**Real-time Stabilization and 3D Reconstruction of Hand Gestures and Finger Movement Traces Using LED-Equipped Gloves**



Software Quality Assurance Plan (SQA)

**Version No. 1.0**

**Project Document Revision History**

| **VersionNumber** | **Date** | **Revision Author** | **Description of Revision** |
| --- | --- | --- | --- |
| 1.0 | 12/08/2024 | Team 5 | Original Published Version |
|  |  |  |  |

# I. Table of Contents

[**I. Table of Contents 3**](#_heading=h.96w65dfa41xf)

[**1.0 Introduction 6**](#_heading=h.5ne63xxnqevh)

[1.1 Scope and intent of SQA activities 6](#_heading=h.v8byampvxy5k)

[1.2 SQA organizational role 6](#_heading=h.62kpug5dro0e)

[**2.0 SQA Tasks 6**](#_heading=h.cjvz7m7rxdwv)

[2.1 Task Overview 6](#_heading=h.e2s81jenn6mu)

[2.1.1 Code Quality Assurance 6](#_heading=h.li1hsoau2yxv)

[2.1.2 Unit Testing 7](#_heading=h.7pkihyehmkyr)

[2.1.3 Integration Testing 7](#_heading=h.w512gft5m7qi)

[2.1.4 System Testing 7](#_heading=h.mgqw8ro95532)

[2.1.5 Performance Testing 7](#_heading=h.oaizvujguvy5)

[2.1.6 Usability Testing 7](#_heading=h.662yl2bobg5k)

[2.1.7 Documentation Review 7](#_heading=h.4k1xps6v8tny)

[2.1.2 Work products and documentation 8](#_heading=h.lgvei44yrm1w)

[2.2 Standards, Practices and Conventions (SPC) 8](#_heading=h.83g57rbvcjiu)

[2.3 SQA Resources 8](#_heading=h.4dlcwiu7k7dt)

[**3.0 Reviews and Audits 8**](#_heading=h.4hxkbauw4dh5)

[3.1 Generic Review Guidelines 8](#_heading=h.of20mwso8s4s)

[3.1.1 Conducting a Review 8](#_heading=h.co67fr4n37bt)

[3.1.2 Roles and Responsibilities 9](#_heading=h.6gfethnvpj6k)

[3.1.3 Review work products 9](#_heading=h.5w5s21d1grav)

[3.2 Formal Technical Reviews 9](#_heading=h.31nyb596338w)

[3.2.1 System Specification Review 9](#_heading=h.v9484uglr5sx)

[3.2.1.1 Description and Focus of the Review 9](#_heading=h.d3wyflrp9uab)

[3.2.1.2 Timing of the Review 9](#_heading=h.8b9naz1qldvv)

[3.2.1.3 Work Products Produced 9](#_heading=h.xnlows2ph505)

[3.2.1.4 Review and Checklist 9](#_heading=h.ndpu2ho2kv40)

[3.2.2 Software Project Plan Review 10](#_heading=h.7ky92r7t98sg)

[3.2.2.1 Description and Focus of the Review 10](#_heading=h.v9znueoh4emp)

[3.2.2.2 Timing of the Review 10](#_heading=h.qlz81jhzv6rb)

[3.2.2.3 Work Products Produced 10](#_heading=h.4bzoagcvmpbv)

[3.2.2.4 Review and Checklist 10](#_heading=h.rnjonuxqdtl3)

[3.2.3 RMMM Review 10](#_heading=h.czrc4tfsfkgt)

[3.2.3.1 Description and Focus of the Review 10](#_heading=h.zegua2mw7bro)

[3.2.3.2 Timing of the Review 10](#_heading=h.f65bqmnu9px)

[3.2.3.3 Work Products Produced 10](#_heading=h.3xht64k8514v)

[3.2.3.4 Review and Checklist 10](#_heading=h.k38n9cpl4cdh)

[3.2.4 Requirements Reviews (Models, Specification) 11](#_heading=h.9exantui93b5)

[3.2.4.1 Description and Focus of the Review 11](#_heading=h.48m6fz6rhzsz)

[3.2.4.2 Timing of the Review 11](#_heading=h.5apdap130tpj)

[3.2.4.3 Work Products Produced 11](#_heading=h.tdkfaj57gznr)

[3.2.4.4 Review and Checklist 11](#_heading=h.mg761m3bll32)

[3.2.5 Data Design Review 11](#_heading=h.n2bmjz7gz7fh)

[3.2.5.1 Description and Focus of the Review 11](#_heading=h.nf60f6150mas)

[3.2.5.2 Timing of the Review 11](#_heading=h.t0zr3ca8xjon)

[3.2.5.3 Work Products Produced 11](#_heading=h.iavx4yet8amu)

[3.2.5.4 Review and Checklist 11](#_heading=h.xwalgufo4cx0)

[3.2.6 Architectural Design Review 11](#_heading=h.agr4b3zcqizp)

[3.2.6.1 Description and Focus of the Review 11](#_heading=h.4dj41fxiwiue)

[3.2.6.2 Timing of the Review 12](#_heading=h.ol1qdira2sqa)

[3.2.6.3 Work Products Produced 12](#_heading=h.qijbyrkxzxju)

[3.2.6.4 Review and Checklist 12](#_heading=h.2lb79wy17qkl)

[3.2.7 Interface (GUI) Design Review 12](#_heading=h.t74lj5vmznzh)

[3.2.7.1 Description and Focus of the Review 12](#_heading=h.3anws8xakacp)

[3.2.7.2 Timing of the Review 12](#_heading=h.boj9ytlpqces)

[3.2.7.3 Work Products Produced 12](#_heading=h.29w3sxbn7wrw)

[3.2.7.4 Review and Checklist 12](#_heading=h.71r4w1n4265o)

[3.2.8 Component Design Review(s) 12](#_heading=h.aobqvdeczq3d)

[3.2.8.1 Description and Focus of the Review 12](#_heading=h.3qnz54wqknhi)

[3.2.8.2 Timing of the Review 12](#_heading=h.mi55myxrt0i2)

[3.2.8.3 Work Products Produced 13](#_heading=h.v7tquhec89hj)

[3.2.8.4 Review and Checklist 13](#_heading=h.mvd0xxf6fjwr)

[3.2.9 Code Reviews 13](#_heading=h.u31mpm13vrgl)

[3.2.9.1 Description and Focus of the Review 13](#_heading=h.i7e7ydh4j3pq)

[3.2.9.2 Timing of the Review 13](#_heading=h.ewiy2y7lyl60)

[3.2.9.3 Work Products Produced 13](#_heading=h.89h4xpethceu)

[3.2.9.4 Review and Checklist 13](#_heading=h.8but23wogfoy)

[3.2.10 Test Specification Review 13](#_heading=h.7ns0r7b8b049)

[3.2.10.1 Description and Focus of the Review 13](#_heading=h.8ixcxvin8fbv)

[3.2.10.2 Timing of the Review 13](#_heading=h.bvt4dza9pka1)

[3.2.10.3 Work Products Produced 13](#_heading=h.esj7ccfbufk0)

[3.2.10.4 Review and Checklist 14](#_heading=h.3ue7excangv)

[3.2.11 Change Control Reviews and Audits 14](#_heading=h.sanujxmzugqg)

[3.2.11.1 Description and Focus of the Review 14](#_heading=h.d1mwfdowoz09)

[3.2.11.2 Timing of the Review 14](#_heading=h.dvbms7aenzl8)

[3.2.11.3 Work Products Produced 14](#_heading=h.2wujdxr2u965)

[3.2.11.4 Review and Checklist 14](#_heading=h.6gahqxnfh8xi)

[3.3 SQA Audits 14](#_heading=h.q83gk2v5kuh3)

[3.3.1 Description of Audits 14](#_heading=h.ixoza04g5us5)

[3.3.2 Types of Audits 14](#_heading=h.n6scd2x6d94i)

[3.3.3 Audit Schedule 14](#_heading=h.mir21cz5pkju)

[3.3.4 Audit Reports 14](#_heading=h.ch4bdrz7bb5u)

[**4.0 Problem Reporting and Corrective Action/Follow-up 15**](#_heading=h.mwu375btqb5z)

[4.1 Reporting mechanisms 15](#_heading=h.vhuy07k742s0)

[4.2 Responsibilities 15](#_heading=h.mrq1pzk71tkw)

[4.3 Data collection and evaluation 15](#_heading=h.ho439pk0gir9)

[4.4 Statistical SQA 16](#_heading=h.imlrdak4m384)

[**5.0 Software Process Improvement Activities 16**](#_heading=h.4d4p1d46di4r)

[5.1 Goal and objectives of SPI 16](#_heading=h.gokjldf3ub8b)

[5.2 SPI tasks and responsibilities 17](#_heading=h.221yubfdt9xf)

[**6.0 Software Configuration Management Overview 17**](#_heading=h.37vwpeoy9uee)

[6.1 SCM Plan Overview 17](#_heading=h.vvjrccv0eha3)

[**7.0 SQA Tools, Techniques, Methods 18**](#_heading=h.6em0dtyoksr3)

[7.1 Tools 18](#_heading=h.ojwx5tqvj6hl)

[7.2 Techniques 18](#_heading=h.a8rcpfbxzdb1)

[7.3 Methods 18](#_heading=h.37grm28tyg2j)

[**8.0 Appendix 18**](#_heading=h.sq20j6thwogc)

[8.1 References 18](#_heading=h.qc4lcti00ptv)

[8.2 Acronyms and Abbreviations 19](#_heading=h.wjdufnvhaxmz)

[8.3 System traceability matrix 19](#_heading=h.gljk5kp8ukjj)

[**9.0 Review and Sign Off 20**](#_heading=h.4d34og8)

# 1.0 Introduction

## 1.1 Scope and intent of SQA activities

The SQA activities for this project aim to ensure that the **Adaptive HCI mobile application** meets the highest standards of quality, reliability, and performance. The primary goal is to provide accurate, real-time stabilization and tracking of hand gestures for patients with Parkinson’s disease using **LED-equipped gloves** and **machine learning algorithms**. The pathways of hand movements will be tracked, smoothed, and visualized within the app, ultimately being ported to VR environments.

**Key Features:**

* **Real-time Gesture Tracking:** Using LED-equipped gloves and video processing.
* **Smoothing Techniques:**
  + Image Processing: **Kalman Filter**, **Moving Average**.
  + Machine Learning: **CNN** and **LSTM Models**.
* **Visualization:** Pathways rendered using **Spline Interpolation**.
* **Portability to VR** for enhanced user experience.

## 1.2 SQA organizational role

The SQA team is embedded within the project team and reports to the **Project Manager**. The SQA team collaborates closely with developers, testers, and stakeholders to ensure quality at each phase. The roles include:

* **SQA Lead**: Oversees all SQA activities and reports findings.
* **Developers**: Implement code and participate in code reviews.
* **Testers**: Execute tests and report defects.
* **Project Manager**: Ensures SQA activities align with project timelines and goals.

# 2.0 SQA Tasks

## 2.1 Task Overview

This section details the key SQA tasks for the project, assigning responsibilities and defining the work products produced.

### 2.1.1 Code Quality Assurance

* **Description**: Ensure all code follows the established coding standards (PEP 8 for Python, Google Java Style for Android). Conduct regular code reviews to identify and fix issues early.
* **Responsibility**: Soham Naik, Deniz K. Acikbas
* **Work Products**: Code review reports, checklist results

### 2.1.2 Unit Testing

* **Description**: Test individual functions and modules to verify they work as intended. Mock objects will be used to isolate functionality.
* **Responsibility**: Soham Naik, Zaynab Mourtada
* **Work Products**: Unit test reports, test logs

### 2.1.3 Integration Testing

* **Description**: Test interactions between system components (e.g., gesture tracking, ML smoothing, VR/AR integration) to ensure seamless operation.
* **Responsibility**: Deniz K. Acikbas, Alan Raj
* **Work Products**: Integration test plans, test results

### 2.1.4 System Testing

* **Description**: Perform end-to-end testing of the entire system under real-time conditions.
* **Responsibility**: Entire Team
* **Work Products**: System test reports, bug logs

### 2.1.5 Performance Testing

* **Description**: Verify that the system meets real-time performance requirements, processing 60Hz video streams with minimal latency.
* **Responsibility**: Soham Naik, Deniz K. Acikbas
* **Work Products**: Performance test reports, latency benchmarks

### 2.1.6 Usability Testing

* **Description**: Ensure the interface is user-friendly and accessible for patients with Parkinson’s disease.
* **Responsibility**: Alan Raj, Zaynab Mourtada
* **Work Products**: Usability test reports, feedback forms

### 2.1.7 Documentation Review

* **Description**: Review project documentation to ensure clarity, accuracy, and completeness.
* **Responsibility**: Entire Team
* **Work Products**: Documentation review notes, checklists

### 2.1.2 Work products and documentation

* Code Review Reports
* Unit Test Reports
* Integration Test Reports
* System Test Reports
* Performance Test Logs
* Usability Test Reports

## 2.2 Standards, Practices and Conventions (SPC)

* **Coding Standards**: PEP 8 (Python), Google Java Style (Android)
* **Version Control**: GitHub with Git Flow branching model
* **Testing**: PyTest (Python), Espresso (Android)
* **CI/CD**: GitHub Actions for automated testing and deployment
* **Documentation**: Clear, versioned documents stored in Google Drive

## 2.3 SQA Resources

* **Personnel**:
  + Soham Naik (Image Processing)
  + Deniz K. Acikbas (Front-End & Hardware Integration)
  + Alan Raj (3D Visualization)
  + Zaynab Mourtada (ML Smoothing)
* **Tools**:
  + GitHub (Version Control)
  + Hive (Project Management)
  + Google Drive (Document Storage)
  + Discord (Communication)

# 3.0 Reviews and Audits

## 3.1 Generic Review Guidelines

### 3.1.1 Conducting a Review

* **Preparation**: Distribute review materials in advance.
* **Execution**: Follow structured agendas.
* **Follow-Up**: Address identified issues promptly.

### 3.1.2 Roles and Responsibilities

* **Author**: Presents the work for review.
* **Reviewer**: Evaluates the work product.
* **Scribe**: Documents findings and action items.

### 3.1.3 Review work products

* Code review checklists
* Design review notes
* Test plans and logs

## 3.2 Formal Technical Reviews

Formal Technical Reviews (FTR) are planned and structured meetings where work products are evaluated by a team of qualified personnel to identify defects and ensure quality standards are met. The following sections provide descriptions, timing, and deliverables for each major review conducted during the software process.

### 3.2.1 System Specification Review

#### 3.2.1.1 Description and Focus of the Review

This review ensures that the system specifications meet all functional, performance, and interface requirements as defined by the project stakeholders.

#### 3.2.1.2 Timing of the Review

The review is conducted after the initial system specification document is drafted and before development begins.

#### 3.2.1.3 Work Products Produced

* System specification review report
* List of identified issues and recommendations

#### 3.2.1.4 Review and Checklist

The following checklist will be conducted:

* Are all functional requirements clearly defined?
* Are performance requirements adequately specified?
* Are external interfaces fully documented?

### 3.2.2 Software Project Plan Review

#### 3.2.2.1 Description and Focus of the Review

Evaluate the completeness and feasibility of the software project plan, including schedules, resources, and risk mitigation strategies.

#### 3.2.2.2 Timing of the Review

The review is conducted after the project plan is drafted and before formal project initiation.

#### 3.2.2.3 Work Products Produced

* Project plan review report
* List of risks and mitigation strategies

#### 3.2.2.4 Review and Checklist

The following checklist will be conducted:

* Is the schedule realistic and achievable?
* Are roles and responsibilities clearly defined?
* Are risks adequately identified and mitigated?

### 3.2.3 RMMM Review

#### 3.2.3.1 Description and Focus of the Review

Review the Risk Mitigation, Monitoring, and Management (RMMM) plan to ensure all potential risks are identified and appropriate mitigation strategies are defined.

#### 3.2.3.2 Timing of the Review

The review is conducted after the project plan is approved and during initial project phases.

#### 3.2.3.3 Work Products Produced

* RMMM review report
* Updated risk mitigation strategies

#### 3.2.3.4 Review and Checklist

The following checklist will be conducted:

* Are high-impact risks identified?
* Are mitigation plans actionable and clear?

### 3.2.4 Requirements Reviews (Models, Specification)

#### 3.2.4.1 Description and Focus of the Review

Ensure that requirements models and specifications meet user needs and system objectives.

#### 3.2.4.2 Timing of the Review

The review is conducted after requirements gathering and documentation.

#### 3.2.4.3 Work Products Produced

* Requirements review report
* Approved requirements document

#### 3.2.4.4 Review and Checklist

The following checklist will be conducted:

* Are requirements complete and unambiguous?
* Are conflicting requirements resolved?

### 3.2.5 Data Design Review

#### 3.2.5.1 Description and Focus of the Review

Evaluate the design of the data structures, database schemas, and data flow to ensure they support system functionality.

#### 3.2.5.2 Timing of the Review

The review will be conducted before implementation of data components begins.

#### 3.2.5.3 Work Products Produced

* Data design review report
* Database schema diagrams

#### 3.2.5.4 Review and Checklist

The following checklist will be conducted:

* Is the data model normalized?
* Are data integrity constraints defined?

### 3.2.6 Architectural Design Review

#### 3.2.6.1 Description and Focus of the Review

Ensure the architecture meets system performance, scalability, and maintainability requirements.

#### 3.2.6.2 Timing of the Review

The review will be conducted before detailed design begins.

#### 3.2.6.3 Work Products Produced

* Architecture design review report
* Architecture diagrams

#### 3.2.6.4 Review and Checklist

The following checklist will be conducted:

* Does the architecture support modular design?
* Are performance and scalability concerns addressed?

### 3.2.7 Interface (GUI) Design Review

#### 3.2.7.1 Description and Focus of the Review

Evaluate user interface designs for usability, accessibility, and consistency.

#### 3.2.7.2 Timing of the Review

The review will be conducted before front-end development begins.

#### 3.2.7.3 Work Products Produced

* GUI design review report
* Wireframes and prototypes

#### 3.2.7.4 Review and Checklist

The following checklist will be conducted:

* Is the UI design intuitive for end-users?
* Are accessibility standards met?

### 3.2.8 Component Design Review(s)

#### 3.2.8.1 Description and Focus of the Review

Review the design of individual software components to ensure they meet functional and performance requirements.

#### 3.2.8.2 Timing of the Review

The review will be conducted before coding of each major component begins.

#### 3.2.8.3 Work Products Produced

* Component design review report
* Design diagrams

#### 3.2.8.4 Review and Checklist

The following checklist will be conducted:

* Is the design modular and reusable?
* Are dependencies minimized?

### 3.2.9 Code Reviews

#### 3.2.9.1 Description and Focus of the Review

Ensure the code adheres to coding standards and best practices.

#### 3.2.9.2 Timing of the Review

The review will be conducted regularly during development.

#### 3.2.9.3 Work Products Produced

* Code review reports
* List of code improvements and fixes

#### 3.2.9.4 Review and Checklist

The following checklist will be conducted:

* Is the code readable and maintainable?
* Are potential bugs identified and fixed?

### 3.2.10 Test Specification Review

#### 3.2.10.1 Description and Focus of the Review

Review test plans and test cases to ensure comprehensive coverage of system functionality.

#### 3.2.10.2 Timing of the Review

The review will be conducted before test execution begins.

#### 3.2.10.3 Work Products Produced

* Test specification review report
* Approved test cases

#### 3.2.10.4 Review and Checklist

The following checklist will be conducted:

* Are all requirements covered by test cases?
* Are test cases clear and executable?

### 3.2.11 Change Control Reviews and Audits

#### 3.2.11.1 Description and Focus of the Review

Review changes to ensure they are necessary and properly managed.

#### 3.2.11.2 Timing of the Review

The review will be conducted whenever significant changes are proposed.

#### 3.2.11.3 Work Products Produced

* Change control review report
* Approved change requests

#### 3.2.11.4 Review and Checklist

The following checklist will be conducted:

* Are changes justified and documented?
* Are impacts on other components evaluated?

## 3.3 SQA Audits

### 3.3.1 Description of Audits

SQA audits assess the effectiveness of SQA activities and compliance with quality standards.

### 3.3.2 Types of Audits

* **Process Audit**: Ensure processes are followed as defined in the project plan.
* **Product Audit**: Verify that work products meet specified standards.
* **Compliance Audit**: Confirm adherence to coding standards and documentation practices.

### 3.3.3 Audit Schedule

Audits will be performed at the end of each major milestone.

### 3.3.4 Audit Reports

* Audit findings
* Recommendations for process improvements

# 4.0 Problem Reporting and Corrective Action/Follow-up

## 4.1 Reporting mechanisms

**GitHub Issues**:

* Use GitHub Issues to track defects, bugs, and tasks throughout the development process.
* Each issue will have labels indicating priority (e.g., "High," "Medium," "Low"), status (e.g., "Open," "In Progress," "Resolved"), and type (e.g., "Bug," "Enhancement").

**Hive Tasks**:

* Use Hive for managing high-level project tasks, milestones, and workflows.
* Tasks in Hive will link to related GitHub Issues for detailed defect tracking.

## 4.2 Responsibilities

**Developers**:

* Fix reported defects and update the status of issues in GitHub.
* Communicate with the SQA team for clarification or additional information.

**SQA Team**:

* Verify that defects are resolved satisfactorily.
* Conduct regression tests to ensure no new issues are introduced.
* Close issues after verification is complete.

**Project Manager**:

* Monitor the progress of defect resolution to ensure it aligns with project timelines.

## 4.3 Data collection and evaluation

**Error/Defect Logs**:

* All defects will be logged in GitHub Issues with the following details:
  + Unique issue ID
  + Description of the defect
  + Steps to reproduce
  + Expected behavior
  + Actual behavior
  + Priority level
  + Assigned developer
  + Date of detection and resolution

**Metrics Collected**:

* Number of defects per module
* Time taken to resolve defects
* Number of defects reopened
* Defect density (number of defects per 1,000 lines of code)

**Storage and Accessibility**:

* Defect data will be stored in GitHub and periodically exported to Hive for evaluation and reporting purposes.

## 4.4 Statistical SQA

**Techniques Applied**:

* **Defect Trend Analysis**: Identify trends in defect occurrence and resolution over time to improve processes.
* **Pareto Analysis**: Identify the most frequent types of defects to focus on key problem areas.
* **Control Charts**: Monitor defect rates to ensure they remain within acceptable limits.
* **Root Cause Analysis (RCA)**: Determine the underlying causes of defects and propose corrective actions.

**Reporting**:

* Monthly SQA reports summarizing defect statistics and trends will be shared with the project team and stakeholders.

# 5.0 Software Process Improvement Activities

The SQA group (and others) is often chartered with responsibility for software process improvement (SPI). This section describes the work associated with SPI.

## 5.1 Goal and objectives of SPI

**Goal**:

* Improve the software development process to enhance quality, reduce defects, and increase efficiency.

**Objectives**:

* Identify process bottlenecks and inefficiencies.
* Implement best practices for coding, testing, and documentation.
* Continuously monitor and refine development processes based on metrics and feedback.
* Achieve compliance with industry standards (e.g., ISO/IEC 25010).

## 5.2 SPI tasks and responsibilities

**SQA Team**:

* Analyze defect trends and identify process improvements.
* Develop and maintain coding standards and best practices.
* Conduct regular process audits and provide recommendations.

**Developers**:

* Follow updated coding standards and process guidelines.
* Provide feedback on process changes.

**Project Manager**:

* Ensure SPI tasks are integrated into project timelines.
* Allocate resources for implementing process improvements.

# 6.0 Software Configuration Management Overview

## 6.1 SCM Plan Overview

The Software Configuration Management (SCM) plan outlines the processes and tools used to manage changes to the project’s software artifacts.

* **Version Control**:
  + **Tool**: GitHub
  + **Branching Model**: Git Flow (feature branches, development branch, main branch)
* **Change Management Process**:
  + All changes must be submitted via pull requests.
  + Pull requests require at least one code review approval before merging.
* **Baseline Management**:
  + Regularly establish baselines for major milestones (e.g., Alpha, Beta, Release Candidate).
* **Backup and Recovery**:
  + Weekly backups of repositories stored on GitHub and mirrored to an external drive or cloud storage.

# 7.0 SQA Tools, Techniques, Methods

## 7.1 Tools

* **Version Control**:
  + GitHub for source code management and collaboration.
* **Project Management**:
  + Hive for task management and high-level planning.
* **Testing Tools**:
  + **PyTest** for Python unit testing.
  + **Espresso** for Android UI testing.
* **CI/CD**:
  + GitHub Actions for automated builds and tests.
* **Documentation**:
  + Google Drive for collaborative document creation and storage.

## 7.2 Techniques

* **Code Reviews**: Regular peer reviews to ensure code quality.
* **Pair Programming**: For complex modules to reduce errors and improve code design.
* **Regression Testing**: Ensure that new changes do not break existing functionality.

## 7.3 Methods

* **Black-Box Testing**: Focus on system functionality without knowing internal code structure.
* **White-Box Testing**: Evaluate internal code logic, branches, and paths.
* **Exploratory Testing**: Unscripted testing to uncover unexpected issues.

# 8.0 Appendix

## 8.1 References

1. **PEP 8 – Style Guide for Python Code**
   * URL:<https://peps.python.org/pep-0008/>
2. **ISO/IEC 25010: Systems and Software Quality Requirements and Evaluation**
   * URL: <https://www.iso.org/standard/35733.html>
3. **LED Glove Specifications**: Detailed technical specifications of the LED gloves, including PWM frequency settings and sensor compatibility.
4. **Model Training Documentation**: Information on the trained CNN-LSTM model used for gesture smoothing, along with training parameters and sample data.
5. **Publication and Research Resources**: Journal and conference recommendations for submitting the research paper, including formatting guidelines and submission timelines.

## 8.2 Acronyms and Abbreviations

* **SQA**: Software Quality Assurance
* **SPI**: Software Process Improvement
* **SCM**: Software Configuration Management
* **FTR**: Formal Technical Review
* **RMMM**: Risk Mitigation, Monitoring, and Management
* **CI/CD**: Continuous Integration/Continuous Deployment

## 8.3 System traceability matrix

The system traceability matrix provides a mapping of the software requirements to corresponding functional and technical specifications, ensuring that each requirement is addressed in the system’s design and implementation. This traceability matrix will support the research objective by confirming alignment between the intended goals (e.g., real-time smoothing, VR/AR integration) and the actual functionality developed for demonstration purposes.

# 9.0 Review and Sign Off

| **Name** | **Project Team Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Soham Naik | Lead Computer Vision & Image Processing Engineer | Soham Naik | 12/10/2024 |
| Deniz Acikbas | Lead Mobile App & VR/AR Engineer | Deniz Acikbas | 12/10/2024 |
| Zaynab Mourtada | Lead Machine Learning Engineer | Zaynab Mourtada | 12/10/2024 |
| Alan Raj | Lead 3D Visualizer Engineer | Alan Raj | 12/10/2024 |
| Xiao Zhang | Client | Xiao Zhang | 12/10/2024 |