ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Different Techniques and Strategies for Software Testing

Divya Bhargava[*1], Aditya Veda[2]

[*1,2]Department of Information Technology, Acropolis Institute of Technology & Research, Indore-452001, India

divya15071992@gmail.com

Abstract

Software testing is one of the promising fields that help to improve the quality of the software by reducing the errors and fault from the software and also help to cut the maintenance and reduce the software cost. Most of the testing techniques, tools and methodologies were emerged out to improve the software quality from last few decades. The main focus of software testing is to provide software reliability, quality and reducing the cost of the software. At every phase of the software development life cycle, we try to apply software testing. In this paper, we will discuss the goal, principles and techniques used in the field of software testing.

Keywords: Software testing, Techniques, strategies, Objective.

Introduction

Software quality is one of the prime concerns for developers. Software is a key component in many of the devices and systems that spread through our society. Software defines the behavior of the infrastructure of modern life that includes telephone, networks etc. Software is an essential component of embedded applications that control unusual applications such as airplanes, spaceships, and air traffic control systems, as well as everyday appliances such as watches, cars, DVD players, garage door openers, cell phones, remote controllers and etc. [1] Testing is the primary method that the Organization uses to evaluate software under development.Software quality is nothing but the fulfillment of the end user requirements. Main areas where developers need to focus upon are quality assurance, quality planning and quality control. Software testing plays an important role in these areas by providing the way of reducing the bugs from the application. We can define “Software Testing” refers to process of evaluating the software with intention to find out error in it. Software testing is a technique intended at evaluating an attribute or capability of a program or product and determining that it meets its quality. [2] Software testing is also used to test the software for other software quality factors like reliability, usability, integrity, security, capability, efficiency, portability, maintainability, compatibility etc.

In section 2: we will see the objective of software testing, in section 3: we will see the software testing techniques, in section 4: we will see software testing strategies in section 5: we will conclude and finally in section 6: references.

Software Testing Objective

The objective of software testing is to eliminate the bugs from the software.

- A superior test case is one that has a prospect of finding an undiscovered error.
- A superior test is not redundant.
- A superior test should neither be too simple nor too complex.
- To check if the system does what it is expected to do.
- To check if the system is “Fit for purpose”.
- To check if the system meets the requirements and be executed successfully in the Intended environment.
- Executing a program with the intent of finding an error.
- Developed test cases for the valid and invalid conditions.
- Apply testing at early stage.
- Testing must be apply by third party for it’s effectiveness.

Software Testing Techniques

In this section we will discuss the software testing techniques. Software testing techniques is categorized in two main areas:

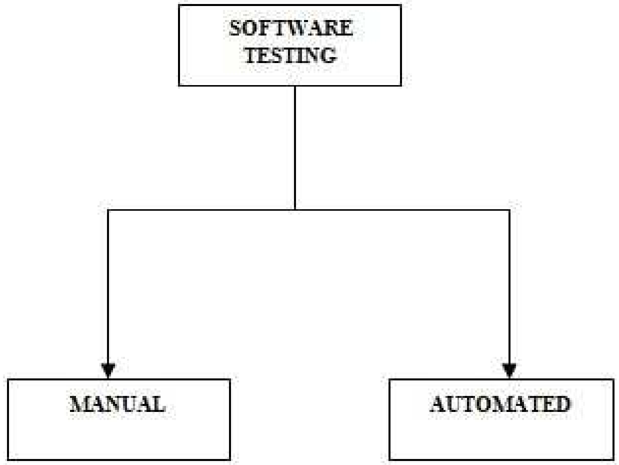

Figure 1: Software Testing Techniques

A. Manual Testing

"Manual Testing" is also known as static testing. Manual testing will take more time than the automated testing and it will also take huge amount of human resources. It will be perform by the Analyst, Developer and the testing team early at the stage.

It will include the different techniques:

- **Walkthrough:** It is the formal review technique. In this technique, the programmer and designer leads the members and interested parties and participants ask questions for intending to finding out the errors, violation of standards and other problems [3].
- **Informal Review:** Informal reviews may often be unnecessarily expensive (because of time-wasting through lack of focus), and frequently provide a sense of security which is quite unjustified by the relatively small number of real defects found and repaired [4].
- **Technical Review:** The purpose of a technical review is to arrive at a technically superior version of the work product reviewed, whether by correction of defects or by recommendation or introduction of alternative approaches. While the latter aspect may offer facilities that software inspection lacks, there may be a penalty in time lost to technical discussions or disputes which may be beyond the capacity of some participants [5].
- **Inspection:** The goal of inspection is to identify the defects in the product. It includes the software requirement specification and test plan. For example, if the inspector is checking the software requirement specification for finding the defects and if it finds than, he disagrees on those defects [6].

B. Automated Testing

Automated Testing required some special software to control the execution of the test and also compare the actual outcomes with the predicted outcomes.

It will include the different techniques:

- **Correctness Testing:** In this technique, the tester may or may not know the internal details of the product. It will tell the behaviour of the product. In this category we can use either black box testing or white box testing.
- **Performance Testing:** It will mainly concern with the performance of the product. The main objective of performance testing is to improve the throughput of the product. The prime goal of performance testing is to focus upon the responsiveness and stability of the system under workload. It will also evaluate the attributes like reliability and scalability. In this category we can perform the load testing, stress testing, soak testing and spike testing.

Software Testing Strategies

Software testing Strategies gives the blue print for testing. A software testing Strategy should be bendable enough to endorse a personalized testing approach at same time it must be accurate enough. Approach is generally developed by project managers, software engineer and testing expert.

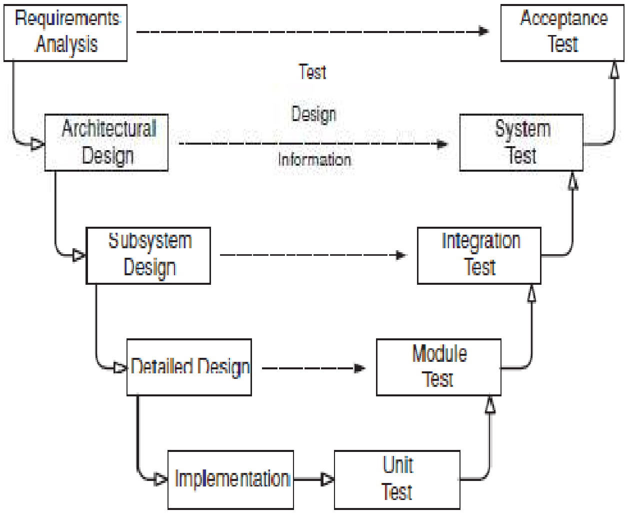

Figure 2: Software Testing Levels

- ❖ **Unit Testing:** In simple words, we can say that **"Unit Testing"** is the testing that tests the unit of code weather they are fit to use. In this testing strategy, individual module is tested. All the modules are independent to each other. This testing is mainly applied by the programmer itself. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

- **Integration Testing:** Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit.

Integration testing is of two types:

- ✓ **Top down Approach:** The top-down approach to integration testing requires the highest-level modules be test and integrated first. This allows high-level logic and data flow to be tested early in the process and it tends to minimize the need for drivers. However, the need for stubs complicates test management and low-level utilities are tested relatively late in the development cycle. Another disadvantage of top-down integration testing is its poor support for early release of limited functionality.
- ✓ **Bottom-Up approach:** The bottom-up approach requires the lowest-level units be tested and integrated first. These units are frequently referred to as utility modules. By using this approach, utility modules are tested early in the development process and the need for stubs is minimized. The downside, however, is that the need for drivers complicates test management and high-level logic and data flow are tested late. Like the top-down approach, the bottom-up approach also provides poor support for early release of limited functionality.

- **System Testing:** System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.
- **Acceptance Testing:** Acceptance testing (also known as user acceptance testing) is a type of testing carried out in order to verify if the product is developed as per the standards and specified criteria and meets all the requirements specified by customer. [4] This type of testing is generally carried out by a user/customer where the product is developed externally by another party. Acceptance testing falls under black box testing methodology where the user is not very much interested in internal working/coding of the system, but evaluates the overall functioning of the system and compares it with the requirements specified by them. User acceptance testing is considered to be one of the most important testing by user before the system is finally delivered or handed over to the end user.

Conclusion

This paper describes the details of software testing, their objective, techniques and strategies of software testing. It required experience to write test cases and testing can also helps to show whether the fault is present or not. By the use of the testing we can assure the quality of the product is meeting with the end user requirement. Test plan, test cases can help the developer as well as client to understand the product flow. We further discuss software testing techniques and software techniques strategies.

References

[1] Mats Grindal, Birgitta Lindstrom, Jeff Offutt, and Sten F. Andler. An evaluation of combination testing strategies. Empirical Software Engineering, 11(4):583– 611, December 2006.

[2] Sahil Batra and Dr. Rahul Rishi,”IMPROVING QUALITY USING TESTING STRATEGIES,” Journal of Gobal Research in Computer Science, Volume 2,No.6,June 2011.

[3] en.wikipedia.org/wiki/Software_walkthrough

[4] http://en.wikipedia.org/wiki/Software_review#Different_types_of_reviews

[5] en.wikipedia.org/wiki/Software technical review

[6] en.wikipedia.org/wiki/Software_inspection

[7] Software Engineering: A Practitioner's Approach by roger pressman.

[8] http://msdn.microsoft.com/en-us/library/ aa 292128.aspx

[9] Mohd. Ehmer Khan,”Different Forms of Software Testing Techniques for Finding Errors,”IJCSI International Journal of Computer Science Issues,Vol. 7, Issue 3, No 1, May 2010.

[10] Ajay Jangra, Gurbaj Singh, Jasbir Singh and Rajesh Verma,”EXPLORING TESTING STRATEGIES,”International Journal of Information Technology and Knowledge Management, Volume 4, NO.1,January-June 2011.