

Software Quality Assurance (SQA) Plan
By Ge Zhuoyan, Li Pingrui, Wang Binli

Date: September 2021

Signature Page

Prepared by 1: Ge Zhuoyan

Date: 10/09/2021

Prepared by 2: Li Pingrui

Date: 10/09/2021

Prepared by 3: Wang Binli

Date: 10/09/2021

Approved by : Ge Zhuoyan

Date: 10/09/2021

Document Change Record

Revision	Description of Change	Approved by	Date

[illegible]

Contents

1. Purpose and Scope
 - 1.1. Purpose
 - 1.2. Scope
2. Reference Documents
3. Management
 - 3.1. Management Organisation
 - 3.1.1. Project Management
 - 3.1.2. Assurance Management
 - 3.2. Tasks
 - 3.2.1. Product Assessments
 - 3.2.2. Process Assessments
 - 3.3. Roles and Responsibilities
 - 3.3.1. QAM
 - 3.3.2. Software Quality Personnel
4. Documents
 - 4.1. Purpose
 - 4.2. Minimum Document Requirements

5. Standards, Practices, Conventions and Metrics
 - 5.1. Purpose
 - 5.2. Software Quality Programme
 - 5.2.1. Standard Metrics
6. Software Reviews
 - 6.1. Purpose
 - 6.2. Minimum Software Reviews
7. Test
8. Problem Reporting and Corrective Action
9. Tools, Techniques and Methodologies
 - 9.1. Software Quality Tools
10. Media Control
11. Record Collection, Maintenance, and Retention
12. Training
13. Risk Management
14. SQA Plan Change Procedure and History

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

StudyHub@NTU is a platform that enhances the learning-based social connectedness among NTU students. It is a web-based forum that provides students with the opportunity of exchanging important course information and opinions and questions about the course content. It also provides students with additional functionalities such as swapping course index, finding teammates and giving feedback to the course.

This project is delivered by a team of six NTU students: GE ZHUOYAN, LI PINGRUI, TAN BOON PING, TANG YUTING, WANG BINLI and WANG YING.

The following are the main software features covered in this Quality Assurance plan.

1. **Generate profile** - This feature allows users to enter personal information and their enrolled courses and view personal profile. Users may update their personal profile whenever they register for a new course and connect with their peers from the same course.
2. **Start a thread** - This feature allows users to start and post a thread to the system. It could be questions about course content or feedback regarding a specific course. The posted threads are readily available for all other users to view and reply.
3. **Find and respond to an existing thread** - This feature allows the user to find and respond to an existing thread. Users may reply/forward/save (as favorite) upon finding a thread of interest.
4. **Swap course index** - This feature allows users to post index swapping requests as well as view requests posted by others. It facilitates the index changing process during the course registration period.
5. **Find teammates** - This feature allows users to look for teammates for course projects.

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
- Project Proposal

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 component will be assessed by SQ personnel:

- Documents
- Project proposal
- Software Requirements Specification (SRS)
- Risk Management Plan
- Low-Fidelity Prototype
- Test Plan
- Prototype
- Configuration Management Plan
- Change Management Plan
- Release Plan
- Forum Module
- Teammate Searching Module

- Swap Index Module
- Account Management Module

Project proposal – The project proposal should define the scope of the project clearly. The solution proposed by the project team should fulfill all functional and nonfunctional requirements to properly address the problem. The technology approach should be feasible. The proposal should also state the deliverables, schedule and budget of the project clearly and avoid any ambiguity.

System Requirement Specification – The SRS should be developed according to the Requirements Elicitation and Analysis process conducted by the project team. The specification should specify the requirements for the system from multiple aspects, including functional requirements, input requirements, process requirements, operational requirements, hardware requirements, software and deployment requirements. All the aspects should be explained in detail and verifiable. The SRS should be trackable and clearly indexed.

Risk Management Plan – The Risk Management Plan should be developed according to the Risk Management Process conducted by the project team. The plan should identify potential risk factors and the degree of impact to the project. The Risk Management Plan should also define the response actions to the risks identified.

Low-Fidelity Prototype – The low-fi prototype should be able to demonstrate the flow of all the use cases and the transition between different states.

Test Plan – The Test Plan should be developed according to the Test Management Process conducted by the Development team and Quality Assurance team. The plan should describe the scope, approach, and schedule of test activities. The plan should clearly define the criteria of pass, fail and suspension. The test deliverables and testing environment should also be included in the plan.

Configuration Management Plan - The Configuration Management Plan should be developed according to the Configuration Management Process conducted by the project team. The plan should document and inform project stakeholders about Configuration Management within a project, what Configuration Management tools will be used, and how they will be applied by the project. The Configuration Management Plan details the methodology that the Project Manager (PM) and lead developer will use to control program documentations and the program baseline (Technical, Functional and Allocated).

Change Management Plan – The Change Management Plan should be developed according to the Change Management process as defined by the project team. The change management plan should define activities and roles to manage and control change during the execute and control stage of the project. Change should be measured against the project baseline.

Release Plan – The Release Plan should be developed according to the Release Management Process as defined by the Release Manager. The Release plan should clearly define the release requirements, release criteria, release goals as well as a release schedule overview.

Forum Module - The forum module should have the function of posting, replying, like and forwarding a thread. The module should fulfill the relevant non-functional requirements specified in SRS.

Teammate Searching Module - The teammate searching module should allow users to post teammate seeking information and apply to existing requests. The module should allow users to accept or decline an application to join the team. The module should fulfill the relevant non-functional requirements specified in SRS.

Swap Index Module - The swap index module should allow users to post index swapping requests and give consent to other users' requests. The module should fulfill the relevant non-functional requirements specified in SRS.

Account Management Module - The account management module should have the function of generating user profile based on the information input by the user. The module should fulfill the relevant non-functional requirements specified in SRS.

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management
- Project Analysis and Design Management
- Risk Management
- Implementation Management
- Project Monitoring and Control
- Test Management
- Release Management
- Maintenance Management

The uniform criteria, ISO 9126 Quality Model (2001) will be used to assess the above mentioned processes.

These processes will be audited and evaluated by SQ personnel during the prior internal audit period with related parties notified in advance and the software review period which will be presented by SQ personnel as well.

Subsequently, if a problem is raised without any visible resolution plan, the issue will then be logged. The list contains all unhandled issues that will be termly reviewed by the Product Manager to check whether it can be eliminated or evaded due to additional or new status reported. When a problem status changes to solved, SQ personnel and PM manger should go through to audit and assess again.

All reported issues together with recommended corrective actions and assessment data will be preceded as stated in the following section “ 8. Problem Reporting and Corrective Action” in this document.

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

- Software Requirement Specification
- Quality Plan
- User Documentation
- Source Code Documentation
- Risk Management Plan
- Design report on software maintainability
- Configuration Management Plan
- Change Management Plan
- Release 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.

As stated in the ISO 9126 Quality Model, the four qualities deemed most important for StudyHub@NTU are Functionality, Reliability, Usability and Maintainability.

- **Functionality**

StudyHub@NTU must be able to carry out the complete and correct functionalities as specified in the SRS. It must provide functions that meet the stated and implied requirements when it is used under the specified conditions.

- **Reliability**

StudyHub@NTU must be able to maintain stable performance under stated conditions during its course of running. It must not be laggy or have long down-time. After a system reboot, it must be able to restore full system functionality and user data.

- **Usability**

StudyHub@NTU must be able to ensure the quality of students' user experience when interacting with the systems. It must be easily understood, easily used and attractive to NTU students.

- **Maintainability**

StudyHub@NTU must be easy to repair and improve. Future modifications may include corrections, system updates and changes in the requirements and functional specification. That is, it must be able to be modified should there be a change in NTU's curriculum or course registration procedure.

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:

- Function point - Functionality
- Functionality Check based on Specified Use Case - Functionality
- Testing - Functionality
- Program Size - Reliability
- MTTF (Mean Time To Failure) - Reliability
- MTTR (Mean Time To Repair) - Reliability
- ROFOD (Probability of failure on demand) - Reliability, Usability
- No. of Error Messages - Reliability, Usability
- Fog Index - Usability

- ISP 9241-11- Usability
- Models like HPMAS - Maintainability
- Principle Components Analysis - Maintainability
- Aggregate Complexity Measure - Maintainability
- Factor Analysis - Maintainability
- Fan-in/Fan-out - Complexity of Overall App
- Depth of Inheritance tree - Complexity of Overall App
- Cyclomatic Complexity - Complexity of Overall App

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
- Quality Plan Review
- Risk Management Review
- Test Plan Review
- Acceptance Review
- Peer Review
- Code 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).

1. Unit Testing

The system will be tested on a functional level to verify the functionality of a specific section of codes. This is to ensure the correct functioning of the building blocks of the software.

2. Integration Testing

The system will be tested to verify the interfaces and interaction between integrated software components against the software design. This is to ensure the software works as a system.

3. System Testing

The completely integrated system will be tested to verify against the Requirement Specification. This is to make sure that the software behaves as expected and fulfills all the functional and non-functional requirements.

4. Acceptance Testing

The system will be tested to evaluate its compliance with the end-user requirement and make sure that it is ready for deployment. This is to understand the user experience and interaction with the system.

SQ personnel will review the test plan and make sure that appropriate testing methodologies are used at all four levels. SQ personnel will review and audit the test cases and test scenarios to ensure testing coverage.

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 generate, track, and trend assessment findings and observations in a EXCEL spreadsheet Reporting and Corrective Action System.

ID	Assesment Domain	Problem Discription	Corrective Action Proposed	Correction status	SQ review	PM review
1	Swap Index	Example 1	Example Action 1			
2	Start a thread	Example 2	Example Action 2			

SQ personnel will come up with corrective actions for each problem. The development team will mark the problem as solved after they execute the corrective action. SQ personnel will then check the status of execution and give feedback on whether the problem is effectively

resolved. If SQ personnel found the problem was not properly resolved, they will mark the problem as not resolved in the spreadsheet and notify the development team for revision.

The Project Manager will check the centralized spreadsheet weekly and check the progress or status.

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)
- Google Drive (For collaborative work)
- UML Modeling and Process Modeling tool (i.e Visual Paradigm)
- Prototype Designing tool (i.e Procreate)
- Progressive Framework (i.e Vue.js, Bootstrap, BootstrapVue and Firebase Realtime Database)
- Appropriate IDE (i.e WebStorm)

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.

1. MediaWiki
2. Git and GitHub
3. SVN
4. Google Drive
5. Visual Paradigm

MediaWiki is a free and popular web-based wiki engine and an extremely powerful and extensible software. Wiki is a hypertext system that is open on the World Wide Web and can be collaboratively authored by multiple people. MediaWiki keeps track of all submitted versions, which guarantee the easy reversion to any previous version. Its good openness allows you to easily view the corresponding documentations and resources from other sites that use the system. Thus even novices can quickly understand the operation method without backgrounds of HTML or CSS.

Git is an open source version control system that can effectively and quickly handle the version management of both small and large projects. Users can clone the complete Git repository (including code and version information) from the server to a single machine, create or merge branches and modify the code on their own machine according to different development purposes. It provides functionalities for users to merge their conflicts before submission as well. Github is a cloud-based Git repository service that provides functionalities such as sharing of codes which ensure well collaboration and easy edition.

Subversion (SVN) is an open source version control system. Subversion manages data that changes over time. These data are placed in a central repository. This archive is much like a normal file server, but it will remember every file change. In this way, you can restore the file to the old version, or browse the changed history of the file.

Google Drive is an online synchronization storage service of Google, which also combines the online document editing functions of Google Docs. It is used for the sharing and storage of documentations which ensure the efficiency of collaboration.

Visual Paradigm is a suite of design, analysis and management tools used for project development and digital transformation. Implemented as a UML Modeling and Process Modeling tool, Visual Paradigm's team collaboration function supports accessing to a central repository for managing, sharing and versioning projects. Team members are allowed to work on their own parts and then commit their work as well as update others' works.

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 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

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.

SQ personnel will pay close attention to the Covid-19 situation daily and follow the latest safety measurement required by the authority. SQ personnel should come up with preplans in response to potential risk caused by Covid-19.

SQ personnel will properly estimate the cost and benefit of switching to new tools during the process.

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.

