## References

### Project References

# Software Development Activities

The section lists and describes the software development activities of CCB software development project.

|  |  |  |
| --- | --- | --- |
| **The Schedule** | | |
| Week 1 | * Software Development Plan (SDP) | Reasonably detailed description of all the activities needed to undertake. |
| Week 2 | * Software Requirements Document (SRS) * Software Test Plan (STP) | Customer requirements for the system will be identified and defined. It will not include detailed information about the system but it will be overview of the system requirements.  In STP part, software and other test preparations for each requirement will be explained. |
| Week 3 | * Software Design Document (v1) (SDD) | Software components, interfaces and data necessary for the implementation phase will be described. ArgoUML diagrams will be added. |
| Week 4 | * Software Design Document (v2) (SDD) | SDD v1 will be improved. |
| Week 5 | * (Implementation) | Implementations which describe in part SSD will be coded. |
| Week 6 | * Software Test Report (STR) * (Demo) | The code generated in implementation part will be tested and after unit, process and system testing, demo will be prepared. |

## Software development process

The waterfall programming model is chosen because of the reasons below:

* The visibility and manageability is easier for the CS320/Group Irregular.
* Plan-driven process provides the group to detect the errors in the beginning of the project and reduces the risks.
* Phases are implemented and completed one at a time.
* It assists the group members to transfer their knowledge efficiently when team members are separated in different locations.

### Overview of process phases

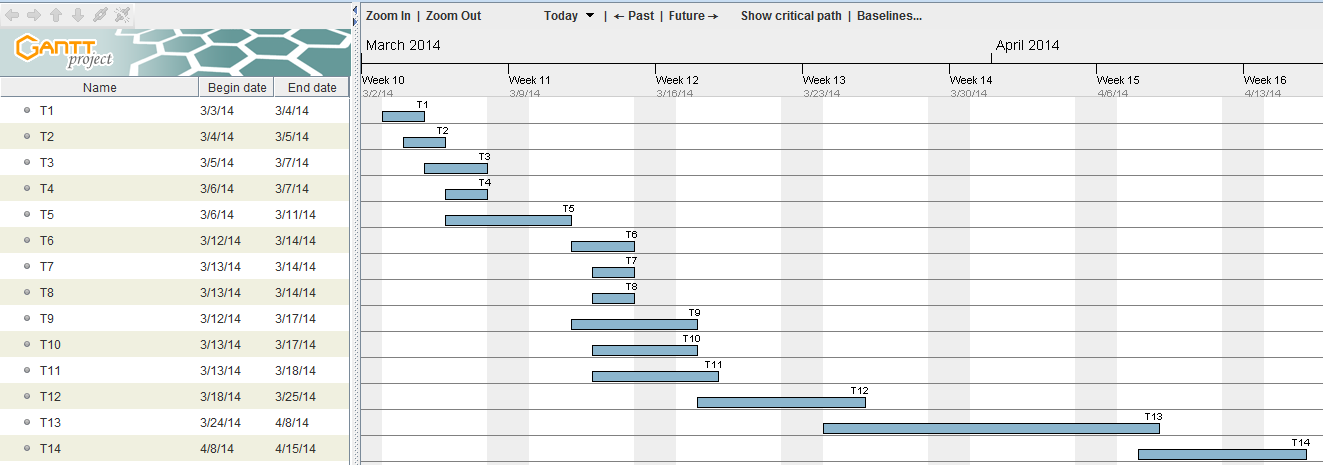
The lifecycle of the software development project is composed of:

* Software specification,
* Software detailed design,
* Software coding and unit tests,
* Software validation and verifications tests
* Software evolution

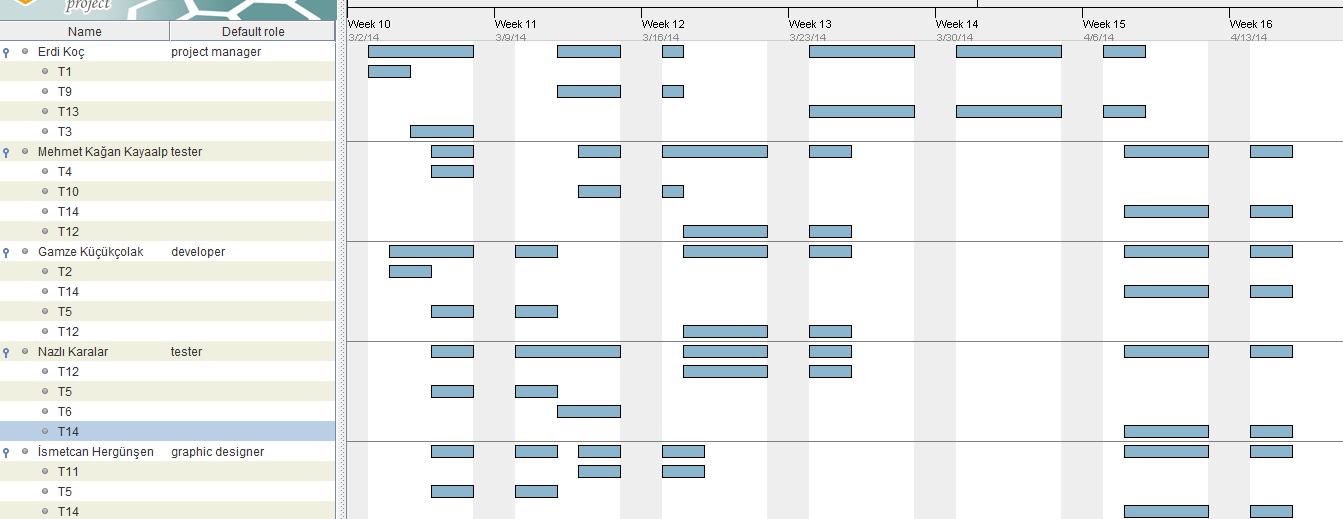
In software specification process, what the system does will be defined. In detailed design phase, the organization and the functions of the system will be explained and after that, it will be implemented in software coding and unit tests part. The software will be tested by two different tests whose names are process and system testing in order to meet the requirements of the customer. Finally, the CCB project will be checked whether the software is flexible and can be easily adapted.

This project is assumed to be completed by the 15th April but it can be finished before the deadline, this is depending on the performance of the team members.

|  |  |
| --- | --- |
| T1 | Preparation of Activities, Phases and Scheduling |
| T2 | Documentation |
| T3 | Tools and Development Environment |
| T4 | Roles and Responsibilities |
| T5 | Risk Management |
| T6 | External Interface Requirements |
| T7 | System Features |
| T8 | Non-functional Requirements |
| T9 | GUI Interface Design |
| T10 | Component Design |
| T11 | Architectural Design |
| T12 | Software Test Plan |
| T13 | Coding and Implementation |
| T14 | Software Testing |



**Staff Allocation Chart**

****

### Technical documentation

The following documentation is produced during the design phases:

* Software specification: SRS, STP,
* Software detailed conception: updated SRS, SDD, updated STP
* Coding and unit tests: STR of unit tests
* Software tests phases: STR, Demo.

### Deliverables:

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

* Technical documentation (SDP,SRS, STP, SDD, STR)
* User documentation: user guide, administration procedures and installation procedure,
* Software and its configuration files.
* Working System (a short demo)
* GitHub records (Project Page, Bugs and Issues ,Time Tracking, Contributions, Commit History)

## Software development tools

### Workstation

In our operating system;

Workstation:

|  |
| --- |
| 1)Windows 7 Ultimate (Erdi Koç) |
| 2)Windows 8/8.1 (Mehmet Kağan Kayaalp, Gamze Küçükçolak, Nazlı Karalar) |
| 3)Mac OS X 10.9.2 (İsmetcan Hergünşen) |

### Requirements management and documentation

Microsoft Office Word

### Software Design

* Argo UML open source tool: It is a tool to analyze and design object-oriented software.

### Coding and automated tests

* Eclipse Standard Kepler 4.3.2 Version
* JDK SE 7

### Configuration management

Configuration management tools:

* GitHub

GitHub is a web-based revision control hosting service for code sharing and software development projects. It offers both private repositories with a charge and free public repositories for open-source projects.

GitHub provides the group members to have a repository where all the files for a particular project are stored and aids them to communicate and share the part of the project that they are responsible for.

GitHub also provides access control for configuration management. So, every change can be managed easily. It keeps track of which version belongs to which configuration.

The most common Git commands are commit, push, pull, clone and diff.

* Commit is used for adding the change to local repository and push sends this change to the remote repository from the local repository.
* Pull is used for update from the remote repository, so basically it synchronizes the Git.
* Clone means checking out a repository, so it creates a working copy of a Git repository.
* Diff is used for view all the merge conflicts.

Bug management tools:

* GitHub Bug Tracker

GitHub has an issue management system for bug tracking. It is flexible and powerful tool for managing large and small software projects.

## Software development rules and standards

**Coding Standards**

The group will follow the Java standards and rules in coding process. The pdf website of the java rules and standards:

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# Responsibilities

* In addition to main roles of each member, every group member also works on multiple tasks throughout the development process.
* Project design and specification will be discussed with the whole team because of the small size of the team.

## Activities and responsibilities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Responsibility** | **Comment** |
| **Project Management** | Erdi Koç | Is responsible for planning, organizing and controlling the project. |
| **Configuration Tools Management** | Mehmet Kağan Kayaalp, Nazlı Karalar | Are responsible for tracking and controlling the changes in the software and implementing the techniques and tools that are required to manage. |
| **Setting up the Development Tools** | Nazlı Karalar, Gamze Küçükçolak | Are responsible for improving the development tools. |
| **Software Specifications** | İsmetcan Hergünşen, Gamze Küçükçolak | Are responsible for completing description of behavior of system to be developed. |
| **GUI Interface Design** | Erdi Koç | Is responsible for interfaces between all components. |
| **Component Design** | Mehmet Kağan Kayaalp | Is responsible for design of each component in software. |
| **Architectural Design** | İsmetcan Hergünşen | Is responsible for structure of the software. |

# Risk Assessment

## Risk Analysis

|  |  |  |
| --- | --- | --- |
| **Risks** | **Probability** | **Effects** |
| Unclear understanding of requirements | High | Tolerable |
| Leaving of a member from project | Low | Serious (Tolerable) |
| Time pressure | High | Serious |
| Technical issues (meet performance requirements and new or untested technologies) | - | - |
| Resource (facilities, attraction, skills limitations) | Low | Tolerable |
| Organizational issues | High | Serious (Tolerable) |
| Financial issues | - | - |

## Risk Planning

* ***Unclear Understanding of Requirements:*** The Irregular Group manages this problem by holding weekly meetings with the instructor.
* ***Leaving of a member from the project:*** Some of the group members may withdraw the course. The other group members will allocate their time to the project more. The remaining parts will be divided into other group members.
* ***Time pressure:*** Due dates for the project’s documents are strict; so, each member gives maximum performance to the project.
* ***Technical issues:*** There are no technical issues that the group will face with. New and untested technologies will not be used in the process.
* ***Resource****:* Facilities of technologic devices and tools
* ***Organizational issues:*** The group members might not gather at the same time. Thus, in order to handle this problem, each member uses GitHub for project documents.
* ***Financial issues:*** There are no financial problems in the CCB project because tools that we use are open source and budget is not needed for coding and testing the project.