Chapter:17

The difference between Verification and Validation is as follow:

| **VERIFICATION** | **VALIDATION** |
| --- | --- |
| It includes checking documents, design, codes and programs. | It includes testing and validating the actual product. |
| Verification is the static testing. | Validation is the dynamic testing. |
| It does *not* include the execution of the code. | It includes the execution of the code. |
| Methods used in verification are reviews, walkthroughs, inspections and desk-checking. | Methods used in validation are Black Box Testing, White Box Testing and non-functional testing. |
| It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. |
| It can find the bugs in the early stage of the development. | It can only find the bugs that could not be found by the verification process. |
| The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| Quality assurance team does verification. | Validation is executed on software code with the help of testing team. |
| It comes before validation. | It comes after verification. |
| It consists of checking of documents/files and is performed by human. | It consists of execution of program and is performed by computer. |

Chapter:18

**What is BLACK Box Testing? Techniques, Example & Types**

**Black Box Testing**

**Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

[](https://www.guru99.com/images/stories/blackbox.png)

The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation. Consider the following video tutorial-

**How to do BlackBox Testing**

Here are the generic steps followed to carry out any type of Black Box Testingnstagram Looks Ahead to the Future at CES

* Initially, the requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

**Types of Black Box Testing**

There are many types of Black Box Testing but the following are the prominent ones -

* **Functional testing** - This black box testing type is related to the functional requirements of a system; it is done by software testers.
* **Non-functional testing**- This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
* **Regression testing**- [Regression Testing](https://www.guru99.com/regression-testing.html) is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

**Tools used for Black Box Testing:**

Tools used for Black box testing largely depends on the type of black box testing you are doing.

[See More](https://eb2.3lift.com/pass?tl_clickthrough=true&redir=https%3A%2F%2Fwww.ibm.com%2Fsecurity%3Futm_content%3D000039JJ%26utm_term%3D10013707%26p1%3DProgrammatic%26p2%3D280910877%26p3%3D132814162%26cm_mmc%3DDisplay_N1114924.2079317MEDIAMATHPROGRAM-_-Portfolio%2BSecurity_Security%2BConversation-_-WW_WW-_-280910877_Security%2BBrand%2BCOVID%2BHealthcare-Gen-V1-1920x1080-Video-VPAID-NULL-Hi-NPB-15sec%26cm_mmca1%3D000039JJ%26cm_mmca2%3D10013707%26cm_mmca4%3D280910877%26cm_mmca5%3D132814162%26cm_mmca6%3D0%26dclid%3DCInmzaD7zOsCFXOCrAIdH3YCDQ&bc=4.0&pr=0.655&brid=561162&bmid=3690&clid=5545520&biid=3690&aid=122062525409140489540&did=17824&tid=5943759&bcud=4000&sid=16982&ts=1606103024&cb=79345)

* For Functional/ Regression Tests you can use - [QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html), [Selenium](https://www.guru99.com/selenium-tutorial.html)
* For Non-Functional Tests, you can use - [LoadRunner](https://www.guru99.com/loadrunner-v12-tutorials.html" \t "_blank), [Jmeter](https://www.guru99.com/jmeter-tutorials.html" \t "_blank)

**Black Box Testing Techniques**

Following are the prominent[Test Strategy](https://www.guru99.com/how-to-create-test-strategy-document.html)amongst the many used in Black box Testing

* **Equivalence Class Testing:** It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
* **Boundary Value Testing:** Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.
* **Decision Table Testing**: A decision table puts causes and their effects in a matrix. There is a unique combination in each column

**What is WHITE Box Testing? Techniques, Example, Types & Tools**

**White Box Testing**

**White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "[Black Box Testing](https://www.guru99.com/black-box-testing.html)" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

531.9K

**What do you verify in White Box Testing?**

White box testing involves the testing of the software code for the following:

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

**How do you perform White Box Testing?**

To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include [Manual Testing](https://www.guru99.com/manual-testing.html), trial, and error testing and the use of testing tools as we will explain further on in this article.

[](https://www.guru99.com/images/2/white-box-testing.jpg)

**WhiteBox Testing Example**

Consider the following piece of code

Printme (int a, int b) { ------------ Printme is a function

int result = a+ b;

If (result> 0)

Print ("Positive", result)

Else

Print ("Negative", result)

} ----------- End of the source code

The goal of WhiteBox testing in software engineering is to verify all the decision branches, loops, statements in the code.

To exercise the statements in the above code, WhiteBox test cases would be

* A = 1, B = 1
* A = -1, B = -3