



# Study of Software Testing and Types of Software Test Design Techniques

<sup>1</sup>Nidhi Kandhil nidhikandhil@gmail.com

<sup>2</sup>Vikas Chahar, Head, Department of Computer Science Vaish (P.G.) College, Rohtak

**Abstract :** Software Testing is the process of identifying the correctness and quality of software program. The purpose is to check whether the software satisfies the specific requirements, needs and expectations of the customer. In other words, testing is executing a system or application in order to find software bugs, defects or errors.

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

**Key Words :** Software Testing, black box, white box, walk through etc.

## Types of Software Test Design Techniques

There are various types of designing techniques, each of which is suitable for identifying a particular type of error. Hence, the challenge lies in selecting the right set of relevant test design techniques for the particular application. Software test design techniques can be broadly classified into two major categories: static and dynamic techniques.

- Static Techniques.
- Dynamic Techniques.

## Static Test Design Techniques:

Static testing refers to the testing of software manually or with the help of tools. Static testing is usually carried out during the early phase of software development life cycle. Static testing is useful for testing multiple aspects of a software, including source code, functional and requirement specifications, and design documents and models.

Static testing can be further divided into two categories based upon whether it is performed manually or with the help of tools.

ISSN 2454-308X



9 770024 543081



1. **Manual Static Test Design Techniques :** These are of the following kinds and differ in the formality of the process.
  - 1) **Walk through:** Walk through is not a formal process of review and is carried out by authors. The authors guide the participants through the document and try and understand their thought processes, in order to gain feedback and arrive at a common understanding.
  - 2) **Informal Review:** An informal review aims to improve the quality of the tested product through discussions and is usually not documented. The preliminary stage of the review may involve assessment by a two-member team whereas during later stages, the reviewing team size may increase and may require more meetings.
  - 3) **Technical Review:** A technical review is carried out by a trained moderator or technical expert. It is less formal as compared to an inspection. It is usually carried out as a peer review without any participation from management.
  - 4) **Audit:** An audit is an independent examination of all the documents related to the software, carried out by an external agent. Audits provide increased assurance to the various stakeholders of the project that the document is up to standards and devoid of any defects.
  - 5) **Inspection:** An inspection is the most formal type of review and is conducted by trained moderators. During an inspection, the documents and products are examined through a process of reviews and defects are identified. The defects identified are documented in a logging list or issue log and a formal follow-up is carried out.
  - 6) **Management Review:** The management review is a cross-functional review undertaken by an organization's top management. It includes analyses of customer satisfaction, determination of the cost of poor quality, performance trends and achievement of objectives defined in the business plan.
2. **Static Test Design Techniques using Tools : Analysis of Coding Standards:** The coding standards consist of a set of naming conventions, programming rules and layout specifications. Tools such as compilers are available that help in the analysis. A check of the coding standards during the early phase of the development cycle saves a lot of time and effort during subsequent stages.



- 1) ***Analysis of Code Metrics:*** Code metrics refer to the measurement of cyclomatic numbers, depth of nesting and number of lines of code. This information helps to keep a check on the code and prevents it from becoming excessively lengthy, unwieldy, difficult to understand and maintain.
- 2) ***Analysis of Code Structure:*** An analysis of the code structure gives an idea about the effort required to write or understand the code and to test it using particular tools and techniques. The analysis includes:
- 3) ***Analysis of the Control Flow Structure:*** The control flow structure refers to the sequence of execution of instructions.
- 4) ***Analysis of the Data Flow Structure:*** Understanding the fate of each data item as it is accessed and modified by the code.
- 5) ***Analysis of the Data Structure:*** Analysis of the organization of the data.

### **Dynamic Test Design Techniques:**

Dynamic testing involves execution of the test object (application) on a computer. The input data is fed into the test object (application) and the program is executed. In dynamic testing, various variable quantities such as memory usage, response time, CPU usage and overall performance of the software are analyzed against the expected output. Dynamic testing is carried out during the validation process. Dynamic test design techniques can be further classified into:

- Specification-based (black-box, also known as behavioral techniques)
- Structure-based (white-box or structural techniques)
- Experience- based Dynamic Testing Techniques:

**Specification-Based / Black Box Test Design Technique:** Also known as specification-based test design technique, the black box test design technique uses the software's external descriptions such as the technical specifications, design, customer requirements, etc. This implies that a tester who does not have any knowledge about the code or internal structure can also perform the test. Some of the well-known methods in this technique are:

- 1) ***Boundary Value Analysis:*** The best method of the black box design technique, boundary value analysis comprises testing the input values at the boundaries. Generally, the input values are put to test at the initial stages to reduce the chances of causing errors



- 2) **Decision Table Testing:** Identifying the test conditions as per the decision tables which are associated with different conditions is known as decision table testing. Each and every decision keeps up a correspondence to predicates, relations, or variables. Decision tables which have a ‘hyphen’ symbol are known to have little influence on the actions that are being performed.
- 3) **State Transition Testing:** Identifying the test conditions from a state table is state transition testing. This test case design method is best used for applications with implanted workflow within them.
- 4) **Equivalence Class Partitioning:** The purpose of this type of test design method is to reduce the number of tests by dividing the different types of test. Once the tests have been divided, the system will behave in a similar way for the different tests with equivalence partition.
- 5) **Use Case Testing:** The method of deriving the test condition from use case is known as use case testing. The test cases designed as per this type of method execute the different business scenarios and user functionalities.

### Structure-based / White Box Test Design Technique

The exact opposite of the black box test design technique, white box test design technique necessitates knowledge of the internal structure of a program. Some of the methods implemented in this type of technique are:

- 1) **Condition Coverage:** Most of the outcomes of test cases designed as per the white box test design technique are derived by following the condition coverage method. By following this method, the test cases are designed in a manner that the condition outcomes (the evaluation of a condition as ‘true’ or ‘false’) are executed automatically.
- 2) **Decision Coverage:** In the decision coverage method, the test cases are designed in a manner that the decision outcomes are executed automatically. It ensures that each possible branch from a decision point is implemented at least once.
- 3) **Statement Coverage :** This method involves calculation of the percentage of executable statements which are being exercised by the test suite.
- 4) **Multiple Condition Coverage:** Every combination of ‘true’ or ‘false’ for the conditions related to a decision have to be tested in this technique.



### **Experience-Based Test Design Technique:**

As the name suggests, the experience-based technique neither involves internal, nor the external structure, but is based on experience. Some of the methods followed are:

- 1) **Exploratory Testing:** This method, generally conducted by business analysts and experts, is followed to test the applications without any documentation.
- 2) **Fault Attack:** One of the widely used method of the experience based test design technique, fault attack involves testers anticipating the errors, defect data availability, etc., as per their experience

**Conclusion :** There are multiple testing techniques belonging to the static and dynamic test design categories. Different types of testing techniques have their own unique benefits. Therefore, the use of any particular technique has to be considered with maximum emphasis on the type of application.

### **References :**

- 1) Test Design Techniques by G C REDDY
- 2) Defect Clustering in Software Testing by Amir Ghahrai
- 3) [https://www.tutorialspoint.com/software\\_testing\\_dictionary/test\\_case\\_design\\_technique.htm](https://www.tutorialspoint.com/software_testing_dictionary/test_case_design_technique.htm)
- 4) <https://sanitytesting.wordpress.com/2014/12/13/defect-clustering-and-pesticide-paradox/>
- 5) <http://www.softwaretestingandistqb.com/7-fundamental-principles-of-software-testing-as-per-istqb/>
- 6) <http://toolsqa.com/software-testing/test-basis/>
- 7) <https://www.testingexcellence.com/istqb-tutorial/>
- 8) <http://softwaretestingtimes.com/2010/04/software-testing-tutorials-for.html>