Top 5 Essentials of Software Testing

Software development cycle is an extremely complex process involving various stages of hidden problems that cannot be pointed out unless the software is tested. These days writing an elegant and clear code is not sufficient. Even the best development teams fail to write a perfect code. Since bugs are inevitable, software testing and quality analysis is an integral part of the development team. But in most cases project and product managers completely underestimate the importance of software testing and quality analysis. These kinds of wrong judgements can cause a lot of issues and result in huge monetary losses.

A study by Mckinsey has revealed that in large software projects, 45% of them run over budget and 7% over time which results in delivered value being 56% less than the predicted value. Any deviation from expected value is regarded as a defect. Defects are usually encountered but not limited to the execution phase of the software development life cycle. In the software program or the software documentation human error will usually result in a fault, a defect or a bug. When this defective code is run, the software systems fails to function in the way that it is supposed to function causing a failure. Testing ensures that the software application adheres to the specifications. No software product should ever go into production without ever getting properly tested.

Here are the top essentials of software testing:

1. A Robust Testing Strategy:

The accuracy of the test estimates and plans and the success of the test effort are largely dependent on the choice of the test strategy which is under the control of the test leaders and testers. The testing strategy forms an essential part of the software development life cycle. The main purpose behind software testing is finding defects rather than passing easy tests. A robust test strategy outlines the testing type that seems best to do, the order in which the tests have to be performed, the recommended execution sequence and the ideal amount of effort that has to be put into each objective so that the testing is made most effective. Business enterprises that do not effectively implement quality control standards and fail to define the test range for a particular application can result in the credibility of the brand being destroyed and disrupting the entire project causing cost overruns. The test strategy must be created when the requirements have settled down at the middle of the design phase.

2. An Efficient Testing Plan:

Test plan is that part of the project plan that deals with the testing work that has to be done. A test plan does not go into much detail and only highlights which tasks have to be done, when they should start and end and how much effort has to be put in with respect to the other tasks. It also clearly highlights the dependencies among the tasks so that a critical path can be created without any surprises. Both test plan and test strategy are closely connected and any changes in one will result in changes in the other. There are three reasons why test plans are written:

- Writing test plans guides the thinking of the people involved in the testing process and helps them to focus on important topics and the challenges that they might encounter.
- The test planning process serves as the means through which communication takes place between members of the project team, peers, testers, managers and various other stakeholders.
- Test plan helps in managing change. In the early phases of the project, plans can be easily revised as more information is gathered. As the project evolves, the plan can be adapted accordingly. Updating the test plan at critical points can help in keeping the tests aligned with the project needs.

3. Unique Test Cases to Meet Requirements:

Test cases that are prepared are usually based on the test strategy which tells the testers how much of each kind of testing has to be done. The execution of the initial functional test requirements is done in the later stages of the first software build. A study by Cereca has found that 78% of IT professionals feel that when it comes to software projects, business enterprises are always out of sync with project requirements. The development team runs the software through a series of tests to make sure that the initial stakeholder requirements are met and paves the way for further beta testing and end user testing. The basis on which the test cases are designed are the acceptance criteria and the listed requirements for the software and the latest risk assessment of the software detailing what could go wrong. Most of the test cases must be prepared well in advance except for a few ad hoc testing. The implementation of the test cases allows for a fresh set of eyes to implement a wide range of 'real world' testing such as functional testing, running the software on various machines having different operating systems and carrying out testing for ease of use which is different from the procedures followed by the development team. A good test case will ensure that every requirement is met and the probability of uncovering defects is also high.

4. Relevant Test Data:

The input feed required for testing a software application is the test data. It consists of input test data as well as database test data that will be used while executing test cases. The quality of a software application is entirely dependent on the test data that is entered while testing. If an ideal set of data is used by the tester while executing the software application then there would be virtually no issues with the quality of the software being assured at 90%. By giving specific test data inputs, the software application can be tested for field and data validations. Along with the different examples of data that will be input, a starting database of data will also be needed to test query functions, delete functions and change functions. Test data development and test case development is usually done side by side.

5. Ideal Testing Environment:

An ideal place and the right equipment are required to carry out the testing in an effective manner. Unless the software that is being tested is very simple, a number of computers will be required. All components of the actual system on which the software application will eventually run on must be present. Test environments are usually the scaled down version of the actual system and therefore must contain all components that is required for testing the software. Building a test environment can be quite challenging as different regions on the mainframe computer or server, PCs and networks have to be set aside which have to be completely dedicated to the testing effort and they must be capable of being reset to the initial settings so that the testing can be carried out as often as needed. For usability or performance testing, lab rooms of equipment can be set aside. Setting up the testing environment comes under testing plan and therefore has to be completed before the testing begins.

Software testing is basically an investigation that is conducted to let the stakeholders know about the quality of the software application that is under test. Software testing also provides business enterprise with a holistic view of the software application allowing the enterprise to appreciate and understand the various risks involved in its implementation. If software testing has to be improved, business enterprise must make sure that the five essentials are well in place. Business enterprises can even ensure that the quality of software is impeccable by outsourcing software testing to an expert software testing service provider. Many software testers struggle with lack of involvement in the development process, inadequate resources and undocumented requirements.