

Welcome to the Class



Department of Computing and Information System



Testing Procedure

Md. Selim Hossain

Senior Lecturer

Department of Computing and Information System

Daffodil International University (DIU), Dhaka, Bangladesh

Outline



- >GreyBox Testing
- >Why GreyBox Testing
- >Procedure of GreyBox Testing
- >Advantages
- >Disadvantages

Greybox Testing



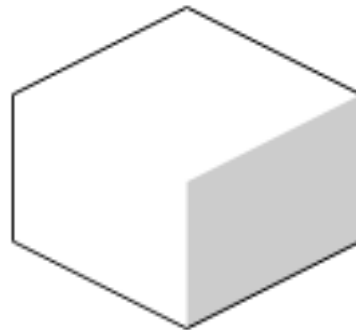
Greybox testing is a software testing method to test the software application with partial knowledge of the internal working structure. It is a **combination of black box and white box testing** because it involves access to internal coding to design test cases as white box testing and testing practices are done at functionality level as black box testing.

Greybox Testing....



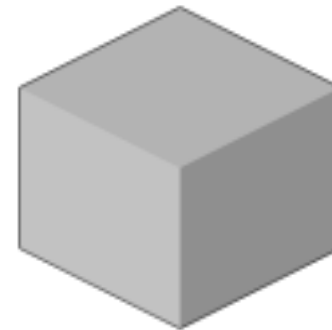
*BlackBox
Testing*

+



*WhiteBox
Testing*

=



*GreyBox
Testing*

Greybox Testing....



GreyBox testing commonly identifies context-specific errors that belong to web systems. For example; while testing, if tester encounters any defect then he makes changes in code to resolve the defect and then test it again in real time. It concentrates on all the layers of any complex software system to increase testing coverage. It gives the ability to test both presentation layer as well as internal coding structure. It is primarily used in integration testing and penetration testing.

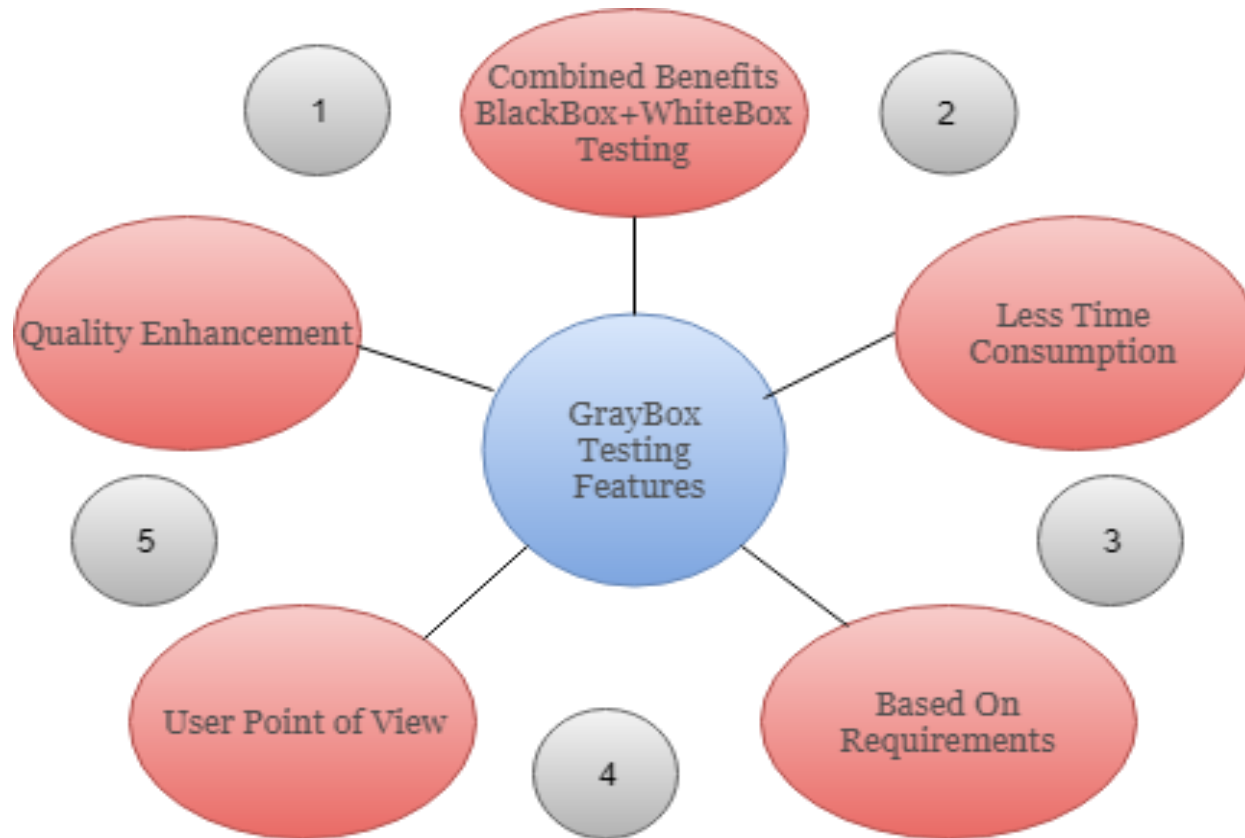


Why GreyBox testing?

Reasons for GreyBox testing are as follows

- It provides **combined benefits of both Blackbox testing and WhiteBox testing.**
- It includes the input values of **both developers and testers** at the same time to improve the overall quality of the product.
- It **reduces time consumption** of long process of functional and nonfunctional testing.
- It **gives sufficient time to the developer to fix the product defects.**
- ◦ It includes **user point of view** rather than designer or tester point of view.

Why GreyBox testing?



GreyBox Testing Strategy



Grey box testing does not make necessary that the tester must design test cases from source code. To perform this testing test cases can be designed on the base of, knowledge of architectures, algorithm, internal states or other high -level descriptions of the program behavior. It uses all the straightforward techniques of black box testing for function testing. The test case generation is based on requirements and preset all the conditions before testing the program by assertion method.

Summary



The **spiral model** is a systems development lifecycle (SDLC) method **used** for risk management that combines the iterative development process **model** with elements of the Waterfall **model**. The **spiral model** is **used** by software engineers and is favored for Mid-large, expensive and complicated projects.

Generic Steps to perform Grey box Testing are:



1. First, select and **identify inputs** from **BlackBox and WhiteBox** testing inputs.
2. Second, Identify **expected outputs** from these selected inputs.
3. Third, **identify all the major paths** to traverse through during the testing period.
4. The fourth task is **to identify sub-functions** which are the part of main functions to perform deep level testing.
5. The fifth task is to **identify inputs for subfunctions**.
6. The sixth task is to identify **expected outputs for subfunctions**.
7. The seventh task includes **executing a test case for Subfunctions**.
8. The eighth task includes **verification of the correctness of result**.



Advantages of Grey Box Testing

Grey box testing gives the following advantages:

- As grey box testing is a combination of black box and white box testing, it provides the best of both worlds i.e. benefits of both the testing techniques.
- Knowledge of the internal mechanisms of the system helps the tester to design test scenarios more extensively.
- For grey box testing, functional specifications and other design documents are used. It does not need the use of the source code which helps in keeping the source code safe from any disruptive changes.

Advantages of Grey Box Testing.....



- It helps in keeping testers and developers separate, which reduces any disagreement between them.
- Even with a partial understanding of the code, testers conduct grey box testing from the end user's perspective. This helps in identifying any issues that the developers might have missed during unit testing.
- It results in the instant fixing of the issues as a tester can change the partially available code to check for the results.
- Even without high-level programming skills, the testers can perform this testing.
- It is platform and language-independent.



Disadvantages of Grey Box Testing

Following are some disadvantages of grey-box testing:

- While doing grey box testing, testers do not have access to the source code, so it becomes difficult to get complete code path coverage and testers might fail to notice some critical vulnerabilities.
- Algorithm testing is not possible as accessing the complete logic of the algorithms is not possible.
- If a developer has already executed a test case, running the same test case in grey box testing may result in redundancy.
- It is usually not suitable for distributed systems.



Thanks to All