Virtual Consultant

Software Quality Assurance (SQA) Plan

Version 1 approved

Prepared by Virtual Consultant

Aditya Chandrasekhar (U1923951A) : Project Manager / Lead Developer
Aratrika Pal (U1922069F) : Backend Developer
Shruthi Srinivas (U1923611G) : Backend Developer / Release Engineer
Chong Zhe Ming (U1920757K) : Frontend Developer
Khush Kothari (U1922279J) : Frontend Developer
Kushal Sai Gunturi(U1923232F): QA Manager
Yi Jia Xin, Joceline (U1920057J) : QA Engineer
Lim Yun Han Darren (U1921275J) : Frontend Developer

5th September 2021

Signature Page

Prepared by:	**Aditya Chandrasekhar**	**Date: 5 September 2021**
	Aratrika Pal	**Date: 5 September 2021**
	Shruthi Srinivas	**Date: 5 September 2021**
	Chong Zhe Ming	**Date: 5 September 2021**
	Khush Kothari	**Date: 5 September 2021**
	Kushal Sai Gunturi	**Date: 5 September 2021**
	Yi Jia Xin, Joceline	**Date: 5 September 2021**
	Lim Yun Han Darren	**Date: 5 September 2021**

Reviewed by 1: _____Aditya_Chandrasekhar_______ Date:10 September 2021

Reviewed by 2: _______Aratrika_Pal_____________ Date:11 September 2021

Approved by : _______Aditya_Chandrasekhar______ Date:13 September 2021

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	**Initial Template**	Kushal Sai Gunturi	**05/09/21**
0.20	**Completed draft**	Aditya Chandrasekhar	**10/09/21**
0.30	**Revised draft**	Aratrika Pal	**11/09/21**
0.40	**Approved draft**	Aditya Chandrasekhar	**13/09/21**

Contents

1. Purpose and Scope

1.1. 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 Virtual Consultant 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.

1.2. Scope

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

Virtual Consultant is a progressive web application that allows patients to connect with doctors virtually to get consultation services, as well as increase patients' awareness on topics of health and diseases. After thorough assessment of consumer needs, we have defined key aspects of our product, which include: connecting users with available doctors virtually, maintenance of patients' medical records, posting of health related content by doctors and a one-way social media account (where users are able to view and like doctors' posts, as well as follow them).

Virtual Consultant will be developed using technological tools such as React.js, which will be used for developing the frontend of the web application; Node.js, for the execution of server-side code; Express.js, which provides a simple and convenient API for developing the application; MongoDB, used as a NoSQL database system that will store the data required for our application; Git and GitHub, for the management and tracking of source code; Visual Studio Code, the Integrated Development Environment; Figma, for production of User Interface prototypes; Notion, an online workspace for task management; Google Drive, for file storage. The SQA Plan is intended to cover the above-mentioned software, its source and verify its authenticity. The Plan is also meant to be as rigorous as possible in order to ensure that the quality of the software development is of the highest and also to avoid future compromises to the project if the software quality is not performing to standard.

Conduction of regular testing will be held for all components of the system and internal logic, which will help find any existing bugs in the software and allow us to resolve them before releasing the application.

Virtual Consultant will be developed by a team of eight members, each assigned a specific role based on their expertise. The roles allotted include: one Project Manager (PM), who will oversee the whole project, one Lead Developer, who will oversee the development of the frontend and backend, two Frontend Developers, two Backend Developers, one Quality Assurance Manager, one Quality Assurance Engineer and lastly, one Release Engineer, who will oversee the release of the application.

2. 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 (https://docs.google.com/document/d/1QHydr2wOKG67FGcxjIb8AzBeXA1htWFgGaY0ewVDYvU/edit?usp=sharing)
- System Requirement Specifications (https://docs.google.com/document/d/1uz3NmF5Cf5cTb-JBC9nPleMj_9obNEWgQIeOB8JPF0Q/edit?usp=sharing)

3. Management

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

3.1. Management Organisation

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

3.1.1. 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 utilized
- The selection of project teams

- The training of project teams

In addition to the above-mentioned roles, the Project Manager will also be responsible for:

- Ensuring that the project stays on schedule
- Resolvement of conflicts, should there arise any

3.1.2. 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.

3.2. Tasks

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

3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- **User Interface (UI)**
  The User Interface of Virtual Consultant will be assessed to ensure that it is user-friendly, yet elegant in designs such that it is visually appealing. The interface should encompass good human-computer interaction principles such as catering to universal usability and consistency. This will allow users to navigate through the different functionalities of our application easily.

- **Patient-Doctor Connection**
  The application must be able to connect users to available doctors in the database upon request. Once a connection has been established, Virtual Consultant must also be able to allow patients to communicate with the doctor through text messages, voice calls and/or video calls (all within the application itself). SQ personnel should assess whether users are able to seamlessly connect with available doctors in the database with minimal downtime. As this is an integral part of our application, it must be assessed carefully by the SQ personnel.

- **Maintenance of Patient Records**

The application should allow for users to be able to input their personal medical history and for doctors to be able to have access to the patient's medical history upon successful connection. This allows for doctors to give a more accurate prognosis. Users and doctors must also be able to update and edit their personal information to ensure that the medical history is updated and accurate. It is essential that the patient records are kept confidential and the SQ personnel will need to thoroughly assess the database at all levels to ensure that leakage of information does not occur.

- **Posts**
  Virtual Consultant aims to increase awareness amongst users on the topics of health and diseases, hence the Post feature is an integral feature of our application. This feature must allow doctors to post articles with regards to health and diseases and users can easily access these articles. SQ personnel should assess whether doctors are able to create posts easily and whether users are able to view and read the most updated posts, especially from doctors that they follow.

- **One-way Social Media Accounts**
  Users must be able to follow doctors on their accounts and view their posts on their 'Post' feed first. Users must be able to like doctors' posts. Doctors are however, not allowed to follow users, hence, making it a one-way social media account. SQ personnel should assess this functionality to ensure that engagement within the application is present.

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- **Requirement Management Process**
  Covered under the scope of Quality Assurance and Management, requirements elicitation and analysis are an integral part of the project. SQ personnel must craft out requirements elicitation procedures and enforce them in order to get a clearer and more comprehensive coverage of the functional requirements of Virtual Consultant. SQ personnel are also required to conduct assessments of the requirements elicitation process and formally finalize the application requirements.

- **Change Management Process**
  Changes within the team and processes may occur during the project duration in order to maximize results. SQ personnel must assess the adaptability of team members to these changes, as well as assess the changing processes and techniques in order to maximize results.

- **Reliability Management Process**

As Virtual Consultant aims to provide medical consultation services to users all around the world, the application should remain reliable. This means that the application must be able to maintain its level of performance under all circumstances. SQ personnel must set a baseline of expectations for the reliability of the application and the steps set out to achieve the same.

- **Maintainability Management Process**
  Virtual Consultant must be maintainable, thus it must be easy to correct faults and improve performance. SQ personnel must come up with a set of guidelines to ensure that the application is maintainable and enforce these guidelines for the duration of the project.

3.3. Roles and Responsibilities

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

3.3.1. 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

3.3.2. 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

4. Documents

4.1. 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.

4.2. Minimum Document Requirements

- Project Proposal
- System Requirements Specification
- Project Plan
- Risk Management Plan
- Configuration Management Plan
- Change Management Plan
- Test Plan
- Test Cases and Requirements Test Coverage Report

5. Standards, Practices, Conventions and Metrics

5.1. Purpose

This section highlights the standards, practices, quality requirements, and metrics to be applied to ensure a successful software quality program.

5.2. Software Quality Programme

These practices and conventions are tools used to ensure a consistent approach to software quality for all programs/projects.

- **Functionality**

  Functionality of the software relates to the capability of the software product to provide functions that meet stated and implied needs when the software is used under specified conditions. It is pivotal that Virtual Consultant is a fully functional web application that is able to connect patients with doctors to gain access to medical consultation services. It must satisfy all functional and non-functional requirements as lined out in the System Requirements Specification document and remain traceable should any errors arise.

- **Usability**

  Usability is the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions. Virtual Consultant must possess a simple user interface that will allow users to easily navigate through the application.

- **Reliability**

  The capability of the software to maintain its level of performance under stated conditions for a stated period of time. Virtual Consultant's code design should not incorporate tight coupling or singletons and the design should utilise more reusable code. This will in turn help to improve performance and minimize downtime.

- **Maintainability**

  Maintainability refers to the capability of the software product to be modified. Modifications may include corrections, improvements or adaptations of the software to changes in the environment and in the requirements and functional specifications. When developing Virtual Consultant, we should ensure that the code is maintainable. Developers should design the software such that it will be easy to modify and change as per changing requirements. Errors and bugs should also be easy to catch and minimal steps should be undertaken to troubleshoot and resolve.

5.2.1. 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:

- **Mean Time to Failure**
  This is the average time between failures. It provides an idea of how frequent errors are presented in the software. In relation to Virtual Consultant, SQ personnel will track the Mean Time to Failure for various scenarios when the application is put under stressors such as excessive load and parallel access of resources used by the application.

- **Defect Density**

  This measures the number of defects relative to the software size expressed as lines of code or function point. SQ personnel are responsible for collecting and tracking the number of defects detected per 250 lines of code in Virtual Consultant and suggest amendments to be made in order to minimize the number of defects.

- **Length of Code**

  Length of code is based on the notion that longer code lengths are harder to understand. This signifies that the length of code must be minimized by improving reusability of code through object-oriented practices such as the usage of functions, inheritance, interfaces and abstract classes. SQ personnel should perform frequent checks to ensure that the length of code is minimized and suggest ways to maintain minimal code length.

- **Problems per User-Month in User Acceptance Testing (UAT)**

  UAT is used to simulate the user experience of the application. If too many problems arise during the UAT phase of the project, those concerns should be addressed immediately as it signifies that the application might not be accepted by users. Hence, when conducting UAT for Virtual Consultant, SQ personnel should take note of all problems encountered and take note of problems that have been reported multiple times. These problems should then be red-flagged and reported to the Project Management team immediately for troubleshooting.

- **Cyclomatic Complexity**

  Cyclomatic complexity is used to indicate the complexity of the program by looking at the source code. It is a quantitative measure of the number of linearly independent paths in the program's source code. SQ personnel must monitor Virtual Consultant's source code and ensure that the cyclomatic complexity of the code is kept low such that most parts of the program will not be interdependent on each other. Should the cyclomatic complexity of the code be high, the Project Management team should be notified of the issue.

- **Percent Delinquent Fixes**

  This is a software maintainability metric and refers to the percentage of all fixes within a time interval that is defective. It is important that this percentage is kept at a minimum through testing of a fix. SQ personnel are responsible for tracking the fixes in Virtual Consultant and relay any defective fixes to the Project Management team as a large number of defective fixes point to a larger issue in the codebase.

6. Software Reviews

6.1. Purpose

In this section, it is listed how many system/subsystem reviews and engineering peer reviews will be conducted by the SQ Personnel. The milestone chart and resource levels determine the number of reviews to be conducted.

Software reviews are an important part of the Software Development Life Cycle (SDLC) that assists software engineers in validating the quality, functionality, and other essential features and components of the software. The test involves testing the software product and ensuring that it meets the client's requirements.

Additionally, software reviews ensure that the code submitted by members of the team meets a coding standard - for example, has appropriate commenting, is low-coupled, has high cohesion, etc.

We review software for the following reasons:

- Increasing productivity of the development team
- Making software testing time and cost-efficient as testing is limited during the initial stage of development
- Lowers the cost of all the processes involved by reducing the number of defects found in the final software.
- Obtaining sufficient reviews since the reviews provided at this stage are found to

be cost-effective if a defect is identified at a much earlier stage. Resolving the defect at a later stage would be more expensive.

• Correcting shortcomings that lead to faults. The removal of faults or errors can greatly benefit the software. A low error rate can be achieved by checking samples of work frequently and identifying small-time defects.

6.2. Minimum Software Reviews

SQ will evaluate the review packages, to ensure that they are being generated in accordance with the prescribed criteria. SQ will also ensure that review material is thorough, accurate, and detailed enough and that Requests for Action are being documented, reviewed, and tracked to completion. SQ will also evaluate the processes used to conduct the reviews to see if the key stakeholders are there, the right information is presented, the entry and exit criteria are met, and the necessary documents are flagged for updates.

The following software reviews will be assessed by SQ:

- Project Plan Review
  - o Software Specification Review
    - ▪ Ensuring the SRS performance requirements are feasible, complete, and consistent with the higher-level specification requirements
    - ▪ Ensuring all derived requirements have been identified and documented
    - ▪ Ensuring the requirements as stated are testable and measurable
    - ▪ Ensuring there are complete verifiable requirements for all performance requirements
    - ▪ Evaluating reserve capacity requirements and scenarios/procedures for measurement
    - ▪ Evaluating agreements on interfaces and boundaries ▪ Evaluating results of functional analyses
    - ▪ Evaluating requirements allocation decisions
    - ▪ Evaluating identified software risks and proposed mitigation methods
    - ▪ Evaluating applicable design constraints
    - ▪ Examining the proposed software development processes
    - ▪ Examining baseline control and configuration management processes
  - o Estimation, Master Schedule, and Project Plan Review

- Requirements Analysis Review
    - Reviewing the software requirement development
    - Determining whether the stated requirements are clear, complete, unduplicated, concise, valid, consistent, and unambiguous, and resolving any apparent conflicts.
- Software Design Review
    - Preliminary Design Review
        - Ensuring that the software requirements are reflected in the software architecture
        - Specifying whether effective modularity is achieved ▪ Defining interfaces for modules and external system elements
        - Ensuring that the data structure is consistent with the information domain
        - Ensuring that maintainability has been considered ▪ Assessing the quality factors
    - Critical Design Review
        - Assuring that there are no defects in the technical and conceptual designs
        - Verifying that the design being reviewed satisfies the design requirements established in the architectural design specifications ▪ Assessing the functionality and maturity of the design critically ▪ Justifying the design to the outsiders so that the technical design is more clear, effective, and easy to understand
    - Program Design Review
        - Assuring the feasibility of the detailed design
        - Assuring that the interface is consistent with the architectural design
        - Specifying whether the design is compatible to the implementation language
        - Ensuring that structured programming constructs are used throughout
        - Ensuring that the implementation team can understand the proposed design
- Peer Reviews (EPR)
    - Code Walkthrough

  - o Design Review

- • Test Plan Review
- • Acceptance Review
- • Release Review
  - o Process Audit: Final Release
- • Project Closing Review
  - o Ensuring all requirements from the project scope document have been met
  - o External review after final delivery
  - o Adding all documents together, including finalizing all project reports, then organizing and archiving them as historical data to be used for future reference.

7. 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 those test anomalies are identified, documented, addressed, and tracked to closure. In addition, SQ will assure 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.

To ensure that our software fulfills the requirements defined in the System Requirement Specifications and to assure our client that the software system will be stable for consumer usage upon releasing it to the market, Software Testing will be conducted as part of our Software Development Process. Several levels of tests will be conducted, as follows:

Unit Testing

All code will be unit tested to ensure that the individual unit (class) performs the required functions and outputs the proper results and data. Proper results are determined by using the design limits of the calling (client) function as specified in the design specification defining the called (server) function.

Unit testing is typically white box testing and may require the use of software stubs and symbolic debuggers. This testing helps to ensure the proper operation of a module because tests are generated with knowledge of the internal workings of the module. To summarize, individual classes will be tested to ensure reliability and functionality within a unit level. Furthermore, a testing module will be created before the tested code to ensure that the code is testable.

Integration Testing

There are two levels of integration testing. One level is the process of testing a software capability e.g., being able to send a message via a DMA port. or the ability to acquire a row of CCD data. During this level, each module is treated as a black box, while conflicts between functions or classes and between different software are resolved.

The second level of integration testing occurs when sufficient modules have been integrated to demonstrate a scenario e.g., type of science mode or the ability to queue and receive commands.

Software documentation is reworked as necessary. To summarize, several classes will be tested together to ensure sufficient execution and compliance with the requirements after integration. Integration testing will be performed before the beta release.

System Testing

System testing comprises a series of different tests where the purpose is to fully exercise the software system. The purpose of system (stress) testing is to identify the operational envelope of the instrument.

Testing will be conducted by the QA team and any bugs or outcome that is found to be different from the expected result will be reported back to the development team.

The whole system shall be used for system testing to ensure all requirements are satisfied, and reliability will be included in the testing to measure the success rate of message delivery. System testing will be performed before the beta release.

User Acceptance Test

User Acceptance Test is being conducted at the last phase of the development where the actual intended users test the software system to ensure that it can handle all the required use cases of their job scopes and the functions are according to the requirement specifications.

8. Problem Reporting and Corrective Action

The Quality Assurance team collects and analyses the QA activity metrics as well as assigns, tracks, and verifies corrective actions resulting from audits, evaluation and monitoring using a centralized Reporting and Corrective Action System implemented on a Kanban board system using Notion Projects Boards. Conditions identified during QC reviews or QA audits as needing corrective action should be addressed promptly.

The lists set up are:

1. **Problems/Issues**
   • Issues unassigned to any SQ personnel
   • Detailed descriptions of issues with a tracking label or tag
   • Priority of the issue
2. **In progress**
   • Current tasks by different SQ personnel are displayed such that others will be able to monitor and be aware of the progress in different parts of the project.
3. **Review**
   • Issues ready to be reviewed by other SQ personnel to check on the quality.
   • If an issue is resolved, it would be moved to the "Completed" list, if it needs more work, it would be moved to the "Progress" list
4. **Completed**
   • Resolved issues.

Furthermore, minimum documentation for any problem should include:

• The location of the failure
• The date and time of the failure
• The operation being performed
• The failure symptom
• The name and number of the reporting individual(s)
• The environment under which the failure occurred
• The impact of the failure on hardware
• For each test-related failure, a "Stress statement" on whether the failure induced any stress in the hardware.

A means should also be available for tabulating the failure information for determining trends and the mean-time-between-failure of the equipment.

The general approach for defining corrective action requirements would involve:

• Identifying corrective action needs and causes
• Establishing appropriate corrective action responses
• Verifying the timely implementation and effectiveness of the corrective action taken

9. Tools, Techniques, and Methodologies

SQA will assure that all purchased or developed tools that directly affect the contents of the software that resides in the DPA are uniquely identified, tested/evaluated.

SQ personnel will require access to the following:

9.1. Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- GitHub Project Boards
- Google Docs
- React Testing Library (RTL)
- Chai and Mocha (JavaScript Testing)

10. 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, except for completed checklists from the process and product assessments. See Section 11 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.

As the project needs to be stored in a secured server, the following services are used:

1. MediaWiki

   Software Quality personnel will also request space on the project's MediaWiki for SQ records. This MediaWiki is password protected and backed up nightly, to ensure any disaster management discrepancies in the documents. There is a wide range of functions that allow users to create their information in different styles. It also allows users to concurrently edit the page at the same time.

2. SVN

   SVN is used for versioning and for revision control. This service is used as it helps us to keep track of the different versions of source code files and documents being uploaded and edited. It allows users to rename, move, remove, and copy the file while keeping the full revision history. Tortoise can be used to integrate windows explorer and hence, allowing us to view and edit the files easily.

3. Google Drive

   Google Drive service is used as file storage and for the backup of documents initially created. This service allows users to share and store files within the group easily. Also, this service allows users to edit documents concurrently. Version control is also available in this service.

4. GitHub

   GitHub is used for versioning and for revision control across all team members. This service helps to update and revise different versions of source code files and documents being uploaded upon pulling request.

All these applications are password protected and allow a single source of truth for the quality control and management documents. Upon project completion, a zip folder containing the entire project document, prototype, and final product is created.

11. Supplier Control

Supplier control is not applicable in the capacity of this project, involving no suppliers associated with the delivery of the software application.

12. 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. Example records include the process and product assessments reports, completed checklists, the SQ Activity schedule, metrics, weekly status reports etc.

Further, there are two types of records that will be maintained: Hardcopy and Electronic. SQ personnel will maintain electronic or hard copies of all reports and findings. SQ Project folders will contain hardcopies of the assessment work products such as completed checklists, supporting objective evidence and notes. In case of the electronically stored documents, internationally accepted security standards will be met.

All the standard and internationally accepted frameworks and guidelines will be followed in the preparation, collection, maintenance and retention of the above mentioned documents. Through version control systems and dedicated software applications, the modification history will be kept intact and ensure traceability and cross validation of all the quality procedures.

All the QA management documents will be available and accessible until 6 months after the project completion, unless stricter rules or regulations do not state a later date. Representatives of relevant authorities and authorised personnel will be entitled to examine the project, all relevant documentation, and accounts of the project also after its closure. 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	6 Months
SQA Checklists	SQ Personnel	6 Months
Deliverable Defects	SQ Personnel	6 Months

13. Training

All the SQ personnel are highly qualified and appropriately trained in software quality management practices and standards. The personnel has 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

14. 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. A comprehensive and robust risk management procedure will be followed which is clearly documented in a separate risk management plan. The salient features are briefly described here.

- **Risk Identification and Analysis**

  Risk will be actively identified, considering potential risks to and from the project team, stakeholders along with extrinsic factors, like environmental, cultural and political factors capable of adversely impacting the project. The Project Proposal, deliverables, constraints, schedule and budget will be scrutinized to minimize risks. Thereafter risks will be analysed in two broad categories of quantitative and qualitative risks. The likelihood and the probability of each identified risk will be calculated and classified into categories of low, medium and high. Further, quantitative metrics to provide an objective numerical rating to the risk will be estimated. All these will be clearly documented in the risk management plan.

- **Risk Response Planning, Monitoring and Control**

  The analysed risk will be categorized into varying levels of criticality of red, yellow and green zones. The red and yellow risks will be assigned to individual personnel, to maintain and ensure each risk is carefully mitigated. The framework of avoid, mitigate, accept and transfer will be used to address the above mentioned risks. The risks prioritized based on their criticality will be tracked, monitored and reported throughout the project lifestyle. The risk impact on key project processes will be analysed and the management will be notified of any important changes to risk assessment.

15. 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. The QAM will then review and approve any necessary changes. All these will be duly recorded, documented and tracked. A version history will be maintained with all the changes verifiable and reverted in case of any errors or discrepancies in the records at a later stage. All the documents are enabled with password protection and stored in secured software applications, as well as version control systems.