Software quality and testing

In any software system, there are always questions raised by the customers on the developed software quality. In simple terms software quality means a software developed should be in line with the customer requirements and specification (Sommerville, 2011). IEEE610.12 defines Software quality in 2 main viewpoints (Lassenius, C. 2014):

1. Validation :The degree to which a system, component or process meets specified requirements
2. Verification : The degree to which a system, component, or process meets customer or user needs or expectations

The International Standards Organisation (ISO) has defined a set of standards for quality management.Standards are the key to effective quality management. Product standards define characteristics that all components should exhibit according to the specifications e.g. the coding standards. Process standards define how the software process should be enacted. The ISO 9001 standard is a generic model of the quality process and must be instantiated for each organisation (Sommerville, 2011). This standards define some Management responsibilities like process control, document control, servicing etc. and also a quality system that involves Inspection and testing, corrective action, quality records, training etc. Customers are increasingly demanding that software suppliers are ISO 9000 certified (Sommerville, 2011). The ISO 9126 lists out some of the quality characteristics like functionality, reliability, usability, efficiency, maintainability and portability that should be considered while developing a software (Lassenius, C. 2014). Functionality is one major characteristic of the product which measures how close the developed product is to the given specification and requirement.

To achieve a good quality software, one must consider different stakeholders viewpoints. Customers, users, developers, project managers and testers have different ways to measure the quality of the software (Lassenius, C. 2014). Testers are interested in the usability and maintenance of the code and similarly developers are interested in the code reusability and the structure of the code. Once the software system is ready, there is a huge checklist that has to be validated to ensure the code is up to the mark (Sommerville, 2011). User scenarios are validated. Requirements, usability and documentation are tested. Quality assurance is an iterative process based on the customer feedback.

Software quality can be ensured in different ways. Execution based testing involves running the code and testing the features. Non- execution based testing are mainly done by inspection and reviews. Reviews and inspection are simple where people read the code or document with the intent to find any problems. Inspection cannot help to test the non-functional characterises such as performance, usability, etc. Pair programming is an effective way to do the inspection testing (Lassenius, C. 2014). In this method, two developers will be developing the code together and promotes knowledge transfer. The formal inspection process contains some pre-review activities like document writing which is an individual process, followed by review meetings with the group. The code refactoring and review changes are a part of post review activity. In Scrum, the review meeting are held after each iteration of the software has

been completed (a sprint review), where quality issues and problems may be discussed (Lassenius, C. 2014).

Execution based testing is used to analyse the software non-functional requirements by executing the code in a known environment with selected input (Sommerville, 2011). Testing should be done by developers, testers and customers. Developers' do a subtle testing of the code, but the majority testing is performed by the testers. They have testing policies defined and prioritize their testing based on the functional requirements (Lassenius, C. 2014). The final acceptance testing is performed by the customer.

Testing does not ensure that the software is bug-free. It focuses on evaluation of the software system. It shows the presence of the errors and not their absence. Test driven development (TDD) is an approach, where in coding and testing sprints are interleaved (Sommerville, 2011). Tests are written prior to the codes and the code have to pass the tests. The code is developed in increments and the testing is done as soon as the increment of the code is ready. If the code fails, then it's sent back to implement and refactor the code functionality. This continuous feedback mechanism helps to improve the code quality. In conclusion, software quality can be achieve using different approaches based on the process that is being followed.

References

Sommerville. I. (2011) Software Engineering, 9th edition, U.S.A: Pearson Education, Inc.

Lassenius, C. (2014), T-76.3601 Introduction to Software Engineering, Lecture 8: Software Quality Assurance and Testing. Aalto University Department of Computer Science and Engineering.

Talby, D., Hazzan, O., Dubinsky, Y., and Arie Keren. (2006). "Agile Software Testing in a Large-Scale Project". IEEE Software, 23 (4), pp. 30-37.