

Software Testing and Quality Management

CS-E3150 Software Engineering

Software quality: definition and existing standards

Software quality can mean different things from different aspects (for example developer and user perspectives). It can be defined in terms of usability of the software, maintainability, efficiency, etc. According to IEEE, it refers to how well a software meets the specified requirements (called verification) and user's expectations and needs (called validation) (Sommerville, 2016). There exist many standards to measure the quality of software. For example, ISO 9126 tends to focus on measuring functionality, reliability, usability, efficiency, maintainability, and portability. But it, of course, depends upon the purpose of the software. For example, a game must be fast, while a banking system must be secure. It also depends upon the perspective of the stakeholder. For example, a developer would want the software to be easily maintainable, while a tester would want it to be easily testable and the customer would want it to be usable.

How to achieve software quality?

First of all, we need to fix our definition of quality for the software we are developing because it's not possible to be the best in all attributes of quality. Then we can develop the software in a way that largely satisfies our definition of quality. **Testing** and **Review/Inspection** are widely used for the same. Testing is an execution based way of ensuring quality, while the review is done by people without actually running the code on a machine.

To improve software quality, developers should focus on writing clean and bug-free code. In agile method, a review (called sprint-review) meeting is held after each iteration to review the progress of the project and fix any possible bugs (Dubinsky et al., 2006). Extreme programming uses pair programming to ensure code review and bug detection in real time (Dubinsky et al., 2006). In the waterfall model, a stage is thoroughly reviewed and the next stage begins only after the previous stage is approved. Even though this hierarchical process is time-consuming, but it helps in developing quality products. In agile context, including customers directly in the development process helps to get a user-perspective which in-turn helps in improving the usability of the product. Testing is also an effective way of finding bugs after code completion. A number of test-cases are given as input to the software and their output is matched with the expected output. There are many existing ways of testing software, namely **development testing**, **unit testing**, **automated testing**, etc and many types of testing like **system testing**, **use-case testing**, **regression testing**, etc (Lassenius, 2013). This method certainly helps in finding bugs, but it is computationally very expensive because of way too many possibilities of paths. Hence exhaustive testing is almost impossible. Another issue is that writing tests itself might have bugs or it may not cover all possible faults. For example, Ariane 5 tests couldn't discover the integer overflow problem. So we have to do manual code inspection anyway.

Overall, fixing bugs is a complicated process. Even a simple typo can take several weeks to be discovered. The best method is to avoid bugs by writing a stable and clean code. For developing a quality product, it is important that everyone working on the project has a high-level view and full information about the project status. A quality manager should work on developing a quality-focused culture. The company should also work on making its process standards adaptable because strictness kills the creativity of developers (Sommerville, 2016). Thus there are many ways through which the quality of software can be ensured.

References

SOMMERVILLE, I., 2016. Software Engineering. 10th ed. Boston: Pearson

DUBINSKY, Y., HAZZAN, O., KEREN, A., TALBY, D. 2006. Agile Software Testing in a Large-Scale Project. IEEE Software [online]. 23(4), pp. 30-37. [viewed 16 March 2020] Available From: https://mycourses.aalto.fi/pluginfile.php/1177234/mod_resource/content/4/Talby_2005.pdf

LASSENIUS, C., 2013. Software Quality Assurance and Testing [lecture online]. [viewed 16 March 2020]. Available from: https://mycourses.aalto.fi/pluginfile.php/1177232/mod_resource/content/3/QualityTesting.pdf