Better than Jira Software Quality Assurance Plan

Authors: Alex Cordova, Grant Carl Jones, Miguel Coelho, Victor, Dominic

Approved by:

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Authors** | **Revision** | **Description** |
| 10-17-2024 | Alex Cordova | 1.0.0 | Initial version. Adding the structure and purpose of the project (Section 1). |
| 10-23-2024 | Alex Cordova | 1.1.0 | Added more details on the purpose, management, and documentation. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Section 1: Purpose

* 1. **Name of Project:**

Better than Jira

**1.2 Intended Use or Feature Set**

The "Better than Jira" project aims to develop a comprehensive project management tool with skill-tracking capabilities. It includes features like project creation, task assignment, progress tracking, skill analytics, burnout monitoring, and gamification elements such as badges, leaderboards, and achievements.

**1.3 Software Dependencies**

Front-End: JavaScript, Angular

Back-End: Node.js, Express.js

Database: MySQL

Additional Dependencies: Any additional frameworks or tools will be finalized during project development.

**1.4 Hardware Target or Targeting Environment**

The project is a web-based application, accessible via modern web browsers on desktop, tablet, and mobile devices.

**1.5 Lifecycle Phases used for this project**

The project will follow an Agile development lifecycle, iterating over planning, development, testing, deployment, and review phases.

**1.6 Organization of the Document**

Section 2: Management

Section 3: Documentation

Section 4: Standards, Practices, and Conventions, and Metrics

Section 5: Software Reviews

Section 6: Test

Section 7: Problem Reporting and Corrective Action

Section 8: Tools, Techniques, and Methodologies

Section 9: Media Control

Section 10: Supplier Control

Section 11: Records Collection, Maintenance, and Retention

Section 12: Training

Section 13: Risk Management

Section 14: Glossary

Section 2: Management

**Communication Plan:** Weekly stand-up meetings and a shared online project board for tracking deliverables.

Section 3: Documentation

Section 4: Standards, Practices, and Conventions, and Metrics

**Coding Standards:** Follow JavaScript, Angular, and SQL best practices.

**Metrics:** Measure bug counts, feature completion rates, and user satisfaction based on surveys and feedback.

Section 5: Software Reviews

**Review Types:** Include code reviews, design reviews, and quality checks at the end of each sprint.

Section 6: Test

**Testing Types:** Include unit testing, integration testing, and end-to-end testing using appropriate tools like Jasmine and Selenium.

Section 7: Problem Reporting and Corrective Action

Section 8: Tools, Techniques, and Methodologies

**Development Tools:** Visual Studio Code, GitHub, MySQL Workbench, and Angular CLI.

**Methodologies:** Follow Agile and SCRUM methodologies for iteration-based project tracking.

Section 9: Media Control

**Data Backup:** Implement regular backups and version control using GitHub.

Section 10: Supplier Control

**Third-Party Libraries:** Monitor and validate third-party libraries before integration.

Section 11: Records Collection, Maintenance, and Retention

**Project Records:** Store documentation, logs, and test reports securely within the GitHub repository.

Section 12: Training

**Training Plan:** Conduct introductory sessions for team members on coding standards and security practices.

Section 13: Risk Management

**Initial Risks:** Identify potential risks like scope creep, database failures, and burnout due to heavy workloads.

Section 14: Glossary