Software Project Management Plan (SPMP) for Computation Security Project

*Baseline version 0.1*

*Issued on : July, 12 2015*

Issued by : Clientle, Inc.

Issued for : Client

**Signature**

The following signature indicates approval of the enclosed Software Project Management Plan.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AnotherBlabla Client Executive Committee representative

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | July 12, 2015 | Devi | initial version |
|  |  |  |  |

**Table of Contents**

List of figures 5

List of tables 6

Chapter 1 Introduction 7

1.1 Project Overview

1.1.1. Purpose, Scope, and Objectives

1.1.2. Assumptions and Constraints

1.2. Project Deliverables

Chapter 2 Project Organization 8

2.1. Software Process Model

2.2. Internal Structure

2.3. Roles and Responsibilities

Chapter 3 Project Management Plan 10

3.1. Work Activities

3.2. Activity Network

**List of Figures**

Activity Network 11

**List of Tables**

Responsibility Assignment Matrix 8

Work Activities Table 10

**Chapter 1**

**INRODUCTION**

* 1. **Project Overview**
     1. **Purpose, Scope, and Objectives**

The project is about creating an encrypted, socket-based chat messenger, as requested by Clientle. Inc. This paper will contain the analysis of requirements for the project as specified by the client.

The scope of this project is the software itself and its functionality – other problems such as hardware failure or network unavailability is considered out of scope.

The objectives of the project are mentioned as follows :

* Complete the project by the due date
* Complete the project within the budget
  + 1. **Assumptions and Constraints**

Here is the list of all assumptions that are made :

* This project is a component of a larger project

Here is the list of all constraints that are made :

* Budget
  + $100
* Time
  + 31 days
* Staff
  + A representative from Clientle will be required to assist in the requirements making. This representative will have the full authority in creating the final requirements, which will be evaluated and agreed by Clientle .Inc.
  1. **Project Deliverables**

Here is the list of all items that will be available by the completion of the project.

* Software program, along with its environment and supporting libraries.
* Software documentation
  + Installation documentation
  + End-user documentation
* Project documentation
  + Software Project Management Plan (SPMP)
  + Software Requirement Specification (SRS)

**Chapter 2**

**PROJECT ORGANIZATION**

1. **Software Process Model**

The software process model used was agile programming. It was used considering that this was only a small project, ran by only a few people.

1. **Internal Structure**

*Use this part to mention who are involved in your team as project participants. Following is an example of a chart that gives that information.*

1. **Roles and Responsibilities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | WBS | Deliverables | Client | Programmer 1 | Programmer 2 |
| 1 | 1 | Initiate project | A |  |  |
| 2 | 1.1 | Select software development process | L | C | C |
| 3 | 1.2 | Identify WBS |  | L | S |
| 4 | 1.2.1 | Identify roles for each member |  | L | S |
| 5 | 1.2.2 | Identify each member's responsibilities |  | L | S |
| 6 | 2.1 | Identify the software's general requirements |  | S | L |
| 7 | 2.2 | Develop functional requirements | R | S | L |
| 8 | 2.3 | Develop non-functional requirements |  | S | L |
| 9 | 2.4 | Identify the software's design |  | L | S |
| 10 | 2.4.1 | Identify use cases |  | L | S |
| 11 | 2.4.1.1 | Create use case scenarios |  | S | L |
| 12 | 2.4.1.2 | Design the sequence diagrams |  | S | L |
| 13 | 2.4.2 | Identify classes/objects |  | L | S |
| 14 | 2.4.2.1 | Create class diagram |  | S | L |
| 15 | 2.5 | Identify the necessary tools |  | L | S |
| 16 | 2.5 | Prepare the necessary tools |  | L | S |
| 17 | 3.1 | Develop classes |  | L | S |
| 18 | 3.2 | Integrate the various classes |  | L | S |
| 19 | 3.3 | Test software |  | L | S |
| 20 | 3.4 | Maintenance | R | L | S |
|  | Key |  |  |  |  |
|  | A | Approval | 1 | 0 | 0 |
|  | L | Lead | 1 | 12 | 6 |
|  | S | Secondary | 0 | 6 | 12 |
|  | C | Contributor | 0 | 1 | 1 |
|  | R | Reviewer | 2 | 0 | 0 |

1. **Tools and Techniques**

Tools :

* Java Development Kit
* Java IDE (optional)

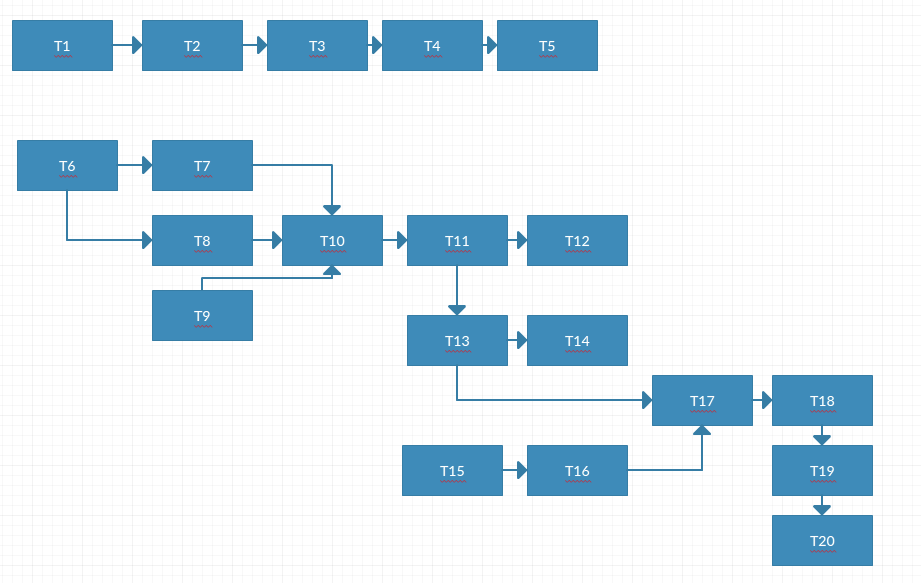
**Chapter 3**

**PROJECT MANAGEMENT PLAN**

1. **Work Activities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | WBS | Deliverables | Duration | Predecessor | Successor |
| 1 | 1 | Initiate project | 1 day |  | 2 |
| 2 | 1.1 | Select software development process | 1 day | 1 | 3 |
| 3 | 1.2 | Identify WBS | 12 hours | 2 | 4 |
| 4 | 1.2.1 | Identify roles for each member | 12 hours | 3 | 5 |
| 5 | 1.2.2 | Identify each member's responsibilities | 12 hours | 4 |  |
| 6 | 2.1 | Identify the software's general requirements | 12 hours |  |  |
| 7 | 2.2 | Develop functional requirements | 12 hours | 6 |  |
| 8 | 2.3 | Develop non-functional requirements | 12 hours | 6 |  |
| 9 | 2.4 | Identify the software's design | 1 day |  |  |
| 10 | 2.4.1 | Identify use cases | 12 hours | 7,8,9 | 11 |
| 11 | 2.4.1.1 | Create use case scenarios | 12 hours | 10 | 12 |
| 12 | 2.4.1.2 | Design the sequence diagrams | 12 hours | 11 |  |
| 13 | 2.4.2 | Identify classes/objects | 1 day |  | 14 |
| 14 | 2.4.2.1 | Create class diagram | 12 hours | 13 |  |
| 15 | 2.5 | Identify the necessary tools | 1 day |  | 16 |
| 16 | 2.5 | Prepare the necessary tools | 1 day | 15 |  |
| 17 | 3.1 | Develop classes | 7 days | 13,16 | 18 |
| 18 | 3.2 | Integrate the various classes | 2 days | 17 | 19 |
| 19 | 3.3 | Test software | 4 days | 18 | 20 |
| 20 | 3.4 | Maintenance | 7 days | 19 |  |

1. **Activity Network**

**