|  |
| --- |
| Department of Software Engineering  Mehran University of Engineering and Technology, Jamshoro |

|  |  |  |  |
| --- | --- | --- | --- |
| Course: SW426 - Software Quality Engineering | | | |
| Instructor | Rabia Iftikhar | **Practical/Lab No.** | 03 |
| Date | 12-08-2020 | **CLOs** | CLO-3 |
| Signature |  | **Assessment Score** | 1 Mark |

|  |  |
| --- | --- |
| Topic | Application of Black box testing |
| Objectives | * To learn black box testing techniques * To apply black box testing techniques |

|  |
| --- |
| Lab Discussion: Theoretical concepts and Procedural steps |

**Black Box Testing**

Black Box Testing is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing.

It is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.

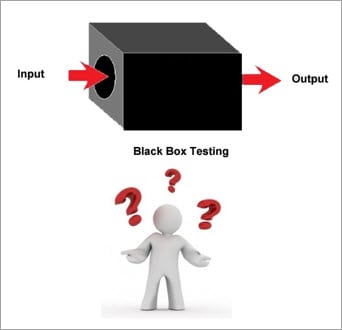
**The main focus in Black Box Testing is on the functionality of the system as a whole.** The term **‘Behavioral Testing'** is also used for Black Box Testing. Behavioral test design is slightly different from the black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged.

Each testing method has its own advantages and disadvantages. There are some bugs that cannot be found using the only black box or only white box technique.

Majority of the applications are tested by Black Box method. We need to cover the majority of test cases so that most of the bugs will get discovered by a [Black-Box](https://en.wikipedia.org/wiki/Black_box) method.

This testing occurs throughout the software development and Testing Life Cycle i.e in Unit, Integration, System, Acceptance, and Regression Testing stages.

This can be both functional or non-Functional.



**Functional Testing**

This type deals with the functional requirements or specifications of an application. Here, different actions or functions of the system are being tested by providing the input and comparing the actual output with the expected output.

**For Example** when we test a Dropdown list, we click on it and verify that it expands and all the expected values are showing in the list.

Few major types of Functional Testing are:

* Smoke Testing
* Sanity Testing
* Integration Testing
* System Testing
* Regression Testing
* User Acceptance Testing

**Non-Functional Testing**

Apart from the functionalities of the requirements, there are several non-functional aspects as well that are required to be tested to improve the quality and performance of the application.

Few major types of Non-Functional Testing include:

* Usability Testing
* Load Testing
* Performance Testing
* Compatibility Testing
* Stress Testing
* Scalability Testing

**Black Box Testing Tools**

Black Box Testing tools are mainly record and playback tools. These tools are used for Regression Testing to check whether new build has created any bug in previous working application functionality. Some black box testing tools are listed below:

1. Selenium
2. Appium
3. Applitools
4. HP QTP
5. Microsoft Coded UI

In order to systematically test a set of functions, it is necessary to design test cases. Testers can create test cases from the requirement specification document using the following Black Box Testing techniques.

* Equivalence Partitioning
* Boundary Value Analysis
* Decision Table Testing
* State Transition Testing
* Error Guessing
* Graph-Based Testing Methods
* Comparison Testing

#### Equivalence Partitioning

This technique is also known as Equivalence Class Partitioning (ECP). In this technique, input values to the system or application are divided into different classes or groups based on its similarity in the outcome.

Hence, instead of using each and every input value we can now use any one value from the group/class to test the outcome. In this way, we can maintain the test coverage while we can reduce a lot of rework and most importantly the time spent.

**For Example**

A “Passing Marks” text field accepts only the numbers 50 and above. There will be two sets of classes or groups **out of which one will be invalid class. These classes are:**

Less than or equal to 49.

One valid class will be the numbers 50 or above.

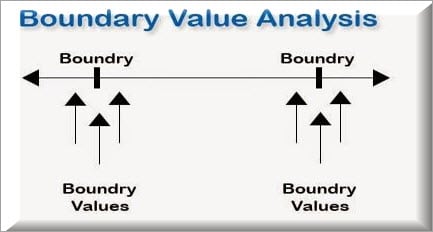
We have thus reduced the test cases to only 2 test cases based on the formed classes thereby covering all the possibilities. So, testing with anyone value from each set of the class is sufficient to test the above scenario.

#### Boundary Value Analysis

From the name itself, we can understand that in this technique we focus on the values at boundaries as it is found that many applications have a high amount of issues on the boundaries.

Boundary means the values near the limit where the behavior of the system changes. In boundary value analysis both the valid inputs and invalid inputs are being tested to verify the issues.

**For Example**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/03/Boundary-Value-Analysis.jpg)

If we want to test a field where values from 1 to 100 should be accepted then we choose the boundary values: 1-1, 1, 1+1, 100-1, 100, and 100+1. Instead of using all the values from 1 to 100, we just use 0, 1, 2, 99, 100, and 101.

#### Decision Table Testing

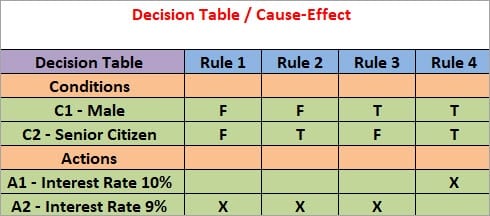
As the name suggests that, wherever there are logical relationships like:

If  
{  
(Condition = True)  
then action1 ;  
}  
else action2; /\*(condition = False)\*/

Then a tester will identify two outputs (action1 and action2) for two conditions (True and False). So based on the probable scenarios a Decision table is carved to prepare a set of test cases.

**For Example**

Take an example of XYZ bank that provides interest rate for the Male senior citizen as 10% and for the rest of the people 9%.

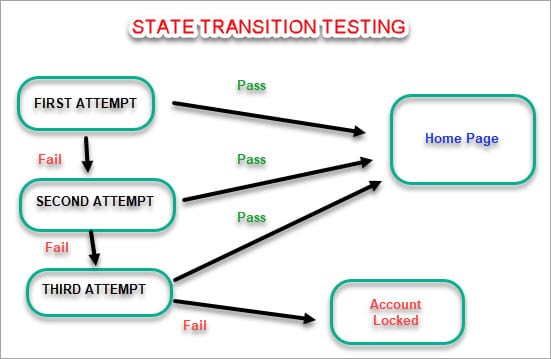
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/03/Decision-Table.jpg)

#### State Transition Testing

State Transition Testing is a technique that is used to test the different states of the system under test. The state of the system changes depending upon the conditions or events. The events trigger states which become scenarios and a tester needs to test them.

A systematic state transition diagram gives a clear view of the state changes but it is effective for simpler applications. More complex projects may lead to more complex transition diagrams thus making it less effective.

**For Example**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/03/State-Transition-Testing.jpg)

#### Error Guessing

This is a classic example of Experience-Based Testing.

In this technique, the tester can use his/her experience about the application behavior and functionalities to guess the error-prone areas. Many defects can be found using error guessing where most of the developers usually make mistakes.

**Few common mistakes that developers usually forget to handle:**

* Divide by zero.
* Handling null values in text fields.
* Accepting the Submit button without any value.
* File upload without attachment.
* File upload with less than or more than the limit size.

#### Comparison Testing

Different independent versions of the same software are used to compare to each other for testing in this method.

|  |
| --- |
| Lab Tasks |

1. An “AGE” text field accepts only the numbers from 18 to 60. How many input classes will be there? Mention the valid and invalid classes. Validate the classes by writing code.
2. Considering the case in task no: 1, test the text field using boundary value analysis.
3. A mobile banking application allows 3 OTP attempts for making transactions. Once it is entered correctly, the transaction succeeds. In the other case the system dismisses the requested transaction and block the mobile account for making the other transaction, for the next 1 hour. Design a state transition diagram for this case.
4. Test the functionality of The National Database and Registration Authority for checking the eligibility of citizens for issuing CNIC and CNICOP using decision table method. The conditions are listed below:

* CNIC: Applicant has Pakistan Nationality and is of age 18 or above.
* CNICOP: An eligible citizen of Pakistan who lives or has reference abroad (i.e. dual nationality)

|  |  |
| --- | --- |
| Lab Tasks Assessment/Rubrics along with Score/Marks | |
| *Rubric Description* | ***Rubric Marks*** |
| 1. Test completeness | 0.25 |
| 1. Test coverage | 0.25 |
| 1. Test correctness | 0.5 |