

COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

SOFTWARE QUALITY ENGINEERING

**Submitted By:**

Laiba Binta Tahir FA21-BSE-019-5A

Arfah Ali FA21-BSE-080-5A

**ASSINGEMENT # 03**

**Submitted To:** Mam Mamoona Atif Swati

Date: 20 November-2023

Contents

[List of Quality Goals 3](#_Toc151138601)

[Review Activities Plan 3](#_Toc151138602)

[Requirements Review: 3](#_Toc151138603)

[Code Inspection: 4](#_Toc151138604)

[Peer Review (General): 4](#_Toc151138605)

[User Interface Review: 4](#_Toc151138606)

[Defect Reporting Review: 4](#_Toc151138607)

[Software Tests 4](#_Toc151138608)

[Acceptance Tests for Software Externally Developed 5](#_Toc151138609)

[Purchased Software: 5](#_Toc151138610)

[Customer-Supplied Software: 6](#_Toc151138611)

[Configuration Management Tools & Procedures 6](#_Toc151138612)

[Version Control: 6](#_Toc151138613)

[Change Control: 6](#_Toc151138614)

[Configuration Identification: 7](#_Toc151138615)

[Verification and Audit: 7](#_Toc151138616)

[Baseline Management: 7](#_Toc151138617)

QUALITY PLAN – LEARNING MANAGEMENT SYSTEM

## List of Quality Goals

Here's the list of quality goals for a learning Management System in a table format:

|  |  |  |
| --- | --- | --- |
| **Quality Aspect** | **Quality Goal** | **Quantitative Measure** |
| Accuracy and Reliability | Achieve a data accuracy rate of 99% in LMS | Conduct regular data validation checks, aiming for a maximum error rate of 1% in data records. |
| Usability | Attain a user satisfaction rate of 90% or above. | Conduct usability testing with a target success rate of 85% for common user tasks. |
| Performance | Ensure a system response time under 3 seconds. | Measure and optimize database query response times to meet the specified performance target. |
| Security | Achieve a security vulnerability rate of less than 2%. | Conduct periodic security audits and aim to address identified vulnerabilities promptly. |
| Scalability | Ensure the system can handle a 20% increase in load. | Conduct scalability testing to verify the system's ability to handle increased load. |
| Maintainability | Maintain a code readability score of at least 80%. | Regularly assess and improve code maintainability using automated tools and peer reviews. |
| Timeliness | Achieve a 95% on-time delivery rate for updates. | Track and analyze delivery timelines for updates, addressing any delays promptly. |
| Compliance | Ensure compliance with relevant standards and regulations. | Regularly audit the system against applicable standards and regulations. |
| Customer Support Responsiveness | Maintain an average response time of less than 24 hours. | Monitor and analyze customer support response times, aiming for continuous improvement. |

## Review Activities Plan

### Requirements Review:

Type: Design Review

Schedule: Before development begins

Procedure: Review requirements documentation for completeness, clarity, and alignment with goals.

Responsibilities: Project Manager, Quality Assurance Manager, Development Team

### Code Inspection:

Type: Code Review

Schedule: After initial code implementation

Procedure: Inspect code for adherence to coding standards, best practices, and potential issues.

Responsibilities: Development Team, Code Reviewers (Peers or Senior Developers)

### Peer Review (General):

Type: Peer Review

Schedule: Regular intervals during development

Procedure: Conduct general peer reviews to ensure code quality and consistency.

Responsibilities: Development Team

### User Interface Review:

Type: Design Review

Schedule: During the design phase

Procedure: Review user interface design for usability, consistency, and alignment with requirements.

Responsibilities: UI/UX Designers, Development Team

### Defect Reporting Review:

Type: Defect Review

Schedule: Throughout the testing phase

Procedure: Review reported defects for accuracy, completeness, and priority.

Responsibilities: Test Team, Development Team, Quality Assurance Manager

## Software Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Unit, Integration, or Complete System** | **Type of Testing** | **Planned Test Schedule** | **Responsible Parties** |
| Unit Testing - Data Validation | Unit | Functional Testing | During development of each module | Developer, Test Team |
| Unit Testing - Module A | Unit | Functional Testing | After completion of module A | Developer, Test Team |
| Integration Testing - Modules A and B | Integration | Integration Testing | After completion of individual unit tests for A and B | Test Team, Developers of Modules A and B |
| System Testing | Complete System | System Testing | Before the start of User Acceptance Testing (UAT) | Test Team, Quality Assurance Manager |
| Usability Testing | Complete System | Usability Testing | During the final stages of development | UI/UX Designers, Test Team, End Users |
| Disaster Recovery Testing | Complete System | Disaster Recovery Testing | Periodically (e.g., quarterly) | IT Team, System Administrators, Test Team |

## Acceptance Tests for Software Externally Developed

### Purchased Software:

**Type of Software:**

ERP System (e.g., Financial Module)

**Acceptance Tests:**

Security Test: Verify the security measures implemented by the purchased software.

Usability Test: Evaluate the user interface and overall user experience.

**Planned Test Schedule:**

Conducted immediately after the software is delivered and integrated into the LMS.

**Responsible Parties:**

Internal Testing Team, Vendor Representatives

### Customer-Supplied Software:

**Type of Software:**

Student Information System (SIS) Integration Module

**Acceptance Tests:**

Data Integration Test: Validate the accurate transfer of data between the SIS and the LMS.

Performance Test: Assess the system's performance with the integrated SIS module.

End-to-End Test: Test the complete workflow, including data input, processing, and output.

**Planned Test Schedule:**

Coordinated with the delivery schedule of the customer-supplied software.

**Responsible Parties:**

Internal Testing Team, Customer's IT Representatives

## Configuration Management Tools & Procedures

### Version Control:

**Tool:** Git

**Procedures:**

Use branches for development features.

Regularly merge branches into the main code.

Tag releases for easy reference.

### Change Control:

**Submission:**

Changes submitted through a Change Request (CR) form.

Include reason, impact, and proposed solutions.

**Evaluation:**

Change Control Board (CCB) evaluates changes.

Prioritize based on urgency and importance.

**Implementation:**

Approved changes implemented and tested.

Documentation updated.

### Configuration Identification:

**Criteria:**

Unique identifier for each configuration item (CI).

Use version numbers and release dates.

**Documentation:**

Maintain a Configuration Management Plan.

Use a Configuration Management Database (CMDB).

### Verification and Audit:

**Verification:**

Periodic checks to ensure configurations match expectations.

Ensure documentation aligns with the actual system.

**Audit:**

Conduct audits at project milestones.

Random audits for ongoing compliance.

### Baseline Management:

**Establish Baselines:**

Create baselines for major project milestones.

**Change Impact Analysis:**

Analyze the impact before making changes.

Evaluate implications on the entire system.