Types of Testing

- White Box Testing
- **Black Box testing**
- **Graph based Testing**
- **Unit Testing**
- **Stress testing**
- **Usability Testing**
- **Compatibility Testing**
- **Alpha testing**
- **Beta Testing:**
- **Smoke testing**
- **Sanity Testing**
- **Regression testing**
- Graphics User Interface Testing
- **Object Oriented Testing**
- **Client server Testing**

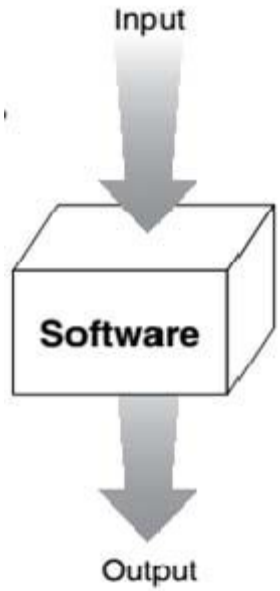
2.1 White Box Testing

1. This is also known as glass box, clear box, and open box testing.
2. In white box testing, test cases are created by looking at the code

to detect any potential failure scenarios.

3. The suitable input data for testing various APIs and the special code paths that need to be tested by analyzing the source code for the application block.
4. Therefore, the test plans need to be updated before starting white box testing and only after a stable build of the code is available.

5. White box testing assumes that the tester can take a look at the code for the application block and create test cases that look for any potential failure scenarios.
6. During white box testing, analyze the code of the application block and prepare test cases for testing the functionality to ensure that the class is behaving in accordance with the specifications and testing for robustness.
7. A failure of a white box test may result in a change that requires all black box testing to be repeated and white box testing paths to be reviewed and possibly changed.


Input
Software
Output

Classifications of white Box testing

Classifications of white Box testing					
1) Static Testing		2) Structural Testing			
1.1) Desk Checking		2.1.1) Unit/ Code Functional Testing		2.2.1) Code coverage	3.1) Code complexity
				2.2.2) Statement coverage	3.2) Cyclomatic complexity
1.2) Code Walkthrough				2.2.3) Path coverage	
				2.2.4)Condition coverage	
1.3) Code Inspection				2.2.5)Function coverage	

Static White box testing

1) This method involves principle of human reading the program to detect errors rather than computers.

2) Type of testing which requires only the source code of the project/software, not binaries or executable.

3) Does not involve executing the programs on computers but involves select people going through the code **to find out whether:**

- The code works according functional requirement.

- The code has been written in accordance with the design developed earlier in the project life cycle.

- The code for any functionality has been missed out.

- The code handles errors properly.

4) This testing can be done by human or with the help of specialized tools

Advantages of static testing

a) Sometimes humans can find errors computers cannot.

b) Possibility of multiple perspectives by making multiple human read and evaluations of program.

c) Compares with specifications and standards, (in reactive testing reveals identifying symptoms rather than root cause) rather its **Proactive Testing**, this is done faster by human being.

d) One can find root cause for the problem.

e) Saves computer resources, as testing the code can be done before execution.

- Desk checking of code
- Code walkthrough
- Code review and checklist
- Code injection

Disadvantages of Static testing

1. It is time-consuming.
2. The logistics and scheduling can become an issue since multiple peoples are involved.
3. It is not always possible to go through every line of code.
4. Required High- skills.

contents of Code review Checklist

- ✔ Data item declaration related
- ✔ Data usage related
- ✔ Control flow related
- ✔ Standard related
- ✔ Style related
- ✔ Miscellaneous
- ✔ Documentation related

Inspection under static testing

Inspections are the most formal type of reviews in white box testing. They are highly structured and require training for each participant.

Inspections are different from peer reviews and walkthroughs in that the person who presents the code, the presenter or reader, isn't the original programmer. These forces someone else to learn and understand the material being presented, potentially giving a different slant and interpretation at the inspection meeting. The other participants are called inspectors. Each is tasked with reviewing the code from a different perspective, such as a user, a tester, or a product support person. This helps bring different views of the product under review and very often identifies different bugs. One inspector is even tasked with reviewing the code backward—that is, from the end to the beginning—to make sure that the material is covered evenly and completely.

In this:

1. Thorough preparation is required before an inspection/review
2. Enlisting multiple diverse views.
3. Assigning specific roles to the multiple participants
4. Going sequentially through the code in a structured manner.

There are four roles in this.

1.Author of the code: the person who had written the code

2.Moderator: who is expected to formally run the inspection according to the process.

3.Inspectors: are the people who actually provide review comments for the code.

4.Scribe: who takes detail notes during the inspection meeting and circulates them to the
inspection team after the meeting.

The author or moderator selects review team. The inspection team assembles at the agreed
time for inspection meeting. The moderator takes the team sequentially through the program code. If any defect is found they will classify it as minor or major. A scribe documents the defects. For major defects the review team meets again to check whether
the bugs are resolved or not.

structural walk through under static testing

One of the static testing methods is structural walkthrough.

In walkthroughs, a set of people look at the program code and raise questions for the author. The author explains the logic of the code and answers the questions. If the author is unable to answer some questions, he or she then takes those questions and finds their answers.

(i) Walkthroughs are the next step up in formality from peer reviews.

(ii) In a walkthrough, the programmer who wrote the code formally presents (walks through) it to a small group of five or so other programmers and testers.

(iii) The reviewers should receive copies of the software in advance of the review so they can examine it and write comments and questions that they want to ask at the review.

(iv) Having at least one senior programmer as a reviewer is very important.

Compare Walkthroughs and inspections in terms of Participants, type of meeting, Training required, Purpose, and Effectiveness

Particulars	Walkthroughs	Inspections
Participants	Author is leader and others (Presenter, Moderator, Reviewers, Scribe, are participants,	Every participant has some decided role.
Type of meeting	Informal	Formal
Training required	No	Yes
Purpose	To find bugs	To improve quality of product
Effectiveness	Low	High