Software Project Plan

**Messenger**

**Client – Server Application**

**Simple Message Protocol (SMP)**

**Version 1**



# Table of Contents

[Table of Contents 2](#_Toc131494068)

[Revision History 4](#_Toc131494069)

[1 Project Overview 5](#_Toc131494070)

[2 Objectives 5](#_Toc131494071)

[2.1 Objective 1 SMP Server 5](#_Toc131494072)

[2.1.1 Description 5](#_Toc131494073)

[2.2 Objective 2 SMP Client Producer 5](#_Toc131494074)

[2.2.1 Description 5](#_Toc131494075)

[2.3 Objective 3 SMP Client Consumer 5](#_Toc131494076)

[2.3.1 Description 5](#_Toc131494077)

[3 Scope 5](#_Toc131494078)

[4 Deliverables 6](#_Toc131494079)

[4.1 Phase 1 6](#_Toc131494080)

[4.1.1 Deliverable 1: Software Project Plan 6](#_Toc131494081)

[4.1.2 Deliverable 2: Software Application User Interface Specification 6](#_Toc131494082)

[4.2 Phase 2 6](#_Toc131494083)

[4.2.1 Deliverable 1: Software Requirements Specification 6](#_Toc131494084)

[4.2.2 Deliverable 2: Software Technical Specification 7](#_Toc131494085)

[4.2.3 Deliverable 3: Software Design Specification 7](#_Toc131494086)

[4.3 Phase 3 7](#_Toc131494087)

[4.3.1 Deliverable 1: SMP Server Prototype 7](#_Toc131494088)

[4.3.2 Deliverable 2: SMP Client Producer Prototype 7](#_Toc131494089)

[4.3.3 Deliverable 3: SMP Client Consumer Prototype 7](#_Toc131494090)

[4.4 Phase 4 8](#_Toc131494091)

[4.4.1 Deliverable 1: Software System Test Plan 8](#_Toc131494092)

[5 Assumptions and Risks 8](#_Toc131494093)

[5.1 Assumption and Risks: Cyber-Security Concerns 8](#_Toc131494094)

[5.1.1 Description 8](#_Toc131494095)

[5.2 Assumption and Risks: Vulnerability Assessment 8](#_Toc131494096)

[5.2.1 Description 8](#_Toc131494097)

[5.3 Assumption and Risks: Penetration Testing Assessment 9](#_Toc131494098)

[5.3.1 Description 9](#_Toc131494099)

[6 Stakeholders 9](#_Toc131494100)

[6.1 Stakeholder 1 Reality Software 9](#_Toc131494101)

[6.1.1 Description 9](#_Toc131494102)

[6.2 Stakeholder 2 Never Crash Software Services 9](#_Toc131494103)

[6.2.1 Description 9](#_Toc131494104)

[7 References 10](#_Toc131494105)

[7.1 Reference: Software Requirements Document 10](#_Toc131494106)

[7.1.1 Description 10](#_Toc131494107)

[7.2 Reference: Software Technical Specification 10](#_Toc131494108)

[7.2.1 Description 10](#_Toc131494109)

[7.3 Reference: Software Design Specification 10](#_Toc131494110)

[7.3.1 Description 10](#_Toc131494111)

[7.4 Reference: SPIF (Software Project Information File) 10](#_Toc131494112)

[7.4.1 Description 10](#_Toc131494113)

[8 Team Member Bios & Team Member Rolls 11](#_Toc131494114)

[9 General Schedule 12](#_Toc131494115)

[10 Appendix 13](#_Toc131494116)

[10.1 Acronyms 13](#_Toc131494117)

[10.2 Engineering Terms 13](#_Toc131494118)

[10.2.1 Software Development 13](#_Toc131494119)

[11 Notes 14](#_Toc131494120)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | Authors | Comments |
| 1 |  |  | 1. Initial release. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

**The Software Development Lifecycle Model**

The Software Development Lifecycle Model begins with a Software Project Proposal. During this phase, the concepts of the software system are first discussed and formalized into a definable system. From the Software Project Proposal, a Software Project Plan document is created. Based on the project proposal shown directly below, this document defines the Software Project Plan.

**Project Proposal**

Miss Pudding, the VP of Engineering for Reality Software, has contracted with Never Crash Software Services to develop a client-server application that communicates using a simple communications protocol. The communications protocol this project develops and utilizes is termed the Simple Message Protocol (SMP). Mr. Pumphrey, a senior project manager with Never Crash Software Services, has been tasked to oversee this project. He will direct his team to develop the application according to the defined project requirements. The project requirements, which have yet to be formally defined, are defined based on feedback from the project’s stakeholders.

# Objectives

## Objective 1 SMP Server

### Description

Implement an SMP server that stores SMP client messages.

## Objective 2 SMP Client Producer

### Description

Implement a SMP client that sends (writes/produces) SMP messages to a SMP server.

## Objective 3 SMP Client Consumer

### Description

Implement as SMP client that retrieves (reads/consumes) SMP messages from an SMP server.

# Scope

This document covers the following categories items:

* Deliverables
* Assumptions and Risks
* Stakeholders
* Project Schedule

All licenses for this software application are owned by the principle contracting agency, Reality Software.

NOTE: This document is NOT intended to serve as a non-disclosure agreement, letter of intent, request for proposal or quotation, joint marketing agreement or purchase agreement. These or other documents that may be required to define any business relationship or agreements between any contracting agency or vendor and Never Crash Software Services shall be executed separately as required.

# Deliverables

## Phase 1

### Deliverable 1: Software Project Plan

The Software Development Lifecycle Model begins with a Software Project Proposal. During this phase, the concepts of the software system are first discussed and formalized into a definable system. From the Software Project Proposal, a Software Project Plan document is created. The primary purpose of The Software Project Plan document is to define the software application development phases. This Software Project Plan document defines the SMP software application development phases.

### Deliverable 2: Software Application User Interface Specification

The Software Application User Interface Specification document defines a prototype of the software application’s user interface. The SMP Software Application User Interface Specification document defines a prototype of the software application’s user interface.

## Phase 2

### Deliverable 1: Software Requirements Specification

Perhaps the most important piece of the software system initiation, definition and planning stage is the definition of the software requirements. The role of the software requirements specification document is to define in detail the functionality of the software system. The Software Requirements Specification document defines the software requirements for each software system component. The software requirements specification document typically includes software configurations, software installation procedures, database installation procedures, and training standards.

Because the software requirements specification document defines the system functionality in great detail, the software requirements definition process is usually a very formal and time-consuming process. Using an agile software development methodology, requirements gathering is an iterative process. During each iteration, the team develops a better understanding of the system and the system’s operating environment.

The SMP Software Requirements Specification document contains client and server application development requirements and other requirements specific to customers contracting with Never Crash Software Services and vendors developing a software or hardware product or software subsystem for Never Crash Software Services.

### Deliverable 2: Software Technical Specification

The SMP Software Technical Specification document contains client and server application technical specifications.

### Deliverable 3: Software Design Specification

A software design document (also known as a software design specification or technical specification documents) is a written report of a software product’s design, describing its overall architecture. Such design documents are usually written by software designers or project managers and are given to the software development team to give them an overview of what needs to be built and how. A software design document helps to ensure the design specs of the software are understood and it’s clear to all. It specifies what is possible with the product and how it can be accomplished. The Software Design Specification defines each software component, and the software component interfaces. It may include any of the following:

• Software Component APIs.

• Software Component Diagrams

• Software Component Interface Diagrams.

• Software/Hardware Interfaces.

• Flowcharts illustrating important use cases.

• Protocols, protocol data formats, and protocol messages.

The SMP Software Design Specification document contains client and server application design specifications.

## Phase 3

### Deliverable 1: SMP Server Prototype

The Requirements & Technical Specification document contains the requirements and technical specifications for the SMP Server. The SMP Server is designed to store SMP client messages.

### Deliverable 2: SMP Client Producer Prototype

The Requirements & Technical Specification document contains the requirements and technical specifications for the SMP Client Producer. The SMP Client Producer is designed to send (write/produce) SMP messages to a SMP server.

### Deliverable 3: SMP Client Consumer Prototype

The Requirements & Technical Specification document contains the requirements and technical specifications for the SMP Client Consumer. The SMP Client Consumer is designed to retrieve (read/consume) SMP messages from an SMP server.

## Phase 4

### Deliverable 1: Software System Test Plan

The Software System Test Plan document defines how each of the software components is tested and validated. It is created by the software test engineering team. The Software System Test Plan must be detailed enough to verify compliance with the Software Design Specification, Software Requirements Specification, and the Software Technical Specification. The Software System Test Plan should note if a software project test automation suite is planned, and how many of the test cases can be automated.

The SMP Software System Test Plan document contains a suite of test cases that covers the entire software system functionality as defined by the requirements document. To rapidly test the functionality of the software system, the test cases should be automated and executed using a test automation suite.

# Assumptions and Risks

## Assumption and Risks: Cyber-Security Concerns

### Description

Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. These cyberattacks are usually aimed at accessing, changing, or destroying sensitive information. A successful cybersecurity approach has multiple layers of protection spread across an organization’s computers, networks, programs, and data.

## Assumption and Risks: Vulnerability Assessment

### Description

A vulnerability assessment is a systematic review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed.

Examples of threats that can be prevented by vulnerability assessment include:

* SQL injection, XSS and other code injection attacks.
* Escalation of privileges due to faulty authentication mechanisms.
* Insecure defaults – software that ships with insecure settings, such as a guessable admin password.

<https://www.imperva.com/learn/application-security/vulnerability-assessment/>

## Assumption and Risks: Penetration Testing Assessment

### Description

Software penetration testing is a method used to identify vulnerabilities in your software. This method is commonly used to test software security and is a very reliable way to check for security risks in the software. It is a simulated attack into the software that is conducted by professional security experts.

Software penetration testing is a type of test that is conducted to check the security of the application or software from malicious attacks. This testing is performed by a professional team consisting of a security tester and a report writer.

Software penetration testers are given an objective to test the security of a software. In some cases, they are given a checklist of what they should check during the test. The penetration test report writer then analyzes the results of the penetration test and creates a report. The penetration test report is then sent to the client, and the client can then decide how and when to fix the vulnerabilities.

<https://www.getastra.com/blog/security-audit/software-penetration-testing/>

# Stakeholders

## Stakeholder 1 Reality Software

### Description

Contact: Miss Pudding – VP of Engineering for Reality Software

Miss Pudding, the VP of Engineering for Reality Software, has contracted with Never Crash Software Services to develop a client-server application that communicates using a simple communications protocol.

## Stakeholder 2 Never Crash Software Services

### Description

Contact: Mr. Pumphrey -- Senior Project Manager for Never Crash Software Services

Mr. Pumphrey, a senior project manager with Never Crash Software Services, has been tasked to oversee this project. He will direct his team to develop the application according to the defined project requirements.

# References

## Reference: Software Requirements Document

### Description

Perhaps the most important piece of the software system initiation, definition and planning stage is the definition of the software requirements. The role of the software requirements document is to define in detail the functionality of the software system.

Because the software requirements document defines the system functionality in great detail, the software requirements definition process is usually a very formal and time-consuming process. Using an agile software development methodology, requirements gathering is an iterative process. During each iteration, the team develops a better understanding the system and the system’s operating environment.

## Reference: Software Technical Specification

### Description

The software technical specification document details a software system’s technical requirements.

## Reference: Software Design Specification

### Description

The software design document is a high-level view of the software system.

## Reference: SPIF (Software Project Information File)

### Description

The Software Project Information File (SPIF) is used to determine the location of the Software Project Repository and other important artifacts. Each software project must contain a Software Project Information File (SPIF). The SPIF must be located in the root directory of the Software Project Repository.

The contents of the SPIF should include:

• The location of the Software Project Repository.

• The name and location of the Software Design.

• The name and location of the Software System Test Plan.

• Instructions on how to build the software, and the names and locations of all tools needed to build the software, or the name of the file that has this information. If a reference to a file is made that has this information, then the Software Project Repository directory should contain this file.

# Team Member Bios & Team Member Rolls

**Mary Lamb: Technical Lead**

* 5 years of TCP/IP software development experience.
* Lead for the TCP/IP subsystem for the (Remote Propagation Gate) RPG transponder.

**Tom Thumb: Developer**

* 7 years of UI design experience.
* Designed the UI for the (Remote Propagation Gate) RPG server.

**Mickey Mouse: Developer**

* New hire. Just graduated with a CS degree.
* Extensive training needed in TCP/IP. Will work under Mary Lamb to help with the SMP TCP/IP subsystem.

# General Schedule

See SMP\_Detailed\_Schedule.xls for a detailed schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Hours | Notes | Comments |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Appendix

## Acronyms

SMP – Simple Message Protocol

## Engineering Terms

### Software Development

Client – A software application that sends and receives messages to and from a server application.

Server – A software application that processes messages from client applications.

Producer – A software application that produces messages.

Consumer – A software application that consumes messages.

# Notes