CareMate: Personalized Healthcare Companion

Jason Browder

CS-3332 Software Engineering I

Mr. Reggie Haseltine

3/2/2025

Software Quality Assurance (SQA) is an essential aspect of software development, ensuring that the final product meets quality standards, is reliable, and functions as intended. This paper outlines the SQA processes that should be implemented for the proposed software application, detailing standards, quality goals, and reliability considerations.

SQA Processes and Implementation To ensure high-quality software, a structured SQA framework must be established. The following processes should be implemented:

1. Requirements Analysis and Verification
    - Utilize formal requirement gathering techniques, including stakeholder interviews and use case development.
    - Conduct Formal Technical Reviews (FTRs) to validate requirements before proceeding with design and development (Pressman & Maxim, 2020).
2. Software Design Reviews
    - Implement design walkthroughs and inspections to ensure adherence to architectural best practices.
    - Use standardized design patterns for modular, maintainable code.
3. Code Quality and Static Analysis
    - Enforce coding standards such as the Google Java Style Guide or PEP 8 (Python) for consistency.
    - Conduct static code analysis using tools like SonarQube to detect potential defects early.

4. Automated and Manual Testing
    - Implement unit testing using frameworks like JUnit (for Java) or PyTest (for Python).
    - Conduct system, integration, and regression testing to validate system-wide behavior.
    - Perform usability testing to ensure a seamless user experience (ISO 25010:2017).
5. Configuration and Change Management
    - Use Git for version control and implement a branching strategy such as GitFlow.
    - Track and manage changes using tools like Jira or Trello.
6. Defect Tracking and Continuous Integration
    - Utilize defect-tracking systems like Bugzilla or Jira to document and resolve software issues.
    - Implement Continuous Integration/Continuous Deployment (CI/CD) pipelines to automate testing and deployment.
7. Formal Technical Reviews
    - Conduct peer reviews and inspections at each phase of the Software Development Life Cycle.
    - Document review findings and track corrective actions.

Software Quality Assurance Plan (SQA Plan) A structured SQA Plan should include:

- Purpose and Scope: Define objectives, stakeholders, and applicability.
- Standards and Guidelines: Include IEEE 730 (SQA Standards) and ISO 9001:2015 (Quality Management Systems).
- SQA Activities: Specify inspections, testing methodologies, and compliance requirements.
- Defect Management Process: Outline reporting, classification, and resolution procedures.
- Risk Management Strategy: Identify potential risks and mitigation measures.
- Metrics and Reporting: Define measurable indicators like defect density, test coverage, and Mean Time Between Failures (MTBF).
- Audit and Compliance: Plan for periodic quality audits and adherence to regulatory requirements.

Availability and Reliability Goals The application should achieve:

- 99.9% availability, with redundancy strategies to minimize downtime.
- Reliability metrics, such as Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR), ensuring software robustness.
- Disaster Recovery Plan, including backup and failover strategies to handle unexpected failures.

Implementing a robust SQA process ensures that the software meets industry standards, achieves high availability, and maintains reliability. By adhering to formal methodologies and best practices, the application can deliver a high-quality user experience while minimizing defects and risks.

References

Purdue Online Writing Lab. (n.d.). General format. Purdue University. https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_formatting_and_style_guide/general_format.html

International Organization for Standardization. (2017). *ISO 25010:2017 Systems and software engineering — Systems and software Quality Requirements and Evaluation System and software quality*

https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). New York, NY: McGraw Hill.