

School of Computer Science and Engineering

CZ3002 Advanced Software Engineering

**Software Quality Assurance (SQA) Plan**

Project Name: HangOut

****

Group Name: Mac & Cheese

Lab Group: TDDP1

| **Group Members** |
| --- |
| Jethro Phuah An Ping (Team Leader) |
| Lam Zhi Fah (Deputy Team Leader) |
| Alicia Chua Jieying |
| Ernest Ang Cheng Han |
| Ong Jing Heng Shaun |
| Ong Sim Hao |
| Shannon Tan Xinyi |

Submitted to:

Dr Althea Liang

Department of School of Computer Science and Engineering, Nanyang Technological University

**Signature Page**

**Prepared by:** Jethro Phuah An Pin **Date: September 2021**

Lam Zhi Fah **Date: September 2021**

Alicia Chua Jieying **Date: September 2021**

Ernest Ang Cheng Han **Date: September 2021**

Ong Jing Heng Shaun **Date: September 2021**

Ong Sim Hao **Date: September 2021**

SHannon Tan Xinyi **Date: September 2021**

**Reviewed by 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 <(Reviewer Name)>**

**Reviewed by 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 <(Reviewer Name)>**

**Approved by : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 (Name)**

# **Document Change Record**

| **Revision** | **Description of Change** | **Approved**  **by** | **Date** |
| --- | --- | --- | --- |
| **1.0** | **Initial Template** | **Ong Sim Hao** | **01/09/2021** |
| **1.1** | **Purpose, Scope, Reference Document and Management** | **Lam Zhi Fah** | **01/09/2021** |
| **1.2** | **Task, Product and Process Assessment** | **Ong Jing Heng Shaun** | **02/09/2021** |
| **1.3** | **Updated Process Assessment** | **Ong Jing Heng Shaun** | **02/09/2021** |
| **1.4** | **Roles and Responsibility, and Document Requirement** | **Lam Zhi Fah** | **03/09/2021** |
| **1.5** | **Standards, Practices, Conventions and Metrics** | **Ong Sim Hao** | **03/09/2021** |
| **1.6** | **Software Reviews** | **Lam Zhi Fah** | **04/09/2021** |
| **1.7** | **Test** | **Ong Sim Hao** | **04/09/2021** |
| **1.8** | **Problem Reporting and Corrective Actions** | **Ong Jing Heng Shaun** | **04/09/2021** |
| **1.9** | **Tool, Techniques and Methodologies** | **Ong Jing Heng Shaun** | **05/09/2021** |
| **2.0** | **Media Control** | **Lam Zhi Fah** | **05/09/2021** |
| **2.1** | **Record Collection, Maintenance and Retention** | **Ong Sim Hao** | **06/09/2021** |
| **2.2** | **Training** | **Ong Sim Hao** | **06/09/2021** |
| **2.3** | **Risk Management** | **Ong Jing Heng Shaun** | **07/09/2021** |
| **2.4** | **SQA Plan Change Procedure and History** | **Lam Zhi Fah** | **08/09/2021** |
| **2.5** | **Revised Quality Plan** | **Ong Jing Heng Shaun** | **08/09/2021** |
| **2.6** | **Finalised SQA** | **Zhi Fah, Sim Hao, Shaun** | **10/09/2021** |
|  |  |  |  |
|  |  |  |  |

# **Contents**

[1.](#_heading=h.gjdgxs) Purpose and Scope 7

[1.1.](#_heading=h.30j0zll) Purpose 7

[1.2.](#_heading=h.1fob9te) Scope 7

[2.](#_heading=h.3znysh7) Reference Documents 7

[3.](#_heading=h.tyjcwt) Management 8

[3.1.](#_heading=h.3dy6vkm) Management Organisation 8

[3.1.1.](#_heading=h.1t3h5sf) Project Management 8

[3.1.2.](#_heading=h.4d34og8) Assurance Management 8

[3.2.](#_heading=h.2s8eyo1) Tasks 8

[3.2.1.](#_heading=h.17dp8vu) Product Assessments 8

[3.2.2.](#_heading=h.3rdcrjn) Process Assessments 11

[3.3.](#_heading=h.26in1rg) Roles and Responsibilities 14

[3.3.1.](#_heading=h.lnxbz9) QAM 14

[3.3.2.](#_heading=h.35nkun2) Software Quality Personnel 14

[4.](#_heading=h.1ksv4uv) Documents 15

[4.1.](#_heading=h.44sinio) Purpose 15

[4.2.](#_heading=h.2jxsxqh) Minimum Document Requirements 15

[5.](#_heading=h.z337ya) Standards, Practices, Conventions and Metrics 15

[5.1.](#_heading=h.3j2qqm3) Purpose 15

[5.1.1. Standards](#_heading=h.4i7ojhp) 15

[5.1.1. Pract](#_heading=h.4i7ojhp)ices 16

[5.1.1.](#_heading=h.4i7ojhp) Conventions 16

[5.2.](#_heading=h.1y810tw) Software Quality Programme 16

[5.2.1.](#_heading=h.4i7ojhp) Standard Metrics 17

[6.](#_heading=h.2xcytpi) Software Reviews 17

[6.1.](#_heading=h.1ci93xb) Purpose 17

[6.2.](#_heading=h.3whwml4) Minimum Software Reviews 17

[7.](#_heading=h.2bn6wsx) Test 18

[8.](#_heading=h.qsh70q) Problem Reporting and Corrective Action 19

8[.1.](#_heading=h.1ci93xb) Possible problems which may arise 19

8[.2.](#_heading=h.3whwml4) Problem reporting procedure 19

[9.](#_heading=h.3as4poj) Tools, Techniques and Methodologies 20

[9.1.](#_heading=h.32hioqz) Software Quality Tools 20

[10.](#_heading=h.1hmsyys) Media Control 20

[11.](#_heading=h.147n2zr) Record Collection, Maintenance, and Retention 21

[12.](#_heading=h.3o7alnk) Training 21

[13.](#_heading=h.23ckvvd) Risk Management 22

[14.](#_heading=h.ihv636) SQA Plan Change Procedure and History 23

# **Purpose and Scope**

## ***Purpose***

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the HangOut project.

The Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the QAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software. The systematic monitoring of products, processes, and services will be evaluated to ensure they meet requirements and comply with policies, standards, and procedures, as well as applicable Institute of Electrical and Electronic Engineers (IEEE) and ISO standards.

## ***Scope***

The purpose of SQA is to ensure that the software developed does not deviate from the original intended product. SQA is also concerned to identify any errors, omissions, inconsistencies, and alternatives, enhancements or improvements that can be made at any stage of development.

The scope of the plan highlights the various methodologies and acceptance standards which will be documented and strictly adhered to in order to ensure the quality of the development of Hangout and that it does not differ from the original intended product. This plan will:

1. Outline the responsibilities of the Project Manager, Quality Assurance Manager and Lead Developers in the Quality assurance team.
2. Assure that software development, evaluation and acceptance standards are developed, documented and followed closely.
3. Ensure that the results of software quality reviews and audits are reported to the appropriate management in the Quality assurance team.

This ensures that the Quality assurance team will be able to confidently identify any errors and inconsistencies, and gather useful feedback that can aid in making improvements in accordance to development standards during the development stage of the project.

# **Reference Documents**

* IEEE STD 730-2002, IEEE Standard for Software Quality Assurance Plans (<http://standards.ieee.org/reading/ieee/std_public/description/se/730-2002_desc.html>)
* ISO IEC 90003:2004 Software Standard (<http://praxiom.com/iso-90003.htm>)
* Project Plan
* System Requirement Specifications

# **Management**

This section describes the management organizational structure, its roles and responsibilities, and the software quality tasks to be performed.

## ***Management Organisation***

The implementation of the quality assurance system is the responsibility of the Quality Assurance Manager (QAM).

### **Project Management**

The Project Manager will be responsible for approving:-

* The system requirement specification document
* The overall time scale for the project
* The choice of system development life cycle
* The choice of software development tools and techniques utilised
* The selection of project teams
* The training of project teams

### **Assurance Management**

The QAM provides Project Management with visibility into the processes being used by the software development teams and the quality of the products being built. The QAM maintains a level of independence from the project and the software developers.

In support of software quality assurance activities, the QAM has assigned and secured Software Quality personnel from the pool of available SQ trainees to coordinate and conduct the SQ activities for the project and report back results and issues.

## ***Tasks***

This section summarizes the tasks (product and process assessments) to be performed during the development of software. These tasks are selected based on the developer’s Project Plan and planned deliverables, and identified reviews.

### **Product Assessments**

The following product assessments will be conducted by SQ personnel:

| **Product Assessments 001** | **Requirements Of System** |
| --- | --- |
| **Entry Criteria** | Post SPRINT meeting. |
| **Description** | The functional and non-functional requirements of the HangOut mobile application. |
| **Exit Criteria** | Software Requirement Specification document |
| **Procedure** | QAM will review the document in detail. |
| **Notes** | Nil |

| **Product Assessments 002** | **Initial Prototype** |
| --- | --- |
| **Entry Criteria** | Post SPRINT meeting after the initial prototype is formed. |
| **Description** | The initial prototype will contain the core features of the HangOut mobile application. |
| **Exit Criteria** | 1. Review and documentation of functionalities checks. 2. Bug reports. |
| **Procedure** | 1. QAM will review product requirements that were stipulated in previous meetings. 2. QAM will then test and identify bugs in the prototype. 3. QAM will update the bug report. |
| **Notes** | Nil |

| **Product Assessments 003** | **Subsequent Prototype** |
| --- | --- |
| **Entry Criteria** | Post SPRINT meeting after improving the initial prototype. |
| **Description** | The new prototype will consist of additional modifications and improvements from the previous prototype. |
| **Exit Criteria** | 1. Review of functionalities checks. 2. Review of previous bugs. 3. Bug Reports. |
| **Procedure** | 1. QAM will review the product requirements that were discussed in previous meetings. 2. QAM will review the previous bugs and identify any new bugs. 3. QAM will take note of resolved bugs and update the bug report. |
| **Notes** | Nil |

| **Product Assessments 004** | **Finalised Product** |
| --- | --- |
| **Entry Criteria** | Post Sprint meeting after the product is fully completed and there is access to all features in the application. |
| **Description** | The finalised product will contain all features in the HangOut application. The features have to be in working condition so that it can fulfil its intended function. |
| **Exit Criteria** | 1. Review of functionalities checks. 2. All previous bugs resolved. 3. No further bugs found. |
| **Procedures** | 1. QAM will review product requirements that were discussed in previous meetings. 2. QAM will check across bug reports to make sure all bugs are resolved. 3. QAM will check for any new bugs. 4. QAM certifies and confirms that the product is free of bugs. |
| **Notes** | Nil |

### **Process Assessments**

The following process assessments will be conducted by SQ personnel:

| **Process Assessment 001** | **Ensuring Conformity between Process life cycle and Requirements** |
| --- | --- |
| **Entry Criteria** | Project Plan. |
| **Description** | This stage ensures that the Process life cycle does not deviate from the schedule that was originally planned.  In case of possible adjustments to the life cycle of the product, procedures such as the change management policy are put in place to ensure changes are properly documented. |
| **Exit Criteria** | 1. Quality assurance policy. |
| **Procedure** | 1. Review list of requirements given by contract and project plan. 2. Set schedule for conducting reviews and creating audits of ex-sell prototypes. 3. Standardise and document policies to manage process changes. 4. Standardise and document policies for making bug report, release control. 5. Every prototype introduced will be checked for bugs and errors. This will be compared with the estimated number of bugs for a given number of lines of code. 6. Compile findings into a report to be sent to the Project Manager, QAM and Lead Developer to go through. |
| **Notes** | Nil |

| **Process Assessment 002** | **Evaluation of Software Environments** |
| --- | --- |
| **Entry Criteria** | Project Plan. |
| **Description** | This stage ensures that the programming libraries and packages used to develop HangOut are conforming with the regulations stipulated in the contract. |
| **Exit Criteria** | Non-conformance report if any. |
| **Procedure** | 1. Prior to every milestone in the project schedule, conduct a meeting with the Lead Developer to discuss and standardise the type of libraries and packages that will be used for developing the HangOut application. 2. Requirements must be explicitly stated and updated accordingly if any changes occur. 3. During the development phase, every resulting prototype will be scrutinized for any discrepancies that are not in conformance to the agreed standards. 4. In the case where there are any discrepancies, SQ personnel will report to QAM and hold an immediate meeting with the Lead Developer to highlight and resolve the discrepancies. |
| **Notes** | Nil |

| **Process Assessment 003** | **Evaluation of Process Measurements for Conformance** |
| --- | --- |
| **Entry Criteria** | Project Plan. |
| **Description** | This stage ensures that the progress of the project is measured in a quantitative manner to maintain consistency.  This stage also ensures that the progress of the project is reviewed regularly to make sure that the project is developing smoothly. |
| **Exit Criteria** | Non-conformance report if any. |
| **Procedure** | 1. For every milestone in the project schedule, QAM will link up with the Lead Developer to estimate the lines of code needed to achieve the goal at the stipulated checkpoint, as well as different components of the milestone. 2. The Lead Developer will pass down relevant information to the developers on the team 3. At the end of every month, developers are required to submit a simple report on the components that they are working or have worked on, as well as the lines of code produced. This can be done through version control commits. 4. Should there be any failures to meet the estimated lines of code or progress, the Lead Developer and QAM will discuss measures to improve on current status. These measures can include a change in story tickets etc. |
| **Notes** | Peer Reviews can be another form of measurement to determine who has made the most and least contributions to the project. |

| **Process Assessment 004** | **Evaluation of Staff Skills and Knowledge for Conformance** |
| --- | --- |
| **Entry Criteria** | At any stage of the project lifecycle. |
| **Description** | This stage ensures that everyone on the project team is skilled enough to handle the technical requirements of the project.  This involves everyone on the project team, including the management, developers and engineering team. |
| **Exit Criteria** | Non-conformance report if any. |
| **Procedure** | 1. Based on the project plan, QAM performs evaluations to determine if personnel have the necessary qualifications to develop the required components of the project. 2. In the event that there is a mismatch in skill sets, the Project Manager, QAM and the Lead Developer will hold a meeting to discuss possible duty changes between the personnels. 3. Should there be any further gaps in knowledge, personnel will be sent for 2-3 weeks of training in the specific domain(s) to level up their skill sets and experience. 4. Steps 1 - 3 will be monitored every month by QAM to ensure every personnel is able to perform their given task well. |
| **Notes** | Nil. |

## ***Roles and Responsibilities***

This section describes the roles and responsibilities for each assurance person assigned to the Project.

### **QAM**

Responsibilities include, but are not limited to:

* Secure and manage SQ personnel resource levels
* Ensure that SQ personnel have office space and the appropriate tools to conduct SQ activities
* Provide general guidance and direction to the SQ personnel responsible for conducting software quality activities and assessments
* Assist SQ personnel in the resolution of any issues/concerns and/or risks identified as a result of software quality activities
* Escalate any issues/concerns/risks to project management

### **Software Quality Personnel**

Responsibilities include, but are not limited to:

* Develop and maintain the project software quality assurance plan
* Generate and maintain a schedule of software quality assurance activities
* Conduct process and product assessments, as described within this plan
* Identify/report findings, observations, and risks from all software assurance related activities to the QAM

# **Documents**

## ***Purpose***

This section identifies the minimum documentation governing the requirements, development, verification, validation, and maintenance of software that falls within the scope of this software quality plan. Each document below shall be assessed (reviewed) by SQ personnel.

## ***Minimum Document Requirements***

In no particular order:

1. System Requirements Specification
2. Quality Management
3. Software Model Prototype
4. Design report on software maintainability
5. Risk Management
6. Release Plan
7. Change Management Plan
8. Configuration Management Plan
9. Test Plan and Documentation

# **Standards, Practices, Conventions and Metrics**

## ***Purpose***

In every software development process, it is often noted that developers have a tendency to maintain a certain bias that might result in the alteration of the desired final products, and may not be well received by the customers at the end of the project. Therefore, by establishing international standards and practices, the above mentioned issue can be mitigated as it ensures that all biases are avoided, and key issues during the development stages are accurately addressed.

### Standards

The standards that we will be adopting are the ISO 9001 and the ISO 25010. The ISO 9001 is vital in bringing our team a competitive advantage over others as it improves the marketability of our product as well as business processes such as Project Management, Software Development Life Cycle, and Quality Assurance and Testing. Moving on, the ISO 25010 is an international standard for software quality assurance. Any potential human biases that can adversely affect the delivery and the developmental process of the software development project can be addressed with the help of this standard.

### Practices

Our project will be adopting the Agile software development methodology. Agile will prioritise on the development of an initial prototype for us to test and obtain essential feedback from. It also provides us the capability to modify existing functions or even include new ones. With the Agile methodology in place, we will be able to streamline the delivery of our software product while maintaining our product quality.

### Conventions

We will be using letter case-separated words for our naming conventions. This helps with ease of typing and readability, and also avoids confusion when an outside developer observes the lines of code while performing modifications per the client request.

## ***Software Quality Programme***

These practices and conventions are tools used to ensure a consistent approach to software quality for all programs/projects. The 4 qualities we will be focusing on are as follows:

1. Efficiency

Having a fast response from the system for every user input will improve the user experience. A product that carries out its function efficiently and provides the appropriate performance will minimise waiting times and prevent frustration.

1. Reliability

After a user has signed up for an event, the system must capture the user input correctly and display it on the user interface. The system needs to be reliable such that it can perform its required function and maintain a level of performance.

1. Maintainability

Allowing for constant modifications to the application ensures that users can be kept up-to-date with the latest improvements implemented by developers. Other modifications include corrections or adaptations to the software, or changes in the environment etc.

1. Portability

The system needs to be able to be portable across different mobile platforms such as the Apple IOS and Android. This allows the user to enjoy the full range of functionalities provided by the application regardless of their operating system.

### **Standard Metrics**

The following standard metrics are the minimum planned metrics that will be collected, reported, and maintained in the area of software quality assurance:

1. Fan in/ Fan-out
2. Cyclomatic complexity
3. Depth of conditional nesting
4. Length of code
5. Weighted methods per class
6. Fog Index

[Note: These are the metrics that SQ personnel will maintain and report to the QAM. Please enter at least 6 suitable metrics]

# **Software Reviews**

## ***Purpose***

This section identifies the number and type of system/subsystem reviews and engineering peer reviews that will be supported by the SQ Personnel. The project milestone chart, and the SQ Personnel resource levels determine the reviews that are supported.

## ***Minimum Software Reviews***

For each review, SQ will assess the review products to assure that review packages are being developed according to the specified criteria, the review content is complete, accurate, and of sufficient detail, and Requests for Action are captured, reviewed, and tracked to closure. In addition, SQ will assess the processes used to conduct the reviews to determine if appropriate personnel are in attendance, correct information is presented, entry and exit criteria are met, and appropriate documents are identified for update.

The following software reviews will be assessed by SQ:

1. Project Plan Review
2. Requirements Analysis Review
3. Software Specifications Review
4. Test Plan Review
5. Acceptance Review
6. Peer Reviews
7. Post-Implementation Reviews

# **Test**

SQ personnel will assure that the test management processes and products are being implemented per Test Plan. This includes all types of testing of software system components as described in the test plan, specifically during integration testing (verification) and acceptance testing (validation). SQ personnel will monitor testing efforts to assure that test schedules are adhered to and maintained to reflect an accurate progression of the testing activities. SQ will assure that tests are conducted using approved test procedures and appropriate test tools, and that test anomalies are identified, documented, addressed, and tracked to closure.

In addition, SQ will ensure that assumptions, constraints, and test results are accurately recorded to substantiate the requirements' verification/validation status. SQ personnel will review post-test execution related artifacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

[Add any additional SQ test activities (Don’t need to repeat test plan here.)]

As part of our Software Development Process, software testing will be conducted on the following levels:

**Unit Testing**

Unit testing is the first level of testing in which components of a software are tested. Unit testing is conducted during the development phase of the project by the development team to ensure that most implementation errors are looked into, and isolates each individual unit of the code to verify their correctness. Unit testing is usually conducted using Black box and White box testing.

**Integration Testing**

Integration testing is the second level of testing conducted after the initial unit tests are completed. Integration testing focuses mainly on integrating the components and ensuring that data/information can flow smoothly between the components. Tests will be conducted to uncover errors associated with interfacing between the components.

**System Testing**

A series of different tests are carried to fully exercise the software system and explore any possible limitations in the system. The tests are conducted by the QA team and if any bugs or unexpected outcomes are observed, they will report back to the development team to carry out the necessary changes.

**User Acceptance Testing**

During the last phase of the development, the actual intended users test the final product to ensure that the software system is able to handle all the required use cases and perform the functions in accordance with the requirement specifications.

# **Problem Reporting and Corrective Actio**n

## Possible Problems which may arise

1. Document problems:

● Non-compliance with other project documents

● Non-compliance with the house style (SCMP)

● Incompleteness

● Documentation errors

1. Code problems:

● Lack of functionality

● Incorrect functionality

● Non-Compliance with coding or commentary standards

1. Requirement changes by the customers.

## Problem reporting procedure

In the event that the documentation (1) contains error or (2) is incomplete, the member who discovered the problem will be required to perform the following procedures:

1. Report the issue to the Project Manager and QA Manager.
2. The Project Manager notifies the corresponding team member and provides suggestions to implement changes accordingly. The member appointed will be responsible for solving the problem.
3. The assigned member will inform the Project Manager and QA Manager when the problem is solved. They will perform checks to ensure that the problem is no longer present.
4. If the assigned member cannot solve the problem, he/she must notify the Project Manager and QA Manager, who will call for a meeting with the team.
5. A final decision will be made on the subsequent steps to rectify the problem.

In the event that the system or code segment (1) contains an incomplete function, (2) non-compliant with the requirements or (3) is missing totally, the member who discovered the problem will be required to perform the following procedures:

1. Report the issue to the Project Manager and Lead Developer.
2. The Project Manager notifies the corresponding team member and provides suggestions to implement changes accordingly. The member appointed will be responsible for solving the problem.
3. The assigned member will inform the Project Manager and QA Manager when the problem is solved. They will perform checks to ensure that the problem is no longer present.
4. If the assigned member cannot solve the problem, he/she must notify the Project Manager and QA Manager, who will call for a meeting with the team.
5. A final decision will be made on the subsequent steps to rectify the problem.

In the event that there are changes in the software requirements requested by the customer, the following procedures should be taken by the Project Manager:

1. The Project Manager will call for a meeting and inform all the team members about the requested changes.
2. The Project Manager will then appoint team members to perform the various changes accordingly.
3. The team members will implement the necessary changes as assigned to the application and documentation respectively.
4. Once the changes are complete, the members will need to notify the Project Manager and carry out the necessary quality checks to ensure that changes comply with the customer’s request.

All revisions to documentation and code lines are to be updated and notified during weekly team meetings. If there is a need to hold an emergency meeting, the issues must be brought up via phone calls or video conferences to determine the best course of action.

# **Tools, Techniques and Methodologies**

SQ personnel will require access to the following:

## ***Software Quality Tools***

* Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
* GitHub
* Flutter
* FireBase Database
* Google Maps

# **Media Control**

SQ deliverables will be documented in one of the following Microsoft software applications: Word, Excel, or PowerPoint. Deliverables will be in soft copy, with the exception of completed checklists from process and product assessments. See Section 12 for additional details on the collection and retention of key records. Software Quality personnel will request space on the project’s secured server for SQ records. This server is password protected and backed up nightly.

For the maintenance and protection of all records, the following platforms are used:

* MediaWiki
* SVN
* GitHub
* Google Drive

MediaWiki is a collaboration and documentation platform that is free, customizable and open source. The ready availability of user support and comprehensive FAQs provide for a low learning curve and easy handling. MediaWiki also supports concurrent user editing, making it an efficient platform for our team to use.

SVN is used for maintaining current and historical versions of our project, and is an open source version control system. We choose to use SVN instead of other tools such as Helix Core because of its relative accessibility; Helix Core is only free for up to 5 users. SVN allows for editing of our files while safely securing the full revision history.

GitHub is used as a repository for sharing files and concurrently revising code without compromising the edits of other users.This service allows for efficient coding in a team environment, without compromising the contents of the code.

Google Drive is used as a common, easily accessible file storage platform for all users to concurrently edit and store files. The low learning curve, yet high security of Google Drive makes it a good platform for the maintenance and protection of our files.

# **Record Collection, Maintenance, and Retention**

SQ personnel will maintain records that document assessments performed on the project. Maintaining these records will provide objective evidence and traceability of assessments performed throughout the project’s life cycle. There are two types of records that will be maintained: Hardcopy and Electronic. SQ personnel will maintain electronic or hard copies of all assessment reports and findings. SQ Project folders will contain hardcopies of the assessment work products such as completed checklists, supporting objective evidence, and notes.

The table below identifies the record types that will be collected, as well as the Record Custodian and Retention period

| **Record Title** | **Record Custodian** | **Record Retention** |
| --- | --- | --- |
| SQA Assessments | SQ Personnel | One Year |
| SQA Checklists | SQ Personnel | One Year |
| Deliverable Defects | SQ Personnel | One Year |

# 

# **Training**

SQ personnel have fundamental knowledge in the following areas through prior experience, training, or certification in methodologies, processes, and standards:

∙ Audits and Reviews (Assessments)

∙ Risk Management

∙ Software Assurance

∙ Configuration Management

∙ Software Engineering

∙ ISO 9001, ISO 9000-3

∙ CMMI

∙ Verification and Validation

# **Risk Management**

SQ personnel will assess the project’s risk management process and participate in weekly risk management meetings and report any software risks to the QAM and the project manager.

| Type of Risk | Risk | Probability of Risk | Effect of Risk | Strategies to Combat Risk |
| --- | --- | --- | --- | --- |
| Cost Risk | Shortage or poor management of project funds | Moderate | Project may be delayed as a result of an insufficient budget. In worst case scenarios, the negative effects may spill over into scope risks | Excel data should be tabulated on the usage of project funds to track and analyse financial data and ensure that costs fall within project means. |
| Scope Creep Risk | Poorly controlled or unauthorised change to initial project scope | Moderate | Our project may see an unsustainable increase in costs, delaying release of product. In other cases, some important features may have to be compromised for other less integral scope functions. | Each and every scope change must be properly documented and agreed upon by the entire team. The budget required for any additional scope features must also be approved by the budgetary team before set in stone. |
| Skills Resource Risk | Staff unavailability or incompetence | Moderate | Staff unavailability or incompetence in various project divisions may lead to the overall delay of the final project, or contribute to additional, unsustainable costs. | Resource and workforce scheduling must be strictly adhered to. For example, a shared repository should be used to make informed decisions on resource allocation. |
| Performance Risk | Discrepancies between product performance and intended performance | Moderate | Late discovery of performance discrepancy may lead to a delay in product release | User testing must be conducted on a consistent, weekly basis. There must be ample communication among team members to ensure no discrepancy in understanding of project requirements |

# **SQA Plan Change Procedure and History**

SQ personnel are responsible for the maintenance of this plan. It is expected that this plan will be updated throughout the life cycle to reflect any changes in support levels and SQ activities. Proposed changes shall be submitted to the Quality Assurance Manager (QAM), along with supportive material justifying the proposed change.