Predicting the Analysis of Software Quality Management Practices in Day-to-Day Life

Halyna Monaco*

Department of Technology, University of Denmark, Lyngby, Denmark

DESCRIPTION

Software has become an indispensable part of our lives, from smartphones and laptops to sophisticated systems in the healthcare, financial, and transportation sectors. The quality of software is paramount, as it directly impacts user experience, safety, and efficiency. Software quality refers to the degree to which software meets specific requirements, expectations, and standards.

It encompasses various aspects, including functionality, performance, reliability, security, and usability. A high-quality software system is one that not only meets its intended purpose but does so efficiently, effectively, and without causing harm.

Dimensions of software quality

Functionality: Functionality is at the core of software quality. It relates to the software's ability to perform its intended tasks accurately and reliably. To assess functionality, software must meet its functional requirements, which can be both explicit and implicit. This dimension encompasses features, correctness, completeness, and compliance with specifications.

Performance: Performance is critical, especially in applications where speed and responsiveness are essential. This dimension includes factors like response times, throughput, and resource utilization. A high-quality software system should offer optimal performance under varying conditions.

Reliability: Reliability refers to the software's ability to maintain its intended functionality over time. High-quality software should be robust, fault-tolerant, and able to recover gracefully from errors. Users should trust that the software will work consistently.

Security: Security is a paramount dimension of software quality, especially in an age where data breaches and cyber-attacks are common. High-quality software should protect user data, guard against vulnerabilities, and maintain the confidentiality, integrity, and availability of information.

Usability: Usability is about user-friendliness. A high-quality software system should have an intuitive interface, provide clear instructions, and ensure an overall positive user experience. Good usability reduces user frustration and the risk of user errors.

Interconnected modules, intricate algorithms, and diverse platforms make it difficult to anticipate all potential issues. Software development often involves evolving requirements, which can lead to scope creep and make it challenging to maintain consistent quality. Agile methodologies attempt to address this challenge by embracing change, but it can still be a struggle. Limited time, budget, and skilled personnel can compromise efforts to maintain high-quality software. Cutting corners to meet deadlines can lead to quality issues. Comprehensive testing is essential for software quality, but it's challenging to predict all real-world scenarios and edge cases. Automated testing and techniques like exploratory testing are used to address this limitation. Modern software often relies on third-party libraries and components, which may not always meet the same quality standards as the in-house development team. This creates a dependency on external entities for software quality. The concept of "shift-left" involves addressing quality from the earliest stages of development. In DevOps, the integration of development and operations facilitates continuous testing, automation, and rapid deployment, enhancing software quality by identifying and addressing issues early in the development cycle. AI and machine learning are increasingly employed to improve software quality. They assist in identifying anomalies, automating testing, and enhancing cyber-security. Open-source software has gained prominence. It fosters collaboration, innovation, and access to high-quality code. However, it also introduces the challenge of evaluating the quality of third-party components. Metrics and benchmarks are used to measure and track software quality. Tools like static code analysis, dynamic analysis, and quality assurance frameworks help assess quality objectively. The focus on user experience has become more prominent in software development. User feedback, usability testing, and the design thinking principles are

Correspondence to: Halyna Monaco, Department of Technology, University of Denmark, Lyngby, Denmark, E-mail: monhal@jsv.dk

integrated into the development process to ensure software meets user expectations. Given the increasing cyber threats, many other organizations now prioritize security from the outset.

Security by design ensures that security considerations are integral to the development process.