**WIF3002 Software Process and Quality**

**Visualization Project**

**MyCovidView**

**Software Quality Assurance (SQA) Plan**

**Table of Contents**

[1.0](#_3eiydpfwths7) Purpose 1

1[.2](#_1t3h5sf) Tasks 1

2[.0](#_mlugqyr8ins) Reference 2

3[.0](#_3dpw5od9pjyn) Management 3

2[.1](#_wo1fb19werlm) Organization 3

2[.2](#_wo1fb19werlm) Resources 5

4[.0](#_mqtxfixiypzh) Problem Reporting and Corrective Action 5

5[.0](#_e6hi5akev1u2) Tools, Techniques and Methodologies 6

5[.2](#_4k668n3) Software Quality Tools 6

5[.3](#_2zbgiuw) Project Tools 6

6[.0](#_b62n8zbyz47u) Code Control 6

7[.0](#_ft5pn72uq2kx) Record Collection, Maintenance, and Retention 7

8[.0](#_e6hi5akev1u2) Testing Methodologies 7

# 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 WIF3002 Visualization project.

The Software Quality Assurance 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 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 the products, processes, and services will be evaluated to ensure they meet requirements and comply with the relevant policies, standards, and procedures, including the ISO 25000 Software and Data Quality standards.

## Tasks

The SQA activities will be conducted throughout the software development life cycle. The following tasks will be performed:

1. Requirements Analysis
   1. Review and analyze the software requirements specification.
   2. Ensure that the requirements are clear, consistent, and testable.
   3. Identify any ambiguities or missing requirements.
   4. Collaborate with stakeholders to clarify requirements, if necessary.
2. Test Planning
   1. Define the test objectives, scope, and test deliverables.
   2. Identify test scenarios, test cases, and test data.
   3. Prioritize testing activities based on risk and criticality.
   4. Develop a test schedule and allocate resources accordingly.
   5. Define the test environment and necessary tools.
3. Test Design and Execution
   1. Design test cases and test scenarios based on the requirements.
   2. Define the expected results and acceptance criteria for each test case.
   3. Execute test cases and record the actual results.
   4. Report and track defects using a defect tracking system.
   5. Conduct regression testing to ensure changes do not impact existing functionality.
   6. Perform system integration testing to verify the interoperability of various components.
4. Performance Testing
   1. Define performance benchmarks and objectives.
   2. Develop performance test scenarios and scripts.
   3. Execute performance tests to measure system response times, throughput, and scalability.
   4. Identify and resolve performance bottlenecks, if any.
5. Documentation and Reporting
   1. Document test plans, test cases, and test results.
   2. Prepare regular status reports to stakeholders.
   3. Maintain traceability between requirements, tests, and defects.
   4. Document lessons learned and recommendations for future improvements.

# Reference

The following documents were used or referenced in the development of this plan:

1. IEEE STD 730-2002, IEEE Standard for Software Quality Assurance Plans
2. ISO/IEC 25000 Standards for Software Data and Quality
3. Whole Project Plan

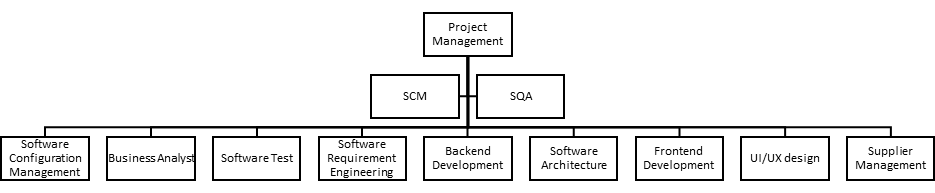
# 

# Management

## Organization

Good software practice requires a measure of independence for the SQA group. This independence provides a key strength to SQA; that is, SQA has the freedom, if the quality of the product is being jeopardized, to report this possibility directly above the level of the project. While in practice this rarely occurs, for almost all problems are correctly addressed at the project level, the fact that the SQA group can go above the project level gives it the ability to keep many of these problems at the project level.

Figure below shows the SQA organization with relation to the MyCovidView project organization



SQA is responsible for ensuring compliance with SQA requirements as delineated in this SQA Plan. The SQA organization assures the quality of deliverable software and its documentation, non-deliverable software, and the engineering processes used to produce software.

The following describes the functional groups that influence and control software quality.

1. Software Quality Assurance
   1. To ensure that the software meets all the specified requirements and is free from bugs and errors.
   2. To ensure the development process comply with the relevant standard practices to minimize the likelihood for an error to occur.
   3. To develop test plans and execute them to find and report defects.
2. Supplier Management
   1. To manage commercial-off-the-shelf (COTS) software
   2. To ensure the goods or services the suppliers provide meet the project’s quality standards and deadlines.
   3. To verify the software processes.
   4. To handle any issues that arise with the software.
3. UI/UX Design
   1. To design the user interface and user experience of the software.
   2. To make the software intuitive, easy to use and visually appealing.
4. Frontend Development
   1. To implement the visual and interactive elements of a web application that users interact with directly.
   2. To embed the visualization graphs into the web application.
5. Software Architecture
   1. To design high-level structure of the software system
   2. To choose the appropriate technologies and patterns.
   3. To ensure the software is scalable, maintainable and meet both technical and business requirements.
6. Backend Development
   1. To develop server-side web application logic which includes core application logic, data and application integration, API and other back end processes.
   2. To help out the teams on building the graphs needed by using the chosen visualization tool.
7. Software Requirement Engineering
   1. To define, document and maintain the software requirements
   2. To communicate with stakeholders to understand their needs and translate them into technical requirements.
8. Software Test
   1. To test the software for bugs, errors and other issues.
   2. To design and execute test plans and report their findings to the development team.
9. Business Analyst
   1. To understand the business needs and translate them into specific software requirements.
   2. To serve as a liaison between the business side and the technical side of a project.
10. Software Configuration Management
    1. To track and control changes in the software which includes managing software releases, maintaining version control systems
    2. To ensure the integrity and traceability of the software configuration.

# Software Quality Policy

These are the policy of quality that we decided on:

* Customer attention

Customer's needs and expectations are the foremost key criteria, which helps in achieving the quality prospects in a product. Therefore, the focus should be made on briefing the present and future demands of the customer, along with their fulfillment. Therefore the visualization of MyCovidView should adhere to the Software Requirement Specification.

* Leadership

The leadership and managerial process should be able to simulate and create a motivational and enthusiastic work environment, to bring the best out of each individual, towards the quality of MyCovidView.

* Constant and continual improvement

Every individual involved in the development should strive for continuous improvement in the procedure and approach, so as to improve the quality of the product. Necessary update to the relevant documentation of processes will be carried out as needed

* Process

Should reflect the adherence and following of all standard practices and processes, which contributes towards the quality. This includes ISO Standards for Quality Assurance and IEEE Software Engineering Standard.

# 

# 

# Tools, Techniques and Methodologies

SQ personnel will be utilizing the following tools:

## Documentation Tools

* Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
* Google Services (i.e., Drive, Docs, Slides)

## Project Tools

* Github Repository
* Figma
* Canva

## Management Overseeing Tool

The development process follows the scrum methodology. [Add description]

* Trello
* ICT-INOV

# SQA Processes

## Task Definition and Distribution/Assignment

The task related to SQA processes are as follows:

| Task | Description of Task | Team Responsible | QA Member Review |
| --- | --- | --- | --- |
| Supplier Management |  | First Minute | aiyin |
| UI/UX Development |  | Group 4 | Kayshav |
| Frontend Development |  | group 5 | Syah |
| Software Architecture Design |  | Last Minute | GanJW |
| Backend Development |  | Galaxy | Ravin |
| Software Requirement Engineering |  | Fantasy | zy |
| Software Testing |  | Impact | Sulaiman |
| Mobile App Development | Figma developer for web user interface (wireframe) | Group 11 | Kai Yang |
| Mobile App Development | Figma Developer for Web User Interface | Group 12 | Zhi Guang |
| Business Analysis |  | Group 13 | cf |
| Software Configuration Management |  | Hell’s Kitchen | Ashiq |

# Code Control

Code control includes the items listed below:

1. Identifying, labeling, and cataloging the software to be controlled
2. Identifying the physical location of the software under control
3. Identifying the location, maintenance, and use of backup copies
4. Distributing copies of the code
5. Identifying the documentation that is affected by a change
6. Establishing a new version
7. Regulating user access to the code.

Visualization Project uses GitHub for code control.

# 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. Example records include the process and product assessments reports, completed checklists, the SQ Activity Schedule, metrics, weekly status reports, etc.

## Documents Related to the Project

1. Software Requirement Specification

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign.

Group Fantasy is responsible to document the requirement of MyCovidView in a comprehensive SRS, the srs will be evaluated by the SQA Member responsible.

1. UML Use Case Diagram

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign.

Group Fantasy is responsible to document the requirement of MyCovidView in a comprehensive SRS, the srs will be evaluated by the SQA Member responsible.

1. Software Configuration Management Plan
2. Software Testing Plan

# Testing

The Testing of the Visualization software will be conducted in accordance with the Software Testing Plan prepared by the Software Testing group.

SQ personnel will assure that the test management processes and products are being implemented per the Software Management Plan and /or Test Plan(s). 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.