**White Box Testing**

**White Box Testing** is a testing technique in which software’s internal structure, design, and coding are tested to verify input-output flow and 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 perspective. On the other hand, White box testing in software engineering is based on the inner workings of an application and revolves around internal testing.

## 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

## White Box Testing Techniques

A major White box testing technique is Code Coverage analysis. Code Coverage analysis eliminates gaps in a [Test Case](https://www.guru99.com/test-case.html) suite. It identifies areas of a program that are not exercised by a set of test cases. Once gaps are identified, you create test cases to verify untested parts of the code, thereby increasing the quality of the software product

There are automated tools available to perform [Code coverage analysis](https://www.guru99.com/code-coverage.html). Below are a few coverage analysis techniques a box tester can use:

**Statement Coverage**: This technique requires every possible statement in the code to be tested at least once during the testing process of [software engineering](https://www.guru99.com/what-is-software-engineering.html).

**Branch Coverage:** This technique checks every possible path (if-else and other conditional loops) of a software application.

Apart from above, there are numerous coverage types such as Condition Coverage, Multiple Condition Coverage, Path Coverage, Function Coverage etc. Each technique has its own merits and attempts to test (cover) all parts of software code. **Using Statement and Branch coverage you generally attain 80-90% code coverage which is sufficient.**

Following are important WhiteBox Testing Techniques:

* Statement Coverage
* Decision Coverage
* Branch Coverage
* Condition Coverage
* Multiple Condition Coverage
* Finite State Machine Coverage
* Path Coverage
* Control flow testing
* Data flow testing

**Types of White Box Testing:**

White box testingencompasses several testing types used to evaluate the usability of an application, block of code or specific software package. There are listed below,

* **Unit Testing:** It is often the first type of testing done on an application.[Unit Testing](https://www.guru99.com/unit-testing-guide.html) is performed on each unit or block of code as it is developed. Unit Testing is essentially done by the programmer. As a software developer, you develop a few lines of code, a single function or an object and test it to make sure it works before continuing Unit Testing helps identify a majority of bugs, early in the software development lifecycle. Bugs identified in this stage are cheaper and easy to fix.
* **Testing for Memory Leaks**: Memory leaks are leading causes of slower running applications. A QA specialist who is experienced at detecting memory leaks is essential in cases where you have a slow running software application.

Apart from the above, a few testing types are part of both black box and white box testing. They are listed below

* **White Box**[**Penetration Testing**](https://www.guru99.com/learn-penetration-testing.html)**:**

 In this testing, the tester/developer has full information of the application’s source code, detailed network information, IP addresses involved and all server information the application runs on. The aim is to attack the code from several angles to expose security threats.

* **White Box Mutation Testing**:

[Mutation testing](https://www.guru99.com/mutation-testing.html) is often used to discover the best coding techniques to use for expanding a software solution.

**White Box Testing Tools:**

Below is a list of top white box testing tools.

* [EclEmma](https://www.eclemma.org/download.html)
* [NUnit](http://nunit.org/)
* [PyUnit](https://www.guru99.com/python-unit-testing-guide.html)
* [HTMLUnit](http://htmlunit.sourceforge.net/)
* [CppUnit](https://sourceforge.net/projects/cppunit/)

**Advantages of White Box Testing**

* Code optimization by finding hidden errors.
* White box tests cases can be easily automated.
* Testing is more thorough as all code paths are usually covered.
* Testing can start early in [SDLC](https://www.guru99.com/software-development-life-cycle-tutorial.html) even if GUI is not available.

**Disadvantages of WhiteBox Testing**

* White box testing can be quite complex and expensive.
* Developers who usually execute white box test cases detest it. The white box testing by developers is not detailed and can lead to production errors.
* White box testing requires professional resources with a detailed understanding of programming and implementation.
* White-box testing is time-consuming, bigger programming applications take the time to test fully.