Software Project Proposal

**Messenger**

**Client – Server Application**

**Simple Message Protocol (SMP)**

**Version 1**



# Table of Contents

[Table of Contents 2](#_Toc184375897)

[Revision History 3](#_Toc184375898)

[1 Introduction 4](#_Toc184375899)

[2 Notes 4](#_Toc184375900)

# 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.

**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.

The SMP protocol allows client applications to send messages to a SMP server and retrieve messages from a SMP server. The server application stores the messages in a datastore, such as an ASCII text file. The datastore is designed to implement the behavior of a message queue. A queue is a first-in, first-out data structure. Messages are sent to the SMP server and inserted into the queue in the order they’re received. The messages are retrieved from the queue in a first-in, first-out fashion. In other words, the first message inserted in the queue is the first message retrieved from the queue. The SMP Message Producer client program is designed to send messages to a server using TCP/IP and the Sockets API (Application Programming Interface). The SMP Message Consumer client program is designed to retrieve messages from a server using TCP/IP and the Sockets API (Application Programming Interface).

# Notes