

# White Box Testing: An Overview

## By Vinod Dandoti

### Overview

This paper introduces white box testing and describes how to perform white box testing, and tools and techniques relevant to white box testing. It follows traditional white box testing techniques. It assumes the reader to be familiar with general concepts of software testing.

### What is White Box Testing?

White box testing is performed based on the knowledge of how the system is implemented. White box testing consists of analysis of data flow, control flow, information flow, coding practices, and exception and error handling within the system to ensure correct software behavior.

Many Organizations perform white box testing any time during the life cycle after the code development, However it is a good practice to start white box testing along with the unit testing.

Testers needs requires thorough knowledge of software under test from the source code point of view. They need to think like a software breaker. Testers also need to have extensive experience of usage of different white box testing tools and techniques.

### How to perform White Box Testing

The common white box testing process includes the following

1. Development of test strategy and define all the testing activities.
2. Development of a detailed test plan based on the strategy.
3. Setup the test environment for test case execution.
4. Execution of test cases and reporting.

The following sections discuss inputs, activities, and deliverable outputs in detail.

### Inputs

Source code, specification/requirements documentation, design documentation, and quality assurance related documentation.

Architectural and design , including test planning, test case creation, test data selection, test technique selection, and test entry/exit criteria selection.

Design documentation is essential to improve program understanding and to develop effective test cases that validate design decisions and assumptions.

Other relevant documents for program understanding should be available to testers.

## **Test Strategy**

Test strategy is normally a management activity. A test manager is responsible for developing and managing a test strategy.

The purpose of a test strategy is to clearly define the major activities, decisions, and challenges in the white box testing effort. This includes defining testing scope, techniques, metrics, environment setup, and skill requirements.

## **Test Plan**

The test plan is manifestation of the test strategy and organizes the testing process including test areas covered, technique implementation, test case and data selection, results validation, cycles, and entry and exit criteria based on metrics. In general, the test plan should incorporate both a high-level outline of which areas are to be tested and what methodologies are to be used and a general description of test cases, including prerequisites, setup, execution, and a description of what to look for in the test results. The high-level outline is useful for administration, planning, and reporting, while the more detailed descriptions are meant to make the test process go smoothly.

## **Test plan provide following few benefits**

- Written record of what is to be done.

- Test plans provide a way to measure progress.
- A test plan allows the priorities the testing.

## **Test Case Development**

Test case is a collection of step by step instruction to test a functionality, including prerequisites, test inputs, expected results. The testing team is responsible for development of test cases.

## **Test Environment**

Before performing any testing it is essential to setup an effective test environment. The test environment generally consists of a single computer for small applications, but for enterprise-level software systems, the test environment is much more complex, and the software is usually closely coupled to the environment.

It is necessary for effective testing, the testers to have more control over the environment. This is because the tester must be able to examine and manipulate software/environment interactions at a greater level of detail.

## **Test Execution**

Test engineers will be responsible for execution and reporting the status.

### **Advantages of White box testing are:**

- i) As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
- ii) The other advantage of white box testing is that it helps in optimizing the code
- iii) It helps in removing the extra lines of code, which can bring in hidden defects.

### **Disadvantages of white box testing are:**

- i) As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, which increases the cost.
- ii) And it is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

### **White Box testing types**

#### *Static and dynamic Analysis*

Static analysis involves going through the code in order to find out any possible defect in the code. Dynamic analysis involves executing the code and analyzing the output.

#### Statement Coverage:

In this type of testing the code is executed in

such a manner that every statement of the application is executed at least once. It helps in assuring that all the statements execute without any side effect.

#### *Branch Coverage:*

No software application can be written in a continuous mode of coding, at some point we need to branch out the code in order to perform a particular functionality. Branch coverage testing helps in validating of all the branches in the code and making sure that no branching leads to abnormal behavior of the application.

#### *Security Testing:*

Security Testing is carried out in order to find out how well the system can protect itself from unauthorized access, hacking – cracking, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing techniques.

#### *Mutation Testing:*

A kind of testing in which, the application is tested for the code that was modified after fixing a particular bug/defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality effectively.

### **Tools**

#### Static Analysis

Static analysis is the process of parsing

the source code to evaluate the code meets the coding standards. Static analyzers parse the source code and automatically detect errors which can be corrected by the developers before releasing to the QA. Static analysis does not find all issues. Some static tools provide data-flow and control-flow analysis support, which can be used during test case development.

### *Program Understanding Tools*

In general, white box testers should have access to the same tools, documentation, and environment as the developers and functional testers on the project do. In addition, tools that aid in program understanding, such as software visualization, code navigation, debugging, and disassembly tools, greatly enhance productivity during testing.

### *Coverage Analysis*

Code coverage tools measure how thoroughly tests exercise programs. There are many different coverage measures, including statement coverage, branch coverage, and multiple-condition coverage. The coverage tool would modify the code to record the statements executed and which expressions evaluate which way (the true case or the false case of the expression). Modification is done either to the source code or to the executable the compiler generates.

## **Profiling**

Profiling allows testers to learn where the

software under test is spending most of its time and the sequence of function calls as well. This information can show which pieces of the software are slower than expected and which functions are called more often than expected. From a security testing perspective, the knowledge of performance bottlenecks help uncover vulnerable areas that are not apparent during static analysis. The call graph produced by the profiling tool is helpful in program understanding. Certain profiling tools also detect memory leaks and memory access errors (potential sources of security violations). In general, the functional testing team or the development team should have access to the profiling tool, and the security testers should use the same tool to understand the dynamic behavior of the software under test.

## **Alternatives**

There are alternatives to white box testing:

Performing only the white box testing is tedious and not cost effective even though it increases the test effectiveness. It is good practice to perform White box testing for high-risk areas, and black box testing for the whole system.

More tests can be developed and executed by complementing both.

## **Skills and Training**

To perform white box testing it is essential for the tester to have the complete software knowledge including design and coding.

## About the Author:

**VINOD DANDOTI** is a consultant, software testing at Hisolve. Hisolve is a global consulting and technology service provider in testing services and in other areas. Vinod Dandoti has extensive experience in software testing like white box testing. He can be reached at [vinod.dandoti@hisolve.com](mailto:vinod.dandoti@hisolve.com).