# Software Quality Assurance (SQA) Plan
# By Think2

**Date: September 2022**

# Signature Page

**Prepared by:** **Tanya Banerjee** **Date:** **7 September 2022**

**Reviewed by 1:** **Siti Nur Umm’aira Phang** **Date:** **14 September 2022**

**Reviewed by 2:** ____________________________ **Date:**________________

**Approved by :** ____________________________ **Date:**________________

# Document Change Record

| **Revision** | **Description of Change** | **Approved by** | **Date** |
|---|---|---|---|
| **1.0** | **Initial Draft** | **Siti** | **14/9/2022** |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Contents

**Purpose and Scope** **6**

Purpose 6

Scope 6

**Reference Documents** **7**

**Management** **7**

Management Organisation 7

Project Management 7

Assurance Management 7

Tasks 8

Product Assessments 8

Process Assessments 9

Roles and Responsibilities 9

QAM 9

Software Quality Personnel 10

Documents **10**

Purpose 10

Minimum Document Requirements 10

Standards, Practices, Conventions and Metrics **10**

Purpose 10

Software Quality Programme 11

Standard Metrics 11

Software Reviews **12**

Purpose 12

Minimum Software Reviews 12

**Test** **13**

**Problem Reporting and Corrective Action** **13**

**Tools, Techniques and Methodologies** **13**

Software Quality Tools 13

**Media Control** **14**

**Supplier Control** **14**

**Record Collection, Maintenance, and Retention** **14**

Training **14**

**Risk Management** **15**

**SQA Plan Change Procedure and History** **15**

# 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 HelpMe 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

This project includes the development of an Android mobile app where users can post the services (e.g. pet sitting, catering) they require, or the services they need. Other users who are interested in the posts may contact them.

QA personnel will thoroughly test all functional and non-functional requirements of the app. 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.

The project HelpMe will be a mobile application that enables customers and contractors to collaborate on a single, user-friendly platform for services. Users will be able to create profiles that showcase their skills and portfolio. Users will be able to both request services from and offer services to the community. Services will be user-defined – users will decide on the prices and description of given services. Other users on the application can then accept these service requests or offers. In addition, users will be informed in real-time through a notification system if there are any offers or requests that are related to their preferences. The application aims to deliver the most user-friendly environment for users to collaborate with zero platform fee.

The Think2 team is involved in this project. The group consists of seven members, who have been assigned a role each in the designing, development, testing and releasing of the application. HelpMe will be developed using Android studio with Java language for the frontend and Firebase for the backend.

# 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 ( HelpMe proposal.docx )
- System Requirement Specifications SRS.docx

## 3. Management

This section describes the management organisational 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 utilised
- The selection of project teams
- The training of project teams

#### 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 summarises 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.

#### 3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- **User Interface**
  The User Interface should be easy to use and able to direct new users to the functions they want to access. This is to make it easier for older people to use the app.

- **Sign up and Login**

  During sign up, an error message should be displayed if the user enters a previously registered email, enters their email in the wrong format, or enters a weak password. The database will be used to store the new user's details after the sign up.

  During login, an error message should be displayed if wrong login credentials are entered, as verified from the database.

- **Database Communication**

  The database will store each user's login details, profile information, all posts written, and post history. Database communication should be fast and error free.

- **Filtering**

  Posts can be filtered by two options - region and categories.

  - Regions - Posts should be filtered according to the region (e.g. Central, North-East) entered by the user.
  - Categories - Posts should be filtered according to the category (e.g. babysitting, elderly care) entered by the user.

- **Sorting by Rating**

  Posts should be sorted in descending order of rating for the particular category selected by the customer.

- **Searching**

  Users should be able to search all posts for words they wish to find.

- **Posting of services and requests**

  Users should be able to post about the service they wish to provide or the service request they wish to make. All users can view the entire list of posts under the *View Request/Offer* page.

### 3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- **Requirement Management Process**

  The requirement elicitation procedures must be drafted and enforced throughout the full lifecycle of the project. Consumers, stakeholders, managers and developers must be involved in the requirements management process. SQ personnel must work to ensure that no changes are made to baselined requirements without performing a risk analysis, re-estimating impacts to cost and schedule, and validation among stakeholders.

- **Change Management Process**
  A significant change in the team and its processes might be needed to achieve certain business outcomes. SQ personnel must assess readiness and ease of changing processes, techniques and people to achieve the expected outcomes.

- **Maintainability Management Process**
  The application HelpMe must be maintainable for ease of changes in the future. A set of baseline expectations for maintainability must be formalised and broken down into steps. During development, the SQ personnel will regularly assess the maintainability management to ensure high quality.

- **Risk Management Process**
  The risk management measures are to be enforced by the SQ personnel. All edge cases and potential risks should be flagged and the team should be prepared for such contingency scenarios in advance.

## *3.3. Roles and Responsibilities*

This section describes the roles and responsibilities for 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*

- System Requirement Specifications
- Use Case Description
- Proposal
- Project Plan
- Risk Management Plan
- Design report on Software Maintainability
- Configuration Management Plan
- Change Management Plan
- Release 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**
  The software products must be able to provide functions which meet stated and implied needs when the software is used under specific conditions.

- **Usability**
  The software product should be understood, learned, used and attractive to the user when used under specific conditions.

- **Efficiency**
  The software product must be able to provide appropriate performance,

relative to a number of resources used, under stated conditions.

- **Reliability**
  The software product must be able to maintain its level of performance under stated conditions for a stated period of time.

### 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:

- **Fan in/Fan-out**

  Fan-in is a measure of the number of functions or methods that call some other function or method (say X). Fan-out is the number of functions that are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components.

- **Length of code**

  This is a measure of the size of a program. Generally, the larger the size of the code of a component, the more complex and error-prone that component is likely to be. Length of code has been shown to be one of the most reliable metrics for predicting error-proneness in components.

- **Cyclomatic complexity**

  This is a measure of the control complexity of a program. This control complexity may be related to program understandability.

- **Mean Time to Failure**

  The average time between two system failures should be sufficiently large.

- **Depth of conditional nesting**

  This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone.

- **Depth of inheritance tree**

  This represents the number of discrete levels in the inheritance tree where subclasses inherit attributes and operations (methods) from superclasses. The deeper the inheritance tree, the more complex the design. Many different object classes may have to be understood to understand the object classes at the leaves of the tree.

# 6. Software Reviews

## *6.1. 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.

## *6.2. 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:

- Project Plan Review
- Requirements Analysis Review
- Software Design Review
- Test Plan Review
- Acceptance Review

# 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 that 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 artefacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

# 8. Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in a centralised Reporting and Corrective Action System that is implemented in a Notion Kankan board.

- Tests that were done
- Which use cases are satisfied and which are not

- Problems which are not use cases (usability)
- In progress, Review, Completed

# 9. Tools, Techniques and Methodologies

SQ personnel will require access to the following:

## 9.1. Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- Firebase
- Google Docs
- Visual Paradigm
- Github
- Android Studio
- Notion

# 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, 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.

# 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. The records will be maintained in electronic form. SQ Project folders will contain softcopies 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 | 4 months |

| | | |
|---|---|---|
| SQA Checklists | SQ Personnel | 4 months |
| Deliverable Defects | SQ Personnel | 4 months |

# 13. 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

# 14. Risk Management

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

# 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.