

## ANALYSIS OF SOFTWARE TESTING

---

Amit Khatri\*

### ABSTRACT

*This paper describes the analysis of software testing in software development. Software testing is a process of finding error and faults in the program so that the product will be bug free. A good quality of software can be developed through means of software testing. Verification and validation are the two types of testing which have major impact in building any software and before its deliver to the customer. This paper aims to discuss the various types of existing testing techniques for the development of better quality of software like verification, validation, black box and white box testing etc.*

---

**Keywords:** Software Testing, Testing Techniques, Quality, Verification, Validation, Automation Testing, Manual Testing.

---

### Introduction

In general, Testing is the process of identifying the completeness, correctness, and quality of software. According to the IEEE, Testing is defined as the process of assessing a system by automated or manual means to check that it satisfies the specified requirements and to identify the difference between the authenticate and expected results. Automated testing is mainly suitable when testing is functionally based and more suitable when test cases are run repeatedly whereas manual testing is mainly suitable when testing is for user experience and when the test case are supposed to run only once or twice. The results generated through automated testing are more accurate while in case of manual these are less accurate.

Software is tested because testing helps in reducing the errors and makes sure that software is developed according to the customer requirements or business needs. It improves the quality of software. The purpose of testing is verification and validation and detecting the error to find out the problems associated with the software.

**Verification** is the process of involving review, analysis, inspection of the software and **Validation** is the process of identifying the software is designed and built as per the customer requirements.

Verification	Validation
<ul style="list-style-type: none"><li>• Are u doing it right?</li><li>• It is done in every phase of the project.</li></ul>	<ul style="list-style-type: none"><li>• Have u build the right thing?</li><li>• It is done at the end after the completion of the protect.</li></ul>
<ul style="list-style-type: none"><li>• It is done by software developer.</li><li>• It concerned with phase of containment of errors.</li></ul>	<ul style="list-style-type: none"><li>• It is done by the tester.</li><li>• Main aim is to make the final product error free.</li></ul>
<ul style="list-style-type: none"><li>• It involves unit, review, inspection, and integration testing.</li><li>• Static and dynamic activities.</li></ul>	<ul style="list-style-type: none"><li>• It involves system testing.</li><li>• only dynamic activities.</li></ul>

---

\* M.Tech, Computer Science.

## Verification vs Validation

### Types of testing levels:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

### Unit Testing

It is a testing in which a individual unit or blocks of the software is tested. It is done by developers. Unit testing checks every part before its integration. The main advantage of this testing is that it helps in identifying the errors at the early stage of software development.

### Integration Testing

It is a testing in which testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after the unit testing. The individual unit are integrated together to form a complete software.

### Types of integration testing:

- Big bang testing
- Bottom-up testing
- Top-down testing
- Sandwich testing

### System Testing

It is a level of testing that validates the complete and fully integrated software product. The purpose is to evaluate the end-to-end system specification (SRS). It is also known as black box testing. It is based on the functional and non- functional requirements.

### Types of system testing:

- Performance testing
- Recovery testing
- Stress testing
- Security testing

**Performance testing** involves speed, stability, and scalability of the product. Recovery testing means if in case of any failure or system crash the software recovers from expected or unexpected failures without loss of data. Stress testing is used to test the system beyond the limits of its requirements. Security testing involves identification of flaws that may harm the software or lead to any failure.

### Acceptance Testing

The purpose of acceptance testing is to allow customer to validate all the requirements (SRS). This type of testing is conducted when the software or product is built for a specified customers and not for a large public. The process follows by the acceptance testing is Customers test the software->feedback->modification->software delivery

### Acceptance testing is mainly of two types:

- Alpha testing
- Beta testing

Alpha testing	Beta testing
• It is done by the customer at the developer site.	• It is done by the customer at the user site.
• It is conducted in a controlled environment.	• It is conducted at the real time environment.
• Developer is present.	• Developer is not present.
• It is carried out before the release of product to the customer.	• It is carried out after the release of product to the customer.
• All the error and failures are recorded.	• All the failures are reported from the customer side.
• It involves white plus black box testing.	• It involves black box testing.

## Alpha vs Beta Testing

### Testing techniques:

The most used types of software testing techniques are:

- White box testing
- Black box testing

**White box testing** is also known as clear box or glass box testing. This testing is more effective in detecting and solving problems because errors can be traced before they create fault. In this testing software developer have knowledge of internal structure and code used inside the program. Tools used in white box testing are Veracode, Nunit, Rctest etc.

**Black box testing** is based on the output requirements and in this method, testing is done without the knowledge of internal structure and code of the program.

White Box Testing	Black Box Testing
<ul style="list-style-type: none"> <li>• What the software is supposed to do and aware of how it does</li> </ul>	<ul style="list-style-type: none"> <li>• What the software is supposed to do but not aware how it does.</li> </ul>
<ul style="list-style-type: none"> <li>• There is need of understanding of programming language used in the program.</li> </ul>	<ul style="list-style-type: none"> <li>• There is no need of understanding of programming language.</li> </ul>
<ul style="list-style-type: none"> <li>• It is done by the developers.</li> </ul>	<ul style="list-style-type: none"> <li>• It is done by the test engineers.</li> </ul>
<ul style="list-style-type: none"> <li>• Developers should know internal design of the code.</li> </ul>	<ul style="list-style-type: none"> <li>• There is no need to know about the internal design of the code.</li> </ul>
<ul style="list-style-type: none"> <li>• It involves unit testing mainly.</li> </ul>	<ul style="list-style-type: none"> <li>• It involves unit, integration, and system testing.</li> </ul>

### White Box vs Black Box Testing

Some general classification of white box testing techniques are:

- Control flow testing
- Data flow testing
- Branch testing
- Statement coverage testing
- Decision coverage testing

### Statement Coverage Technique

It is used to design white box testing. It involves execution of all statement of the source code at least once. It covers total no of executed statements in the source code out of total statements presents in the source code. It also covers dead code and unused code.

For example:

```
main()
{
int a1, a2, a3;
if (a1>=2 && a1>=3)
printf("%d a1 is the largest number."a1);
else if (a2>=a1 && a2>=a3)
printf("%d a2 is the largest number."a2);
else
printf("%d a3 is the largest number."a3);
}
```

### Condition Coverage Technique

It is also used to design white box testing. It is used to cover all the conditions. This technique is also known as predicate coverage in which each one of the Boolean expression have been evaluated to both true and false.

```

For example:
read x, y, z;
if(x==0|| y==0)
{
print 1;
}
else
{
If(z==0 && d==0)
{
print 2;
}
}
}

```

### Data Flow Testing

It is one of the white box testing techniques. It focuses on mainly two points:

- In which statement the variables are defined
- In which statement the variables are used

It also designs the test cases that covers control flow paths around variable definitions and their uses in the modules.

For example:

```

read a, b, c;
if(a>b)
x=a+1
print x;
else
x=b-1;
print z;

```

### Boundary Value Testing

it is black box testing technique. It is helpful for detecting threats or any error that happened at the boundary values of valid or invalid partitions rather than focusing on the center of the input data.

Let us assume a test case that takes the speed of a bike from 40 to 80 to get the best fuel efficiency.

Invalid test case = Min value-1 (39)

Invalid test case = Min, +Min, Max, -Max (40, 41, 79, 80)

Invalid test case = Max value + 1 (81)

### Conclusion

Software testing is the activity of software engineering whose aim is to detecting errors or faults. software testing reduces the cost and time and help in building the error free product. Testing plays a vital role in developing the best quality product or software and in developing software according to the need of customer. This paper aims to describe the various software testing techniques like black box testing, white box testing, verification, and validation testing.

### References

1. P. Ron. Software testing. Vol. 2. Indianapolis: Sam's, 2001.
2. S. Amland, "Risk-based testing:" Journal of Systems and Software, vol. 53, no. 3, pp. 287–295, Sep. 2000.
3. Redmill and Felix, "Theory and Practice of Risk-based Testing", Software Testing, Verification and Reliability, Vol. 15, No. 1, March 2005.

4. B. Agarwal et al., "Software engineering and testing". Jones & Bartlett Learning, 2010.
5. Available from [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing).
6. Available from: <http://technav.ieee.org/tag/1579/software-testing>
7. Bansal, A; (2014) "A Comparative Study of Software Testing Techniques", <http://www.ijaprr.com/download/1440480921.pdf>.
8. Stacey, D. A., "Software Testing Techniques"
9. Guide to the Software Engineering Body of Knowledge, Swebok – A project of the IEEE Computer Society Professional Practices Committee, 2004.
10. R. Chauhan and I. Singh, "Latest Research and Development on Software Testing Techniques and Tools", INPRESSCO, International Journal of Current Engineering and Technology, 2014.
11. S. P and D. N, "Automation of Software Testing in Agile Development – An Approach and Challenges with Distributed Database Systems", GJRA - GLOBAL JOURNAL FOR RESEARCH ANALYSIS, vol. 3, no. 7, 2014.

