

SOFTWARE TESTING - METHODS

http://www.tutorialspoint.com/software_testing/software_testing_methods.htm

Copyright © tutorialspoint.com

There are different methods that can be used for software testing. This chapter briefly describes the methods available.

Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

The following table lists the advantages and disadvantages of black-box testing.

Advantages	Disadvantages
<ul style="list-style-type: none">• Well suited and efficient for large code segments.• Code access is not required.• Clearly separates user's perspective from the developer's perspective through visibly defined roles.• Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems.	<ul style="list-style-type: none">• Limited coverage, since only a selected number of test scenarios is actually performed.• Inefficient testing, due to the fact that the tester only has limited knowledge about an application.• Blind coverage, since the tester cannot target specific code segments or error-prone areas.• The test cases are difficult to design.

White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

The following table lists the advantages and disadvantages of white-box testing.

Advantages	Disadvantages
<ul style="list-style-type: none">• As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.• It helps in optimizing the code.• Extra lines of code can be removed which can bring in hidden defects.• Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.	<ul style="list-style-type: none">• Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased.• Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested.• It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools.

Grey-Box Testing

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

Advantages	Disadvantages
<ul style="list-style-type: none">• Offers combined benefits of black-box and white-box testing wherever possible.• Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications.• Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling.• The test is done from the point of view of the user and not the designer.	<ul style="list-style-type: none">• Since the access to source code is not available, the ability to go over the code and test coverage is limited.• The tests can be redundant if the software designer has already run a test case.• Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

A Comparison of Testing Methods

The following table lists the points that differentiate black-box testing, grey-box testing, and white-box testing.

Black-Box Testing	Grey-Box Testing	White-Box Testing
The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledge of the internal workings of the application.
Also known as closed-box testing, data-driven testing, or functional testing.	Also known as translucent testing, as the tester has limited knowledge of the insides of the application.	Also known as clear-box testing, structural testing, or code-based testing.
Performed by end-users and also by testers and developers.	Performed by end-users and also by testers and developers.	Normally done by testers and developers.
Testing is based on external expectations - Internal behavior of the application is unknown.	Testing is done on the basis of high-level database diagrams and data flow diagrams.	Internal workings are fully known and the tester can design test data accordingly.
It is exhaustive and the least time-consuming.	Partly time-consuming and exhaustive.	The most exhaustive and time-consuming type of testing.
Not suited for algorithm	Not suited for algorithm testing.	Suited for algorithm

testing.

This can only be done by trial-and-error method.

Data domains and internal boundaries can be tested, if known.

testing.

Data domains and internal boundaries can be better tested.