Concordia Institute for Information Systems Engineering (**CIISE**)

Team Noble

Software Quality Assurance Plan (SQAP)

Project – Electrification of STM bus route 211 from Terminal McDonalds to Lionel-Groulx

Prepared By –Yu Jin (23753034), Gurpreet Kaur (40106011), Daniel Zakerifar (40054453)

Date of Issue – April 28th, 2020

Title – Software Quality Assurance Plan (SQAP)	
Authors – Yu Jin , Gurpreet Kaur, Daniel	**Date – 28th April**, 2020
Approved By – Prof. Chun Wang	**Date –**

Introduction 3
1. Purpose 3
2. Reference Documents 4
3. Management 4
3.1 Organization 4
3.2 Roles 5
3.3 Tasks and Responsibilities 5
Quality Assurance Estimated Resources 6
4. Documentation 6
5. Standards, practices, conventions, and metrics 7
5.1 Standards 7
5.2 Metrics 7
6. Software Reviews 9
6.1 Minimum requirements 10
6.1.1 Software Development Plan Review (DPR) 10
6.1.2 Software requirement specifications review (SRSR) 10
6.1.3 Design review of preliminary design (PDR) 10
6.1.4 Detailed design review (DDR) 10
6.1.5 Verification and validation plan review (TPR) 10
7. Test 11
8. Problem reporting and corrective action 19
9. Tools, techniques and methodologies 20
10. Media control 20
11. Supplier Control 20
12. Records collection, maintenance and retention 21
13. Training 21
14. Risk management 21
15. Glossary 21
16. SQAP change procedure and history 21

Introduction

This document is the deliverable “Software Quality assurance Plan (SQAP)” for electrification of STM bus route 211 from terminus McDonalds to Terminal Lionel-Groulx.

The document is organized in the following sections:

1. Purpose
2. Reference documents
3. Management
4. Documentation
5. Standards, practices, conventions, and metrics
6. Software reviews
7. Test
8. Problem reporting and corrective action
9. Tools, techniques, and methodologies
10. Media control
11. Supplier control
12. Records collection, maintenance, and retention
13. Training
14. Risk management
15. Glossary
16. SQAP change procedure and history

1.Purpose

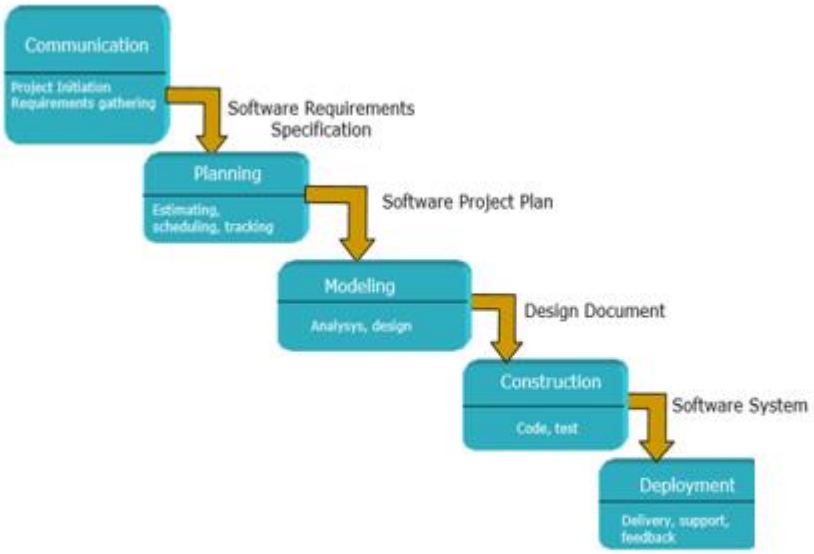
We will be developing the software that will help the STM Team of Montreal to plan the electrification of bus route 211 in their system i.e. from Lionel-Groulx to Sainte-Anne/Terminus McDonals. The software will provide the optimal number of buses and chargers to be obtained at each terminus in order to electrify the bus route 211.

Definition of the critically of the need for this software solution:

Because the price of the electrifying devices is so expensive, the investors should know how much they should spend, and what is the outcome of the investment.

Development Life Cycle Model used:

Development lifecycle model to be used for the development of whole project is waterfall model.



2.Reference Documents

List of documents that form the basis of this SQAP

The documents used in order to build the SQAP is as follows:

1. IEEE Std 730™-2002 (Revision of IEEE Std 730™-1998)

(This standard assists in determining the content and preparation of Software Quality Assurance Plans and provides a standard against which such plans can be prepared and assessed. It is directed toward the development and maintenance of software.)

2. Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 8th edition, Jan 2014, ISBN-10: 007802212

3.Management

This section describes the project organization structure, its tasks and its roles and responsibilities.

3.1 Organization

This project is developed by a team of two (Gurpreet Kaur, Daniel) under the supervision of professor Chun Wang. Therefore, it the responsibility of both the team members to review the product usability, efficiency, reliability and accuracy. The professor will however conduct inspections, reviews and walk-through of the work on a regular basis.

3.2 Roles

1. **Developer** – Daniel, Gurpreet Kaur
2. **QA** – Yu Jin
3. **Manager** – Dr. Chun wang

3.3 Tasks and Responsibilities

The responsibility of developers are as follows:

- Develop the requirement specification document.
- Develop the design plan and testing plan for testing the software.
- Implement and test the application and deliver the application along with necessary documentation.
- Give a formal document and presentation (optional) to the manager on completion of the analysis, design and testing phases. The manager will review the developer's work and provides feedback.
- Planning, coordinating, testing and assessing all aspects of quality issues.

The responsibility of manager are as follows:

- Review the work performed by the developer
- Provide feedback and advice

RACI chart (R - Responsible, A – Accountable, C - Consulted, I – Informed)

Activities/Tasks	Yu	Daniel	professor
Quality Assurance	C,R	C,R	A,I,C
Software Engineering	A,C,R	A,C,R	I,C
Quality Control	C,R	C,R	I,C

3.4 SQA Implementation in different phases of waterfall model

Quality assurance will be implemented through all the software life cycles of the tool's development process, until the release of the software product. The following are the quality assurance tasks for each phase of the software development:

Requirements phase: When the SRS is being developed, the developer must ensure that it elucidates the proposed functionality of the product and to keep refining the SRS until the requirements are clearly stated and understood.

Specification and Design phase: Due to the great importance for accuracy and completeness in these documents, weekly reviews shall be conducted between the developer and the professor to identify any defects and rectify them. Also, software design specification document developed during this phase.

Implementation phase: The developer shall do code reviews when the construction phase of the Tool begins.

Software testing phase: The developer shall test each case. The final product shall be verified with the functionality of the software as specified in the Software Requirements Specification (SRS) for the Tool.

Through all these phases of the software development, the following shall also be conducted to improve the software quality:

· Develop and generate SQAP: Generate a finalized SQAP plan

· Communication and Feedback: The developer is encouraged to freely express disagreements, suggestions and opinions about all aspects of the weekly process of software development.

· Internal audits and evaluations: The professor is expected to do evaluations at the end of each phase in the project.

Quality Assurance Estimated Resources

A team of two is assigned for the completion of this project for four months.

4. Documentation

In addition to this document, additional documentation will include:

- Software Specification Requirement
  This document consists of all the requirement which needs to be there in the software.
- Project Development Plan

This document consists of the schedule project, each activity should be given start date along with the expected end date in order to track the progress of the project.

- Software Design Description
  This document consists of the blueprint of the design of product/software. In this project this document contains the algorithm for the software.
- Verification and Validation Plans
  This document contains the detailed description of the steps to verify that product developed is the right or not, along with validation steps in order to make sure that all the processes needed to develop the product should be right. For example, plans to review SRS document, testing plans etc.
- Verification Result Report and Validation Result report
  This document consists of the result for verification and validation activities. For example, result of SRS document review, result of testing.
- User Documentation
  This document consists of user manual that helps the end user to understand the operation of product and software.

5. Standards, practices, conventions, and metrics

5.1 Standards

- Coding standards: All functions will follow proper naming convention; comments will be written in order to increase the readability of code.
- Testing Standards: Follow IEEE standard for software test documentation.

5.2 Metrics

- LOC: Line of Code is used to measure the size of the software
- Function-based Metrics: Used to forecast human resources required for the project.
- Error Density Metrics: Provide info about the number of code errors detected in the software code by code inspections and testing.
- Error Severity Metrics: Provide info to detect situations of increasing number of severe errors in situations where errors and weighted errors are generally decreasing.

We have utilized function-based metrics for finding out the complexity of our product along with forecasting human resources required for the development of the project.

Function points calculation for determining the complexity:

Information Domain Value	Count	Weighting Factor			Result
		Simple	Avergae	Complex	
External inputs (EIs)	0	3	4	6	0
External outputs (EOs)	3	4	5	7	21
External inquiries (EQs)	5	3	4	6	30
Internal Logical Files (ILFs)	2	7	10	15	30
External Interface Files (EIFs)	3	5	7	10	30
Count Total					111

To compute function points, the following relationship is used:

$$FP = \text{count total} \times [0.65 + 0.01 \times \Sigma (F_i)]$$

The F_i (i = 1 to 14) are *value adjustment factors* (VAF)

Calculations for VAF:

For our project VAF is 44 because only following factors are true among the 14,

Factor 9. Are the inputs, outputs, files or inquiries complex?

Factor 10. Is the internal processing complex?

Factor 11. Is the code designed to be reusable?

Factor 14. Is the application designed to facilitate change and ease of use by the user?

Therefore, the FP for our project is,

FP = 111 * [0.65 + (0.01*44)] = 120.99 ~ 121

Therefore, based on FP computation and historical information, we need two human resources for four months to complete whole project along with documentation.

6. Software Reviews

The people responsible for doing the review are Producer, Review leader, Reviewer, and recorder. They will be chosen from senior members of the project team, senior professional assigned to other projects, customer, and software development consultant. To prepare the review meeting, review leader appoints the team members, schedules the review sessions and distributes the design document and checklists among the team members.

In our project review team will typically consist of professor, classmates. Output of each review meeting will give us review document that will contain the summary of meeting like what was reviewed, who did review, what were the comments, time to resolve the comment or date of next review meeting.

The following table describes the document to be reviewed for our project along with dates and summary document of every review meeting.

Topic to Review	Date	Document to be produced after the review
Review of SRS	Jan-28-2020	SRSR
Review of Development Plan	Feb-3-2020	DPR
Design review of preliminary design	Feb-15-2020	PDR
Detail design Review	Feb-30-2020	DDR
Review of test plan	Apr-20-2020	TPR

6.1 Minimum requirements

6.1.1 Software Development Plan Review (DPR)

In this review meeting we try to find that our plan is according to the Standards and will it be able to deliver the project within time and quality constraints.

6.1.2 Software requirement specifications review (SRSR)

In this review meeting we tried to make sure that all the requirements of customer are clear and documented in Requirement specification Document.

6.1.3 Design review of preliminary design (PDR)

In the review meeting we tried to make sure that customer is satisfied with the initial design of the product.

6.1.4 Detailed design review (DDR)

In this review meeting will try to make sure that blueprint of the algorithm will be able to develop the product or not.

6.1.5 Verification and validation plan review (TPR)

The purpose of this review meeting is to make sure that plans for verification and validation of the product is good enough to meet quality standards for satisfying the customer needs.

Template used for reviewing all above mentioned documents is given below:

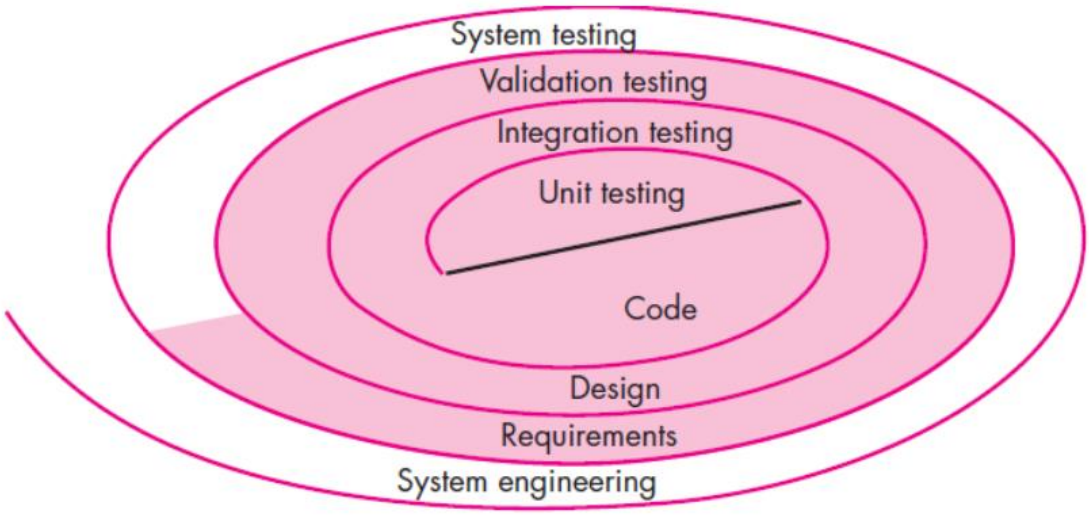
Review Record				
Name of Activity for which review meeting is planned for.	Date reviewed	Date Approved	Comment	Signature

7. Test

There are various types of software testing techniques available in the market like Integration testing, Unit testing, Stress testing and many more. In this project, team consist of two individuals, both will be responsible for the development as well as testing.

Our project will follow below approach for testing:

1. **Which strategic approach to be used for software testing?**
   Our approach to test project will be start from “testing in the small” and move towards “testing in the large”.



2. **Which approach to be used to do validation testing?**
   For the validation testing, SRS document will be base. Tester should design the test cases in order to meet all the requirements mentioned in SRS document. If any requirement is not met in the final product, this needs to communicate to developer before final submission.

Following are the test cases written for our project:

Unit Testing

Test case	Unit case Bus	
Name of state class	Battery	
Name of the states	stateOfCharging	
Scenario	We create one bus.The has a battery with capacity of 300Kwh. Then we do the following methods:	
Expected Result	Methods	StateOfCharging
	Taveling	260Kwh
	TravelingEmpty	220Kwh
	FastCharging	227
	overNightCharging	300

Integration Testing

Test case	Integration Test
Name of state class	Battery
Name of the states	stateOfCharging
Scenario	For integrated Testing we are using the random testing. The way is that we put one simple symmetric schedule in the software and then the we see the results. Because it is symmetric, we can calculate the everything with hand. This is the schedule for west and east 05,00,00 05,30,00 08,30,00 10,30,00 12,30,00 14,00,00 16,00,00 18,00,00 19,30,00 21,30,00

Expected Results

AT_SOCkm	Charger_ID	Start_Time_quickCharg	End_Time_quickCharge	AT_SOC(ON)	Charger_ID2	Start_Time_overnight	End_Time_overnight	BT_SOC_trip	Trip_ID	Start_Time_Trip	End_Time_Trip	Reach
0km	l						294	east05:00:00	05:00:00	06:00:00	east	
254km	east2OC450kw	06:00:00	06:01:00				261	west08:30:00	08:30:00	09:30:00	west	
221km	west1OC450kw	09:30:00	09:31:00				229	east10:30:00	10:30:00	11:30:00	east	
189km	east2OC450kw	11:30:00	11:31:00				196	west12:30:00	12:30:00	13:30:00	west	
156km	west1OC450kw	13:30:00	13:31:00				164	east14:00:00	14:00:00	15:00:00	east	
124km	east2OC450kw	15:00:00	15:01:00				131	west16:00:00	16:00:00	17:00:00	west	
91km	west1OC450kw	17:00:00	17:01:00				99	east18:00:00	18:00:00	19:00:00	east	
59km	east2OC450kw	19:00:00	19:01:00	66Km	east3DC50KW	19:01:00	00:01:00	end of trip				

bus ID: 2east05:00:00

AT_SOCkm	Charger_ID	Start_Time_quickCharg	End_Time_quickCharge	AT_SOC(ON)	Charger_ID2	Start_Time_overnight	End_Time_overnight	BT_SOC_trip	Trip_ID	Start_Time_Trip	End_Time_Trip	Reach
0km	null	null		0Km	null	null	294	west05:00:00	05:00:00	06:00:00	west	
254km	west1OC450kw	06:00:00	06:01:00	0Km	null	null	261	east08:30:00	08:30:00	09:30:00	east	
221km	east2OC450kw	09:30:00	09:31:00	0Km	null	null	229	west10:30:00	10:30:00	11:30:00	west	
189km	west1OC450kw	11:30:00	11:31:00	0Km	null	null	196	east12:30:00	12:30:00	13:30:00	east	
156km	east2OC450kw	13:30:00	13:31:00	0Km	null	null	164	west14:00:00	14:00:00	15:00:00	west	
124km	west1OC450kw	15:00:00	15:01:00	0Km	null	null	131	east16:00:00	16:00:00	17:00:00	east	
91km	east2OC450kw	17:00:00	17:01:00	0Km	null	null	99	west18:00:00	18:00:00	19:00:00	west	
59km	west1OC450kw	19:00:00	19:01:00	66Km	west4DC50KW	19:01:00	00:01:00	0	null	null	null	

bus ID: 3west05:30:00

AT_SOCkm	Charger_ID	Start_Time_quickCharg	End_Time_quickCharge	AT_SOC(ON)	Charger_ID2	Start_Time_overnight	End_Time_overnight	BT_SOC_trip	Trip_ID	Start_Time_Trip	End_Time_Trip	Reach
0km	null	null		0Km	null	null	294	east05:30:00	05:30:00	06:30:00	east	
254km	east2OC450kw	06:30:00	06:31:00	0Km	null	null	261	west19:30:00	19:30:00	20:30:00	west	
221km	west1OC450kw	20:30:00	20:31:00	0Km	null	null	229	east21:30:00	21:30:00	22:30:00	east	
0km	null	null	0Km	null	null	0	null	null	null			

bus ID: 4east05:30:00

AT_SOCkm	Charger_ID	Start_Time_quickCharg	End_Time_quickCharge	AT_SOC(ON)	Charger_ID2	Start_Time_overnight	End_Time_overnight	BT_SOC_trip	Trip_ID	Start_Time_Trip	End_Time_Trip	Reach
0km	null	null		0Km	null	null	294	west05:30:00	05:30:00	06:30:00	west	
254km	west1OC450kw	06:30:00	06:31:00	0Km	null	null	261	east19:30:00	19:30:00	20:30:00	east	
221km	east2OC450kw	20:30:00	20:31:00	0Km	null	null	229	west21:30:00	21:30:00	22:30:00	west	

Validation Testing

Use Case Name	Normal Usage
Use case ID	UC – 1
Description	In this test will check the output of the software when user provides all the necessary inputs as mentioned in precondition.
Precondition	User should provide inputs like manufacture for charger, price for both slow speed and high-speed charger, Please input the kind of battery, Please input the price of bus
Expected Result	Software should display number of buses, number of both types of charger at both terminuses, and total price and generate three files (bus schedule, charging schedule, state of chargers along with bus schedule)

Use Case Name	Validate Allotment of Buses (charger type is ABB and Battery size is small)
Use case ID	UC – 2
Description	In this test will validate the allotment of buses by validating the bus schedule file generated by software when battery size is small and charger type is ABB
Precondition	User should provide inputs like manufacture for charger, price for both slow speed and high-speed charger, Please input the kind of battery, Please input the price of bus
Expected Result	Bus schedule file should show allotment of 19 buses

Use Case Name	Validate Allotment of Buses (charger type is ABB and Battery size is big)	
Use case ID	UC – 3	
Description	In this test will validate the allotment of buses by validating the bus schedule file generated by software when battery size is big and charger type is ABB	
Precondition	User should provide inputs like manufacture for charger, price for both slow speed and high-speed charger, Please input the kind of battery, Please input the price of bus	
Expected Result		Bus schedule file should show allotment of 24 buses

Use Case Name	Validate Allotment of Buses (charger type is HELIOX and Battery size is small)	
Use case ID	UC - 4	
Description	In this test will validate the allotment of buses by validating the bus schedule file generated by software when battery size is small and charger type is HELIOX	
Precondition	User should provide inputs like manufacture for charger, price for both slow speed and high-speed charger, Please input the kind of battery, Please input the price of bus	
Expected Result		Bus schedule file should show allotment of 23 buses

Use Case Name	Validate Allotment of Buses (charger type is HELIOX and Battery size is small)
Use case ID	UC - 5
Description	In this test will validate the allotment of buses by validating the bus schedule file generated by software when battery size is small and charger type is HELIOX
Precondition	User should provide inputs like manufacture for charger, price for both slow speed and high-speed charger, Please input the kind of battery, Please input the price of bus
Expected Result	Bus schedule file should show allotment of 17 buses

Use Case Name	Validate input format for first input
Use case ID	UC - 6
Description	In this test will validate the input format for first input of our software which is “please enter the manufacture of charger (ABB/HELIOX)”. This field should accept only these two values.
Precondition	User should provide input either ABB or HELIOX
Expected Result	Upon provided the input which is either ABB or HELIOX only then, software should accept the input else give error message and stop the flow.

Use Case Name	Validate input format for second input	
Use case ID	UC - 7	
Description	In this test will validate the input format for second input of our software which is “please input the price of over-night charger”. This field should accept only numeric values.	
Precondition	User should provide input only in numerical format	
Expected Result		Upon provided the input, which is number value only then, software should accept the input else give error message and stop the flow.

Use Case Name	Validate input format for third input	
Use case ID	UC - 8	
Description	In this test will validate the input format for third input of our software which is “please input the price of fast charger”. This field should accept only numeric values.	
Precondition	User should provide input only in numerical format	
Expected Result		Upon provided the input, which is number value only then, software should accept the input else give error message and stop the flow.

Use Case Name	Validate input format for fourth input
Use case ID	UC - 9
Description	In this test will validate the input format for fourth input of our software which is “please input the “kind of battery: small(294kwh), big(394kwh)”. This field should accept either small or big.
Precondition	User should provide string input which should be either small or big.
Expected Result	Upon provided the input, which is either ABB or HELIOX only then, software should accept the input else give error message and stop the flow.

Use Case Name	Validate input format for fifth input
Use case ID	UC - 10
Description	In this test will validate the input format for fifth input of our software which is “please input the price of bus”. This field should accept only numeric values.
Precondition	User should provide input only in numerical format
Expected Result	Upon provided the input, which is number value only then, software should accept the input else give error message and stop the flow.

Use Case Name	Validate input format for second, third and fifth input should not be too large nor zero
Use case ID	UC - 11
Description	In this test will validate the input format for second, third and fifth input of our software which is “please input the price of over-night charger, fast charger and price of bus. This field should accept only numeric values that should be neither too long nor zero.
Precondition	User should provide input only in numerical format
Expected Result	Upon provided the input, which is number value greater than zero, non-negative number, number not too large only then, software should accept the input else give error message and stop the flow.

8. Problem reporting and corrective action

Our project will follow certain procedures to track, report and resolve problems in the software product as well as in the process used for the development.

1. Each document discussed in the documentation section (section 4) should be reviewed either by the professor or by the peers and review result will be documented. If, everything seems good to review team then only that document will be considered as complete.
2. Any comments made by review team during the review process needs to be resolved.
3. Project development plan document is created in order to keep the track of the progress of the project as each activity required for the project is written there along with start and end date.
4. Every document should have version at the start along with status and who approved it.

9. Tools, techniques and methodologies

In our project will follow proper SQA management plan in order to carry out SQA activities. Activities involved in SQA is as follows:

1. SQA management plan: In this will defined how SQA activities should be carried out, examine whether we have proper skill set to carry out this activity or not.
2. Set up checkpoints: Team will set up certain checkpoints according to which it evaluates the quality of the project activities at each stage.
3. Apply software engineering techniques: Applying software engineering techniques helps software designer to gather requirement more precisely. For example, for gathering information, developer may use techniques such as interviews and later once information has been gathered software designer can prepare the project estimation using KLOC.
4. Execute formal technical reviews: A FTR is done in order to check the quality of each and every product whether its product or document.
5. Proper testing plan: Multi strategy testing approach will be used in order to test the product from each angel.
6. Controlling Change and measure change impact: If there is any change in the requirements it should be come from proper channel and before implementing the change, proper evaluation shall be done that how much more time will be required to do the change, how it impact our testing schedule etc.
7. Maintain records and reports: All document should be kept for future reference and should be available to share with the stakeholders whenever needed.

10. Media control

This section is not applicable to this plan.

11. Supplier Control

This section is not applicable to this plan.

12. Records collection, maintenance and retention

All the documents required for the project will kept for 1 year after the project is completed and will be available to share to stakeholders within the specified time. Since we are keeping the documents only for 1 year after the project id done therefore, the retention period for our project will be one year and this will be communicated with all the SME's, stakeholders.

13. Training

This section is not applicable to this plan.

14. Risk management

This section is not applicable to this plan.

15. Glossary

SRS – system requirement specifications
SRSR – system requirement specification review
SDS – system design specification
SDSR – system design specification review
FP – Function Point
VAF – Value Adjusted Factor

16. SQAP change procedure and history

Following tables should be used in order to maintain the history/version of any document.

CHANGE LOG

Reason for Change	Issue	Revision	Date

CHANGE RECORD

Reason for Change	Date	Pages	Paragraphs