

# **SeniorConnect Project**

## **Quality Plan**

Version 3

30/09/2015

GruFamily Project Team

Nanyang Technological University  
50 Nanyang Avenue Singapore  
639798

## Revision History

Version No.	Date	Description
1	16/09/15	Initial version 1
2	22/09/15	Merged content. Ensured consistency and removed duplicated content.
3	30/09/15	Version 2: Add in Forms and Checklists
4	10/10/15	Version 3: Finalized version

# SeniorConnect System

## Quality Plan

22/09/2015

### SQA Plan Approvals:

L1 YISHAN

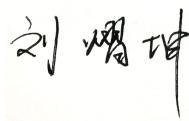
10/10/15

---

SQA Manager

---

Date



10/10/15

---

Project Manager

---

Date

# Table of Contents

<b>1 Purpose and Scope.....</b>	<b>1</b>
<b>1.1 Purpose .....</b>	<b>1</b>
<b>1.2 Scope .....</b>	<b>1</b>
<b>1.3 Product Introduction.....</b>	<b>2</b>
1.3.1 Description of Product.....	2
1.3.2 Intended Market.....	2
<b>2 Reference documents .....</b>	<b>3</b>
<b>3 Management .....</b>	<b>4</b>
<b>3.1 Organization .....</b>	<b>4</b>
<b>3.2 Responsibilities Overview .....</b>	<b>5</b>
<b>4 Overview of Quality Assessment Tasks .....</b>	<b>6</b>
<b>4.1 Reviews.....</b>	<b>6</b>
<b>4.2 Tracking.....</b>	<b>7</b>
<b>4.3 Estimated Resources.....</b>	<b>8</b>
4.3.1 Facilities and Requirements.....	8
4.3.2 Personnel.....	8
<b>5 QA Tasks.....</b>	<b>8</b>
<b>5.1 Task: Review Software Products .....</b>	<b>8</b>
<b>5.2 Task: Evaluate Software Products Review Process .....</b>	<b>8</b>
<b>5.3 Task: Evaluate Project Planning, Tracking and Oversight Processes .....</b>	<b>9</b>
<b>5.4 Task: Evaluate Software Requirements Analysis Process .....</b>	<b>9</b>
<b>5.5 Task: Evaluate Software Design Process.....</b>	<b>10</b>
<b>5.6 Task: Evaluate Software Implementation and Unit Testing Process .....</b>	<b>10</b>
<b>5.7 Task: Evaluate Unit Integration and Testing .....</b>	<b>11</b>
<b>5.8 Task: Conduct and Verify Project Reviews.....</b>	<b>11</b>
5.8.1 Task: Conduct Technical Reviews .....	12
5.8.2 Task: Verify Technical Reviews .....	13
5.8.3 Task: Conduct Other Deliverable Reviews.....	14
5.8.4 Task: Verify Management Reviews .....	17
5.8.5 Task: Conduct Process Audits .....	17
5.8.6 Task: Conduct Configuration Audits.....	17
<b>5.9 Task: Audit Software Configuration Process .....</b>	<b>17</b>
<b>5.10 Task: Verify Software Quality Assurance .....</b>	<b>18</b>
<b>5.11 QA Task Responsibilities .....</b>	<b>18</b>
<b>5.12 QA Schedule .....</b>	<b>19</b>
<b>5.13 QA Review Forms and Checklists .....</b>	<b>19</b>
5.13.1 Review Meeting Minutes.....	19
5.13.2 Review Form Template .....	20
5.13.3 Software Requirement Review Checklist .....	22
5.13.4 Software Design Review Checklist.....	24
5.13.5 Test Readiness Review Checklist .....	26
5.13.6 Post-Project Review Checklist.....	27
5.13.7 Technical Review Audit Form.....	28
5.13.8 Management Review Form.....	30

5.13.9	Code Review Checklist.....	31
<b>5.14</b>	<b>QA Audit Checklists and Forms.....</b>	<b>34</b>
5.14.1	Software Requirement Analysis Process Audit Checklist .....	34
5.14.2	Software Design Process Audit Checklist.....	35
5.14.3	Software Implementation and Unit Testing Process Audit Checklist .....	36
5.14.4	Software Unit Integration and Testing Process Audit Checklist .....	37
5.14.5	Process Audit Form.....	38
<b>6</b>	<b>Documentation .....</b>	<b>39</b>
6.1	Purpose .....	39
6.2	Minimum Documentation Requirements & Guidelines.....	39
6.3	Check for Adequacy .....	40
<b>7</b>	<b>Standards, Practices, Conventions, and Metrics .....</b>	<b>41</b>
7.1	Purpose .....	41
7.2	Content .....	41
<b>8</b>	<b>Reviews and Audits .....</b>	<b>42</b>
8.1	Purpose .....	42
8.2	Minimum requirements.....	42
8.2.1	Software Requirements Review (SRR).....	42
8.2.2	Preliminary Design Review (PDR) .....	43
8.2.3	Critical Design Review (CDR) .....	43
8.2.4	Functional audit.....	43
8.2.5	Physical audit.....	43
8.2.6	In-process audits.....	44
8.2.7	Managerial reviews.....	44
8.2.8	Software Configuration Management Plan Review (SCMPR).....	44
8.2.9	Post-mortem review.....	45
8.3	Review Plans & Standards .....	46
8.4	Other.....	50
<b>9</b>	<b>Test .....</b>	<b>51</b>
<b>10</b>	<b>Problem reporting and corrective action .....</b>	<b>52</b>
<b>10.1</b>	<b>Process Audit Report.....</b>	<b>52</b>
10.1.1	Submittal and Disposition of Process Audit Report.....	53
10.1.2	Escalation Procedure for Resolution of Non-Concurrence on Process Audit Report	53
<b>10.2</b>	<b>Recording Problems in Software Code or Documentation.....</b>	<b>53</b>
<b>11</b>	<b>Tools, techniques, and methodologies .....</b>	<b>53</b>
<b>12</b>	<b>Code Control.....</b>	<b>53</b>
<b>13</b>	<b>Media control .....</b>	<b>54</b>
<b>14</b>	<b>Supplier control .....</b>	<b>55</b>
<b>15</b>	<b>Records collection, maintenance, and retention .....</b>	<b>55</b>
<b>16</b>	<b>Training .....</b>	<b>56</b>
<b>17</b>	<b>Risk Management .....</b>	<b>56</b>
<b>17.1</b>	<b>SQA for Project Risk Management .....</b>	<b>57</b>

**17.2 Risk Management for SQA Activities..... 57**

## **List of Tables**

Table 1 Roles and Responsibilities .....	5
Table 2 Issue Critical Level Classifications.....	8
Table 3 Reviews associated with SDLC Stage .....	12
Table 4 Minimum Documentation Requirements & Guidelines .....	39
Table 5 Project Plan Review .....	46
Table 6 Software Requirements Review.....	47
Table 7 Software Design Review.....	48
Table 8 Test Readiness Review .....	49
Table 9 Post-Project Review .....	50
Table 10 Training Activities Overview.....	56
Table 11 Identified Risk List.....	58

## **List of Figures**

Figure 1: Major Activities in SDLC.....	2
Figure 2 Project Team Organization Structure .....	5
Figure 3 Unit Test Flow .....	51
Figure 4 Integration Test Flow.....	52

# **1 Purpose and Scope**

## **1.1 Purpose**

This Software Quality Assurance Plan (SQAP) aims to provide standards and widely accepted procedures to ensure quality assurance during the software development life cycle of the SeniorConnect System. This system is designed to be a social networking tool for the elderly. The ultimate goal is to provide an elderly-targeted communication environment and to encourage more interactive social activities for silver-aged people, which hence help them to live a healthier life.

This SQAP sets out standards, practices and conventions that shall be followed closely, which provides instructions to SQA planning, defines major procedures, and also offers guidelines to resolve the problems that may be encountered during SQA process.

Besides the process aspect, this SQAP has additional sections to enforce the SQA functions for the product aspect. It sets out clear quality goals for the software product by stating the metrics and measurements for each quality aspect of the product.

Integrating the process aspect and product aspect together, this SQAP aims to serve as a comprehensive rulebook, guideline, reference and specification for the SQA processes throughout the whole SW project, which aids the achievement of more disciplined, well-planned, well-documented, thoroughly-implemented and soundly-validated SQA processes.

## **1.2 Scope**

For every stage of SDLC, SQA activities are specified and defined. The SW project is to be implemented using the Agile software development life cycle (SDLC). Figure 1 lists out the major phrases involved in the SDLC and these processes are iterative under Agile method.

The sections under process aspect of this plan follow the IEEE Std 730-1998, which is a standard for Software Quality Assurance Plans by the Institute of Electrical and Electronics Engineers (IEEE). In these sections, the SQAP describes the major SQA activities that should be carried out during the life cycle of the project. It also defines the practices of other major processes like documentation, test, code control, training and risk management.

The sections under product aspects follow the ISO/IEC 9126-1 Quality Model, which is a model for quality assurance by International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). In these sections, the SQAP sets out main quality goals for the product, provides metrics and measurements for each quality attribute.

The goal of this SQAP is to verify that all the SDLC processes, products, and non-product deliverables are able to meet requirements and achieve a certain level of quality. Thus, all processes, product and non-product deliverables should be examined, reviewed

or audited to ensure that they have closely follow the standards, practices and conventions specified in this document.



Figure 1: Major Activities in SDLC

## 1.3 Product Introduction

### 1.3.1 Description of Product

SC system is a social media application catering to the elderly, in order to satisfy their emotional, social, mental and learning needs, and to enrich their life. Specifically, it provides a platform to engage the elderly in social life by allowing chatting, sharing and joining events, and to help the elderly manage their life for daily activities. It will be designed and released as a mobile application so that elderly people can access to it anytime and anywhere. The design emphasizes on convenient voice channel, simplified GUI with enlarged fronts, accessible functionalities within 4 steps and security, to provide a reliable and effective social platform for the elderly.

The ultimate goal of this software is to assist elderly people to communicate with each other easily and to get involved in social activities actively, which will lead them to healthier life according to scientific materials.

### 1.3.2 Intended Market

SC system specifically targets at the elderly who wish to connect with their family and friends easily, and get involved in community social activities, in order to live a healthier life. The SC system embeds necessary design principles to reduce the learning cost for the elderly.

The SC system shall be first launched in Singapore market for test run. After gathering the market feedbacks, the project team may schedule for improvements. The subsequent launch would be in Asian market and then gradually penetrate into American and European markets. During the penetration, the project team shall adapt the system to specific needs of a particular market.

After the client-side software come to a mature phrase, the project team shall then consider the launching of the community manager software so that the community manager can create the group and upload events to SC on their own instead of going through the SC project team.

## 2 Reference documents

IEEE-SA Standards Board. (1998). *IEEE Standard for Software Quality Assurance Plans*

IEEE Computer Society. (2008). *IEEE Standard for Software Reviews and Audits*

*Post Project Review Questions.* (n.d.). Retrieved from naomistanford:  
[http://www.naomistanford.com/\\_files/live/PIRquestions.pdf](http://www.naomistanford.com/_files/live/PIRquestions.pdf)

*POST-IMPLEMENTATION PROJECT REVIEW REPORTS.* (n.d.). Retrieved from NaomiStanford: <http://www.employeeservices.gov.sk.ca/reviewchecklist>

Reyna, J. (n.d.). *SOFTWARE QUALITY ASSURANCE PLAN TEMPLATE*. San Diego.

Schneider, D. (n.d.). *Software Quality Assurance Plan*.

Software & Systems Engineering Standards Committee of the IEEE Computer Society. (2008). *International Standard Systems and software engineering — Software life cycle processes (2nd edition)*

Singh, B. (2010). *FORMAL TECHNICAL REVIEWS*. Retrieved from Soft Engineering: <http://soft-engineering.blogspot.sg/2010/12/formal-technical-reviews.html>

*Software Quality ISO Standards.* (n.d.). Retrieved from www.arisa.se: <http://www.arisa.se/compendium/node6.html>

Statz, J. (n.d.). *Requirements Review Checklist*. TeraQuest Metrics, Inc. Wikipedia. (n.d.). *ISO/IEC\_9126*. Retrieved from Wikipedia: [http://en.wikipedia.org/wiki/ISO/IEC\\_9126](http://en.wikipedia.org/wiki/ISO/IEC_9126)

The Gru Family. *SeniorConnect Proposal*.

Wikipedia. (n.d.). *Sample Project Quality Plan*. Retrieved from Wikipedia: <https://wiki.cac.washington.edu/display/pmportal/Sample+Project+Quality+Plan>

Wikipedia. (n.d.). *Capability Maturity Model Integration*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Capability\\_Maturity\\_Model\\_Integration](https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration)

## **3 Management**

This section will describe the organizational structure that influences and controls the quality of the software, especially in great details of major elements and interdependencies. Responsibilities will be allocated and recorded in a clear manner for future reference. Major quality assurance (QA) tasks will be explained in section 5.

### **3.1 Organization**

The organization structure of the project team plays an important role in organizing standardized and auditable quality assurance procedures. Team members should be assigned with responsibilities of SQA Tasks with special regards to their concurrent roles in the development team. Organizational dependencies and independencies are taken into consideration to ensure proper supervision for SQA process. The structure for this project is shown in Figure 2.

Project manager is in charge of overseeing and coordinating the whole software development and SQA processes. In order to ensure the quality of the delivered software, SQA team and the development team are kept separate and both report directly to the project manager. This allows the quality team to be completely independent of the development team thereby preventing any unexpected bias in the quality process.

The QA engineer is responsible for designing processes and carrying out tests to certify that the product system meets the requirements specified in the SRS.

The QA manager should interpret SQA test results, weight the severity of any issues found and coordinate solutions. Besides, the QA manager is in charge of quality management and ensures that all quality processes are followed. The QA manager will alert the project manager to any lapses and recommend corrective actions.

The lead engineer is responsible for coordinating the work of the front-end and back-end engineer and performing system integration.

The front-end engineer is responsible for the User Interface (UI) of the system while the back-end engineer focuses on implementing the server-end functions necessary for the system to work. Both engineers should collaborate together to establish the communication channel between front-end and back-end.

SQA team should also assess the quality of project processes carried out by the project manager. If the process quality is jeopardized, the SQA team should alert the project manager and suggest corrective enforcement of the project processes.

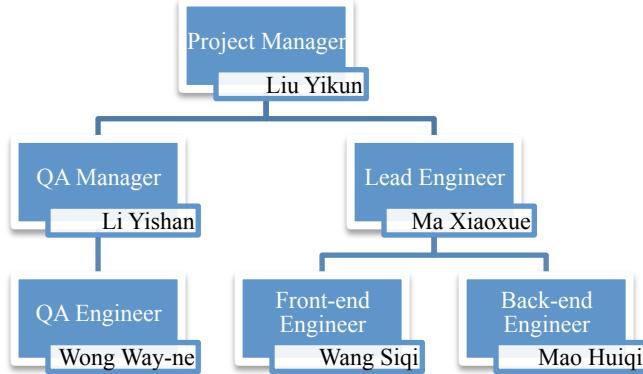


Figure 2 Project Team Organization Structure

### 3.2 Responsibilities Overview

A summary of the responsibilities of different roles regarding quality assurance for SC project is tabulated in Table 1 below.

Table 1 Roles and Responsibilities

Role	Quality Assurance Responsibility
Project Manager	<ul style="list-style-type: none"> <li>Review and Approve of the SQAP.</li> <li>Provide formal sign-off on all SQA deliverables, documents and processes.</li> <li>Review SQA issues to determine/assess impacts to overall project plan and project schedule.</li> <li>Enforce the SQA standards, practices and conventions specified in SQAP together with QA team.</li> <li>Resolve SQA issues and follow-up on any quality issues raised during the quality activities.</li> </ul>
Quality Assurance Team (QA Manager and QA Engineer)	<ul style="list-style-type: none"> <li>Identify the SQA activities that need to be performed for SC system at each stage of the SDLC.</li> <li>Prepare, develop and maintain quality and testing plans with reference to SQAP.</li> <li>Enforce the SQA standards, practices and conventions specified in SQAP together with project manager; monitor the implementation of SQA activities.</li> <li>Verify and audit SQA processes.</li> <li>Update the project manager with product/process quality assessment result.</li> <li>Follow-up and resolve SQA issues raised during reviews and audits.</li> <li>Facilitate weekly Quality Assurance meetings and maintains agenda.</li> </ul>
Development Team (Lead Engineer, Front-end Engineer and	<ul style="list-style-type: none"> <li>Review and provide feedback on SQAP.</li> <li>Participate in review meetings and provide feedback on the deliverables, documents and processes.</li> <li>Follow the standards, practices and conventions specified in SQAP in the development activities.</li> <li>Follow-up on and resolve any quality issues related to software design and</li> </ul>

Back-end Engineer)	development.
-----------------------	--------------

Table 1 provides a high level description of quality assurance responsibilities for each role within SC project team. The responsibilities for each role will be further clarified in section 5.11 of this SQAP, where the responsibilities are delegated and described at the task level.

## 4 Overview of Quality Assessment Tasks

This section serves as an overview of the quality assessment tasks to be performed by SQA team throughout the whole SDLC. It describes the general activities and techniques in the SQA. A more detailed definition of respective tasks within each SDLC stage is to be referred in section 5.

### 4.1 Reviews

Review is a proven effective technique for software quality assurance, especially in the areas of detecting design faults, code faults, and documentation faults. To ensure the effectiveness and efficiency of review, this section refers to IEEE Standard for Software Reviews and Audits (2008).

Throughout the project, five types of reviews are planned to assure standardized process and product quality. There are management review, technical reviews, Inspections, Walk-throughs, and audits.

Management reviews involve progress monitoring, schedule statuses checking, requirements and system allocation confirmation, and management approaches effectiveness evaluation. The review will be carried out every fortnight to ensure that the project direction is discussed by the management team, passed to the project team and that the technical team is able to meet their expectations. This will allow the project schedule to be kept realistic.

Technical reviews evaluate a software product systematically by examining the suitability of the software product for its intended use and identifying discrepancies from specifications and standards.

Major technical reviews will be performed at each milestone in the SDLC. These reviews will focus on the coding aspects of the project to validate that the product software meets both the functional and non-functional requirements written in SRS, as well as successful integration of the modules.

Minor technical reviews will be carried out at the completion of each sub module to verify the functional correctness to meet the expected requirements and system robustness to handle invalid user actions and other unexpected cases.

Inspections will conduct a systematic peer examination on the software product to detect and identify software anomalies, including errors and deviations from standards and specifications. Inspections will be carried out to resolve anomalies and inspect product quality.

Walk-through is a static analysis technique in which a designer or programmer educates audience regarding a software product. The audience can be members of the development team and other interested parties. The audience may ask questions and make comments about possible anomalies, violation of development standards, and other problems. Walk-through will be carried out to examine alternatives and to improve product.

Audit provides independent evaluation from a third party to access product conformance to regulations, standards, and plans. Audits will be carried to conform to objective standards and regulations

Before reviews, review information including objectives will be notified together with distribution of materials. Responsibilities will be assigned. During reviews, participants will focus on detecting anomalies, and file them in review form. An anomaly checklist can be prepared beforehand to ensure a complete review. Review minutes will be kept for further reference. After review, the detected anomalies will follow a standardized process for investigation, resolution and re-review.

The process for reviews of different types and different objectives are explained in the sequence of occurrences in software development life cycle in Section 5 QA Tasks and Section 6 Standards, Practices, Conventions and Metrics.

## 4.2 Tracking

Project tracking should be done periodically throughout the whole project lifecycle by regular progress inspections and product reviews.

Issue tracking should be done systematically. At the time of discovery, issues will be recorded in review form and classified according to the critical levels of being critical, severe, moderate or minor for prioritization. The criteria for deciding critical levels are explained in Table 2. After that, resolution tasks are added to backlog in the sequence of prioritization. Along the process, the issue status of being discovered, outstanding, or resolved is traceable via issue report. The purpose of this is to ensure that any issues found will be kept in records, and resolved in a timely manner. Besides, the overall number of occurrences of anomalies and issues is also indicative of the performance of the technical team and can be a reference input for management review.

Table 2 Issue Critical Level Classifications

Issue Classification	
Critical	Defects that may cause the system to hang, crash, produce incorrect results or behavior, or corrupt user data. No known alternatives.
Severe	Defects that cause incorrect results or behavior with known alternatives. Large and/or important areas of the system are affected.
Moderate	Defects that affect limited areas of functionality that can either be worked around or ignored.
Minor	Defects that can be overlooked with no loss of functionality.

## 4.3 Estimated Resources

### 4.3.1 Facilities and Requirements

SC QA team will have access to computer resources needed to perform the SQA activities specified in this SQAP, e.g. project deliverable testing and document reviews. SC QA team will also have access to the venues needed to hold review and re-review meetings.

### 4.3.2 Personnel

The estimated personnel, time and effort needed to complete the SQA activities defined in this SQAP are analyzed in details in the project plan. QA team members are required to have reasonable knowledge on the SDLC, the related software design and development activities in order to conduct SQA effectively.

## 5 QA Tasks

### 5.1 Task: Review Software Products

Project plan lists out the software products to be developed and the schedules for developing them. SQA team should help to identify the standards, general practices and conventions to be followed during the development process. The details regarding standards, practices and conventions shall be discussed in Section 7. The review and verification process for intermediate and final software products are specified in the following part of this section, Section 8. Section 6 Documentation specifies the minimum documentation requirements for the software products.

### 5.2 Task: Evaluate Software Products Review Process

SQA team needs to ensure that adequate review and verification processes are in place for the software products and the processes. SC SQA team has developed a relatively

complete and sound review and verification scheme for the project, under which the issues and errors can be uncovered, tracked and resolved.

For critical software deliverables and development processes, major technical reviews and verifications will be conducted in each stage of the SDLC. These reviews need to be verified and audited by SC SQA team, and then approved by the project manager. There are also other deliverable reviews planned in between the major technical reviews. The technical review process, related measurements and metrics are defined in later parts of this section, section 5.8.1 and 5.8.2.

For other processes, such as project management and configuration management, general reviews shall be conducted periodically. Some guidelines for management reviews are provided in section 5.8.4. Suggested processes for general audits are defined in section 8.

### **5.3 Task: Evaluate Project Planning, Tracking and Oversight Processes**

Project planning, tracking and oversight are the main activities of project management.

Project planning should be done upstream at the beginning of the project. The plans and schedules for development, SQA, testing and system configuration are documented in the project plan. Project tracking and oversight should be done periodically throughout the whole project lifecycle by regular progress inspections and product reviews.

SC SQA team should review the project plan to ensure it covers all the major products and activities involved in SC project. The detailed review process is defined in Section 5.8 with reference to guidelines in Section 8.2.

SC SAQ should also conduct periodical management reviews to monitor project tracking and oversight processes. The management review process is defined in section 5.8.4 and can be done with the help of Management Review Form template (Section 5.13). The results and findings of the review should be documented in the form and reported directly to the project manager. All the problems/issues raised should be tracked and resolved.

### **5.4 Task: Evaluate Software Requirements Analysis Process**

Software requirement analysis mainly involves discovering, itemizing, formulating and documenting the software related requirements. Not only functional requirements, other software requirements like non-functional requirements, interface requirements and user requirements all needed to be considered and factored into the software design. All these requirements together serve as the basis of downstream development activities.

Requirement analysis is not only done during the initial phase. Requirements are dynamic. Requirement management needs to be done throughout the project timeline to effectively deal with requirement changes.

To ensure the requirements are well managed, SC SQA team need to check following aspects:

- a. Documentation
- b. Personnel
- c. Review and Verification

The detailed evaluation criteria are listed in the Software Requirement Analysis Process Audit Checklist (Section 5.14), which can be used by SC SQA team to conduct the evaluation. The result of the evaluation shall be documented using the General Process Audit Form in Section 5.14. Minimum documentation requirement for software requirement analysis process is defined in Section 6.2.

## **5.5 Task: Evaluate Software Design Process**

High-level design activities are conducted after the requirement analysis phase to conceptually divide and capture requirements into different models. The requirements are re-represented in software engineering language and can be communicated across different designers.

Low-level design activities are conducted after high level design are reviewed and stabilized. The functions in low-level designs should establish one-to-one correspondence with a specific requirement. This kind of correspondence should be well documented so it's traceable and can be understood by every developer. Low-level design documents should be detailed enough so that it's designer or developer independent.

SC SQA team should evaluate software design process from the following aspects:

- a. Documentation
- b. Consistency and Traceability
- c. Review and Verification

The detailed evaluation criteria are listed in the Software Design Process Audit Checklist (Section 5.14), which can be used by SC SQA team to conduct the evaluation. The result of the evaluation shall be documented using the General Process Audit Form in Section 5.14. Minimum documentation requirement for software design process is defined in Section 6.2.

## **5.6 Task: Evaluate Software Implementation and Unit Testing Process**

After the software design processes, the designs are implemented using selected language and platform. The small software modules, also called units, are tested individually with

the test cases specified in the test plan before they are integrated together. SC SQA team should evaluate the process for implementation and unit testing as two different aspects.

SC SQA team may use the Software Implementation and Unit Testing Process Audit Checklist (Section 5.14) as a guide for conducting the evaluation. The result of the evaluation shall be documented using the General Process Audit Form in Section 5.14.

## **5.7 Task: Evaluate Unit Integration and Testing**

Software integration and testing are conducted after the unit testing, when it is verified that the individual units can function correctly by themselves. The task of integration and testing is to combine the pieces to form complete software and to make it functions as specified in the requirements.

Unlike the unit testing which focuses on the testing of internal functioning within a single unit, testing after integration focuses more on the inter-unit functioning, meaning how the units interact with each other to perform the desired functions.

SC SQA team may use the Software Unit Integration and Testing Process Audit Checklist (Section 5.14) as a guide for conducting the evaluation. The result of the evaluation shall be documented using the General Process Audit Form in Section 5.14.

## **5.8 Task: Conduct and Verify Project Reviews**

The project reviews and audits are closely tied with the deliverables and processes in each SDLC stage. They are the most direct control over the quality of the product as well as the processes. Enforce, monitor and track these review and audit processes are one of the main tasks for SC SQA team.

For reviews, both technical reviews and management reviews will be held within the project team. The technical reviews shall mainly focus on the inspection, walk-through and feedback on the software product itself. The technical reviews are the critical checkpoints for product quality at each SDLC stage. They are to ensure that the software product is consistent with the requirements and meet the specified standard at the end of each period. Management reviews shall mainly focus on the verification and reflection on the project management practices. They ensure that correct management activities are carried out as planned at each stage, trying to spot management problems and improve the management practice.

For the verifications, both verifications for technical reviews and management reviews shall be carried out periodically, preferably not longer after the review, to make sure that the review processes are carried out in a standard, well-documented and planned manner.

Table 3 shows the major technical, managerial reviews, audits and other deliverable reviews associated with each SDLC stage. Other deliverable reviews are the reviews for important deliverables, such as SQAP and project plan. They are not classified under

technical reviews since they are not strictly focused on the software product. The detailed practices for these reviews shall be further discussed in the following sections.

Table 3 Reviews associated with SDLC Stage

SDLC Stage	Reviews and Audits
Project Planning	SQAP Review CMMI Process Definition Review Software Configuration Plan Review Test Plan Review Risk Management Plan Review Project Plan Review Management Review Process Audit
Requirement Elicitation and Analysis	Use Case Model Review Software Requirements Review Management Review Process Audit
Design	Class Diagram & Sequence Diagram Review Software Design Review Management Review Process Audit
Implementation	Code Review Prototype Review Release Plan Review Test Readiness Review Management Review Process Audit
Launch and Maintenance	Post-Project Review Management Review Process Audit

### 5.8.1 Task: Conduct Technical Reviews

The technical reviews shall mainly focus on the inspection, walk-through and feedback on the software product itself. As listed by Xicron Technology, there are several objectives for a formal technical review (Xicron Technology, 2010):

- a. To verify that the software product is consistent with the requirements;

- b. To ensure that the software has met the specified standard (especially standard in SQAP);
- c. To reveal errors in software design, implementation and testing;
- d. To provide periodical assessment that makes project more manageable

The technical review materials shall be prepared by the development team. Once the materials are ready, the development team should communicate with SQA team to arrange the detailed review meeting time. SQA team shall prepare the review minutes, related forms and checklists. The review meetings should be held by SQA team and be supervised by the project manager. The Review Meeting Forms need to be signed by the SQA team members. SC SQA team directly reports to project manager regarding the review results. The major reviews to be held within SC project team are further described in following part of this section.

For the four reviews described in the previous tables, the documentation templates/measurements needed for each meeting can be found in the Section 5.13. The review forms are mainly used for documenting the errors found or issues raised during the review meetings. These review forms are designed by SC SQA team to cater the needs for SQA activities during the review meetings.

The checklists are mainly used as a reference or guideline to verify that the software product under review meets standards and follows general software engineering conventions. These review forms are designed according to related IEEE/ISO standards. They may also be from good industrial practices. SC SQA team has combined some existing checklist templates to form a template for the use of SC project team.

### **5.8.2 Task: Verify Technical Reviews**

All major technical reviews will need to be verified and audited by the project manager to make sure that the reviews have been carried out with due care. The manager needs to verify the following aspects:

- a. The review is held according to planned schedule.
- b. All the intermediate reviews that are planned to be ahead of the major technical review have been conducted.
- c. All defined processes for the review have been carried out in appropriate manner
- d. The carried out processes are consistent with the specification in SQAP.
- e. All required related documents, including Review Meeting Minutes, Review Form and Review Checklist, have been completed; all the issues/errors uncovered are documented.
- f. All the issues/errors uncovered during the review has been followed up and resolved. The Technical Review Audit Form (Section 5.13) can be used as a guideline during the verification process. The audit form needs to be signed by the project manager.

### 5.8.3 Task: Conduct Other Deliverable Reviews

Other than the formal technical reviews, other deliverable reviews are also scheduled to serve as additional checkpoints to closely monitor the software product quality and the progress of the processes. Unlike formal technical reviews, the additional deliverable reviews are not milestone reviews for the SDLC stages, thus verifications are not required for these reviews. These reviews can be moderated by the QA team, but still need to be supervised by the project manager. During the verification of the milestone review for each SDLC stage, it needs to be verified that intermediate deliverable reviews during that stage has been completed. For example, during the verification process for Test Readiness Review, SQA team needs to verify that the Prototype Review has already been completed. The additional deliverable reviews scheduled are described in the following part of this section.

Only processes of selected deliverables reviews are shown here. Other deliverable review should follow similar procedures. Other deliverable reviews are added and scheduled as required.

SQAP Review	
Objectives	Ensure that the SQA framework has covered all the major product and process areas for SC project  Determine whether the processes and product measurements defined in SQAP are enforceable, traceable and verifiable
Roles & Responsibilities	Reader: QA Engineer  Author: QA Team  Inspector: Development Team & Project Manager  Recorder: 3 Team members  Moderator: QA Manager
Input	SQAP  CMMI Process Definition
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the document  Fill in SQAP Review Form (Section 5.13)
Exit Criteria	The inspection of the whole document is finished  Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13)  SQAP Review Form

Use Case Model Review	
Objectives	Ensure the consistency of high-level design with the requirements  Determine the adequacy of the model representation
Roles & Responsibilities	Reader: QA Engineer  Author: QA Engineer  Inspector: Development Team & Project Manager  Recorder: 3 Team members  Moderator: QA Manager
Input	Use Case Diagram  Use Case Description
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the document  Fill in Use Case Model Review Form (Section 5.13)
Exit Criteria	The inspection of the whole document is finished  Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13)  Use Case Model Review Form

Code Review	
Objectives	Ensure that the coding practices has been carried out according to basic coding principles  Ensure the implementation follows the designs  Examine the adequacy of design, coding and implementation
Roles & Responsibilities	Reader: Lead Developer  Author: Development Team  Inspector: QA Team & Project Manager  Recorder: 3 Team members  Moderator: Project Manager
Input	Implementation Code  Test Cases
Entry Criteria	The input documents are completed. Only implementation code for the prototype is required, not full implementation code.

Procedures & Measurements	Walk-through and inspect the code according to the checklist Fill in Code Review Form (Section 5.13) Fill in the Code Review Checklist (Section 5.13) Conduct sanitary tests (only for major functions)
Exit Criteria	The inspection of the whole document is finished Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13) Code Review Form Code Review Checklist

Prototype Review	
Objectives	Ensure the consistency of high-level design with the requirements Ensure the consistency of lower-level design with the requirements
Roles & Responsibilities	Reader: Lead Developer Author: Development Team Inspector: QA Team & Project Manager Recorder: 3 Team members Moderator: QA Manager
Input	Hi-fi Prototype
Entry Criteria	The input documents are completed. Only implementation code for the prototype is required, not full implementation code.
Procedures & Measurements	Walk-through and inspect the code according to the checklist Fill in Prototype Review Form (Section 5.13)
Exit Criteria	The inspection of the whole document is finished Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13) Prototype Review Form

#### **5.8.4 Task: Verify Management Reviews**

SC SQA team should periodically audit management review processes to ensure that they have been conducted in accordance to corresponding standards and practices in project plan and SQAP. The project manager should examine following aspects during a management review:

- a. Project status/ progress
- b. Issues identified
- c. Risk analysis

The project manager can use the Management Review Form template (Section 5.13) in this SQAP to document the review findings. Detailed sections under each aspect are shown in the review form. The SQA team should use the Process Audit Form in Section 5.14 to audit management review process and report to the project manager.

#### **5.8.5 Task: Conduct Process Audits**

Software development processes are carried out according to the schedule specified in the project plan. The development processes are also periodically reviewed and audited as specified in SQAP section 5 (this section).

#### **5.8.6 Task: Conduct Configuration Audits**

SC SQA team should also participate in both Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA).

During FCA, the built software is examined to determine that whether it covers and satisfies all the requirements specified in SRS. If the software covers other requirements that are not specified in the SRS, the software's compliance to the additional requirements should not hinder its compliance to the requirements stated in the SRS.

During PCA, the built software is examined to determine whether its documentations are consistent with what is being built.

### **5.9 Task: Audit Software Configuration Process**

SC SQA team should conduct status accounting and configuration audit regularly. SQA team should examine the integrity of current status of configuration items and evaluate the configuration status records of each configuration item against change requests. Any non-compliance issues should be documented and conveyed to relevant personnel.

Detailed process and documents are defined in Software Configuration and Change Management Plan.

## **5.10 Task: Verify Software Quality Assurance**

The project manager should assess the SQA activities periodically, independent from the SQA team members. The project manager should assess both the reviews and audit conducted by the SQA team. The following aspects should be examined:

- a. Whether the reviews and audits have been carried out in accordance to the processes definition and schedule in SQAP and project plan
- b. Whether the review and audit results have been well-documented according to the specification in SQAP
- c. Whether the issues/errors uncovered during reviews have been tracked to closure
- d. Whether re-review is conducted when needed
- e. Whether the problems discovered during audits that require corrective action have been processed in accordance to the specification in SQAP

The project manager should provide feedback and suggestions to SQA team after the assessments to point out the areas of improvement.

## **5.11 QA Task Responsibilities**

As described previously in section 4, SC SQA team should conduct the SQA activities in accordance to the definition and specification in this SQAP, including regular management reviews.

The project manager should periodically review the SQA processes, including reviews and audits conducted; ultimately responsible for the quality of SC project and related non-deliverable products, including documentations.

The development team should complete the intermediate and final software products according to planned schedule and prepare them for review. The development team should also actively participate in review meetings to present their efforts and give feedbacks.

QA manager is responsible for ensuring timely completion and auditable compliance regarding the quality process and the documentation as part of the development process. Besides, QA manager is also in charge of internal review teams and reports to the project manager with all findings for hazardous anomalies and major improvements.

QA engineer is in charge of technical reviews to assure technical soundness and identify discrepancies from approved standards. This includes review items according to checklist. Technical review should follow the review descriptions mentioned in Section 5.8.1 and Section 5.8.2. Besides, QA engineer is in charge of maintaining the backlog and reporting summarized important information to the QA manager, e.g. information about severe issues found.

## **5.12 QA Schedule**

Table 3 shows the major reviews and audits that are scheduled in each stage of the SDLC.

At the completion of each stage of the SDLC, the intermediate reviews should have already been conduct. The SQA team shall then conduct all the major technical review and relate audit process for this stage. The detailed schedules for the SQA activities are stated in the project plan.

## **5.13 QA Review Forms and Checklists**

### **5.13.1 Review Meeting Minutes**

Review Meeting Minutes			
General Information			
Review Title			
Review Date			
Review Meeting Start Time		Review Meeting End Time	
Recorder			
Attendee			
Summary			
Details			
	Decision	Action By	Deadline
1			
2			
3			
4			
5			
Related Documents			

	Document Name	Document ID
1		
2		

### 5.13.2 Review Form Template

Review Form Template						
Record						
ID	Section / Page / Use Case... No. <sup>1</sup>	Description of Error / Issue	Classification (C, S, M, M') <sup>2</sup>	Recommended Action (A, M, D) <sup>3</sup>	Recommended Action Description	Re-review Status (S, NS) <sup>4</sup>
1						
2						
3						
4						
5						
Follow-up						
ID	Action Item			Assignee	Deadline	
1						
2						
Re-review						
Re-review Date <sup>5</sup>						
Re-review Summary <sup>6</sup>						

Checked by:	Approved by:		
Assignee	Signature: _____	Project Manager	Signature: _____

QA Engineer      Signature: \_\_\_\_\_      QA Manager      Signature: \_\_\_\_\_

\*Note:

1. The Section/ Page/ Use Case... No. records the location of the error/issue within the document under review. Depending on the document under review, the denotation may be different.
  - 1) Using Section No. -- SQAP Review/ Project Plan Review/ Software Requirement Review
  - 2) Using Use Case No. -- Use Case Model Review
  - 3) Using Page Name -- Prototype Review/ Software Design Review (for GUI)
  - 4) Using Class Name -- Prototype Review/ Software Design Review (for implementation code)
  - 5) Using Test Case No. -- Test Readiness Review
  - 6) Using Document ID + Section No. -- Post-Project Review
2. Issue Classification

C – Critical

S – Severe

M – Moderate

M' – Minor
3. Recommended Action

A – Add

M – Modify

D – Delete
4. Re-review Status: the status of this error/issue during re-review, to be filled on during re-review

S – Solved

NS – Not solved
5. Re-reviewed Date: to be decided during review meeting
6. Re-review Summary: to be filled in during re-review

### 5.13.3 Software Requirement Review Checklist

ID	Y/N	Aspect to check	Comments
Minimum Requirements			
1		Has PM and SQA, reviewed and signed off SRS?	
2		Are there any conflicts in the requirements?	
3		Is any requirement ambiguous or not clear?	
4		Testability: Is each requirement testable and verifiable?	
5		Does the SRS cover Non-functional requirements like - performance, usability, safety, reliability, security and privacy and statutory and regulatory norms?	
6		Are the interfaces with other software/hardware products fully specified in the SRS?	
Organization and Structure of the Documentation			
7		Have appropriate requirements documentation standards been followed?	
8		Are all figures, tables, and diagrams labelled and referenced?	
9		Are all requirements written at a consistent and appropriate level of detail?	
10		Do the requirements provide an adequate basis for design and system test?	
Completeness and Correctness			
11		Are all internal cross-references to other requirements correct? [For modifiability, minimize cross-references.]	
12		Does each functional requirement specify input and output, as well as function, as appropriate?	
13		Are user documentation and training requirements addressed?	
14		Are all reliability, recoverability (business continuity), and performance requirements properly specified?	

15		Are all security requirements properly specified?	
16		Have all quality attributes (characteristics) been properly specified (i.e. efficiency, flexibility, interoperability, maintainability, portability, reusability, usability, availability)	
17		Have the human interface requirements been addressed? Are they correct?	
18		Are all external hardware, software, and communication interfaces defined? Are they correct?	
Consistency and Clarity			
19		Are the requirements free of duplication and conflict with other requirements?	
20		Is each requirement written in consistent, clear, concise language?	
21		Is each requirement verifiable by testing, demonstration, review, or analysis?	
Traceability			
22		Is each requirement uniquely and correctly identified?	
Special Issues			
23		Are all requirements actually requirements, not design or implementation solutions?	

**Checked by:**

QA Engineer      Signature: \_\_\_\_\_

**Approved by:**

QA Manager      Signature: \_\_\_\_\_

#### 5.13.4 Software Design Review Checklist

ID	Y/N	Aspect to check	Comments
General Design			
1		Does the design support both product and project goals?	
2		Is the design feasible from a technology, cost, and schedule standpoint?	
3		Is any requirement ambiguous or not clear?	
4		Testability: Is each requirement testable and verifiable?	
5		Does the SRS cover Non-functional requirements like - performance, usability, safety, reliability, security and privacy and statutory and regulatory norms?	
6		Are the interfaces with other software/hardware products fully specified in the SRS?	
Organization and Structure of the Documentation			
7		Have appropriate requirements documentation standards been followed?	
8		Are all figures, tables, and diagrams labelled and referenced?	
9		Are all requirements written at a consistent and appropriate level of detail?	
10		Do the requirements provide an adequate basis for design and system test?	
Completeness and Correctness			
11		Are all internal cross-references to other requirements correct? [For modifiability, minimize cross-references.]	
12		Does each functional requirement specify input and output, as well as function, as appropriate?	
13		Are user documentation and training requirements addressed?	
14		Are all reliability, recoverability (business continuity), and performance requirements properly specified?	

15		Are all security requirements properly specified?	
16		Have all quality attributes (characteristics) been properly specified (i.e. efficiency, flexibility, interoperability, maintainability, portability, reusability, usability, availability)	
17		Have the human interface requirements been addressed? Are they correct?	
18		Are all external hardware, software, and communication interfaces defined? Are they correct?	
Consistency and Clarity			
19		Are the requirements free of duplication and conflict with other requirements?	
20		Is each requirement written in consistent, clear, concise language?	
21		Is each requirement verifiable by testing, demonstration, review, or analysis?	
Traceability			
22		Is each requirement uniquely and correctly identified?	
Special Issues			
23		Are all requirements actually requirements, not design or implementation solutions?	

**Checked by:**

QA Engineer      Signature: \_\_\_\_\_

**Approved by:**

QA Manager      Signature: \_\_\_\_\_

### 5.13.5 Test Readiness Review Checklist

Note: Adapted from Test Readiness Assessment Checklist (Texas Department of Information Resources)

ID	Y/N	Aspect to check	Comments
1		Is the Test Plan comprehensive and complete?	
2		Are the test scenarios, data, procedures, descriptions, and other test information comprehensive and complete?	
3		Have test plan, test scenarios, procedures, descriptions, and other test information been reviewed and approved by the designated approvers?	
4		Was the prerequisite test phase completed successfully?	
5		If the prerequisite test phase was not completely successful, have workarounds or contingency plans for proceeding with testing been identified?	
6		Was the build verification test on the release to be tested executed successfully?	
7		Has the entry criteria for the test phase to be initiated been met?	
8		Is the test data available?	
9		Have software, documentation, data, and other appropriate configuration items been placed under configuration control?	
10		Have testing resources participated in test plan reviews, training or orientation?	
11		Are there risks associated with initiating the test phase?	
12		Are the scheduled resources available to execute and support the test phase activities?	
13		If there are risks associated with initiating the test phase, are there plans to accept or mitigate those risks?	

Checked by:

Approved by:

QA Engineer      Signature: \_\_\_\_\_      QA Manager      Signature: \_\_\_\_\_

### **5.13.6 Post-Project Review Checklist**

\* Note: Adapted from Post-implementation Project Review Report (Government of Saskatchewan). This reference document isn't really a template of post project review for a software project. It broadly defines the post project review process for general projects. However, it does offer good examples on how general reviews should be done.

ID	Y/N	Aspect to check	Comments
<b>Cost Aspect</b>			
1		Does the total project expenditure agree with the amount that was included in the project plan?	
2		If there are discrepancies, do the discrepancies appear reasonable?	
3		If any discrepancies, what caused the discrepancies?	
<b>Schedule Aspect</b>			
4		Are there any situations where significant delays in the completion of the project were experienced?  What are the reasons and/or factors that caused the delays?  What are the impacts of those delays?	
5		If there were delays, what were the steps taken during the course of the project to address delays? Summarize their effectiveness.	
6		If there were delays, which stage in the SDLC did the delay occur?	
<b>Risk Aspect</b>			
7		For each risk that occurred, consider the following:  Was the risk identified in the project plan or SQAP for the project? What was the impact?  Are mitigation steps taken reasonable and effective?	
<b>Tools</b>			
8		Are the software tools used for this project appropriate?	

		How effective were these tools?	
9		How effectively have the project team utilized the tools?	

<b>Checked by:</b>	<b>Approved by:</b>
QA Engineer      Signature: _____	QA Manager      Signature: _____

### 5.13.7 Technical Review Audit Form

Technical Review Audit Form			
General Information			
Auditor			
Date of Audit			
Date of Report			
Review Process Audited			
Audit Checklist			
	Y/N	Audit Aspect	Comment
1		The review is held according to planned schedule.	
2		All the intermediate reviews that are planned to be ahead of the major technical review have been conducted.	
3		All defined processes for the review have been carried out in appropriate manner	
4		The carried out processes are consistent with the specification in SQAP.	

5	All required related documents, including Review Meeting Minutes, Review Form and Review Checklist, have been completed; all the issues/errors uncovered are documented.		
6	All the issues/errors uncovered during the review has been followed up and resolved.		
<b>Audit Findings</b>			
<input type="checkbox"/> Review Process Acceptable <input type="checkbox"/> Review Process Conditionally Acceptable (Subject to satisfactory completion of action items listed below) <input type="checkbox"/> Review Process Unacceptable			
<b>Follow-up (if any)</b>			
	Action Item	Assignee	Deadline
1			
2			
<b>Corrective Action</b>			
<b>Disposition</b>		<i>APPROVE</i>	<i>CANCEL</i>
			<i>DEFER</i>

<b>Checked by:</b>	<b>Approved by:</b>
QA Engineer    Signature: _____	QA Manager    Signature: _____
	Project Manager    Signature: _____

### **5.13.8 Management Review Form**

<b>Management Review Form</b>		
General Information		
Review Objective		
Review Date		
Project Status/Progress		
	Area to examine	Comment
1	What SDLC stage is the project currently in?	
2	Referring to the project schedule in project plan, have all planned activities been completed so far?	
3	Are there any delays in the project schedule? What caused the delays?	
4	If any, what are the plans to catch up?	
Issues Identified		
	Description	Comment
1		
2		
3		
Risk Analysis		
	Risk Description	Comment
1		
2		
3		

Follow-up (if any)			
	Action Item	Assignee	Deadline
1			
2			
Corrective Action			
Disposition	<i>APPROVE</i>	<i>CANCEL</i>	<i>DEFER</i>

Checked by:	Approved by:
QA Manager      Signature: _____	Project Manager      Signature: _____

### 5.13.9 Code Review Checklist

\* Note: Adapted from Code Review Check List For Java Language (*by Hung Nguyen*). This code review should be used for Java code review for Android development. For the implementation for iOS which is in Objective-C, the checklist should be similar to this.

ID	Y/N	Aspect to check	Comments
Deviation Objective			
1		Does the code correctly implement the design?	
2		Does the code implement more than the design?	
3		Is every parameter of every method passing mechanism (value or reference) appropriate?	
4		Does every method return the correct value at every method return point?	
5		Are there any requirements of design that were not implemented?	

Defect Objective			
6		Do all attributes have appropriate access modifiers (private, protected, public)?	
7		Are there static attributes that should be non-static or vice-versa?	
8		Do all methods have appropriate access modifiers (private, protected, public)?	
9		Is every method parameter value checked before being used?	
10		Are there static methods that should be non-static or vice-versa?	
11		Do any subclasses have common members that should be in the superclass?	
12		Can the class inheritance hierarchy be simplified?	
13		Does the code avoid comparing floating-point numbers for equality?	
14		Is every three-way branch (less, equal, greater) covered?	
15		Will all loops terminate?	
16		When there are multiple exits from a loop, is each exit necessary and handled properly?	
17		17 Does each switch statement have a default case?	
18		Are all I/O exceptions handled in a reasonable way?	
19		Is there a low level of coupling between modules (methods and classes)?	
20		Is there a high level of cohesion within each module (methods or class)?	
21		Is there repetitive code that could be replaced by a call to a method that provides the behavior of the repetitive code?	
22		Are the Java class libraries used where and when appropriate?	

23		Are arrays large enough?	
24		Are object and array references set to null once the object or array is no longer needed?	
25		Can better data structures or more efficient algorithms be used?	
26		Are logical tests arranged such that the often successful and inexpensive tests precede the more pensive and less frequently successful tests?	
Inconsistency and Ambiguity Objective			
27		Are there any code implemented in inconsistent way?	
28		Are there variables with confusingly similar names?	
29		Are all variables properly defined with meaningful, consistent, and clear names?	
30		Are any modules excessively complex and 30 should be restructured or split into multiple routines?	
Redundancy Objective			
31		Can any code be replaced by calls to external reusable objects?	
32		Are there any blocks of repeated code that could be condensed into a single method?	
33		Are there any leftover stubs or test routines in the code?	

<b>Checked by:</b>	<b>Approved by:</b>
QA Engineer      Signature: _____	QA Manager      Signature: _____

## 5.14 QA Audit Checklists and Forms

### 5.14.1 Software Requirement Analysis Process Audit Checklist

\* Note: Adapted from *Software Requirement Analysis Process Audit Checklist (SQA\_plan\_template)*

ID	Y/N	Aspect to Audit	Comments
Documentation			
1		Software requirements are documented in a SRS or other approved format and are traceable.	
2		The SRS is maintained under configuration management.	
3		The SRS changes undergo a peer review before they are incorporated into the requirements baseline.	
4		Software development plans, work products, and activities are changed to be consistent with changes to the software requirements.	
Personnel			
5		Software engineering group is trained to perform requirements management activities.	
Review and Verification			
6		Peer review validates testability of requirements and that appropriate metrics are established to validate measurable performance requirements.	
7		Requirement analysis and associated requirements reviews are conducted in accordance with the process definition in SQAP and project plan.	
8		Action items resulting from reviews of the software requirements analysis are resolved.	

<b>Checked by:</b>	<b>Approved by:</b>	
QA Engineer	Signature: _____	QA Manager
	Signature: _____	

### 5.14.2 Software Design Process Audit Checklist

\* Note: Adapted from *Software Requirement Analysis Process Audit Checklist (SQA\_plan\_template)*

ID	Y/N	Aspect to Audit	Comments
Documentation			
1		Software design document and software test plan are documented and are traceable.	
2		Software design document and software test plan are maintained under configuration management.	
3		Changes must go through peer review. Changes to the software design are identified, reviewed, documented, and tracked to closure.	
4		Methods, such as the Software Development File (SDF) or Unit Development folder (UDF), used for tracking and documenting the development of a software unit is implemented and is kept current.	
Consistency and Traceability			
5		Software designs are consistent based on approved software requirement changes.	
Review and Verification			
6		Software design activities and associated design reviews are conducted in accordance with the process definition in SQAP and project plan.	
7		Action items resulting from reviews of the software designs are	

		resolved.	
--	--	-----------	--

<b>Checked by:</b>	<b>Approved by:</b>
QA Engineer      Signature: _____	QA Manager      Signature: _____

### **5.14.3 Software Implementation and Unit Testing Process Audit Checklist**

\* Note: Adapted from *Software Requirement Analysis Process Audit Checklist (SQA\_plan\_template)*

ID	Y/N	Aspect to Audit	Comments
Implementation			
1		Code is maintained under configuration management.	
2		Changes to code are identified, reviewed, and tracked to closure.	
3		Methods used for tracking and documenting the development of a software unit is implemented and is kept current.	
Unit testing			
4		Ensure that passing criteria for unit test is documented, and that compliance has been recorded.	
5		Results of unit testing are documented.	
Review and Verification			
6		Coding process, associated code reviews, and software unit testing are conducted in accordance with the process definition in SQAP and project plan.	
7		Action items resulting from reviews of the software designs are resolved.	

<b>Checked by:</b>	<b>Approved by:</b>	
QA Engineer	Signature: _____	QA Manager
	Signature: _____	

#### **5.14.4 Software Unit Integration and Testing Process Audit Checklist**

\* Note: Adapted from *Software Requirement Analysis Process Audit Checklist (SQA\_plan\_template)*

ID	Y/N	Aspect to Audit	Comments
1		Software test activities are identified, test environments have been defined, and guidelines for testing have been designed.	
2		A plan for the integration of the CIs exists, which specifies the order and schedule in which the CIs are integrated.	
3		The integrated CIs have completed unit testing. Any required corrections have been completed.	
4		The CIs have been retested.	
5		Test procedures are defined for CI integration.	
6		Test cases are defined.	
7		Test pass/fail criteria are defined.	
8		The test results are documented.	
9		Software integration process, software integration testing activities and the software performance testing activities are conducted in accordance with the process definition in SQAP and project plan.	
10		Action items resulting from reviews of the software designs are resolved.	
11		The software unit's design, code, and test are updated based on the results of software integration testing, software performance testing, and corrective action process.	

<b>Checked by:</b>	<b>Approved by:</b>
QA Engineer      Signature: _____	QA Manager      Signature: _____

### 5.14.5 Process Audit Form

<b>Process Audit Form</b>			
General Information			
Auditor			
Date of Report			
Date of Audit			
Process/ Procedure Audited			
Issues			
	Classification	Issue Description	Comment
1			
2			
3			
4			
Audit Findings			
<input type="checkbox"/> Process/Procedure Acceptable <input type="checkbox"/> Process/Procedure Conditionally Acceptable (Subject to satisfactory completion of action items listed below) <input type="checkbox"/> Process/Procedure Unacceptable			
Follow-up (if any)			
	Action Item	Assignee	Deadline

1			
2			
Corrective Action			
Disposition	<i>APPROVE</i>	<i>CANCEL</i>	<i>DEFER</i>

<b>Checked by:</b>	<b>Approved by:</b>		
QA Engineer	Signature: _____	QA Manager	Signature: _____
		Project Manager	Signature: _____

## 6 Documentation

### 6.1 Purpose

The documentations that are created during the lifecycle of SW project for all purposes should be maintained, tracked and kept to date. In the light of IEEE Std 730-1998 for Software Quality Assurance Plans, the documentation section includes:

- a) Identify the documentation governing the development, verification and validation, use, and maintenance of the software.
- b) State how the documents are to be checked for adequacy with respect to Section 8.

### 6.2 Minimum Documentation Requirements & Guidelines

The below documentation is required as a minimum to ensure that the implementation of the software satisfies requirements. The documentation is developed along every stage of SDLC and follows respective standards.

Table 4 displays documents categorized according to SDLC stages with industrial standards and guidelines. Documents should be drafted according to the structure and requirements mentioned in the standards, finalized with respect to project specifics and approved by the project management.

Table 4 Minimum Documentation Requirements & Guidelines

Minimum Documentation Requirements & Guidelines		
SDLC Stage	Document Name	Standard, Guideline

Project Planning	Software Project Management Plan	IEEE 1058-1998
	Software Quality Assurance Plan	IEEE Std 730-1998
	CMMI Process Definition	CMMI Version 1.3
	Risk Management Plan	IEEE Std 1540
	Software Test Plan	IEEE Std 829
	Configuration Management Plan	IEEE Std 828-1998
	Change Management Plan	IEEE Std 828-1998
Requirement Analysis	Use Case Model	IEEE Std 830-1998
	System Requirement Specification	IEEE Std 830-1998
Design	Software Design Description	IEEE Std 1016-1998
	Conceptual Model	ISO/IEC/IEEE 42010
	User Interface Design	Shneiderman's "Eight Golden Rules of Interface Design"
	Entity Relationship Model	Integration DEFinition for Information Modeling (IDEF1X)
Implementation & Testing	Test Report	IEEE Std 829
Launch & Maintenance	User Documentation	IEEE Std 1063

### 6.3 Check for Adequacy

This section describes how the adequacy of the documentations shall be checked.

- a. Every document should keep a Record of Change at the beginning of the document, which clearly states the following of a specific change: version number, changed date, action performed, brief description of the change and any documents related to this change.
- b. For general documents, the review and audit process shall follow the description in section 5.8.3. The review result should be documented using Review Form Template (Section 5.13). The review process for a general document can be verified during the audit for major technical reviews or can be conducted independently using the Process Audit Form found in section 8. After the document has been reviewed and the review process has been audited, the adequacy of the documentation can be deemed checked.
- c. For the major technical documentations, the document should be reviewed in accordance to section 5.8.1. The review process should be audited in accordance to

section 4.8.2. After the document has been reviewed and the review process has been audited, the adequacy of the documentation can be deemed checked.

## 7 Standards, Practices, Conventions, and Metrics

### 7.1 Purpose

Concerned with ensuring the required level of quality in the software product, appropriate quality standards, practices and conventions ought to be defined and stated so that they can be followed and complied in the software development life cycles. Well-defined quality standards and procedures also play a necessary role in developing a ‘quality culture’ where quality is seen as everyone’s responsibility.

It is vital to deliver a product whose qualities and the processes carried out are widely recognized. Thus, existing industrial standards and practices should be referred when applicable to ensure a wide acceptance and recognition.

This section shall

- . Identify the standards, practices, conventions, and metrics to be applied
- . State how compliance with these items is to be monitored and assured

### 7.2 Content

This SC project mainly refers to IEEE/EIA Standard 12207 Series, an international standard that establishes a common framework for software life cycle process, with well-defined terminology. Any reference should be stated within the project plan.

In this SQAP, the software quality assurance activities for the various activities during SDLC are defined in section 5 SC SQA team shall follow the definitions and ensure that the quality standards are met and processes are carried out as planned in the project plan.

Standards for logic structure, coding, and code comments are defined in the project plan. SC SQA team shall verify the source code adhering to these stated standards. Details for code control are defined in section 12 Standards and practices for testing are defined in the project plan. SC SQA team shall verify that the coding activities are carried out as specified in the project plan. The major reviews and audits planned for implementation phase are defined in section 4.8. These reviews and audits aim to ensure the quality of coding practices.

For the definition of the major software development activities, SC project follows the Capability Maturity Model (CMM). SC project thrives to achieve level 2 of CMM. The detailed process definition, related measurements and metrics can be found in CMM Process Definition document. SC SQA is responsible for reporting these measurements to the project manager periodically.

## **8 Reviews and Audits**

### **8.1 Purpose**

Review is a commonly used method for effective verification against standards and requirements. Reviews are especially efficient in detecting faults in design, code and documentation.

As one of the most useful reviews, audit provides independent evaluation of product conformance to regulations, standards and plans.

Reviews and audits shall be conducted and at least meet the minimum requirements. The processes shall be followed to achieve a great verification effect.

This section shall

- . Define the technical and managerial reviews and audits to be conducted;
- . State how the reviews and audits are to be accomplished;
- . State what further actions are required and how they are to be implemented and verified.

### **8.2 Minimum requirements**

Minimum requirements span the software development life cycle to ensure sufficient quality assurance.

The minimum is that the reviews and audits in 8.2.1 through 8.2.10 shall be conducted.

#### **8.2.1 Software Requirements Review (SRR)**

The SRR is held to ensure the adequacy of the requirements stated in the SRS.

The requirements shall be verified via reviews considering the criteria listed below:

The system requirements are consistent, feasible, and testable.

The system requirements have been appropriately allocated to hardware items, software items, and manual operations according to design criteria.

The software requirements are consistent, feasible, testable, and accurately reflect system requirements.

The software requirements related to safety, security, and criticality are correct as shown by suitably rigorous methods.

## **8.2.2 Preliminary Design Review (PDR)**

The PDR (also known as the top-level design review) is held to evaluate the technical adequacy of the preliminary design (also known as the top-level design) of the software as depicted in the preliminary software design description.

The PDR shall be verified considering the criteria listed below:

- . The design is correct and consistent with and traceable to requirements.
- . The design implements safety, security, and other critical requirements correctly as shown by suitably rigorous methods.

## **8.2.3 Critical Design Review (CDR)**

The CDR (also known as detailed design review) is held to determine the acceptability of the detailed software designs as depicted in the detailed software design description in satisfying the requirements of the SRS.

The CDR shall be verified considering the criteria listed below:

- . The design implements proper sequence of events, inputs, outputs, interfaces, logic flow, allocation of timing and sizing budgets, and error definition, isolation, and recovery.
- . Selected design can be derived from requirements.

## **8.2.4 Functional audit**

This audit is held prior to the software delivery to verify that all requirements specified in the SRS have been met.

Functional audits shall be conducted to ensure that:

- . Software products were successfully tested and meet their specifications in SRS.
- . Test data comply with the specification.
- . Test reports are correct and discrepancies between actual and expected results have been resolved.
- . Activities have been conducted according to applicable requirements.

## **8.2.5 Physical audit**

This audit is held to verify that the software and its documentation are internally consistent and are ready for delivery.

Physical audits shall be conducted to ensure that:

- . As coded, software products (such as a software item) reflect the design documentation.

- . The acceptance review and testing requirements prescribed by the documentation are adequate for the acceptance of the software products.
- . User documentation complies with standards as specified.
- . Activities have been conducted according to applicable plans, and contract.
- . The costs and schedules adhere to the established plans.

### **8.2.6 In-process audits**

This audit is held to verify consistency of the design.

In-process audits shall be conducted to ensure that:

- . Coding plans and patterns (such as a software item) reflect the design documentation.
- . Hardware and software interfaces implemented meet required Interface specifications.
- . Design implementations fulfill functional requirements.
- . Test descriptions match with functional requirements.

### **8.2.7 Managerial reviews**

Managerial reviews are held periodically to assess the execution of all of the actions and the items identified in the SQAP.

Project status shall be evaluated relative to the applicable project plans, schedules, standards, and guidelines. The outcome of the review should be considered by appropriate management and should provide for the following:

- . Making activities progress according to plan, based on an evaluation of the activity or software product status.
- . Maintaining global control of the project through adequate allocation of resources.
- . Changing project direction or determining the need for alternate planning.
- . Evaluating and managing the risk issues that may jeopardize the success of the project.

### **8.2.8 Software Configuration Management Plan Review (SCMPR)**

The Software Configuration Management (SCM) focuses on identifying and controlling major software changes. SCM should limit the impact changes may have on the entire system by eliminating unnecessary changes, and monitoring any necessary changes to ensure proper implementation. Changes should be reported to any other personnel or clients who may have an interest.

Therefore, SCMPR should consist of the following criteria.

- . Configuration identification identifies configurations, configuration items and baselines.
- . Configuration control implements a controlled change process. This is usually achieved by setting up a change control board whose primary function is to approve or reject all change requests that are sent against any baseline.
- . Configuration status accounting records and reports all the necessary information on the status of the development process.
- . Configuration auditing ensures that configurations contain all their intended parts and are sound with respect to their specifying documents, including requirements, architectural specifications and user manuals.
- . Build management manages the process and tools used for builds.
- . Process management ensures adherence to the organization's development process.
- . Environment management manages the software and hardware that host the system.
- . Teamwork facilitates team interactions related to the process.
- . Defect tracking makes sure every defect has traceability back to the source.

### **8.2.9 Post-mortem review**

This review is held at the conclusion of the project to assess the development activities implemented on that project and to provide recommendations for appropriate actions.

This is an essential step in evaluating and learning from the project. The short-term benefit of a project post-mortem is that it defines for the customer and the project team exactly what has been achieved by the investment of money, time and sanity into the project.

The project post-mortem should discuss the following questions.

- . What went right?
- . What went wrong?
- . What should we do differently next time?

These questions should be applied to every area of project management: from planning to delivery, from teamwork and communication to risk and change management, from stakeholder input to support staff efficiency.

- . The different areas can be broken down through the following rules.
- . Project stages or phases (e.g. determining need, project planning, creating deliverables)
- . Project processes (e.g. planning, risk management, configuration management)
- . Project roles (e.g. support staff, change manager, project board)
- . Key skill areas (e.g. communication, teamwork, management, organization)
- . Products (e.g. budget, schedule, plans, reports, post-project review questionnaire)

### 8.3 Review Plans & Standards

This section presents the standards for various reviews, which are derived and summarized from *IEEE Standard for Software Reviews and Audits* (2008).

Table 5 Project Plan Review

Project Plan Review	
Objectives	<ul style="list-style-type: none"><li>- Ensure that the project plan has covered all the major activities for SC project</li><li>- Examine the adequacy of the planned timeline and estimated resources</li><li>- Determine whether the steps defined in project plan are executable</li></ul>
Roles & Responsibilities	Reader: Project Manager Author: Project Manager Inspector: Development Team & QA Team Recorder: 3 Team members Moderator: QA Manager
Input	Project Plan
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the document Fill in Project Plan Review Form (Section 5.13)
Exit Criteria	The inspection of the whole document is finished Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13) Project Plan Review Form

Table 6 Software Requirements Review

Software Requirements Review	
Objectives	Examine the adequacy of the generated software requirements and developer's effort  Ensure that SRS completely and accurately documents the software requirements and is able to act as a basis for subsequent design activities
Roles & Responsibilities	Reader: QA Engineer  Author: QA Manager & Engineer  Inspector: Development Team & Project Manager  Recorder: 3 Team members  Moderator: Project Manager
Input	System Requirement Specification (including both software and hardware requirements)  Use Case Model
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the document  Fill in System Requirement Specification Review Form (Section 5.13)  Go through System Requirements Specification Review Checklist (Section 5.13))
Exit Criteria	The inspection of the whole document is finished  Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13)  System Requirement Specification Review Form  System Requirement Specification Review Checklist

Table 7 Software Design Review

Software Design Review	
Objectives	Evaluate the consistency of software high-level design with the software requirements Evaluate the adequacy of high-level design
Roles & Responsibilities	Reader: Lead Developer Author: Development Team Inspector: QA Team & Project Manager Recorder: 3 Team members Moderator: Project Manager
Input	SRS Use Case Model ER Diagram Conceptual Model
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the document Fill in Software Design Review Form (Section 5.13) Go through Software Design Review Checklist (Section 5.13)
Exit Criteria	The inspection of the whole document is finished Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13) Software Design Review Form Software Design Review Checklist

Table 8 Test Readiness Review

Test Readiness Review	
Objectives	Determine whether the test procedures are complete and the product is fully tested
Roles & Responsibilities	Reader: QA Manager Author: QA Team Inspector: Development Team & Project Manager Recorder: 3 Team members Moderator: Project Manager
Input	Implementation Code Test Plans and Test Cases Test Results
Entry Criteria	The input documents are completed
Procedures & Measurements	Walk-through and inspect the test cases and test results Fill in Test Readiness Review Form (Section 5.13) Go through Test Readiness Review Checklist (Section 5.13)
Exit Criteria	The inspection of the whole document is finished Issues identified are documented in the review form and resolved later
Output	Review Meeting Minutes (Section 5.13) Test Readiness Review Form Test Readiness Review Checklist

Table 9 Post-Project Review

Post-Project Review	
Objectives	Reflect on the whole life cycle of the project Think of possible improvements on both product and process aspects
Roles & Responsibilities	Reader: Project Manager Author: Project Manager Inspector: Entire Project Team Recorder: 3 Team members Moderator: Project Manager
Input	Documentations Implementation Codes
Entry Criteria	The input documents are completed
Procedures & Measurements	Hold discussions and reflections held by the project manager Fill in Post-Project Review Form (Section 5.13) Go through Post-Project Review Checklist (Section 5.13)
Exit Criteria	Reflections and possible improvements are documented Project Manager dismiss the meeting
Output	Review Meeting Minutes (Section 5.13) Post-Project Review Form Post-Project Review Checklist

## 8.4 Other

Other reviews and audits may include the user documentation review (UDR). This review is held to evaluate the adequacy (e.g., completeness, clarity, correctness, and usability) of user documentation.

Project manager should organize other reviews and audits extensively to ensure quality assurance.

## 9 Test

The testing activities of SC project mainly consist of unit testing, system testing and integration testing. Before entering into each testing phase, SC SQA team need to prepare the test cases according to the test plan stated in project plan. Figure 3 shows how the testing shall be conducted. The figure displays a typical unit testing process flow. System test flow shall be defined closely according to different system structure and flows. Figure 4 shows a typical integration testing process flow.

The tasks for SQA team members during the testing phase are:

- . Ensure the readiness of starting and conducting a test phase
- . Perform the testing with designed test cases.
- . Verify that the testing results are documented and complied with the configuration management processes defined in CMM process definition and project plan.
- . Coordinate the maintenance of Software Trouble Report.
- . Evaluate the testing processes in accordance to relevant processes in section 4.
- . Report the audit findings to the project manager
- . Witness the regression testing to verify that the errors found by unit and integration testing are corrected.

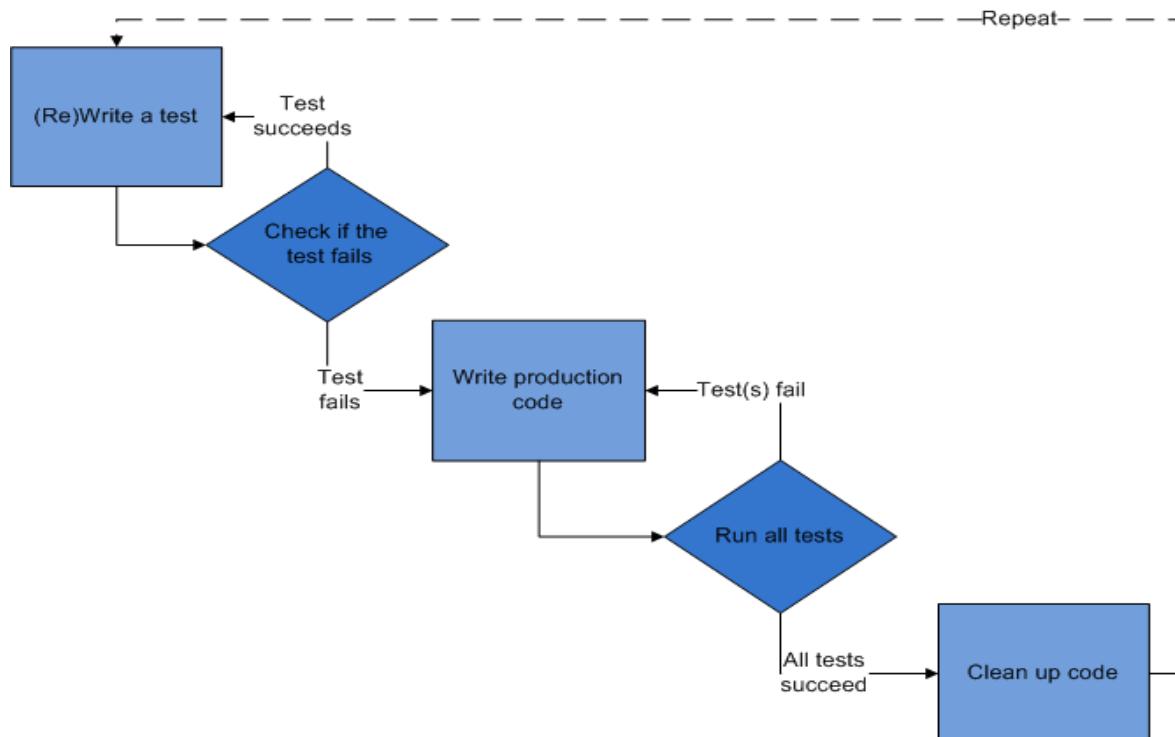


Figure 3 Unit Test Flow

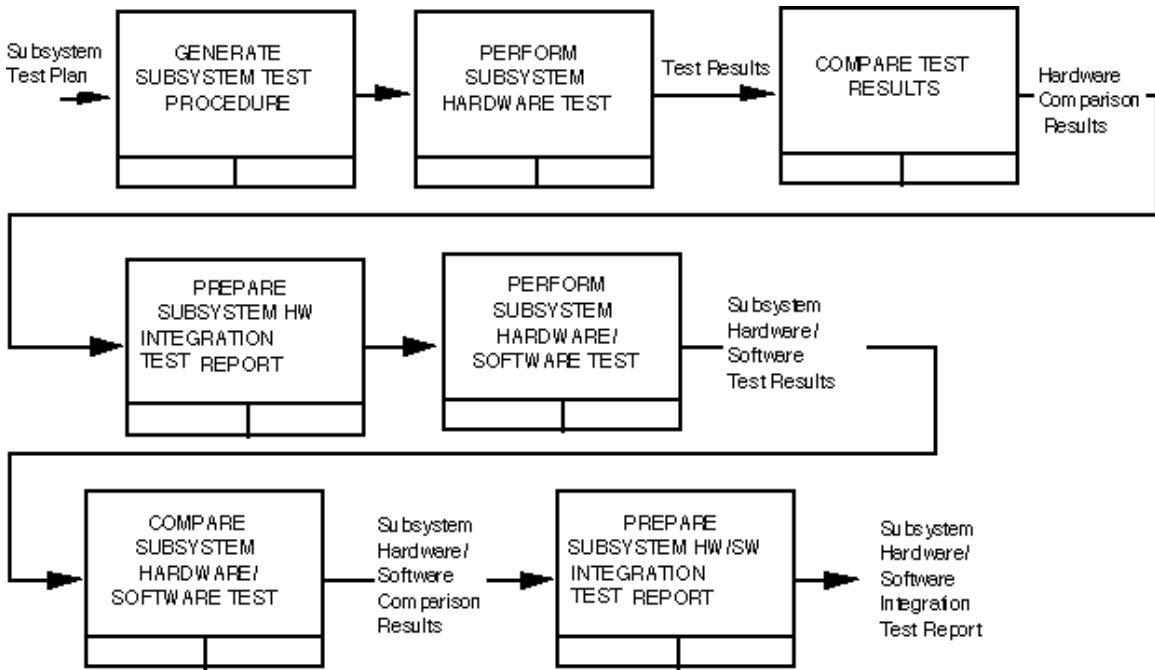


Figure 4 Integration Test Flow

## 10 Problem reporting and corrective action

This section describes the reporting and control practices and procedures exercised by SC SQA team to examine processes, record audit results, report audit results and track the raised issues to closure.

This section shall

- . Describe the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance process;
- . State the specific organizational responsibilities concerned with their implementation.

### 10.1 Process Audit Report

SC SQA team conducts the process audits and reports the audit result to the project manager using the Process Audit Report. Specially, for the audit of technical review processes, special Process Audit Report is used – the Technical Review Audit Form which explicitly provide an audit checklist to aid the audit process. A process shall be classified into one of the following categories after audit:

- . Being followed correctly and conducted effectively.
- . Being followed but is not conducted effectively.
- . NOT being followed.

The auditor may suggest corrective actions to rectify the process. The project manager need to sign the Process Audit Form, follows up the issue raised and resolve the issue. The template for Process Audit Form can be found in *Appendix III-E*.

### **10.1.1 Submittal and Disposition of Process Audit Report**

After the Process Audit Report being submitted to the project manager, the manager should:

- . Indicate whether to approve, cancel or defer the suggestions given by the auditor. Since SC project manager is ultimately responsible for the quality of products and processes, the project manager shall direct the final disposition for the corrective actions.
- . Analyze the result and examine whether the processes has complied with the standards, practices and conventions stated in SQAP and project plan and been conducted effectively. If necessary, the project manager shall enforce the planned process/procedure or initiate actions to rectify the situation.

### **10.1.2 Escalation Procedure for Resolution of Non-Concurrence on Process Audit Report**

In the event that there are disputes over the results and suggestions in the Process Audit Form, SC SQA team shall discuss with the affected personnel to resolve the disputes. If the disputes remain unsolved, the project manager shall direct the final disposition. Disregarding the disposition, the original report should be retained.

## **10.2 Recording Problems in Software Code or Documentation**

Issues and problems uncovered in the audit process (for both code and documentation) should be documented in STR. SC SQA team member should examine the pattern of the issues and problems, identify weak processes or teams and suggest preventive solutions to the project manager.

## **11 Tools, techniques, and methodologies**

This section shall identify the special software tools, techniques, and methodologies that support SQA, state their purposes, and describe their use.

## **12 Code Control**

It is the LD's responsibility to assure the correct handling of the code and all other configuration items due to the standards described in SCMP.

Code control includes the items listed below:

- a. Identify, label, and catalogue the software to be controlled
- b. Identify the physical location of the software under control
- c. Identify the location, maintenance, and use of backup copies
- d. Distribute copies of the code
- e. Identify the documentation that is affected by a change
- f. Establish a new version
- g. Regulate user access to the code.

The following has to be valid:

- Documents are available to all people who are authorized to access them and to no one else.
- All versions of a document are available.
- No file is unnecessarily locked.
- Naming conventions are consequentially used.

SC project uses SVN for code version control. SC SQA team should review the code control requirements and regularly audit the code control processes to ensure that they are carried out properly in accordance to specified standards and practices. This is done by reviews and random checks. Problems are reported to the LD and PM.

## **13 Media control**

This section shall state the methods and facilities to be used to

- a) Identify the media for each computer product and the documentation required to store the media, including the copy and restore process; and
- b) Protect computer program physical media from unauthorized access or inadvertent damage or degradation during all phases of the software life cycle.

This may be provided as a part of the SCMP. If so, an appropriate reference shall be made thereto.

Media control has both software and hardware aspects:

- a. Software
  1. Maintenance of the inventoried media filed in storage place according to security requirements
  2. Control and monitor access to those files
  3. Prevent media degradation or damage
  4. Schedule backups
- b. Hardware

1. Prevent unauthorized access
2. Regular physical maintenance
  - a. Regularly scheduled backup of the media.

SC SQA team should review the security requirements and regularly audit the media control processes to ensure that they are carried out in accordance to specified standards and practices.

## **14 Supplier control**

This section shall state the provisions for assuring that software provided by suppliers meets established requirements. In addition, this section shall state the methods that will be used to assure that the software supplier receives adequate and complete requirements. For previously developed software, this section shall state the methods to be used to assure the suitability of the product for use with the software items covered by the SQAP. For software that is to be developed, the supplier shall be required to prepare and implement an SQAP in accordance with this standard. This section shall also state the methods to be employed to assure that the developers comply with the requirements of this standard.

All external software components in the program code, that have an unreliable source, will be tested according to the IEEE 9126 standards. Software components that have reliable sources will undergo some quick tests. These tests will be focused on the parts of this software that are of importance to the project. Whether an external software component is reliable or not is to be decided by the developer that wants to use the specific component and at least one other member of the SC project team that wants to use the component.

The software tools that will be used for development of the program code (such as Ionic) are available to all project members.

## **15 Records collection, maintenance, and retention**

This section shall identify the SQA documentation to be retained; shall state the methods and facilities to be used to assemble, safeguard, and maintain this documentation; and shall designate the retention period.

SQA activities should be documented using respective forms and reports, such as minutes of meeting and review forms, maintained throughout the lifecycle of SC project. All the documentation should be stored in a central depository for easy control and management. The documents are to be added after all SQA members have approved them. Respective documents are to be delivered within 3 working days after the SQA activity has been done.

For SC project, the central depository is realized by SVN and Wiki. The upload and change of any SQA related documents should be coordinated by the PM and QA manager.

All the review and audit information shall be collected, examined to study for the pattern and device preventive solutions.

## 16 Training

This section shall identify the training activities necessary to meet the needs of the SQAP.

This section identifies the training activities planned to be conducted to meet the need of this SQAP. There are various skills required for performing SQA. However, considering the available resources, timeline and budget for SC project, only the training activities tabulated in the following table shall be conducted. Other required skills such as testing methodology, configuration management and risk management process shall be selflearned by the SQA team.

Table 10 Training Activities Overview

Skills	Training Type	Source
Peer Reviews	Classroom	Advanced Software Engineering Lectures
Software Development and Documentation standards and guidelines		
Project Management		
Review Techniques		
SQA Management	Consulting and Sharing	Experienced QA Manager

## 17 Risk Management

This section shall specify the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle covered by the SQAP.

There are two aspect of risk management that shall be discussed in this section:

- a. The review and audit processes that the SQA team should follow to ensure the risk management activities for SC project are carried out in accordance to the risk management plan and project plan.
- b. The identification and analysis of the potential risk involved in SQA processes/procedures.

## **17.1 SQA for Project Risk Management**

To assure the risk management activities, SC SQA team should actively monitor, review, audit and report the risk management activities. The methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle, which shall be enforced by the SQAP, are identified and defined by the project plan.

The responsibilities of SC SQA team:

- a. Review risk analysis documents and any risk avoidance and reduction plan.
- b. Periodically audit the risk management processes to ensure that they are carried out in accordance to the risk management plans.
- c. Report the audit result to project manager and suggest corrective actions.
- d. Ensure that the issues/errors raised are resolved and tracked to closure.
- e. Document the successful/failed risk management cases.

The responsibilities of project manager:

- a. Examine whether the risk management process has been carried out with due care according to the report of SQA team.
- b. Direct disposition regarding the suggested corrective action.
- c. Initiate enforcement action if necessary. Direct resources and time wisely and make appropriate decisions to mitigate risk impact when risk cannot be avoided.

## **17.2 Risk Management for SQA Activities**

There are various potential risks involved in SQA activities themselves. For example, the development team may fail to deliver functionally complete software at the scheduled delivery time, thus delaying the scheduled technical review. SC SQA team should adopt a proactive approach in risk management. By doing so, the risk involved can be controlled and mitigated to a lower level, reducing the potential timeline delays and additional costs.

To achieve proactive risk management, SC SQA team should identify the potential risks, actively monitor the risk level, periodically review risk factors, establish backup plans upfront and act agilely when the risk cannot be avoided.

The following part of this section identifies the potential risks involved in SQA activities.

There are three types of impact severity: low, moderate and high. Low severity risks do not require special management attention. They will not result in significant timeline delays and additional costs. Moderate severity risks may require certain level of management attention. Corrective action might be required since they may negatively affect the schedule and cost aspect. High severity risks require high level of management

attention. Corrective action against them should be enforced as soon as possible to avoid significant adverse impact.

The risks identified are listed in the following table.

Table 11 Identified Risk List

Risk Description	Assurance processes defined in SQAP not feasible
Impact Severity	High
Probability	Moderate
Impact	The infeasible processes cannot be implemented. SQA scheme is not complete. Some processes/procedures actually don't have SQA associated with them. This may lead to poor product/process quality.
Corrective action	<ol style="list-style-type: none"> <li>1. Reference existing industrial standards and good practices when drafting SQAP.</li> <li>2. Ensure that SQAP is reviewed by the whole project team.</li> <li>3. Submit SQAP to project manager for checking and approval.</li> </ol>
Risk Description	Standard, practices and conventions stated in SQAP not followed or carried out effectively
Impact Severity	High
Probability	Moderate
Impact	Poorly conducted SQA may lead to unsatisfactory product/process quality.
Corrective action	<ol style="list-style-type: none"> <li>1. Educate the project team regarding the importance of SQA.</li> <li>2. Build a SQA team that is independent from other project sub-teams.</li> <li>3. Project manager should periodically audit SQA processes.</li> </ol>
Risk Description	Delayed review or audit due to delay in reviewed

	or audited process/procedure
Impact Severity	High
Probability	High
Impact	Due to the delays, the time planned for SQA processes may be reduced to meet the final delivery deadline. Poorly conducted SQA may lead to unsatisfactory product/process quality.
Corrective action	<ol style="list-style-type: none"> <li>1. Monitor the progress of the project closely and assess the project status frequently. Spot any possible delays early and plan in advance.</li> <li>2. Launch backup plans if necessary.</li> <li>3. If no backup plans, act agilely when risk cannot be avoided. In this case, SQA team should report to project manager and develop a contingency SQA plan.</li> </ol>
Risk Description	Approved corrective action are not taken or implemented
Impact Severity	High
Probability	Low
Impact	If the approved corrective action is not implemented, the issues and errors uncovered by the process audit remain unresolved and jeopardize the product/process quality. The process audit does not meet its purpose.
Corrective action	<ol style="list-style-type: none"> <li>1. Maintain good documentation for all audits; track uncovered issues and errors to closure.</li> <li>2. Conduct re-reviews.</li> <li>3. Discuss with the action assignee to resolve this problem.</li> <li>4. If cannot be resolved after step 3, report to project manager.</li> </ol>
Risk Description	Standard, practices and conventions stated in SQAP not followed or carried out effectively
Impact Severity	High

Probability	Moderate
Impact	Poorly conducted SQA may lead to unsatisfactory product/process quality.
Corrective action	<ol style="list-style-type: none"> <li>1. Educate the project team regarding the importance of SQA.</li> <li>2. Build a SQA team that is independent from other project sub-teams.</li> <li>3. Project manager should periodically audit SQA processes.</li> </ol>