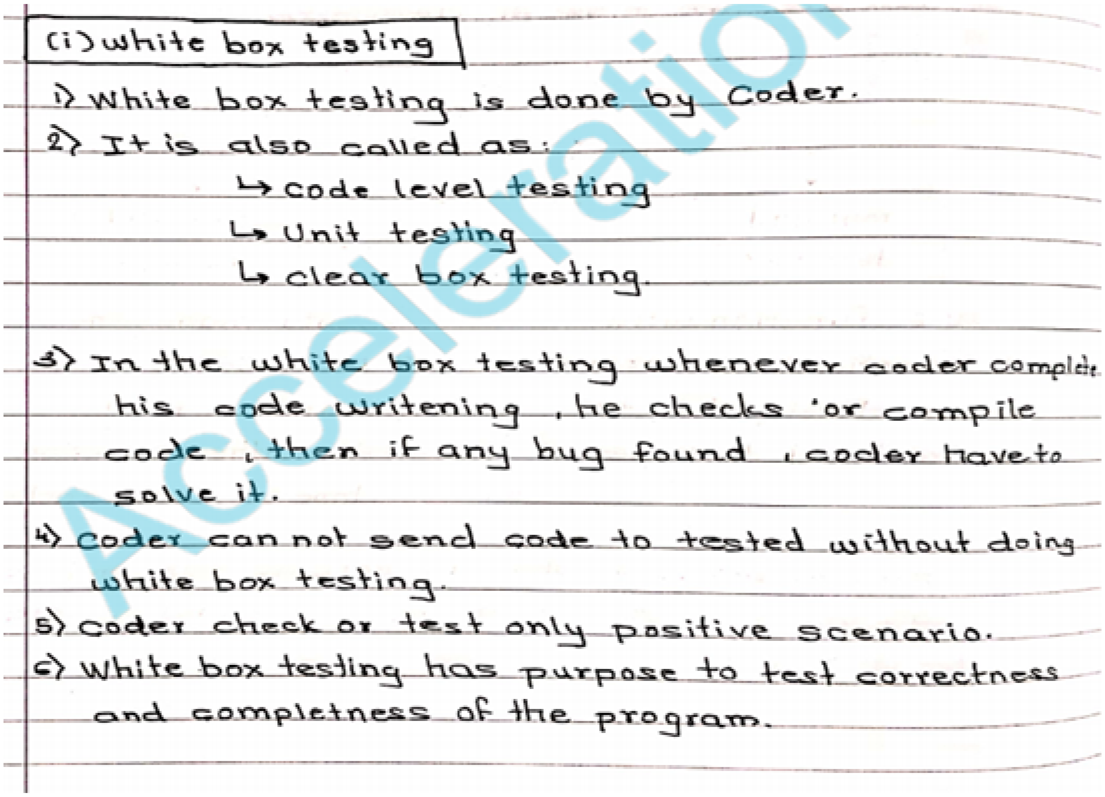
White Box Testing (WBT)



(i) white box testing

1) White box testing is done by Coder.

2) It is also called as:

↳ code level testing

↳ Unit testing

↳ clear box testing.

3) In the white box testing whenever coder complete his code writening, he checks 'or compile code, then if any bug found, coder have to solve it.

4) Coder can not send code to tested without doing white box testing.

5) coder check or test only positive scenario.

6) White box testing has purpose to test correctness and completness of the program.

WHITE BOX TESTING (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested

the entire WBT is done by developers. It is the testing of each and every line of code in the program. Developers do WBT, send the s/w to the testing team. The testing team does black box testing and checks the s/w against requirements and finds any defects and sends it to the developer. The developers fix the defect and WBT and send it to the testing team. Fixing defect means the defect is removed and the feature is working fine

Test engineers should not be involved in fixing the bug because,

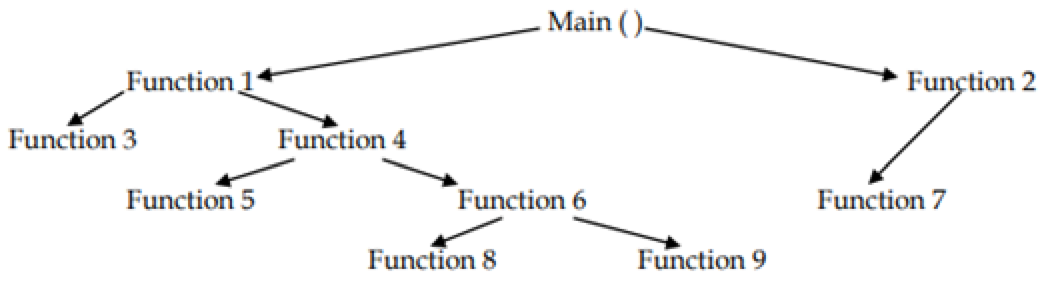
1) if they spend time in fixing the bug, they lose time to catch some more other defects in the s/w

2) fixing a defect might break a lot of other features. Thus, testers should always identify defects and developers should always be involved in fixing defects

WBT consists of the following tests :

Path testing

Write flow graphs and test all the independent paths. Writing flow graphs means – flow graphs means representing the flow of the program, how each program is interlinked with one another.



Test all independent paths – Consider a path from main( ) to function 7. Set the parameters and test if the program is correctly in that path. Similarly test all other paths and fix defects.

Condition testing

Test all the logical conditions for both true and false values i.e, we check for both “if” and “else” conditions.

If(condition) - true

{

……

}

Else - false

{

….

}

The program should work correctly for both conditions i.e, if condition is true, then else should be false and vice-versa

Loop testing

Test the loops(for, while, do-while, etc) for all the cycles and also check for terminating conditions if working properly and if the size of the condition is sufficient enough.

Developers do a lot of WBT automatically rather than manually because it saves time

Unit Testing

Customer will give the requirement and the developer will write the main program and corresponding test program in the same language.

Test program will invoke the main program and test the main program and give the result pass or fail

The objective of unit testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure

Building unit Test Cases

Unit testing is commonly automated, but may still be performed manually. The IEEE does not favour one over the other. A manual approach to unit testing may employ a step-by-step instructional document.

Under the automated approach-

- A developer could write another section of code in the application just to test the function. They would later comment out and finally remove the test code when the application is done.
- They could also isolate the function to test it more rigorously. This is a more thorough unit testing practice that involves copy and pasting the function to its own testing environment other than its natural environment. Isolating the code helps in revealing unnecessary dependencies between the code being tested and other units or data spaces in the product. These dependencies can then be eliminated.

A coder may use a Unit Test Framework to develop automated test cases. Using an automation framework, the developer codes criteria into the test to verify the correctness of the unit. During execution of the test cases, the framework logs those that fail any criterion. Many frameworks will also automatically flag and report in a summary these failed test cases. Depending upon the severity of a failure, the framework may halt subsequent testing.

Testing from memory point of view

BY changing the logic of the program, we can reduce the memory

By converting the duplicate code into the generic function

By removing the dead code, unused commands and unwanted functions from the program

By using built in functions

By removing the unused variables from the program

Difference between White Box Testing and Black Box testing

White Box Testing

White Box Testing Done by developers

Look into the source code and test the logic of the code

Should have knowledge of internal design of the code

Should have knowledge of programming

Black Box Testing

Black Box Testing Done by test engineers

Verifying the functionality of the application against requirement specifications

No need to have knowledge of internal design of the code

No need to have knowledge of programming

Q.21> Why it is known as white box testing.

Ans:- 1> white box testing is transperent testing.

2> White box testing is aka clear box testing & Glass box testing.

3> developer do the white box testing.

4> when coding get over for the project. coder review the code.

5> In the review process coder compile the code

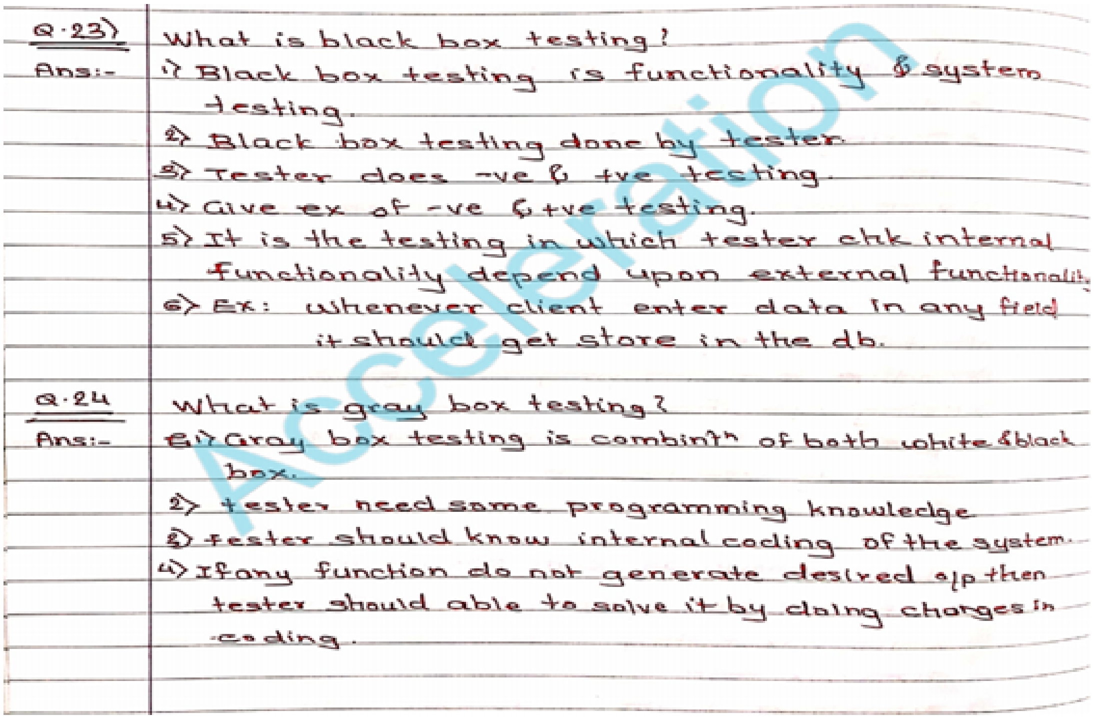
6> if He found some bug, error then coder resolve that problem.

7> It is also called as unit testing.

8> Also known as code level testing.

9> It is the process of testing correctness & complete of the program.

Black Box testing



Q.23) What is black box testing?

Ans:-

i> Black box testing is functionality & system testing.

2> Black box testing done by tester.

3> Tester does -ve & +ve testing.

4> Give ex of -ve & +ve testing.

5> It is the testing in which tester chk internal functionality depend upon external functionality.

6> Ex: whenever client enter data in any field it should get store in the db.

Q.24 What is gray box testing?

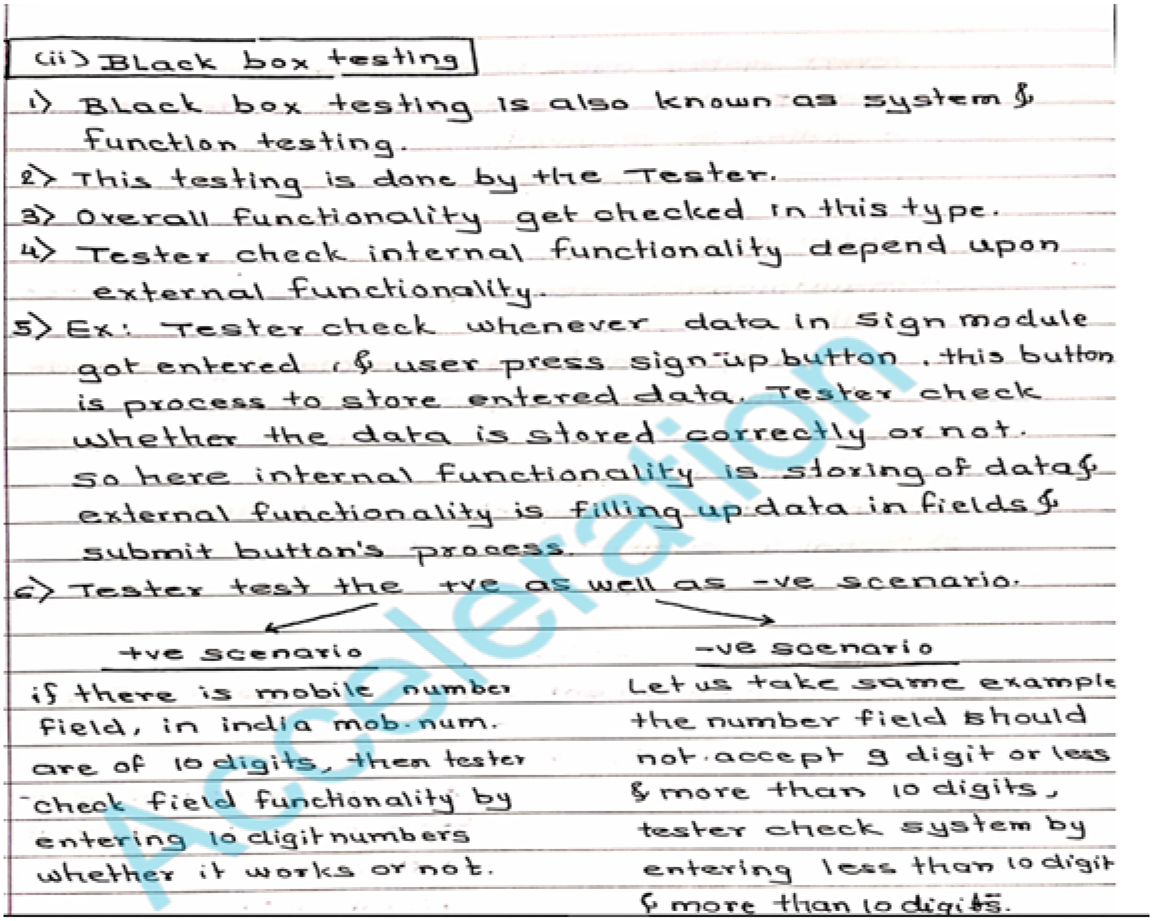
Ans:-

1> Gray box testing is combintn of both white & black box.

2> tester need some programming knowledge

3> tester should know internal coding of the system.

4> If any function do not generate desired o/p then tester should able to solve it by doing changes in coding.



(ii) Black box testing

1> Black box testing is also known as system & function testing.

2> This testing is done by the Tester.

3> Overall functionality get checked in this type.

4> Tester check internal functionality depend upon external functionality.

5> Ex: Tester check whenever data in sign module got entered, & user press sign-up button. this button is process to store entered data. Tester check whether the data is stored correctly or not. So here internal functionality is storing of data & external functionality is filling up data in fields & submit button's process.

6> Tester test the +ve as well as -ve scenario.

+ve scenario

if there is mobile number field, in india mob. num. are of 10 digits, then tester check field functionality by entering 10 digit numbers whether it works or not.

-ve scenario

Let us take same example the number field should not accept 9 digit or less & more than 10 digits, tester check system by entering less than 10 digit & more than 10 digits.

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.

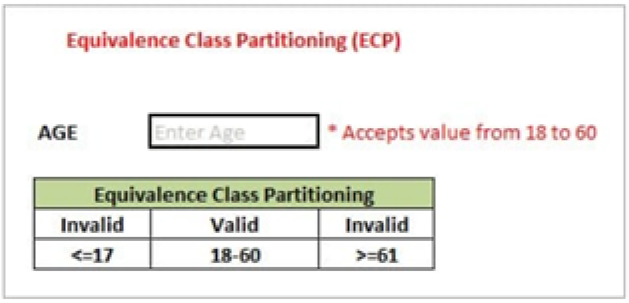
The main focus in Black Box Testing is on the functionality of the system as a whole.

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 test case designing technique checks the input and output by dividing the input into equivalent classes. The data must be tested at least once to ensure maximum test coverage of data. It is the exhaustive form of testing, which also reduces the redundancy of inputs.



Equivalence Class Partitioning (ECP)

AGE Enter Age * Accepts value from 18 to 60

Equivalence Class Partitioning		
Invalid	Valid	Invalid
<=17	18-60	>=61

Two invalid classes will be:

a) Less than or equal to 17.

b) Greater than or equal to 61.

One valid class will be anything between 18 to 60.

Boundary Value Analysis

It is the widely used black-box testing, which is also the basis for equivalence testing. Boundary value analysis tests the software with test cases with extreme values of test data. BVA is used to identify the flaws or errors that arise due to the limits of input data.

For example: Taking inputs for a test case data for an age section should accept a valid data of anything between 1-100. According to BVP analysis, the software will be tested against four test data as -1, 1, 100, and 101 to check the system's response using the boundary values.

Decision Table

A decision table is a tabular representation of inputs vs cases, rules and test conditions.

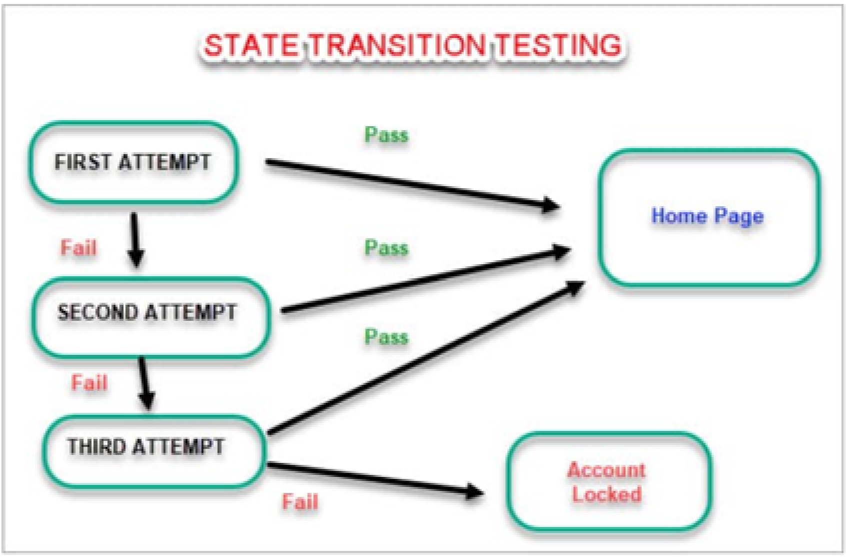
The condition here is that the user will be redirected to the homepage if he enters the correct user name and password, and an error message will be displayed if the input is wrong.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	T	T
Output (E/H)	E	E	E	H

- T- Correct username or password
- F- Wrong username or password
- E- Error message is displayed.
- H – Home screen is displayed.

State Transition Testing

This testing technique uses the inputs, outputs, and the state of the system during the testing phase. It checks the software against the sequence of transitions or events among the test data.



Error Guessing:

This is the type of Experience-Based Testing. This method of designing test cases is about guessing the output and input to fix any errors that might be present in the system. It depends on the skills and judgment of the tester.

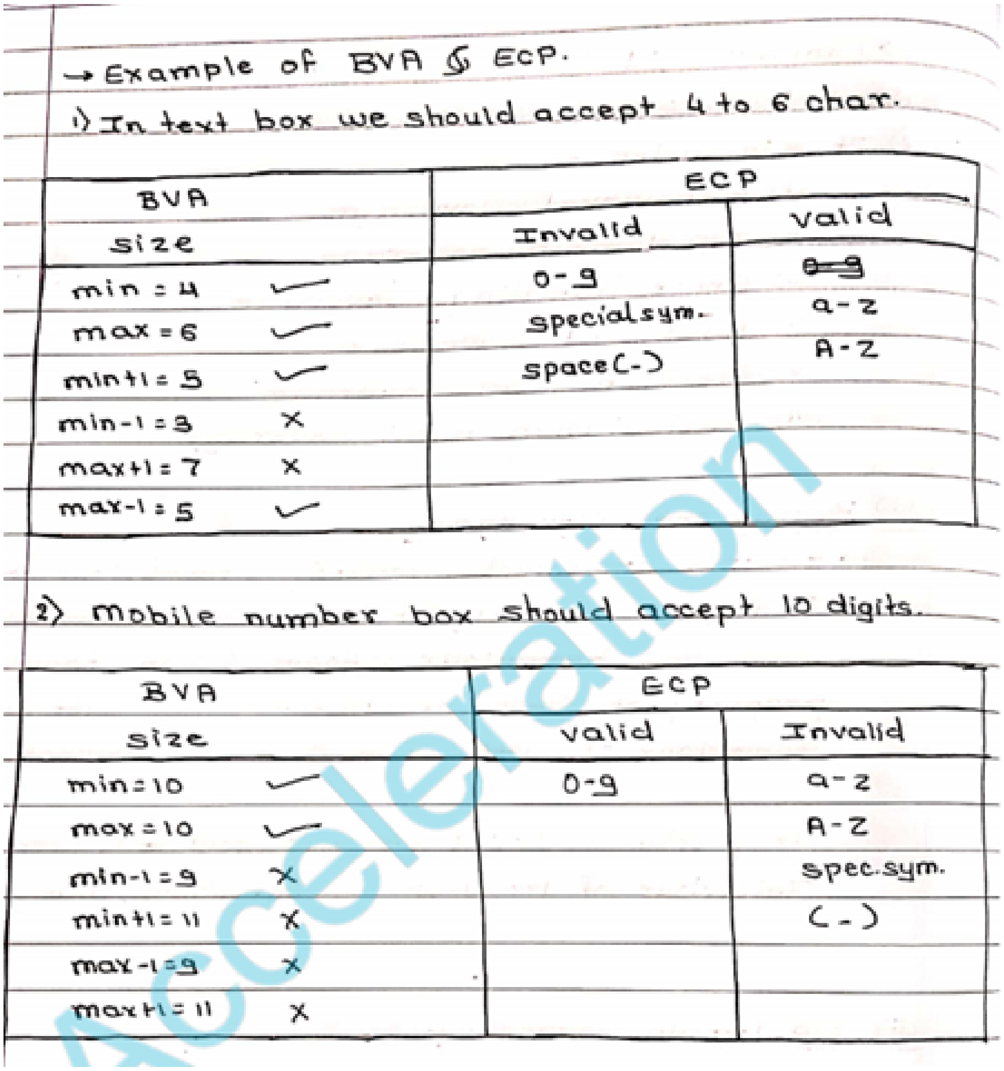
Graph-Based Testing:

It is similar to a decision-based test case design approach, where the relationship between links and input cases are considered.

Comparison testing

This method uses the two different versions of the same software to compare and validate the results.

More examples of BVA, ECP



→ Example of BVA & ECP.

1) In text box we should accept 4 to 6 char.

BVA		ECP	
size		Invalid	valid
min = 4	✓	0-9	~~0-9~~
max = 6	✓	specialsym.	a-z
min+1 = 5	✓	space(-)	A-Z
min-1 = 3	✗		
max+1 = 7	✗		
max-1 = 5	✓		

2) mobile number box should accept 10 digits.

BVA		ECP	
size		valid	Invalid
min=10	✓	0-9	a-z
max=10	✓		A-Z
min-1=9	✗		spec.sym.
min+1=11	✗		(-)
max-1=9	✗		
max+1=11	✗		

3) Password should accept betⁿ 8 to 14 digit, allow 1 cap char, 1 small char, spc.sym, & digit, No space is allowed.

BVA	ECP	
size	valid	Invalid
min = 8 ✓	0 - 9	(-)
max = 14 ✓	a - z	
min+1 = 9 ✓	A - Z	
min-1 = 7 ✗ max+1 = 15 ✗ m -1 = 13 ✗	spc.sym	

Scanned by CamScanner

Lecture - 13
10 - 7 - 19

4) check BVA & ecp for cycle stand having 100 cycles.

BVA	ECP	
size	valid	Invalid
min = 1 ✓	0 - 9	a - z
max = 100 ✓		A - Z
min-1 = 0 ✗		spcl. sym
min+1 = 2 ✓		(-)
max+1 = 101 ✗		
max-1 = 99 ✓		

5) check for 3 digit number which should not start with 0 & 1.

BVA	ECP	
size	valid	Invalid
min = 200 ✓	0 - 9	a - z
max = 999 ✓		A - Z
min-1 = 199 ✗		spcl.sym
min+1 = 201 ✓		(-)
max -1 = 998 ✓		
max+1 = 1000 ✗		

(iii) Error handling Coverage.

1) Error handling Coverage include checking whether system show error message or not.

2) If in customer requirement, mobile number should accept the 10 digits & developer developed build according to it then if customer enter 8 digits & click on ok button then system should highlight text box with red color with error message " Please Enter 10 digit mobile number ".

3) So this is the process of validation whether system show error message or not.

How to do Black Box testing?

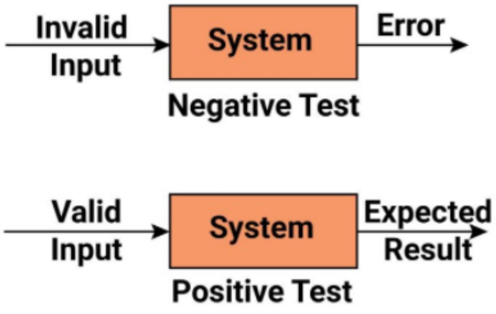
- The first step to black-box testing is to understand the requirement specifications of the application under test. An accurate and SRS Document should be there.
- The next step is to evaluate the set of valid inputs and test scenarios to test the software. The goal is to save time and get good test coverage.
- Prepare the test cases to cover a maximum range of inputs.
- The test cases are run in the system to generate output, which is validated with the expected outcome to mark pass or fail.
- The failed steps are marked and sent to the development team to fix them.
- Retest the system using various testing techniques to verify its recurring nature or to pass it.

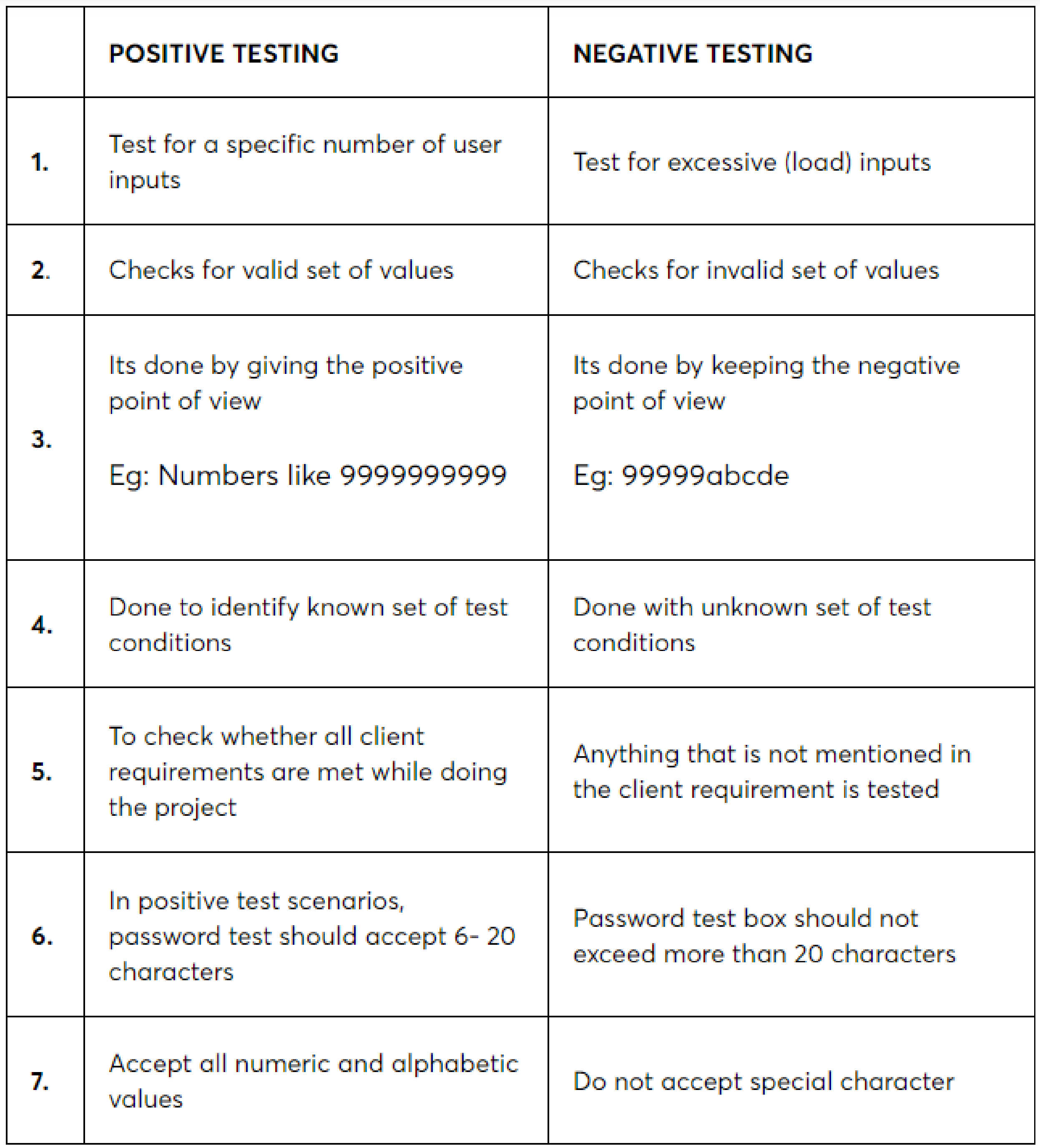
What are the benefits of Black Box testing?

- The tester doesn't need any technical knowledge to test the system. It is essential to understand the user's perspective.
- Testing is performed after development, and both the activities are independent of each other.
- Test cases can be generated before development and right after specification.
- Black box testing methodology is close to agile.

What is the difference between positive and negative testing?

Positive testing refers to all the happy paths where the application performs as expected with the correct input. Negative tests, on the other hand, involves giving incorrect data to the system to ensure it responds appropriately.





	POSITIVE TESTING	**NEGATIVE TESTING**
1.	Test for a specific number of user inputs	Test for excessive (load) inputs
2.	Checks for valid set of values	Checks for invalid set of values
3.	Its done by giving the positive point of view Eg: Numbers like 9999999999	Its done by keeping the negative point of view Eg: 99999abcde
4.	Done to identify known set of test conditions	Done with unknown set of test conditions
5.	To check whether all client requirements are met while doing the project	Anything that is not mentioned in the client requirement is tested
6.	In positive test scenarios, password test should accept 6- 20 characters	Password test box should not exceed more than 20 characters
7.	Accept all numeric and alphabetic values	Do not accept special character