

A STUDY ON SOFTWARE TESTING AND SOFTWARE TESTING TECHNIQUES

Dr.G. Manivasagam¹

ABSTRACT

Software testing is an analysis of the software quality and, depending on the testing outcome, information is provided to the investors to understand about its quality. It can also offer an unbiased, self-determining view about the software which allows businesses to understand the risk issues and advantages in implementation of the software. This paper mainly focuses on various testing techniques.

1. INTRODUCTION

In general, testing techniques never limit or control the program execution process with the intention of finding software errors or defects [1].

Software testing can be defined as the procedure of authorizing and confirming that the software concerned has met the following criteria's [2]:

·Fulfills the necessities which guide its design and development

- It functions as anticipated,
- It can be used with the same characteristics
- Fulfills the requirement of the investor.

Depending on the methodology used for testing the software, it can be applied at any time in the software development process. But, conventionally after the requirement of the software that has to be developed is defined and the coding procedure is over, testing efforts are taken up. Depending on the software development methodology the test methodology varies. The task of the software testing extends to fix significant errors, flaws or defects in the application code.

The software testing process involves three main factors, namely verification process, validation process and finding of defects.

• Process of verification

During validation situation instructions are used to verify whether the software accomplishes the technical requirements specified. Here, specification means whether the software for a given input produces the expected output or not, and it is determined under various conditions.

• Process of validation

The validation metric is used to determine whether the requirement of the business is fulfilled by the software [3].

• Finding of Defects

The variation among the real and predictable output of the software is known as defect. The source of defect can be identified by finding whether the cause is due to specification, design or coding phase.

1.1 Software Test Life Cycle

In general the software testing life cycle consists of six different stages. They are used to determine what kind of testing activity has to be carried out during the process of software testing, which is well planned in an efficient way as shown in figure 1.1.

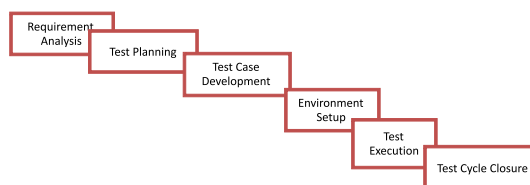


Figure 1.1: Software Test Life Cycle

1. Requirements Analysis:

In this stage the software testers evaluate the requirement of the customer and work in coordination with the developers to determine what requirements can be possibly applied for testing.

¹Asst.Professor, Department of Computer Science
Karpagam Academy of Higher Education, Coimbatore.

2. Test Planning:

In this stage planning on what kind of testing has to be done is made. It determines what has to be tested, how to perform the testing, strategy to be followed during software testing, environment in which tests must be conducted, methodology for testing and the availability of hardware and software.

Test plan defines the subsequent

- Testing Scope
- Resources Identification
- Strategies for testing identification
- Risk-involved identification
- Scheduling timeline
-

3. Test Case Development:

In this period tester creates test cases for testing the quality of the software. The tester creates the test script for generating automatic test cases. The activities involved in this stage are:

- Reconsideration and agreement of functional validation using matrix
- Reconsideration and agreement of environment of software testing
- Reconsideration of test scripts and test cases during baseline testing.

4. Test Environment setup:

In setting up environmental setup the subsequent factors to be considered are:

- Knowing the environment and architecture set-up
- Preparing a list of hardware and software that are required
- Confirming requirement of connectivity testing
- Arrangement of test data and the testing environment
- Preparing a check list for environment setup
-

5. Test Execution and Bug Reporting:

In this stage, after the execution of test cases defects are reported to the error-tracing system; then the programmers fix the bugs reported by the tester. The tester

then performs regression testing to confirm that the bugs are fixed and that they don't occur in other parts of the software.

6. Test Cycle closure

In this test cycle closure stage the following tasks are done:

- It tracks the closure defects
- Preparing metrics of testing based on test coverage, cost, software quality, objectives of professional after the completion of evaluation cycle
- Report of test closure will be prepared
- The severity of defects will be determined based on their distribution and type.

2. Software Testing Techniques

Different types of software testing techniques and approaches are useful in various applications. This section discusses basic dissimilarities among various approaches to testing software [4].

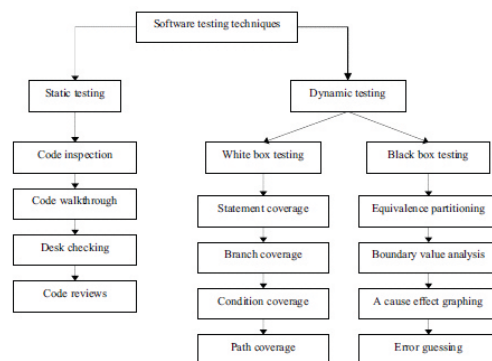


Figure 1.2: Software testing techniques

In general, techniques for software testing are divided into two main classes, namely static testing and dynamic testing [4].

In the approach of static analysis, a code reviewer will read the code of the program statement by statement and follow the logical program flow visually by passing an input. This kind of testing is extremely dependent on the skill of the reviewers. For visual review, it uses the requirement of program and document design. While in contrast the

technique of dynamic testing executes the program that is under test.

2.2.1 Static Analysis

For a long time most of the programmers assumed that programs were only for machine execution and not for humans so that the only way to test a program was by its execution on a machine justifying its output. This trend changed in the 1970s, and static-analysis-based methods began to be applied amid code completion, and execution-based testing was started. Some of the methods of static analysis are inspection of code, walkthroughs of code, checking desk and reviewing code. The two primary methods in static analysis are inspection of codes and walkthroughs. It is done by a team of people who are involved in preparatory work. During the meeting errors are determined and resolved by debugging process.

Code Inspections

It is a sequence of techniques and uncovering of errors for reading the source code. Here the focus is on procedures, filling the forms, etc. Two different activities are done in the period of inspection; they are narration and examination of code. According to the check list history codes are read and analyzed statement by statement.

Code Walkthroughs

The starting process is same as inspection process, but the difference is that instead of just reading the program against the checklist a person is assigned as a tester who comes with some test cases that represent a set of input and estimated output for the established segment. During the gathering session each test case is executed by walkthrough of the logic of the program and the values of the variables.

Desk Checking

It may be viewed as single person assessment. A tester is involved in reading a program and checking it against a bug list and test data through walks. The

disadvantage of this method is that it is undisciplined, ineffective and has no competition as in team work.

Code Reviews

It is a kind of calculating the unidentified process based on its excellence, maintenance, usability, extensibility and clearness. In this, programs are rated by a group of programmers in terms of an assured scale given in the analysis forms.

2.2.2 Dynamic Testing

In dynamic testing the program is tested by executing it under a test input data and the observed output. Generally, testing refers to dynamic testing which is of two types, namely black box and white box testing.

White box testing

The most-widely used technique is white box testing, which is also known as structural testing or logic coverage testing because it views the program structure [2, 3, 4]. The aim of white box testing is to work out various logic structures and program flows. The testing criteria followed in white box testing are [5]:

- **Statement testing**

Each statement in the software for testing has to be executed at least once. It can be applied to the object code directly.

- **Branch testing**

It is stronger than the statement coverage. Here, all the possible decisions have to be worked out at least once for finding out all types of outcomes. That is to say, the transfer of control has to be executed including coverage of statement. Anyway, some of the bugs can be identified if the statements and branches are deployed in a specific order, which may result in path testing.

- **Condition coverage**

In this case the test cases should be produced in such a way

that each state results in taking all probable outputs at least once.

- **Path testing**

In this testing all possible paths in software to be tested are executed, and it leads to increase the probability of detection of errors and it is stronger than branch and statement testing. It can be defined as conjunction of software's input and conjunction of predicates [6, 7]. Path is set to be feasible only when there exists an input for which the path is travelled or else the path is considered unfeasible.

Black Box Testing

Black box testing tests only the functionalities of the software under test but not its construction. There are four different kinds of black box testing, namely Boundary value analysis, equivalence partitioning, error-guessing and create-result graphing.

- **Equivalence partitioning**

It partitions the input domains into a finite set of equivalence classes; likewise one can assume that a descriptive rate of each class is equal to test of several additional values belonging to the respective session. It can also be applied for white box testing also.

- **Boundary value analysis**

There may be situations when the classes of test classes arise on the edges of input and output equivalence classes. The test cases which lie on the boundaries have higher payoff than other test cases do.

- **Cause-effect graphing**

This method is used to translate the natural language specification to formal language which denotes the incompleteness and uncertainties in the specification. In this,

simpler notation is used to represent the digital logic circuit

- **Error-guessing**

It is an ad-hoc process which is very difficult to formalize. This kind needs the assistance of experts to identify errors. The general idea is to compute a list of possible bugs and then write a test case created on the list obtained.

3.CONCLUSION

There are a number of testing techniques that are different from a software process, based on the specification and the software language. This paper summarizes software testing techniques, their significance in software development life cycle and the various processes involved in software testing life cycle. It also explains the various techniques of software testing and the working model of each of the approaches. Advance research is required to implement the most recent algorithm like Genetic Algorithm, which is likely to reduce the test cases in a given testing process.

References

- [1] Dr. K. Meenakshi Sundaram, D.Sasikala, P.Aarthi Rani, "A study on preprocessing a mammogram image using Adaptive Median Filter", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 3, March 2014, pp 10333 – 10337.
- [2] R.Ramani, Dr.S.Suthanthiravanitha, S.Valarmathy, "A survey of current image segmentation techniques for detection of breast cancer", International Journal of Engineering, Vol. 2, Issue 5, September- October 2012, pp.1124-1129.
- [3] Belal K. Elfarra, Ibrahim S. I. Abuhaiba, "New feature extraction method for mammogram computer aided diagnosis", International Journal of Signal

Processing, Image Processing and Pattern Recognition, Vol. 6, No. 1, February, 2013.

- [4] Shelly Gupta, Dharminder Kumar, Anand Sharma, “Data mining classification techniques applied for breast cancer diagnosis and prognosis”, Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2 No.2 Apr-May 2011, pp 188-195.
- [5] M.Vasantha, V. Subbiah Bharathi, “Classification of mammogram images using hybrid feature”, European Journal of Scientific Research ISSN 1450 – 216, Vol.57No. 1 (2011), 87-96.
- [6] Jwad Nagi, “Automated breast profile segmentation for ROI detection using digital mammograms”, IEEE EMBS Conference on Biomedical Engineering & Sciences (IECBES 2010), Kuala Lumpur.
- [7] Juan Shan, Wen Ju, Yanhui, Guoa, Ling Zhang, H.D. Cheng, “Automated breast cancer detection and classification using ultrasound images: A survey”, Pattern Recognition 43 (2010) 299-317.