**What is White Box Testing and Why is it Done?**

White Box Testing is a software testing technique where we test the design, internal structure and software coding to validate the flow of input-output and to improve the software usability, security and design. It’s given the name “white box testing” because the code is made visible to the penetration testers. For this reason, it is also called Open Box Testing, Clear Box Testing, Glass Box Testing or Transparent Box Testing.

White Box testing is one of the two part of the Box Testing method of software testing. The other part, Black Box Testing, is done from an end-user’s perspective. However, White box testing revolves around internal testing and is based on an application’s inner working.

**Why Do We Perform White Box Testing?**

We can perform a white box test at unit, system and integration levels of application development. The basic goal of this test is to authenticate a working flow for a software application. When we test a number of predefined inputs against desired or expected outputs, we can identify bugs in the system if a particular input does not show an expected output.

Hence, in a nutshell, white box testing involves testing a software program for:

* Predefined outputs
* Internal security loopholes
* Flow of input through the code
* Poorly constructed or broken paths in coding process
* Functionality of conditional loops
* Individually testing each object, statement and function
* Existence of bugs and errors such as
  + Design errors that arise due to difference between the logical program flow and its actual implementation
  + Logical errors that creep into our program during designing and implementation of functions and conditions
  + Typographical and syntax errors

**Steps in Performing White Box Testing**

The white box testing process can be simply described in two steps:

**Step 1: Understand the Source Code**

The first step is to understand the source code of a software application. Since in a white box test, the tester works with the inner workings of the application, they must be proficient in the programming languages used. They must also be knowledgeable in secure coding practices and be able to find security issues within the application. This is important to prevent cyber attacks who may inject malicious code into the software.

**Step 2: Create Test Case and Execute**

In the second step, the tester will test the source code of the application for its flow and structure. They create tests for each process or a series of processes in the application and execute it. Other methods include trial and error testing or the use of white box testing tools.

White box tests can be complex, depending upon the application you are testing. A small software application with simpler operations can be tested quickly, whereas bigger applications may take days, weeks or even longer.