

Project Cloudbase

Software Requirements Specification

Version 2.0

24 February 2015

Harryson Tun, Joshua Greer, Michael Mastantuono

Hood College, Department of Computer Science, Frederick MD, 21701

Revision History

Date	Description	Author	Comments
2/10/2015	Version 1.0	Team Cloudbase	Original SRS Document
2/24/2015	Version 2.0	Team Cloudbase	Updated SRS Document

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

Table of Contents

Revision History	2
Document Approval	2
1. Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5-6
1.4 References	6
1.5 Overview	6-7
2. General Description	7
2.1 Product Perspective	7
2.2 Product Functions	7
2.3 User Characteristics	7
2.4 General Constraints	8
2.5 Assumptions and Dependencies	8
3. Specific Requirements	8
3.1 External Interface Requirements	8
3.1.1 User Interfaces	8
3.1.2 Hardware Interfaces	8
3.1.3 Software Interfaces	8
3.1.4 Communications Interfaces	8
3.2 Functional Requirements	9
3.2.1 Time Format	9
3.2.2 Authenticated Login	9
3.2.3 Admin User	9
3.2.4 Flight Sheet	9
3.2.5 Add / Remove Pilots	9
3.2.6 Update Pilot Names	9
3.2.7 Add / Remove Aircraft	10
3.2.8 Update Aircraft Name	10
3.2.9 Create / Edit Payment	10
3.2.10 Reports	10
3.3 Use Cases	11
3.3.1 Use Case #1	11
3.4 Classes / Objects	11
3.4.1 User	11
3.4.2 DAC (Database Access Component)	11
3.4.3 Flight Record	11-12
3.5 Non-Functional Requirements	12
3.5.1 Performance	12
3.5.2 Reliability	12
3.5.3 Availability	12
3.5.4 Security	12
3.5.5 Maintainability	12
3.5.6 Portability	13
3.6 Inverse Requirements	13
3.7 Design Constraints	13
3.8 Logical Database Requirements	13
3.9 Other Requirements	13

4. Analysis Models	13
4.1 Sequence Diagrams	13-14
4.2 State-Transition Diagrams (STD)	14-15
4.3 Data Flow Diagrams (DFD)	15
5. Change Management Process	16

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of Cloudbase, a custom web based application for the Mid-Atlantic Soaring Association (M-ASA). The document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the restrictions under which it must operate and how the system will react to user input. The document is intended for the developers of the system.

1.2 Scope

The software product will be called Cloudbase. Cloudbase will be used by one or more users to record flight information and generate reports necessary for the daily operations of M-ASA. Cloudbase will be used to:

1. Track the number of hours an aircraft was in flight
2. Track the number and associated times of flights a pilot was flying
3. Track the names, contact information and amount paid by each client who flew
4. Produce reports for each aircraft and it's total flight time
5. Produce reports for each pilot and their total flight time
6. Produce reports for each client and their associated information

1.3 Definitions, Acronyms, and Abbreviations

AOD	Assistant Operations Director
Cloudbase	The software product
CSS	Cascading Style Sheets
DBMS	Database Management System. A software suite that allows a user to create, modify and access databases.
GPL	GNU General Public License version 3 is a copyright license. Among its restrictions, the license requires that all licensed products convey the source code in a public manner. Anyone is free to modify the source code provided all modifications are also licensed under the GPLv3 and therefore made available to the public.
GUI	Graphical User Interface. The part of an application that allows a user to interact with an application
HTML	HyperText Markup Language
LAMP	Linux Apache MySQL PHP/Perl/Python. This is a software stack which is

	commonly used to host web based applications.
M-ASA	Mid-Atlantic Soaring Association
MySQL	A relational database software product. SQL is an acronym for Structured Query Language.
OD	Operations Director
PHP	PHP: Hypertext Preprocessor is a server-side scripting language
Soaring	The act in which an engineless aircraft is towed into the air and let to glide back down to the ground.

1.4 References

Source	Website	Description
Mid-Atlantic Soaring Association (M-ASA)	www.midatlanticsoaring.org	M-ASA provides information about its soaring organization
Open Web Application Security Project (OWASP)	www.owasp.org	OWASP provides information and resources about developing a secure web application

1.5 Overview

The remainder of this document is organized as follows:

Section 2: General Description

The section provides a general description of the system, its purpose, and its functionality.

Section 3: Detailed Requirements

This section outlines all of the specific functional and nonfunctional requirements that describe the product.

Section 4: Analysis Models

This section clarifies the manner in which the product will function and be used. It will contain models of state transition, data flow and use cases.

Section 5: Change Management Process

This section will outline the structure of the team and the procedures governing changes that may be made to this document in the future.

2. General Description

2.1 Product Perspective

The product currently in use by M-ASA is called Foxpro. Foxpro is a very dated general purpose DBMS and a predecessor to Microsoft Access. Currently M-ASA has a single workstation dedicated to hosting Foxpro. Only one individual may access the application at a time. The Foxpro database suffers from a GUI that is dated and difficult to learn. It is necessary for M-ASA employees to backup the Foxpro files in case of data loss.

In contrast Cloudbase will be a web application. This means that it will run in a hosted environment and allow multiple users to access the database simultaneously over a network. Cloudbase will not suffer the same GUI constraints as Foxpro resulting in a more appealing GUI that is easier to use. Because Cloudbase will be a web app it will be possible to host it anywhere that supports PHP 5.6.5 and MySQL 5.6. Should the user decide to host Cloudbase with a hosting service such as HostGator the burden of data security will shift away from the user and onto the datacenter. However, another consequence is the increase in the need of a highly secure application that will be exposed to the internet at large.

2.2 Product Functions

Cloudbase will provide two main functions namely recording and reporting. Cloudbase will record flight information for every single M-ASA flight. Flight records include data concerning engine and engineless planes such as pilot names, passenger names, take-off times, landing times, instructors, and applicable fees.

Cloudbase will also contain a mechanism for producing reports. There are three categories of reports: Flight times, pilot times, and customer fees. The flight times report conveys the cumulative flights for each aircraft; the pilot times report conveys cumulative pilot times; and the customer report conveys customer information. Users will be able to control the time frames each report represents.

2.3 User Characteristics

The average user of Cloudbase may have limited experience with computers. The user must be able to retrieve forgotten passwords from an email account and understand how to login to the application. The average user will be more versed in aeronautics and the daily operations of M-ASA than computing. The average user will understand how to write and read time in 24-hour notation.

At least one administrator user will be required to host and administer the software. This individual will be expected to understand how to navigate a hosting environment. The administrator user will be expected to read and understand documentation that will accompany the product to set up other standard users, reset passwords when necessary and perform any other maintenance necessary on the software.

2.4 General Constraints

Cloudbase will be designed to run on the LAMP software stack. Cloudbase will not incorporate additional software not compatible with the GPL software license as it will be released under this license.

2.5 Assumptions and Dependencies

2.5.1. The user is internet literate: The user is able to login with an account

2.5.2. The flight information provided is accurate and available: The user is provided with correct flight information before entering the flight data into the database.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 Web accessible : The product must be accessible and function properly on Internet Explorer 11+, Firefox 35+, Google Chrome 40+ and Safari 7+. The product will be optimized for use on tablets and mobile devices.

3.1.1.2 Login: Users will be provided the opportunity to login ensuring only authorized individuals make changes to the database.

3.1.1.3 Account Page: This page will allow users to change their password and profile information. It will also enable admin users to create edit or remove other user profiles.

3.1.1.4 Search Flight Sheets Page: This page allows the user to search for flight sheets within a date range and displays all flightsheets within the range. This page also allows the user to create a new flight sheet.

3.1.1.5 Flight Sheet Input Page: This page allows the user to input new flights into the database.

3.1.1.6 Reports Page: This page allows the user to view or print reports.

3.1.2 Hardware Interfaces

The product will not feature or require any hardware interfaces.

3.1.3 Software Interfaces

3.1.3.1 Database Connector: The product will interface with a database in order to store and retrieve data.

3.1.4 Communications Interfaces

The system will not have any communication interfaces. All communications functions are handled by the operating system and are not governed by the product.

3.2 Functional Requirements

3.2.1 Time Format

All times will be recorded in 24-hour format.

3.2.2 Authenticated Login

Access will be restricted to users with login credentials.

3.2.2.1 Encrypted Password Storage: Passwords will not be stored in plain text, but a hash of the password (with salt) will be stored in the database. Every time a user logs in the salt will be retrieved and the provided password will be hashed and compared to the stored hash value.

3.2.2.2 Password Change: Users will have the ability to change their own passwords.

3.2.2.3 Password Reset: Users will have the ability to reset their passwords if forgotten.

3.2.3 Admin User

An administrator user will have the ability to create new users and reset passwords for other users.

3.2.4 Flight Sheet

Users will be able to record daily flight operations on flight sheets. Flight Sheets will contain the following information.

3.2.4.1 Date: The date on which the flight took place.

3.2.4.2 Operations Director: Who was the operations director during this flight.

3.2.4.3 Assistant Operation Director (OD): Optional Assistant OD.

3.2.4.4 Tow Plane: Designation of the tow plane on this flight.

3.2.4.5 Tow Plane Pilot: Who was the tow plane pilot for this flight

3.2.4.6 Comments: This allows the user to input any comments for this flight.

3.2.4.7 Glider Plane: Designation of the glider plane used on this flight.

3.2.4.8 Glider Plane Pilot: Who was the glider pilot for this flight.

3.2.4.9 Instructor: Optional. Who was the instructor on this flight.

3.2.4.10 Takeoff Time: Takeoff time of both the tow and glider planes.

3.2.4.11 Towplane Landing Time: Time that the towplane landed.

3.2.4.12 Glider Plane Landing Time: Time that the glider plane landed.

3.2.4.13 Release Altitude: Release altitude of the glider plane.

3.2.4.14 Penalty: Any additional costs incurred on this flight.

3.2.4.15 Cost: Based on release altitude plus any additional penalties incurred.

3.2.5 Add / Remove Pilots

The user will have the ability to add names to the list of pilots. It also allows the user to mark a pilot inactive so that the pilot does not appear in the drop-down list of available pilots on the flight sheet page. Users will have the ability to update

3.2.6 Update Pilot Names

The user will have the ability to update the pilot names. The name change will be reflected throughout the entire database.

3.2.7 Add / Remove Aircraft

The user will have the ability to add aircraft to the list of tow and glider planes. It also allows the user to mark an aircraft inactive so that the aircraft does not appear in the drop-down list of available aircraft on the flight sheet page.

3.2.8 Update Aircraft Name

The user will have the ability to update the aircraft names. The name change will be reflected throughout the entire database.

3.2.9 Create / Edit Payment

The user will have the ability to create and apply payments to flights. Payments will consist of the following information.

3.2.7.1 Flight Number: Uniquely identifies flight to which this payment will be credited.

3.2.7.2 Payment Method: Cash, Card or Check.

3.2.7.3 Amount: Amount to be credited to flight

3.2.7.4 Individual: Identifies the person making this payment.

3.2.10 Reports

The user will have the ability to display reports which may be printed from the browser or downloaded in PDF format. The following reports and their descriptions follow.

3.2.8.1 Aircraft Flight Record: An individual aircraft may be selected and a start and end date provided. The report will display all flight numbers this aircraft flew between these dates and provide the total flight time accrued on these flights.

3.2.8.2 Pilot Flight Record: An individual pilot may be selected and a start and end date provided. The report will display all flight numbers this pilot flew between the two dates and provide the total flight time accrued.

3.2.8.3 Accounts Receivable: This report will display flight numbers and their associate outstanding balance for all flights with an outstanding balance. It will also display a total outstanding balance for all flights with a non-zero balance.

3.3 Use Cases

3.3.1 Use Case #1

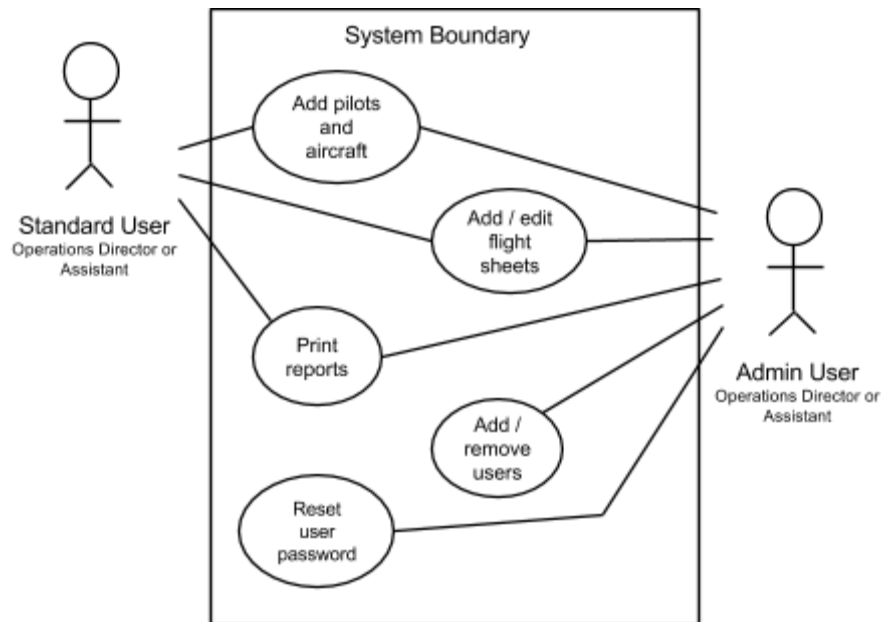


Figure 3.3.1 Cloudbase Use Cases

3.4 Classes / Objects

3.4.1 User

3.4.1.1 Attributes:

3.4.1.1.1 username: Stores username of logged in user.

3.4.1.1.2 name: Stores full name of logged in user.

3.4.1.1.3 password: Stores the password entered by the user.

3.4.1.2 Functions: None

3.4.1.3 Related Requirement: 3.2.2 and 3.2.3.

3.4.2 DAC (Database Access Component)

3.4.2.1 Attributes:

3.4.2.1.1 username: Name of the database user.

3.4.2.1.2 password: Password of the database user.

3.4.2.2 Functions:

3.4.2.2.1 connect: Connects to the database.

3.4.2.2.2 disconnect: Disconnects from the database.

3.4.1.3 Related Requirements: 3.2.2 - 3.2.8

3.4.3 Flight Record

3.4.3.1 Attributes:

3.4.3.1.1 Date: The date on which the flight took place.

3.4.3.1.2 Operations Director: Who was the operations director during this flight.

3.4.3.1.3 Assistant Operation Director (OD): Optional Assistant OD.

3.4.3.1.4 Tow Plane: Designation of the tow plane on this flight.

3.4.3.1.5 Tow Plane Pilot: Who was the tow plane pilot for this flight

3.4.3.1.6 Comments: This allows the user to input any comments for this flight.

3.4.3.1.7 Glider Plane: Designation of the glider plane used on this flight.

3.4.3.1.8 Glider Plane Pilot: Who was the glider pilot for this flight.

3.4.3.1.9 Instructor: Optional. Who was the instructor on this flight.

3.4.3.1.10 Takeoff Time: Takeoff time of both the tow and glider planes.

3.4.3.1.11 Towplane Landing Time: Time that the towplane landed.

3.4.3.1.12 Glider Plane Landing Time: Time that the glider plane landed.

3.4.3.1.13 Release Altitude: Release altitude of the glider plane.

3.4.3.1.14 Penalty: Any additional costs incurred on this flight.

3.4.3.1.15 Cost: Based on release altitude plus any additional penalties incurred.

3.4.3.2 Functions:

3.4.3.2.1 save: Writes attributes to database.

3.4.3.2.2 retrieve: Reads attributes from database.

3.4.1.3 Related Requirements: 3.2.4

3.5 Non-Functional Requirements

3.5.1 Performance

3.5.1.1 Maximum Response Time: The product may be accessed simultaneously by multiple users at the same time. The maximum recommended number of users is five. Barring any network or server related delays (which are beyond the scope of the product) the maximum allowable response time with five active users will be 1 second.

3.5.2 Reliability

3.5.2.1 Input Validation: The product shall validate data on input to ensure errors do not arise as a result of incorrect data types or values. Every field will require appropriate validation.

3.5.2.2 Absence of errors: The product will function free of errors (barring server, client machine and network related errors) so long as valid input is entered.

3.5.3 Availability

3.5.3.1 Always On: The product shall be accessible 100% of the time barring downtime related to the server or network.

3.5.4 Security

3.5.4.1 User Authentication: The product shall require authentication in order to progress past the login page.

3.5.4.2 Secure Password Storage: The product shall store passwords in a secure manner by generating a random salt and hashing input passwords before storing them in the database.

3.5.5 Maintainability

3.5.5.1 Comments: At the beginning of each file will be a comment explaining the following code's function. Code segments exceeding 5 lines will be preceded by a comment explaining their purpose and include the developer's name and the date it was written.

3.5.5.2 Documentation: Documentation outlining installation procedures and program usage will be included.

3.5.5.3 Logging Subsystem: Records database access events and errors.

3.5.6 Portability

3.5.6.1 Database Compatibility: The product will run on any version of MySQL at or above version 5.6.

3.5.6.1 PHP Compatibility: The product will run with any version of PHP at or above version 5.6.5.

3.6 Inverse Requirements

3.6.1 Finances: Although the product will track some information related to finances, it is not intended to replace a fully featured accounting solution.

3.6.2 Maintenance: Although flight hours for each aircraft are reported, it will not track maintenance performed on the aircraft.

3.6.3 Administrative Password Reset: The administrator will have the ability to initiate a password reset for any account. The administrator, however, will not have the ability to specify the password for any account other than his/her own.

3.7 Design Constraints

3.7.1 Time Format: As required by M-ASA, all times will be displayed in 24-hour format.

3.8 Logical Database Requirements

3.8.1 License: As required by M-ASA, the database must be an open source variant.

3.9 Other Requirements

3.9.1 Analytic Graph: An analytic graph is required for each flight. Each analytic graph records and displays flight time for specific flights.

4. Analysis Models

4.1 Sequence Diagrams

Complete flight information is of paramount importance to the client's needs. Of all flight data, the takeoff and landing times for planes and gliders may have the greatest organizational impact, because plane maintenance is scheduled according to cumulative flight time. Complete flight information is not available until both the plane and glider land. The easiest way to avoid incomplete flight information is for the users to enter the complete record once that information is available. Some users may prefer this method of data entry. However, other users may prefer entering flight information as it becomes available. For Cloudbase to handle this capability, it must have facilities for tracking and updating incomplete records.

Flight 4.1.1 is a sequence diagram that demonstrates the functionality of Cloudbase in both aforementioned scenarios. Once flight #1 takes off, a user enters all available data for flight #1. The data entry processing engine will record the available data in the database and then notify the user, via web interface, that information is missing. Next, the user enters complete flight information. Cloudbase will record the data and then confirm its storage. Later on, the complete record for flight #1 becomes available. The user will then update the flight #1 database entry to make it complete.

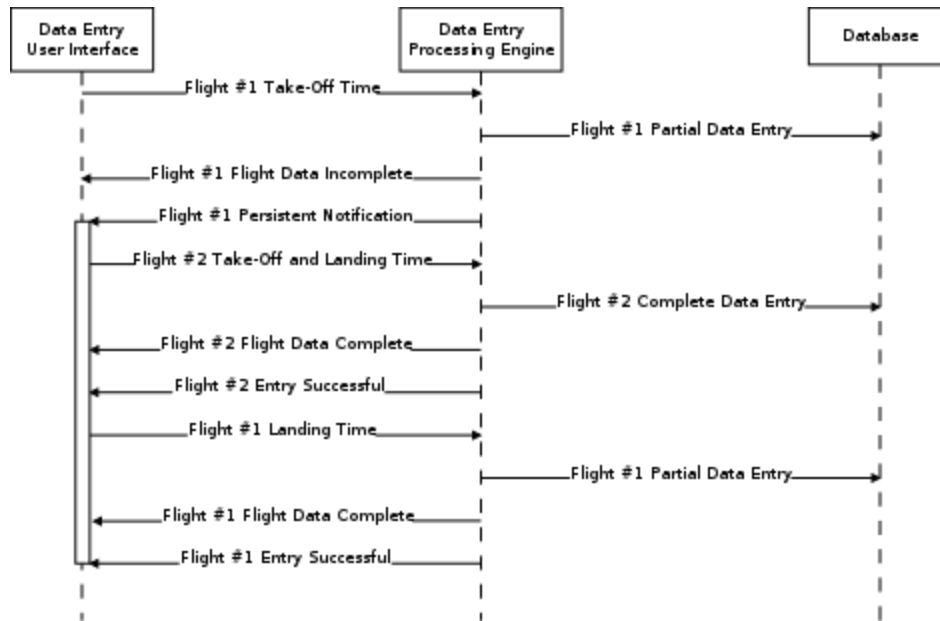


Figure 4.1.1 Cloudbase Flight Entry Sequence Diagram

4.2 State-Transition Diagrams (STD)

Because Cloudbase must be a secure application, its initial state is authentication. The authentication state checks login credentials and establishes user permissions accordingly.

From the authentication state, the state will change according to user choice. Cloudbase will offer three categories of services, and each category corresponds to the program state. In the administration state, users can modify account settings and update available options for recording flight information. In the flight data entry state, Cloudbase will store the flight data that the user enters and give appropriate feedback. In the flight summary state, Cloudbase will produce summary reports according to user-selected options.

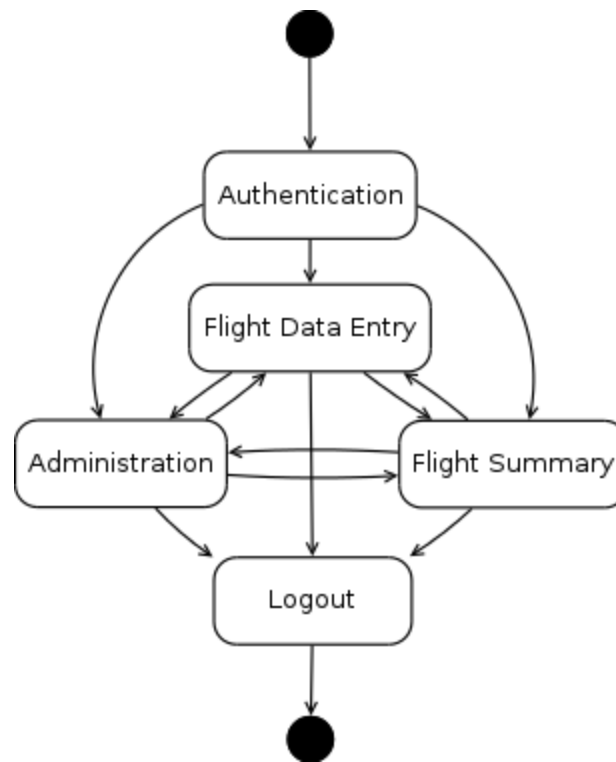


Figure 4.2.1 Cloudbase State Transition Diagram

4.3 Data Flow Diagrams (DFD)

Figure 4.3.1 demonstrates the flow of incomplete and complete flight data. At the web interface, users enter flight data. The data then flows to the flight data processing engine. At this point, the data processing engine examines the data for completeness. It stores both complete and incomplete entries in the database. However, the data processing engine retains a key for identifying incomplete flight records. A list of incomplete entries flows to the web interface where a table of incomplete flight records will appear. Each entry on the table will persist until the user has completed its record.

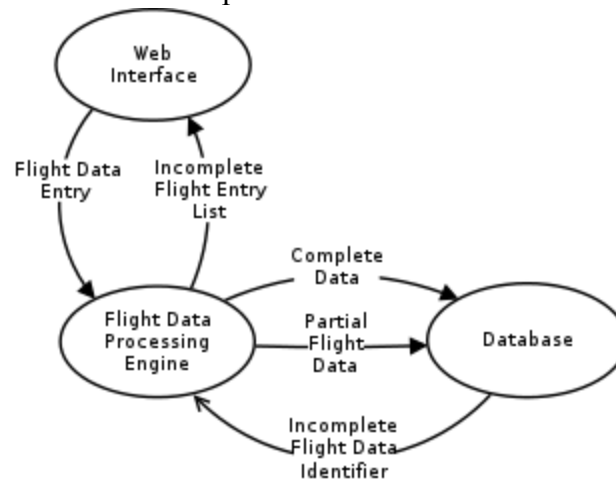


Figure 4.3.1 Cloudbase Flight Entry Data Flow Diagram

5. Change Management Process

If a team member recognizes a need to change the SRS he will contact the other group members. If all group members agree on the change they will add the changes to a draft version of the SRS. The team liaison will then contact the client and seek approval for the change. With the client's approval the draft SRS becomes the current working SRS.