# **Software Quality Assurance Plan (SQAP)**

**Project Name**: Project Golf

**Prepared By:** Bryer Johnson

**Prepared For:** Golf Galaxy

**Approved by:** Brother Scott Wood

**Authors:** Bryer Johnson

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Software Quality Assurance Plan (SQAP) is to define the processes, methods, and tools that will be used to ensure the quality of the Project Phoenix software product.

## **1.2 Scope**

This SQAP applies to all phases of the software development lifecycle for Project Phoenix, including requirements gathering, design, implementation, testing, deployment, and maintenance.

## **1.3 Definitions and Acronyms**

* **SQA:** Software Quality Assurance
* **SDLC:** Software Development Life Cycle
* **QA:** Quality Assurance
* **QC:** Quality Control

# **2. Management**

## **2.1 Organization**

Describe the organizational structure for the project, including key roles and responsibilities.

* **Project Manager:** Responsible for overall project management and delivery.
* **SQA Manager:** Oversees the SQA activities and ensures compliance with the SQAP.
* **Development Team:** Responsible for the design, coding, and unit testing of the software.
* **QA Team:** Responsible for test planning, execution, and reporting.

## **2.2 Tasks**

List the major SQA tasks and their descriptions.

* **SQA Planning:** Develop and maintain the SQAP.
* **Requirements Review:** Ensure requirements are complete, clear, and testable.
* **Design Review:** Verify that design meets requirements and follows best practices.
* **Code Review:** Conduct peer reviews of code for compliance with coding standards.
* **Testing:** Develop test plans, test cases, and execute tests to validate software functionality.
* **Metrics Collection:** Collect and analyze metrics to monitor quality and process effectiveness.

## **2.3 Schedule**

Provide a high-level schedule of SQA activities, including key milestones and deliverables.

* **Project Kickoff:** 10-5-2024
* **Requirements Review** 10-7-2024
* **Design Review:** 10-10-2024
* **Code Review:** 11-10-2024
* **Testing:** 11-17-2024
* **Deployment:** 01-01-2025

# **3. Documentation**

## **3.1 SQA Documents**

List all documents related to SQA activities.

* **SQAP**
* **Requirements Specification**
* **Design Documents**
* **Test Plan**
* **Test Cases**
* **Test Reports**

## **3.2 Control of SQA Documents**

Describe the process for document control, including versioning, storage, and access.

* **Version Control:** Use a version control system (e.g., Git) for all SQA documents.
* **Storage**: Store documents in a centralized repository.
* **Access:** Restrict access to authorized personnel only.

# **4. Standards, Practices, and Conventions**

## **4.1 Software Development Standards**

Specify the standards and practices to be followed during the software development lifecycle.

* **Coding Standards:** Follow industry best practices and project-specific guidelines.
* **Design Standards:** Ensure designs are modular, maintainable, and scalable.
* **Documentation Standards:** Maintain clear, concise, and up-to-date documentation.

## **4.2 Testing Standards**

Outline the standards for testing activities.

* **Test Plan:** Develop a comprehensive test plan covering all aspects of testing.
* **Test Cases:** Write test cases that are clear, concise, and traceable to requirements.
* **Test Execution:** Execute tests according to the test plan and document results.

# **5. Reviews and Audits**

## **5.1 Types of Reviews**

Describe the types of reviews that will be conducted.

* **Requirements Reviews:** Verify that requirements are complete and testable.
* **Design Reviews:** Ensure design meets requirements and adheres to standards.
* **Code Reviews:** Check for compliance with coding standards and best practices.
* **Test Plan Reviews:** Validate that the test plan covers all necessary aspects of testing.

## **5.2 Audits**

Describe the audit process, including the frequency and scope of audits.

* **Internal Audits:** Conduct internal audits to ensure compliance with the SQAP.
* **External Audits:** Engage external auditors for independent verification of quality processes.

# **6. Testing**

## **6.1 Test Levels**

Define the different levels of testing to be conducted.

* **Unit Testing:** Verify the functionality of individual components.
* **Integration Testing:** Ensure that components work together as expected.
* **System Testing:** Validate the complete and integrated software system.
* **Acceptance Testing:** Confirm that the software meets user requirements and is ready for deployment.

## **6.2 Test Environment**

Describe the test environment, including hardware, software, and network configurations.

* **Hardware:** List the hardware specifications required for testing.
* **Software:** Specify the software tools and versions used for testing.
* **Network:** Detail the network setup required for testing.

# **7. Problem Reporting and Corrective Action**

## **7.1 Problem Reporting**

Describe the process for reporting and tracking defects.

* **Defect Tracking System:** Use a defect tracking system (e.g., JIRA) to log and manage defects.
* **Defect Reports:** Generate defect reports to provide visibility into the status and progress of defect resolution.

## **7.2 Corrective Action**

Describe the process for implementing corrective actions.

* **Root Cause Analysis:** Conduct root cause analysis to identify the underlying cause of defects.
* **Corrective Measures:** Implement corrective measures to prevent recurrence of defects.
* **Verification:** Verify that corrective actions have been effective.

# **8. Tools, Techniques, and Methodologies**

## **8.1 Tools**

List the tools that will be used for SQA activities.

\* Version Control: Git

* **Defect Tracking:** JIRA
* **Testing:** Selenium, JUnit
* **Documentation:** Confluence

## **8.2 Techniques and Methodologies**

Describe the techniques and methodologies to be used.

* **Agile Methodology:** Follow Agile principles and practices for iterative development.
* **Continuous Integration:** Use continuous integration tools (e.g., Jenkins) to automate builds and tests.
* **Code Coverage:** Utilize code coverage tools (e.g., JaCoCo) to measure test coverage.

# **9. Training**

## **9.1 Training Plan**

Describe the training plan for the project team.

* **SQA Training:** Provide training on SQA processes and best practices.
* **Tool Training:** Conduct training sessions on the tools used for SQA activities.
* **Ongoing Education:** Encourage ongoing education and professional development.

# **10. Risk Management**

## **10.1 Risk Identification**

Identify potential risks to software quality.

* **Requirements Changes:** Changes in requirements during development.
* **Resource Constraints:** Limited availability of skilled resources.
* **Schedule Slips:** Delays in project schedule impacting quality.

## **10.2 Risk Mitigation**

Describe the strategies for mitigating identified risks.

* **Change Control:** Implement a change control process to manage requirements changes.
* **Resource Planning:** Ensure adequate resource allocation and planning.
* **Schedule Monitoring:** Monitor project schedule closely and take corrective actions as needed.

# **11. Metrics and Reporting**

## **11.1 Metrics Collection**

Describe the metrics that will be collected to monitor quality.

* **Defect Density:** Number of defects per unit of code.
* **Test Coverage:** Percentage of code covered by tests.
* **Pass/Fail Rates:** Percentage of passed and failed test cases.

## **11.2 Reporting**

Describe the reporting process for SQA activities.

* **Status Reports:** Provide regular status reports on SQA activities.
* **Metrics Reports:** Generate metrics reports to provide visibility into software quality.
* **Review Meetings:** Conduct review meetings to discuss progress and address issues.

# **12. Continuous Improvement**

## **12.1 Process Improvement**

Describe the process for continuous improvement of SQA processes.

* **Feedback Loop:** Establish a feedback loop to gather input from team members.
* **Process Evaluation:** Regularly evaluate SQA processes for effectiveness.
* **Improvement Actions:** Implement improvement actions based on evaluation results.

**Approval:** Brother Scott Wood

Brother Scott Wood

Professor of CSE474

10-5-2024