UCLM Escuela Superior de Informática

Quality Management

The quality management strategy defines the key quality characteristics that our Library Management System must meet to ensure long-term efficiency, user satisfaction, and system stability.

After analyzing the system's objectives and scope, we identified four main quality characteristics as the most relevant for our project: **Security, Maintainability, Usability, and Reliability.**

These characteristics were selected because they directly influence the system's performance, sustainability, and user experience.

1. Security

Justification:

Security is essential to protect the system from unauthorized access, data breaches, and malicious actions. Applying security by design ensures trust, data integrity, and system stability, safeguarding both users and institutional resources.

Non-Functional Requirements Related to Security:

- **Two-Step Authentication:** Users must verify their identity using more than one authentication method.
- **Data Encryption:** Sensitive data must be encrypted both during transmission and storage.
- **Backup and Recovery Mechanisms:** The system must perform automatic backups and provide recovery procedures in case of corruption or data loss.
- **Role-Based Access Control:** Permissions are assigned according to user roles (students, faculty, staff, administrators).

Impact on Project Planning:

Integrating advanced security mechanisms will increase development time and testing effort, as encryption, authentication, and access control must be carefully implemented and validated. This may also involve additional costs for security tools and maintenance.

2. Maintainability

Justification:

Maintainability guarantees that the system can be easily updated, corrected, and extended throughout its lifecycle. A maintainable system simplifies future improvements, reduces long-term costs, and supports collaborative development.

Non-Functional Requirements Related to Maintainability:

- **Reusable Architecture:** System components should be designed for reuse in other functionalities.
- **Code Documentation:** The code must include clear comments, naming conventions, and developer documentation.
- **Version Control Management:** A version control system must be used to track changes and support team collaboration.
- **Modular Structure:** The system should be divided into independent modules to simplify maintenance and updates.

Impact on Project Planning:

Ensuring maintainability will require extra time during the design and coding phases to implement a modular structure and proper documentation. However, it will reduce future maintenance costs and facilitate scalability.

3. Usability

Justification:

Usability measures how easy and intuitive the system is for users. Since students, faculty, and staff will interact with the system daily, providing a clear and accessible interface is vital to ensure efficiency and satisfaction.

Non-Functional Requirements Related to Usability:

- **Consistent User Interface:** A unified interface across all platforms (web, mobile, desktop).
- **Accessibility Compliance:** The interface must meet accessibility standards (WCAG) for users with disabilities.
- **Clear Navigation Structure:** Users should easily locate resources, reservations, and account details.
- **User Feedback and Guidance:** The system should provide meaningful messages for errors, confirmations, and help.

Impact on Project Planning:

Usability improvements require additional time for interface design, prototype testing, and user feedback sessions. This may increase development effort but ensures a more successful and user-friendly product.

4. Reliability

Justification:

Reliability ensures the system performs correctly and consistently over time, minimizing interruptions and failures. Given that the library system supports academic operations, reliability is crucial for continuous service availability.

Non-Functional Requirements Related to Reliability:

- **High Availability:** The system should remain accessible even during peak usage periods.
- **Fault Tolerance:** The system must recover gracefully from minor failures.
- **Performance Monitoring:** Ongoing monitoring should detect performance issues early.
- **Recovery Time Objective (RTO):** Services should be restored within an acceptable timeframe after failure.

Impact on Project Planning:

Ensuring reliability requires additional testing phases, such as stress tests, performance analysis, and backup validation. These activities may extend the testing schedule and slightly increase infrastructure costs but will significantly improve system robustness.

Conclusion

By incorporating **Security, Maintainability, Usability, and Reliability** as core quality characteristics, the Library Management System will achieve a high standard of performance, safety, and user satisfaction.
However, these characteristics also introduce **additional time, effort, and cost** into the project plan. Therefore, the development schedule should be **revised** to include:

- Security testing and validation phases.
- Dedicated iterations for documentation and modularization.
- Usability evaluations and user feedback sessions.
- Performance and reliability testing before deployment.

Implementing these adjustments ensures a realistic, sustainable, and high-quality software product aligned with user needs and institutional standards.