
Software Testing Strategies to Improve Software Quality

Rakesh Kumar

Asstt. Prof. Deptt. Of Computer Science C.R.A.College Sonipat
rkphor@gmail.com.

ABSTRACT

It is my belief that software testing as a discipline is ripe for theoretical breakthroughs. Researchers are considering the right questions, and there are promising new approaches and exciting new results. It seems that new understanding of the testing process can lead to practical tools and techniques that revolutionize software development. I don't believe that testing will become easier or cheaper; rather, it will be more rational in the sense that expending effort will more dependably lead to better software. In this introductory essay I provide a personal view of testing, testing research, and their roles in software quality assurance.

In this paper, various types of software testing technique and various attributes of software quality are explained. Identifying the types of testing that can be applied for checking a particular quality attribute is the aim of this paper. All types of testing can not be applied in all phases of software development life cycle. Which testing types are applicable in which phases of life cycle of software development is also summarized.

Keywords: SDLC, Testing, Quality, Efficiency, Software.

INTRODUCTION

Software testing is both a discipline and a process. It is a separate discipline from software development. Software development is the process of coding functionality to meet defined end-user needs. While Software testing tends to be considered a part of development, it is really its own discipline and should be tracked as its own project. Software testing, while working very closely with development, should be independent enough to be able to hold-up or slow product delivery if quality objectives are not met. The objective of software testing is to find problems and fix them to improve quality. Software testing typically represents 40% of a software development budget. There are four main objectives of testing: Demonstration: It show that the system can be used with acceptable risk, demonstrate functions under special conditions and show that products are ready for integration or use. Detection: It discovers defects, errors, and deficiencies.



Determine system capabilities and limitations quality of components, work products, and the system. Prevention: It provides information to prevent or reduce the number of errors clarify system specifications and performance. Identify ways to avoid risks and problems in the future. Improving quality: By doing effective testing, we can minimize errors and hence improve the quality of software.

NEED FOR TESTING

Well, while making food, it's ok to have something extra, people might understand and eat the things we made and may well appreciate our work. But this isn't the case with software project development. If we fail to deliver a reliable, good and problem free software solution, we fail in our project and probably we may lose our client. So in order to make it sure, that we provide our client a proper software solution, we go for testing. We check out if there is any problem, any error in the system, which can make software unusable by the client. We make software testers test the system and help in finding out the bugs in the system to fix them on time.

TESTING AND DEBUGGING

The purpose of debugging is to locate and fix the offending code responsible for a symptom violating a known specification. Debugging typically happens during three activities in software development, and the level of granularity of the analysis required for locating the defect differs in these three. The first is during the coding process, when the programmer translates the design into an executable code. During this process the errors made by the programmer in writing the code can lead to defects that need to be quickly detected and fixed before the code goes to the next stages of development. Most often, the developer also performs unit testing to expose any defects at the module or component level. The second place for debugging is during the later stages of testing, involving multiple components or a complete system, when unexpected behavior such as wrong return codes or abnormal program termination may be found. A certain amount of debugging of the test execution is necessary to conclude that the program under test is the cause of the unexpected behavior.

SOFTWARE QUALITY



Everyone is committed to quality; however, the following statement shows some of the confusing ideas shared by many individuals that inhibit achieving a quality commitment: Quality requires a commitment, particularly from top management. Close cooperation of management and staff is required in order to make it happen.

a. Many individuals believe that defect-free products and services are impossible, and accept certain levels of defects as normal and acceptable. b. Quality is frequently associated with cost, meaning that high quality equals high cost. This is confusion between quality of design and quality of conformance. c. Quality demands requirement specifications in enough detail that the software produced can be quantitatively considered alongside those requirements. Many organizations are not capable or willing to expend the effort to produce specifications at the level of detail required. d. Technical personnel often believe that standards stifle their creativity, and thus do not abide by standards compliance. However, for quality to happen, well-defined standards and procedures must be followed. Quality cannot be achieved by assessing an already completed product. The aim therefore, is to prevent quality defects or deficiencies in the first place, and to make the products assessable by quality

assurance measures. Some quality assurance measures include: structuring the development process with a software development standard and supporting the development process with methods, techniques, and tools.

PROBLEM STATEMENT

The main purpose of software testing is to uncover errors which are not simply syntax errors in code but various other types of errors in all the documents produced during the software development e.g. software requirements document, design document, test plan etc. Various types of software testing techniques have been developed till date, but which type of testing technique will be suitable and sufficient for checking a particular document in which phase of software development life cycle is not yet clear. So here the problem is to a. Identify the testing techniques which can be applied at different levels and phases of software development life cycles. b. Identify the testing techniques which can be applied to measure which software quality attribute. Proposed Scheme Application of Testing to Measurement of Quality Attributes Different quality attributes need different types of testing to measure software quality.

SOFTWARE TESTING

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: blackbox testing and whitebox testing.

Blackbox Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

Whitebox Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

Types of Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the

specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Beta Testing

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

CONCLUSION

Quality is the main focus of any software engineering project. Without measuring, we cannot be sure of the level of quality in software. So the methods of measuring the quality are software testing techniques. This thesis report relates various types of testing technique that we can apply in measuring various quality attributes. Also which testing are related to various phase of SDLC. General SDLC processes are applied to different type of projects under different conditions and

requirements. There are various type of SDLC model. But in all these models, testing is applied after a particular stage and not in all the phases. In this thesis report, it is concluded that testing should be applied in all the phases of SDLC and not at a particular stage. Which type of testing technique can be applied to which type of SDLC phase is also summarized.

REFERENCES

- Jain Deepak, "Software Engineering Principle and Practices" First edition by Oxford University Press, ISBN-13: 978-0-19-569484-0, (2009).
- Bersoff, E.H. and A.M. Davis, "Impact of Lifecycle Modes on Software Configuration Management", ACM, pp104-108 (1991).
- Boris Beizer, "Software testing techniques", Second edition, (1990).
- Cem Karner, "Testing Computer Software", (1993). [9] IEEE "Standard Glossary of Software Engineering Terminology" (IEEE Std 610.12-1990), IEEE Computer society, (dec.10, 1990).