

Software Quality Assurance (SQA) Plan

By Team Syan

Date: 19 September 2021

Signature Page

Prepared by: Ryan Tan Jinn-En

Date: 19 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
0.10	Initial Draft		10/09/21

Table of Contents

1. Purpose and Scope	5
1.1 Purpose	5
1.2 Scope	5
2. Reference Documents	6
3. Management	7
3.1. Management Organisation	7
3.1.1. Project management	7
3.1.2. Assurance Management	7
3.2. Tasks	7
3.2.1. Product Assessments	7
3.2.2. Process Assessments	7
3.3. Roles and Responsibilities	8
3.3.1. QAM	8
3.3.2. Software Quality Personnel	8
4. Documents	9
4.1. Purpose	9
4.2. Minimum Document Requirements	9
5. Standards, Practices, Conventions and Metrics	9
5.1. Purpose	9
5.2. Software Quality Programme	9
5.2.1. Standard Metrics	10
6. Software Reviews	10
6.1. Purpose	10
6.2. Minimum Software Reviews	10
7. Test	11
8. Problem Reporting and Corrective Action	11
9. Tools, Techniques and Methodologies	12
9.1. Software Quality Tools	12
9.2. Techniques and Methodologies	12
10. Media Control	12
11. Supplier Control	12
12. Record Collection, Maintenance, and Retention	12
13. Training	13

14. Risk Management	13
15. SQA Plan Change Procedure and History	14

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 Planit 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.

The Planit application will be an interactive visual piece of software which will assist users in raising a home-garden from scratch through an interactive plot of land, up-to-date information and monitoring tools. The user will also be able to search for specific gardening forums for either specialised consultation or to find a community that suits their interests.

The team consists of 7 members each in charge of a specific role with some crossover. 3 of the members are on backend development 1 of whom is dedicated and oversees the specific task. There also exists 3 people on front end development 1 of whom is dedicated to overseeing the task. Included in the team is also a quality assurance engineer. Finally, our respective managers and leaders are the Quality Assurance Manager, Release Manager, Project Manager and Lead Developer. The team will be responsible for all the development and documentation surrounding the project in order to ensure the product meets sufficient standards.

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
- System Requirement Specifications
- React documentation (<https://reactjs.org/docs/getting-started.html>)
- Node.js documentation (<https://nodejs.org/api/index.html>)
- MongoDB documentation (<https://docs.mongodb.com/>)
- Github documentation (<https://help.github.com/en/github>)
- 1008-1987-IEEE Standard for Software Unit Testing (<https://standards.ieee.org/standard/1008-1987.html>)
- 829-2008 - IEEE Standard for Software and System Test Documentation (<https://ieeexplore.ieee.org/document/4578383>)
- Learning node.js by Alex Zanfir (<https://www.linkedin.com/learning/learning-node-js-2/>)
- Kanban board (<https://www.atlassian.com/agile/kanban/boards>)

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 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 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.

3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

- CRUD operations on plant info component
- User Interface Pages
- User Registration/Login Component
- Warnings and Alerts Components
- Database system

- Searching external sources/communities

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management Process
- Change Management Process
- Maintainability Management Process
- Risk Management Process

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
- Project proposal
- Quality plan
- Project plan
- Risk Management
- Test plan
- Test cases and test coverage report
- Release plan
- Change management plan
- Configuration management plan
- Design report on software maintainability

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. The four most important qualities that we have identified in our application are: Functionality, Usability, Reliability and Consistency.

Firstly, to ensure that our web application is up to standard, we must ensure that the functions it provides are suitable, accurate, interoperable, compliant to industry guidelines and secure. The three main functions we have identified earlier should be well-suited to our customers' needs. The information given in our guides as well as our statistical predictions and visualization should be accurate. Our application also needs to conform to industry standards and have a secure database for storing private login credentials.

The second quality we have identified is Usability. As our web application is targeted towards beginner gardeners with little to no experience in gardening or growing their own food, it is crucial that

the system functions are designed in such a way that users can easily understand the application's functions and easily learn how to use the application.

The third quality identified is Reliability. Our system should be able to maintain its service levels for once the system is put into service. In case of component failure such as loss of internet connection, the system should also be able to recover within 15 seconds. Our system should have a very low frequency of failure, high fault tolerance and should be able to easily recover operations when failure occurs.

The last quality identified is Consistency. Conflicts between system requirements and confusion from users such as date-time format or units of measurement must be avoided and alternative use cases created when there are specific conditions that will affect consistency. (E.g. When the user chooses to use the Imperial System for units of measurement, or when the user specifically states that they are overseas and therefore do not have Singapore's characteristic temperate weather.)

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
- Unit Test Coverage
- Number of defects in production
- Problems per User-Month during User Acceptance Testing
- Length of Code / Program Size in Number of Lines
- Length of User Manual

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

We will be making use of a process of Test Driven Development (TDD) for the testing of our react components. By writing tests out in advance we can use the errors to guide our development making sure that problems and features are sufficiently considered by looking at the bigger picture. The strategy is as follows.

First, we will write a test that fails, followed by writing the minimum amount of code required to get the test passing. Lastly, we will refactor the code. As long as the code stays green we know that there is no regression and can write more efficient solutions with a more refined approach. There are many libraries that we can utilize for the testing of react components but we will be using React Testing Library, Jest.

8. Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in a centralized Reporting and Corrective Action System. We will be making use of our project backlog excel file, drive for file sharing and discord for voice meetups.

1. Problems / Issues
 - a. Task Description
 - b. Task Priority
 - c. Task Deadline
 - d. Task Assignee / Task Helpers
2. In progress
 - a. Google Drive and Discord meetings to be conducted at the mid and end of a task deadline

3. Review

- a. Issues to be reviewed by other SQ personnels
- b. Mark task as completed in project backlog otherwise leave blank

9. Tools, Techniques and Methodologies

9.1. Software Quality Tools

SQ personnel will require access to the following:

- Google Docs
- GitHub
- Microsoft Office tools (i.e., Excel)

9.2. Techniques and Methodologies

The primary methodology to be followed when developing the application is Test Driven Development (TDD). This involves writing tests of what we want a certain module of the software to do, and then implementing the code to make the tests pass.

10. Media Control

SQ deliverables will be documented in one of the following Google software applications: Google Docs and Google Sheets. Deliverables will be in soft copy, with the exception of completed checklists from process and product assessments. Google Docs track the document history and version changes. See Section 12 for additional details on the collection and retention of key records. Software Quality personnel will request space on the project's MediaWiki for SQ records. This MediaWiki is password protected and backed up nightly.

11. Supplier Control

Not applicable for this project

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. 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

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 bi-weekly risk management meetings and report any software risks to the QAM and the project manager.

We will be using the Scrum model to integrate risk management as part of the project life cycle. The project will be done in a series of sprints that spans 2 weeks. In every sprint, the team will complete a sub-task of the project. The sub-tasks will be reviewed and communicated to the project owner for feedback on areas of improvement. By doing so, we will reduce the risk of producing a product that does not match user's needs, identifying defects in the later stage of the project and not meeting project deadline/budget.

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.