Software Quality and Testing

2018-03-10

Software Quality

Software quality is defined as how well the products conformes its specification with certain allowed tolerance since products will never meet their specifications exactly. Software quality is subjective quantity and its left for the quality management team to assess. The quality management team can assess the quality using the quality characteristics; functionality, reliability, usability, efficiency, maintainability and portability, and decide if the software is fit for its intended purpose (Sommerville 2010).

Achieving Software Quality

The selection of **software standards** play an important part on quality assurance. Software standards provide a framework of the definition of quality in particular setting and they also assist in the continuity of the work ensuring that engineers within the ogranization adopt the same practices. The two types of software standards that are used are *product* standards, which apply to the product, and *process* standards, which apply to the development process (Sommerville 2010).

Reviews and inspections are another quality assurance activity to assess that the quality standards have been followed by examining the software, the documentation and the records of the process. The purpose of reviews and inspections is to improve the quality of the software (Sommerville 2010).

Software measurement means deriving a numeric value known as **metric**, from an attribute of software component, system, or process. These value can be used to quantify the quality of the software, by comparing them to each other (Sommerville 2010). For instance, Talby et al. (2006) discusses how they defined the product size as the number of regression test step run at each iteration in their agile project. They note that the test size correlates better with the complexity than lines of code or specifications and therefore is a better approximation of code size, as well as sending a clear message to the team that only feature with full regression tests at each iteration are counted as delivered product size.

Software Testing

Software testing attempts to demonstrate that the program meets its requirements and to discover defects within the program. Testing relates to quality which is defined as conformance to specification, by attempting to find out situations when the software does not conform to its specification (Sommerville 2010).

Development testing is testing activities carried out developers. Development testing includes *unit testing*, *component testing* and *system testing*. A tester can be the programmer who wrote the code or programmer/tester pair (Sommerville 2010). For instance, Talby et al. (2006) describes that in their agile project everyone wrote tests. This eliminated the reliance of single project tester, which would constitute a major bottleneck. They note that this approach also increased developers test-awareness and prevented or quickly caught more edge cases during work. Developers also designed their code to be testable. Tester was assigned to work with pairs of developers and write tests in parallel with coding after they had all inspected the feature's specifications and generated a set of test cases accordingly.

Test-driven development is development model in which test are written before the code and run automatically. Extreme programming is well known, agile test-driven driven development model.

Release testing is done when the system is release outside of the development to customers or other teams.

User testing involves the user in the testing process to provide input on the testing process. User testing includes *alpha testing*, *beta testing*, and *acceptance testing*.

Bibliography

Sommerville, I., 2010. *Software Engineering: 9th Edition*. 9th ed. Addison-Wesley Publishing Company.

Talby, D., Keren, A., Hazzan, O., and Dubinsky, Y., 2006. Agile software testing in a large-scale project. *IEEE Software*, 23 (4), 30–37.