Software Quality Assurance (SQA) Plan

By: Team Runtime Error

Das Atrik (U1823950C)
Chua Chong Wei Melvin (U1921924G)
Chua Chong Hung Kelvin (U1921404D)
Rajuravi Vishal Raj (U1822268B)
Goh Hong Xiang, Bryan (U1920609E)
Padhi Abhinandan (U1823860D)

Date: 06 February 2021

Signature Page

Prepared by: **Melvin, Kelvin, Vishal, Abhinandan, Atrik, Bryan** **Date:** **06/02/2021**

(Name)

Reviewed by 1: **Das Atrik** **Date:** **11/02/21**

<(Reviewer Name)>

Reviewed by 2: **Rajuravi Vishal** **Date:** **11/02/21**

<(Reviewer Name)>

Approved by : **Rajuravi Vishal** **Date:** **11/02/21**

(Name)

Document Change Record

Revision	Description of Change	Approved by	Date
0.1	**Initial Template**	Das Atrik	**06/02/2021**
0.2	**Intermediate Template**	Das Atrik	**09/02/2021**
0.3	**Final Template**	Vishal	**14/02/2021**

Contents

1. 1 Purpose and Scope

1. 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 MALL-E 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.

2. 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 to identify any errors, omissions, inconsistencies, and alternatives, enhancements or improvements that can be made at any stage of development.

The project MALL-E is a mobile application that tracks the crowd within a mall in Singapore, specific to each level and each shop. The application can also provide waiting/queue times for restaurants in malls.The user can also browse the top 5 covid-related news for shoppings malls in Singapore. This allows users to consider the risks of such Covid infected Shopping Malls and plan their trips accordingly.

Software	Intended use
React Native (frontend framework for mobile applications)	It is a JavaScript framework for building natively rendering mobile applications that runs on both iOS and Android. It is based on the web framework ReactJS made by Facebook.
NodeJS (backend run-time environment for web applications)	It is an open-source development platform used to execute server-side JavaScript code. It uses HTTP actions to act as a REST server which reacts appropriately to events fired off at the client-side.
MongoDB (NoSQL database)	It is a database used to store documents/images/other forms of data with good scalability and ease-of-access which help in seamless querying and indexing of data.

2. 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)
- APA is the style of documentation of sources used by the American Psychological Association (https://apastyle.apa.org/)
- Project Plan
- System Requirement Specifications

3. 3 Management

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

1. 3.1. Management Organisation

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

1. 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

2. 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. 3.2. 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.

4. 3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- Software Requirement Specifications Review

- Software Design Document Review
- Product Assessments
- Technology Assessments
- Code Review
- Code Inspection

5. 3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Change management process
- Requirement management process
- Product Assessment following the ISO 9126 Quality Model,

2. 3.3. Roles and Responsibilities

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

1. 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

2. 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. 4 Documents

1. 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.

2. 4.2. Minimum Document Requirements

- Use Case Description

- Use Case Diagram
- System Requirement Specifications
- Project Proposal

5. 5 Standards, Practices, Conventions and Metrics

1. 5.1. Purpose

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

2. 5.2. Software Quality Programme

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

Quality Characteristics	**Description**
Functionality	The application must be able to provide the 4 main functions: 1. Login Function 2. Crowd Density 3. Waiting Time of Restaurants 4. Covid news Shopping Malls The application must be able to provide these 4 functions in Singapore based on the functional requirements provided in the SRS.
Usability	The application must be easy to be understood, learned, used and attractive to all age groups.The application will cater to older age groups especially elderly by having the feature to enlarge text. Our Usability also follows some guidelines of the Shneiderman's "Eight Golden Rules of Interface Design" 1. Offer Informative Feedback. For every screen Press/scroll, there should be some interactive feedback on the screen. Press on an UI component will result in some UI changes. For example, a user presses a Button to move on to the next page of the application. 2. Reduce short-term memory load. The limitation of human information

	processing short-term memory load requires that application displays and functionalities are to be kept simple. As said by Shneiderman it is easier to recognise than to recall. Minimize the user's memory load by making objects, actions, and options available. The user should not have to remember information from one part of the dialogue to another.
Maintainability	The application should be easy to maintain and update as new features would be added in the future. It should also be easy to resolve any highlighted bugs found during the usage of the applications by users.
Portability	The application can be used on both IOS and Android devices. This allows us to provide the service to almost all smartphone users in Singapore.

1. 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:

- **Cyclomatic Complexity.** This is the measure of the control complexity of a program. This control complexity may be related to program understandability.
- **Length of Identifiers.** This is the measure of the average length of distinct identifiers in a program. The longer the identifiers, the more likely they are to be meaningful and hence the more understandable the program
- **Fan-in/Fan-out.** Fan-in is a measure of the number of functions or methods that call some other function or methods. Fan-out is the number of functions or methods that are called by the function.
A high value for fan-in or fan-out would mean that the overall complexity of the program is high and some function is tightly-coupled to the rest of the design.
- **Depth of conditional nesting.** This is the 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.
- **Length of code.** This is the measure of this 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 shown to be one of the most reliable metrics for predicting error-proneness in components.

6. 6 Software Reviews

1. 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.

2. 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:

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

7. 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 artifacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

8. 8 Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in a Reporting and Corrective Action System. We will be using Google Excel Sheet in the Team's share folder on Google Drive to act as Reporting and Corrective Action System.

8.1. Assessment Data

Each reported problem shall include the reporting date, the reporter, the corrective action status, and the assigned SQ personnel to track the problem.

1. 8.2. Auto Notification

When SQ has made any updates to the problem, the QAM will be notified automatically on the Google Excel Sheet and get the latest update on the matter.

For each newly reported problem, the QAM will assign a SQ personnel to track the problem.

The QAM will decide whether to escalate the updated findings to the PM.

2. 8.3. Update Frequency

SQ personnel shall report the updated status of the assigned problem(s) to the QAM every four days, or at a higher frequency if specified by the QAM.

The QAM will then decide whether to escalate the updated findings to the PM.

9. 9 Tools, Techniques and Methodologies

SQ personnel will require access to the following:

1. 9.1 Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- Tortoise SVN
- GitHub
- Google Drive
- MediaWiki

10. 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 store the SQ deliverables in a shared SVN folder. The folder will be password protected and only accessible to authorized individuals. A soft copy will also be uploaded to Wiki.

11. 11 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 Softcopy. SQ personnel will maintain soft 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	Six Months
SQA Checklists	SQ Personnel	Six Months
Deliverable Defects	SQ Personnel	Six Months

12. 12 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 9126

· CMMI

· Verification and Validation

13. 13 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 PM.

Should any high-level risk arise, the QAM will schedule a meeting with the PM within 3 working days to ensure appropriate actions are taken to delegate the risk.

14. 14 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.