

Software Quality Assurance (SQA) Plan

By Drawdiculous

Date: September 2021

Signature Page

Prepared by: He Yinan Date: 15/09/21

Reviewed by 1: Jin Han Date: 15/09/21

Reviewed by 2: Renganathan Ramasamy Date: 15/09/21

Approved by : Sannabhadti Shipra Deepak Date: 15/09/21

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	Initial Template	Hengwei	15/09/21

Contents

1.	5
1.1.	5
1.2.	错误!未定义书签。
2.	5
3.	6
3.1.	6
3.1.1.	6
3.1.2.	6
3.2.	6
3.2.1.	错误!未定义书签。
3.2.2.	错误!未定义书签。
3.3.	7
3.3.1.	7
3.3.2.	7
4.	7
4.1.	7
4.2.	7
5.	错误!未定义书签。
5.1.	8
5.2.	8
5.2.1.	错误!未定义书签。
6.	9
6.1.	9
6.2.	9
7.	9
8.	错误!未定义书签。
9.	错误!未定义书签。
9.1.	错误!未定义书签。
10.	错误!未定义书签。
11.	错误!未定义书签。
12.	11
13.	12
14.	12

15. 错误!未定义书签。

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 “Drawdiculous” 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.

The primary purpose of the mobile application, “Drawdiculous”, is to cater to the social and cognitive needs of the elderly through a social drawing game. Design considerations have been specifically made to reflect this objective.

Firstly, the design ensures that it is as interactive as possible and that 8 people are able to participate in a single game. This social feature is specifically integrated to promote communication in order to effectively bolster the elderly user’s sense of well-being. An additional benefit of this feature is the competitive element which is designed to provide an auxiliary incentive for certain individuals. The game will also function in real-time, to ensure that it runs smoothly.

Second, the game will specifically include relatable and familiar words. The intent behind this is to aid the elderly users’ memory and recall. In addition, we ensure that there are at least 1000 unique words to decrease the chance that the game participants encounter certain words repeatedly.

Lastly, the design layout will be clean and easy to understand in order to prevent confusion while playing. This will allow the elderly users to focus purely on the

game. To achieve a minimalist design, we will design our user interface using the Shneiderman's Eight Golden Rules.

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

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:

- Drawdiculous application that meets all functional and nonfunctional requirements
- Server script supporting the application

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement management process
- Change management process
- Code quality 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
- Quality Plan
- Software Model Prototype
- Risk Management Report
- Design report on software maintainability
- Configuration Management Plan
- Change Management Plan
- Release Plan
- Test Plan

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.

Functionalities should be correctly implemented as a basic requirement to ensure that all users of Drawdiculous can play the game as we designed and provide them with a good gaming experience.

Efficiency is very important in our product in the performance of real-time synchronization of drawings. Our product should be efficient enough to ensure that the drawing is updated at least 10 frames per second.

Simplicity of all pages of the application will be maintained as we are targeting the elderly. The application will be easy to navigate, and all functions will be kept simple to prevent any difficulties of using the application.

Consistency in pages also helps the elderly users by minimizing the UI operations they have to learn. It also keeps the design neat and thus enhances the user's experiences.

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:

- Length of code
- Fog index
- Cyclomatic complexity

- Weighted methods per class
- Number of overriding operations
- Depth of inheritance 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 artifacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

To ensure our application is able to meet its specifications, customer needs and is defect free, SQ personnel will conduct both functional testing and non-functional testing.

Functional testing:

1. Unit Testing:

Unit testing tests individual modules to check if there are any issues. It tests the functional correctness of the standalone modules. The SQ personnel will follow the Unit Testing Lifecycle to conduct Black Box Testing and White Box Testing. Unit testing could reduce defects in newly developed features at an early stage to reduce internal failure cost.

2. Integration Testing:

Software modules are integrated logically and tested as a group in integration testing. This could expose defects in the interaction between different software modules. To do so, The SQ personnel will adopt a Bottom Up Approach (Incremental Approach) whereby lower level modules are tested first. The tested modules will later facilitate testing the higher level modules. The SQ personnel may start testing once a lower level module is developed and does not need to wait for all modules to be developed.

3. User Acceptance Testing:

User Acceptance Testing will be conducted in the last phase of the software development. Target users will test the application to ensure that all requirements and specifications are met.

Non-functional testing:

1. Compatibility Testing

Application will be tested to check that it is capable of running on different Android mobile devices with different Android Operating System versions. To do so, SQ personnel will conduct Backward Compatibility Testing whereby compatibility of application is tested on newer versions of Android Operating System first before testing on older versions.

8. Problem Reporting and Corrective Action

SQ personnel generate, track, and trend assessment findings and observations in a centralized Reporting and Corrective Action System.

In the event error or defects are discovered due to an error in the application code, the person that discovered the problem will follow the following procedure:

1. Report the problem details to the Product Manager.
2. The product manager notifies the Lead Developer and corresponding developer of the problem.
3. Team members will be assigned to recreate and solve the problem. The Lead Developer will provide technical suggestions to team members in locating the problem.
4. Team members solve the problem and notify the Product Manager and Lead Developer when the problem is solved. If team members have any trouble in

solving the problem, they will meet with the Product Manager and Lead Developer to devise a plan.

5. Project Manager will assign Quality Assurance Engineers to test the application to check that the problem is solved.

In the event that documentation requires rectification because it is outdated or there are errors in the documentation, the person that discovered the problem will follow the following procedure:

1. Report the problem details to the Product Manager.
2. Product Manager will assign the task of document rectification to team members that are familiar with the area of problem.
3. Team members will solve the problem and notify the Product manager once the problem is solved. If the team members encounter any difficulties, team members will notify the Product Manager and Lead Developer.

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)
- Github
- Android Studio
- Jira Software
- Visual Paradigm
- IntelliJ IDEA

10. Media Control

SQ deliverables related to Drawdiculous Application will be documented in Microsoft Word and through Google Drive Cloud Storage. 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

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	3 months
SQA Checklists	SQ Personnel	3 months
Deliverable Defects	SQ Personnel	3 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 weekly risk management meetings and report any software risks to the QAM and the project manager.

A preliminary list of risk factors have been identified and will be monitored closely. These preliminary risk factors are as follows:

- Lack of time due to other commitments
 - Some group members may be unable to finish their allocated tasks by the deadlines as they have tests and projects from other courses
- Refresher course for developers who have not worked with Android Studio/Java in some time
 - Our project uses Java as our main programming language
 - Additional time may be needed to refamiliarize with the language

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. The QAM will then view the proposal and approve any necessary changes. In order to keep track of all the changes, a version history will be maintained. If any of the changes need to be reverted in the future, old copies of the plan will be restored.