



## A Study of Software Testing Techniques

Shabeeh Ahmad Siddiqui

Sr Lecturer, Al-Ahgaff University, Yemen

### **Abstract:**

*Software Testing is major part of test plan where successful completeness of the project depends. Software Testing decides and overcome the errors and deviations of the software product from the requirements. In this paper we have described and compare the causes of white box and black box testing.*

**Keywords:** Product; White Box; Black Box; SoftwareTesting ;

---

### **Introduction:**

Software testing is an important component of software quality assurance, and many software organizations are spending up to 40% of their resources on testing. For life-critical software (e.g., flight control) testing can be highly expensive. Because of that, many studies about risk analysis have been made. This term means the probability that a software project will experience undesirable events, such as schedule delays, cost overruns, or outright cancellation, and more about this in [10]. There are a many definitions of software testing, but one can shortly define that as: A process of executing a program with the goal of finding errors . So, testing means that one inspects behavior of a program on a finite set of test cases (a set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement, see [11]) for which valued inputs always exist. In practice, the whole set of test cases is considered as infinite, therefore theoretically there are too many test cases even for the simplest programs. In this case, testing could require months and months to execute. So, how to select the most proper set of test cases? In practice, various techniques are used for that, and some of them are correlated with risk analysis, while others with test engineering expertise. Testing is an activity performed for evaluating software quality and for improving it. Hence, the goal of testing is systematical detection of different classes of errors (error can be defined as a human action that produces an incorrect result) in a minimum amount of time and with a minimum amount of effort.

Software Testing is a part of software development life cycle. Software Testing works parllely with design phase of the project which is part of software testing life cycle. We have described the black box and white

box methods of software testing. Equivalent Partitioning, Boundary Value Analysis, Cause-Effect Graphing Techniques, and Comparison Testing are techniques of black box testing method and Basis Path Testing, Loop Testing, and Control Structure Testing are techniques of white box testing methods.

## TESTING METHODS

Test cases are developed using various test techniques to achieve more effective testing. By this, software completeness is provided and conditions of testing which get the greatest probability of finding errors are chosen. So, testers do not guess which test cases to choose, and test techniques enable them to design testing conditions in a systematic way. Also, if one combines all sorts of existing test techniques, one will obtain better results rather if one uses just one test technique.

Software can be tested in two ways, in another words, one can distinguish two different methods:

1. Black box testing, and
2. White box testing.

### **White box testing:**

*White box testing* is highly effective in detecting and resolving problems, because bugs (*bug or fault is a manifestation of an error in a software*, see [12]) can often be found before they cause trouble. We can shortly define this method as *testing software with the knowledge of the internal structure and coding inside the program* (see [13]). White box testing is also called white box analysis, clear box testing or clear box analysis. It is a strategy for software **debugging** (*it is the process of locating and fixing bugs in computer program code or the engineering of a hardware device*, see [14]) in which the tester has excellent knowledge of how the program components interact. This method can be used for Web services applications, and is rarely practical for debugging in large systems and networks. Besides, white box testing is considered as a **security testing** (*the process to determine that an information system protects data and maintains functionality as intended*, see [6]) method that can be used to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities .

### **Black box testing:**

*Black box testing* is testing software based on output requirements and without any of the internal structure or coding in the program . In another words, a black box is any device whose workings are not understood by or accessible to its user. For example, in telecommunications, it is a resistor connected to a phone line that

makes it impossible for the telephone company's equipment to detect when a call has been answered. In data mining, a black box is an algorithm that doesn't provide an explanation of how it works.

In film-making, a black box is a dedicated hardware device: equipment that is specifically used for a particular function, but in the financial world, it is a computerized trading system that doesn't make its rules easily available.

In recent years, the third testing method has been also considered – **gray box testing**. *It is defined as testing software while already having some knowledge of its underlying code or logic*. It is based on the internal data structures and algorithms for designing the test cases more than black box testing but less than white box testing. This method is important when conducting integration testing between two modules of code written by two different developers, where only interfaces are exposed for test. Also, this method can include reverse engineering to determine boundary values. Gray box testing is non-intrusive and unbiased because it doesn't require that the tester have access to the source code.

The main characteristics and comparison between white box testing and black box testing are follows.

### **Black Box Testing Versus White Box Testing**

#### ***Black Box Testing:***

Performing the tests which exercise all functional requirements of a program;

Finding the following errors:

1. Incorrect or missing functions;
2. Interface errors;
3. Errors in data structures or external database access;
4. Performance errors;
5. Initialization and termination errors.

#### **Advantages of this method:**

1. The number of test cases are reduced to achieve reasonable testing;
2. The test cases can show
3. presence or absence of classes of errors.

#### ***White Box Testing:***

- Considering the internal logical arrangement of software;
- The test cases exercise certain sets of conditions and loops;

▪ **Advantages of this method:**

- ✓ All independent paths in a module will be exercised at least once;
- ✓ All logical decisions will be exercised;
- ✓ All loops at their boundaries will be executed;
- ✓ Internal data structures will be exercised to maintain their validity.

## CONCLUSION

Software testing is a component of **software quality control (SQC)**. *SQC means control the quality of software engineering products, which is conducting using tests of the software system* (see [6]). These tests can be: *unit tests* (this testing checks each coded module for the presence of bugs), *integration tests* (interconnects sets of previously tested modules to ensure that the sets behave as well as they did as independently tested modules), or *system tests* (checks that the entire software system embedded in its actual hardware environment behaves according to the requirements). SQC also includes formal check of individual parts of code, and the review of requirements documents.

SQC is different from **software quality assurance (SQA)**, *which means control the software engineering processes and methods that are used to ensure quality* (see [6]). Control conduct by inspecting quality management system. One or more standards can be used for that. It is usually *ISO 9000*. SQA relates to the whole software development process, which includes the following events: software design, coding, source code control, code reviews, change management, configuration management, and release management.

Finally, *SQC is a control of products, and SQA is a control of processes*.

Eventual bugs and defects reduce application functionality, do not look vocational, and disturb company's reputation. Thence, radically testing is very important to conduct. At that way, the defects can be discovered and repaired. Even if customers are dissatisfied with a product, they will never recommend that product, so product's cost and its popularity at the market will decrease.

Besides, *customer testing* is also very important to conduct. Through this process one can find out if application's functions and characteristics correspond to customers, and what should be changed in application to accommodate it according to customer's requests.



Large losses can be avoided if timely testing and discovering bugs in initial phases of testing are conducting. Deficits are minor if the bugs are discovered by testing within the company, where developers can correct errors rather than if the bugs are discovered in the phase of customer testing, or when the application is started “live” in some other company or system for which the application is created. In that case, the losses can be enormous.

Therefore software testing is greatly important, and test techniques too, because they have the aim to improve and make easier this process.

There is considerable controversy between software testing writers and consultants about what is important in software testing and what constitutes responsible software testing.

## REFERENCES

- [1] [http://www.his.sunderland.ac.uk/~cs0mel/comm83wk5\\_doc](http://www.his.sunderland.ac.uk/~cs0mel/comm83wk5_doc), February 08, 2009.
- [2] Stacey, D. A., “Software Testing Techniques”
- [3] Guide to the Software Engineering Body of Knowledge, Swebok – A project of the IEEE Computer Society Professional Practices Committee, 2004.
- [4] “Software Engineering: A Practitioner’s Approach, 6/e; Chapter 14: Software Testing Techniques”, R.S. Pressman & Associates, Inc., 2005.
- [5] Myers, Glenford J., IBM Systems Research Institute, Lecturer in Computer Science, Polytechnic Institute of New York, “The Art of Software Testing”, Copyright 1979. by John Wiley & Sons, Inc.
- [6] Wikipedia, The Free Encyclopedia, <http://en.wikipedia.org/wiki/>
- [7] [http://www2.umassd.edu/CISW3/coursepages/pages/CI\\_S311/outline.html](http://www2.umassd.edu/CISW3/coursepages/pages/CI_S311/outline.html)
- [8] Parezanovic, Nedeljko, “Racunarstvo i informatika”, Zavod za udzbenike i nastavna sredstva – Beograd, 1996.
- [9] Wei–Tek, Tsai, “Risk – based testing”, Arizona State University, Tempe, AZ 85287
- [10] Redmill, Felix, “Theory and Practice of Risk-based Testing”, Software Testing, Verification and Reliability, Vol. 15, No. 1, March 2005.
- [11] IEEE, “IEEE Standard Glossary of Software Engineering Terminology” (IEEE Std 610.12-1990), Los Alamitos, CA: IEEE Computer Society Press, 1990.
- [12] [http://www.testingstandards.co.uk/living\\_glossary.htm#\\_Testing](http://www.testingstandards.co.uk/living_glossary.htm#_Testing), February 08, 2009.