

# Software Testing - Methods

- 1. Black-Box Testing**
- 2. White- Box Testing**
- 3. Grey-Box Testing**

# 1. Black-Box Testing



- Closed-box testing, Data-driven testing.
- Interior workings of the application is not known.
- The tester is unaware to the system architecture and does not have access to the source code.
- 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.

## **Advantages**

- Well suited and efficient for large code segments.
- Code access is not required.

## **Disadvantages**

- Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
- The test cases are difficult to design.

## 2. White-Box Testing



- **Glass testing or open-box testing.**
- White-box testing is the detailed investigation of internal logic and structure of the code.
- Tester knows the internal working 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.

## **Advantages**

- 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.

## **Disadvantages**

- 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 corner to find out hidden errors that may create problems, as many paths will go untested.



### 3. 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.
- In grey-box testing, the tester has access to design documents and the database.
- Testing is done on the basis of high-level database diagrams and data flow diagrams.

## Advantages

- Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications.
- The test is done from the point of view of the user and not the designer.

## Disadvantages

- Since the access to source code is not available, the ability to go over the code and test coverage is limited.
- Testing every possible input stream is unrealistic