Software Quality Assurance (SQA) Plan

By Kenneth Goh

Team Mawsters

Date: 22 February 2024

Signature Page

Prepared by: Kenneth Goh________________

Date:18 February 2024

Reviewed by 1: Muhammad Sufyan Bin Mohd Jais__

Date: 21 February 2024

Reviewed by 2: Nyan Maw Htun_____________

Date:22 February 2024

Approved by : Nyan Maw Htun____________

Date:22 February 2024

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	Initial Template	Kenneth Goh	19/2/24

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 shelvd 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.

Shelvd is a user-friendly book catalogue application made to simplify book discovery and management. The software items included in this plan encompass all components of the application. The front-end interface is developed with ViteJS, back-end services using Python and MySQL hosted on PlanetScale, and integration with external APIs such as Google Books API. The intended use of this application is to offer a user-friendly interface, efficient reading list management features, and personalised book recommendations via machine learning algorithms.

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
- ViteJS (https://vitejs.dev/guide/)

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 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:

- **User Interface (UI)**
  - The UI of shelvd must be user-friendly and responsive, ensuring ease of navigation and optimal performance. Features should be purposefully and clearly arranged, allowing users to intuitively understand their functionalities without extensive guidance. The interface should be designed to minimise the user's learning curve, enabling them to become comfortable with the application in a short time frame.
- **Recommended Page**
  - The recommended page should display accurate recommendations based on the user's data such as previous books read.
- **Book Details Page**
  - The book details page should fetch accurate information from the Open Library API, presenting key details like title, author, and cover image.

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- **Requirement Management Process**
  - The task of creating and executing requirement elicitation protocols, especially for the shelvd falls to the SQ staff. To fully address the functional requirements of shelvd, these protocols are necessary. To guarantee adherence and complete the definition of both functional and non-functional requirements, they must be integrated with regular team meetings.
- **Change Management Process**
  - Should team composition or procedures change, SQ staff members need to be prepared to evaluate the viability, simplicity, and readiness of putting these changes into practice. Changes to shelvd's technology, procedures, or team members fall under this category.
- **Maintainability Management Process**
  - Shelvd needs to be maintainable, which means that it should be easy for developers to read, allowing simple debugging, and be flexible enough to change as need be. The SQ team is in charge of setting baseline standards for maintainability and delineating the procedures necessary to meet them. To guarantee the caliber of the shelvd product, the SQ team must enforce these maintainability standards during the development phase.
- **Risk Management Process**
  - To ensure shelvd's resilience, the SQ team must oversee a robust risk management process. This involves identifying, assessing, and prioritising potential risks to functionality, security, and user experience. Through proactive measures and contingency plans, the SQ team must aim to minimise risks and maintain the app's reliability and performance.

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

1. Project Proposal
2. Project Plan
3. System Requirement Specifications
4. Risk Management Plan
5. Configurations Management Plan
6. Release Plan
7. Test Plan
8. Test Cases, and Test Coverage Report
9. Change Management Plan
10. Design Report on Software Maintability

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.

1. **Usability**
    a. Usability is crucial as the application's main goal is to provide a user-friendly book catalogue. This includes a user-friendly

interface that takes little time to learn so users can explore and take advantage of the application’s features.

2. **Maintainability**
    a. Maintainability is essential as the application will be a dynamic platform with potential updates and enhancements in the future. This feature makes sure the code is clean, well-documented, and easily modified for updates or fixes.
3. **Efficiency**
    a. Efficiency is key, ensuring that users have a seamless and responsive, especially when managing user data, book recommendations, and search functions.
4. **Reliability**
    a. To keep users engaged and trusting, shelvd must be dependable. The application should reliably and flawlessly carry out its intended tasks in a variety of scenarios.

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:

1. Test case pass rate
2. User satisfaction ratings
3. Issues per month

6. Software Reviews

6.1. Purpose

The purpose of this section is to outline the types of software reviews and peer evaluations that will be conducted and supported by the Software Quality (SQ) personnel. Based on the project milestone chart and the resource levels of SQ personnel, these reviews will ensure that the project adheres to its quality benchmarks and milestones.

6.2. Minimum Software Reviews

SQ personnel will assess the completeness, accuracy, and detail of the review packages to ensure they meet specified criteria and that all Requests for Action are documented and followed through to resolution. Reviews will verify that the process is conducted with the necessary stakeholders, appropriate documentation is presented, and entry and exit criteria are met. The following software reviews assessed by SQ include:

- 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.

8. Problem Reporting and Corrective Action

SQ personnel will leverage GitHub Project Boards as the centralized system for tracking and managing assessment findings and corrective actions. GitHub Project Boards, integrated seamlessly with the development workflow, serve as the primary platform for real-time collaboration and visibility into the project's status. Issues within the repository are utilized to represent individual assessment findings, allowing for detailed documentation including descriptions, labels, assignees, due dates, and comments. The Project Board columns, such as "To Do," "In Progress," and "Done," or customized to fit the workflow, provide a visual representation of the assessment lifecycle. Automation through GitHub Actions streamlines workflows, automatically moving issues across columns and triggering notifications based on predefined conditions. This approach ensures a transparent and efficient process for addressing and resolving identified problems in the shelvd development process, aligning with the collaborative nature of GitHub. Regular reporting and insights are facilitated through the visualization tools inherent in GitHub Projects, contributing to a proactive and streamlined corrective action strategy.

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)
- Version Control Systems (i.e., Git)

10. Media Control

SQ deliverables will be documented in the MediaWiki platform hosted on the school's secured server. All documentation will be in digital format, accessible to authorised personnel. Completed checklists from process and product assessments will be stored electronically on the MediaWiki platform. Refer to Section 12 for additional details on the collection and retention of key records. Software Quality personnel will ensure that all SQ records are appropriately organised and maintained on the MediaWiki platform. Access to the platform is restricted to authorised users and is password protected. Regular backups of the MediaWiki database are performed to ensure data integrity and availability.

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. Electronic copies will be maintained. SQ personnel will maintain electronic copies of all assessment reports and findings. SQ Project folders will contain copies 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

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 9001, ISO 9000-3

· CMMI

· Verification and Validation

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 project manager.

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.