Positive and Negative Testing

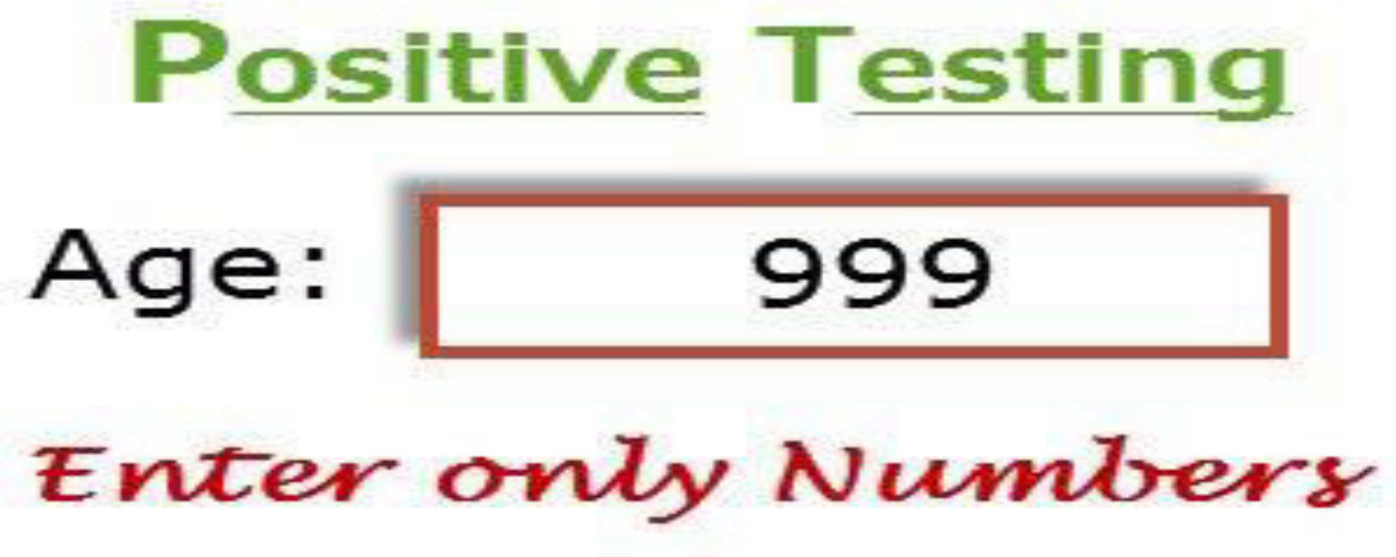
DR. D. P. MOHAPATRA
ASSOCIATE PROFESSOR
DEPARTMENT OF CSE
NIT, ROURKELA

Positive Testing

- Positive testing tries to prove that a given product does what it is supposed to do.
- Positive Testing is testing process where the system is validated against the valid input data.
- In this testing tester always check for only valid set of values and check if an application behaves as expected with its expected inputs.
- The main intention of this testing is to check whether software application not showing error when not supposed to and showing error when supposed to.
- Such testing is to be carried out keeping positive point of view and only execute the positive scenario.
- When a test case verifies the requirements of the product with a set of expected output, it is called *positive test case*.
- The purpose of positive testing is to prove that the product works as per specification and expectations.
- A product delivering an error when it is expected to give an error, is also a part of positive testing.
- Positive testing can thus be said to check the product’s behaviour for positive and negative conditions as stated in the requirement.

Example of Positive testing

- Consider a scenario where you want to test an application which contains a simple textbox to enter age and requirements say that it should take only integer values.
- So here provide only positive integer values to check whether it is working as expected or not is the Positive Testing.
- Most of the applications developers implement Positive scenarios where testers get less defects count around positive testing.



Positive Testing

Age: 999

Enter only Numbers

An example

- Let us take a lock and key.
- We do not know how the levers in the lock work, but we only know the set of inputs (the number of keys, specific sequence of using the keys and the direction of turn of each key) and the expected outcome (locking and unlocking).
- For example, if a key is turned clockwise it should unlock and if turned anticlockwise it should lock.
- To use the lock one need not understand how they work.
- However, it is essential to know the external functionality of the lock and key system.
- Some of the functionality that you need to know to use the lock are given as follows.

Some functionalities to use the lock

Functionality	**What you need to know to use**
Features of a lock	It is made of metal, has a hole provision to lock, has a facility to insert the key and the keyhole ability to turn clockwise or anticlockwise.
Features of key	It is made of metal and created to fit into a particular lock's keyhole.
Actions performed	Key inserted and turned clockwise to unlock. Key inserted and turned anticlockwise to lock.
States Inputs Expected outcome	Locked. Unlocked. Key turned clockwise or anticlockwise. Locking. Unlocking.

Sample requirements specification for lock and key system

Sl. No.	Requirements identifier	Description	Priority (High, med, low)
1	BR-01	Inserting the key numbered 123-456 and turning it clockwise should facilitate locking	H
2	BR-02	Inserting the key numbered 123-456 and turning it anticlockwise should facilitate unlocking	H
3	BR-03	Only key number 123-456 can be used to lock and unlock	H
4	BR-04	No other object can be used to lock	M
5	BR-05	No other object can be used to unlock	M
6	BR-06	The lock must not open even when it is hit with a heavy object	M
7	BR-07	The lock and key must be made of metal and must weigh approximately 150 grams	L
8	BR-08	Lock and unlock directions should be changeable for usability of left-handers	L

Example of positive test cases

Req. No.	**Input 1**	**Input 2**	**Current state**	**Expected output**
BR-01	Key 123-456	Turn clockwise	Unlocked	Locked
BR-01	Key 123-456	Turn clockwise	Locked	No change
BR-02	Key 123-456	Turn anticlockwise	Unlocked	No change
BR-02	Key 123-456	Turn anticlockwise	Locked	Unlock
BR-04	Hairpin	Turn clockwise	Locked	No change

Example of positive test condition for positive testing

- Take the first row in the table.
- When the lock is in an unlocked state and you use key 123-456 and turn it clockwise, the expected outcome is to get it locked.
- During test execution, if the test results in locking, then the test is passed.

Example of negative test condition for positive testing

- In the fifth row of the table, the lock is in locked state.
- Using a hairpin and turning it clockwise should not cause a change in state or cause any damage to the lock.
- On test execution, if there are no changes, then this positive test case is passed.

Negative Testing

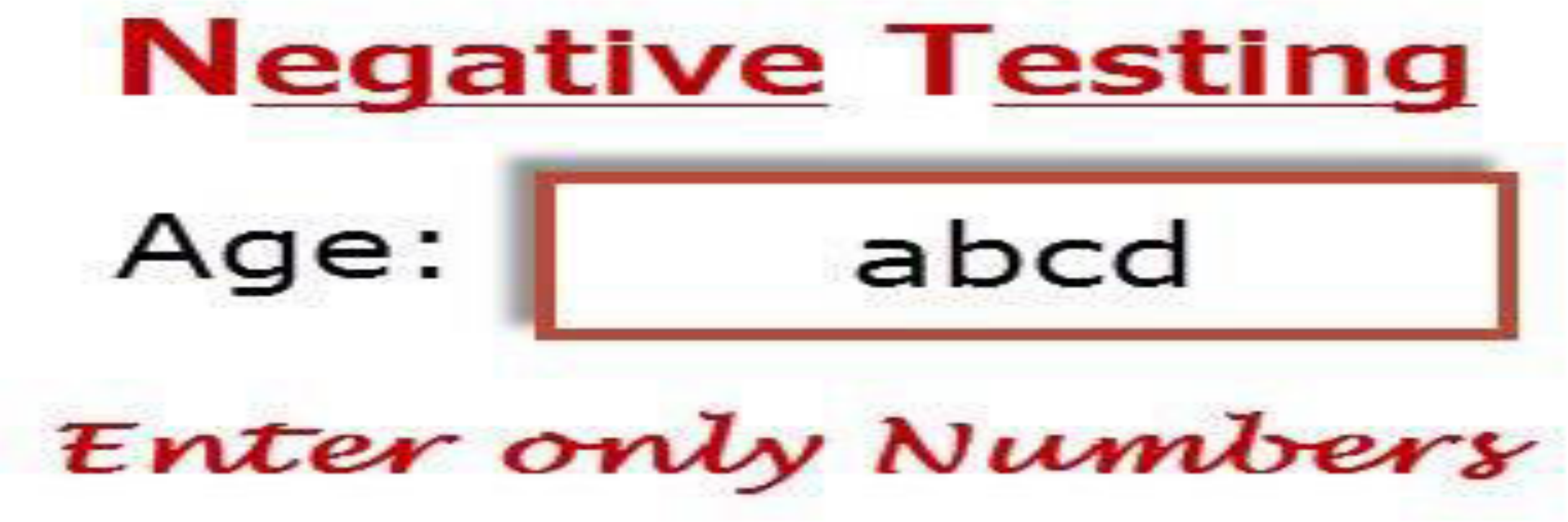
- Negative testing is done to show that the product does not fail when an unexpected input is given.
- The purpose of negative testing is to try and break the system.
- Negative testing covers scenarios for which the product is not designed and coded.
- In other words, the input values may not have been represented in the specification of the product.
- These test conditions can be termed as unknown conditions for the product as far as the specifications are concerned.
- But, at the end-user level, there are multiple scenarios that are encountered and that need to be taken care of by the product.
- It becomes even more important for the tester to know the negative situations that may occur at the end-user level so that the application can be tested and made foolproof.
- A negative test would be a product *not delivering an error when it should* or *delivering an error when it should not*.

Negative Testing

- Negative Testing is testing process where the system is validated against the invalid input data.
- A negative test checks if an application behaves as expected with its negative inputs.
- The main intention of this testing is to check whether software application not showing error when supposed to and showing error when not supposed to.
- Such testing is to be carried out keeping negative point of view and only execute the test cases for only invalid set of input data.
- Negative testing is a testing process to identify the inputs where system is not designed or un-handled inputs by providing different invalid inputs.
- The main reason behind Negative testing is to check the stability of the software application against the influences of different variety of incorrect validation data set.
- The Negative testing helps to improve the testing coverage of your software application under test.
- Both positive and negative testing approaches are equally important for making your application more reliable and stable.

Example of Negative Testing

- Consider a textbox example which should accept only integer values.
- So here provide the characters like “abcd” in the age textbox and check the behavior of application, either it should show a validation error message for all invalid inputs (for all other than integer values) or system should not allow to enter a non integer values.



Negative Testing

Age: abcd

Enter only Numbers

Example of negative test cases

S. No.	Input 1	Input 2	Current state	Expected output
1	Some other lock’s key	Turn clockwise	Lock	Lock
2	Some other lock’s key	Turn anticlockwise	Unlock	Unlock
3	Thin piece of wire	Turn anticlockwise	Unlock	Unlock
4	Hit with a stone		Lock	Lock

- In the above table, there are no requirement numbers.
- This is because negative testing focuses on test conditions that lie outside the specification.
- Since all the test conditions are outside the specification, they cannot be categorized as positive and negative test conditions.
- Some people consider all of them as negative test conditions, which is technically correct.

Positive and negative testing scenarios

- If the requirement is saying that password text field should accepts 6 – 20 characters and only alphanumeric characters.

- **Positive Test Scenarios**
  - Password textbox should accept 6 characters
  - Password textbox should up to 20 characters
  - Password textbox should accepts any value in between 6-20 chars length.
  - Password textbox should accepts all numeric and alphabets values.
- **Negative Test scenarios**
  - Password textbox should not accept less than 6 characters
  - Password textbox should not exceeds more than 20 characters
  - Password textbox should not accept special characters

Positive and Negative Test Scenarios with Example

- You are doing the testing on login form which have following fields like Username field, Password field, and Sign In, Sign Up, Cancel, Login Button etc.
- Now positive scenario of login form is that you enter the valid username and password in the username and password field, and then click on Login Button to check whether the user is able to login or not.
- Negative scenario of login form is that you leave the password field blank and fill the username field, and then click on Login Button to check whether the user is able to login or not.

Consideration for both the testing

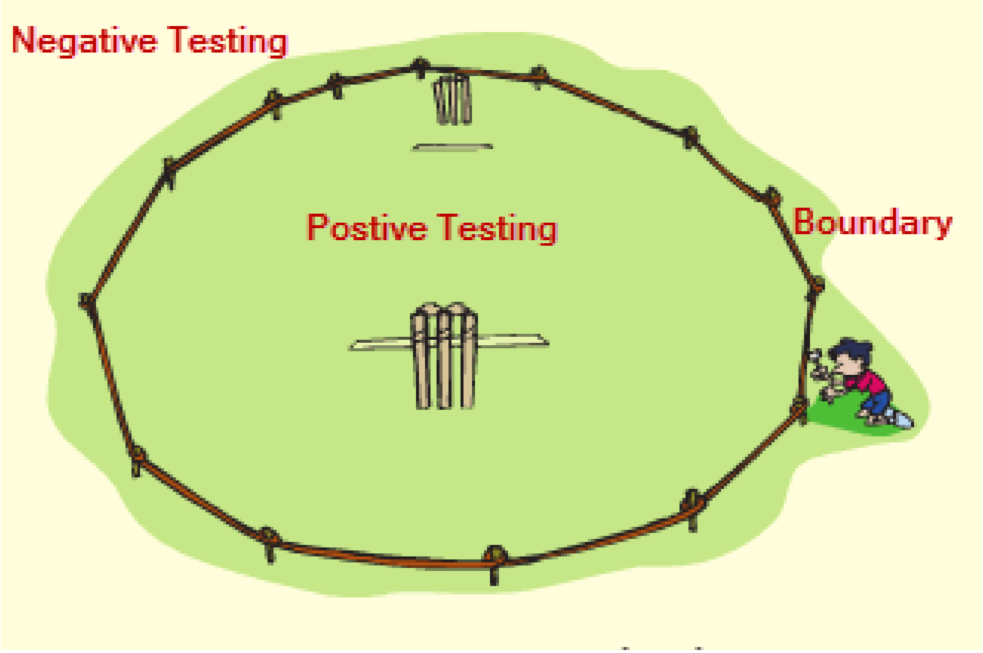
- In both the testing, following needs to be considered:
  - Input data
  - Action which needs to be performed
  - Output Result

Testing Technique used for Positive and Negative Testing

- Following techniques are used for Positive and negative validation of testing is:
  - Boundary Value Analysis
  - Equivalence Partitioning

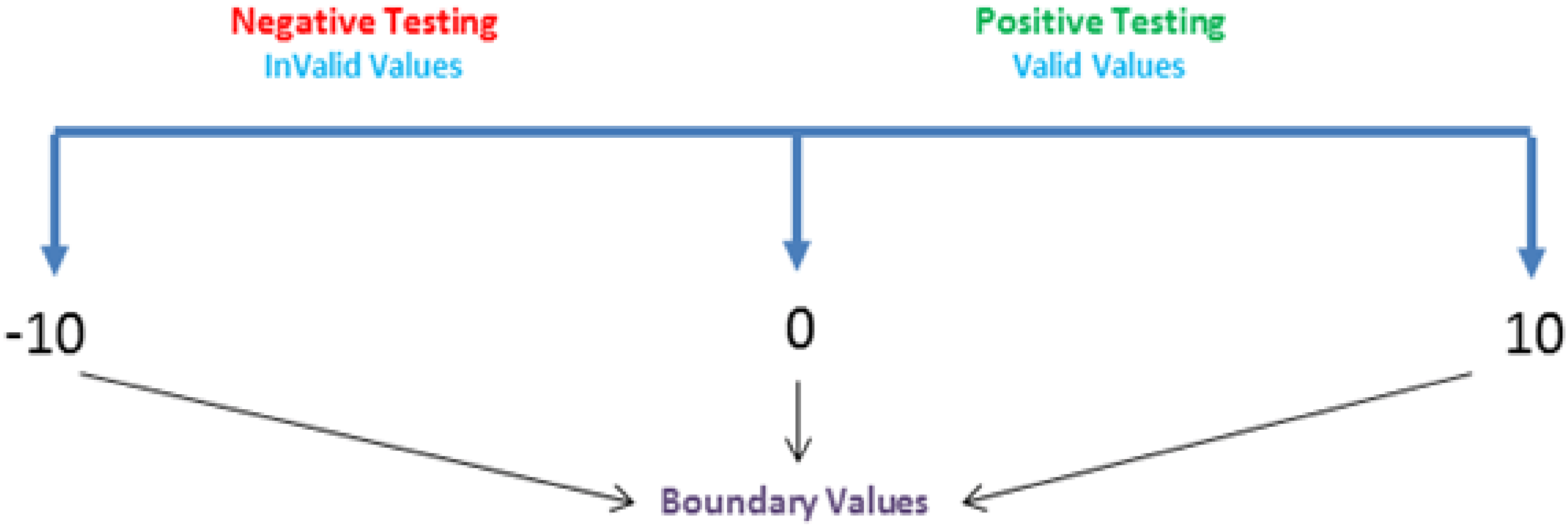
Boundary Value Analysis

- This is one of the software testing technique in which the test cases are designed to include values at the boundary.
- If the input data is used within the boundary value limits, then it is said to be Positive Testing.
- If the input data is picked outside the boundary value limits, then it is said to be Negative Testing.



Example

- A system can accept the numbers from 0 to 10 numeric values.
- All other numbers are invalid values.
- Under this technique , boundary values 0 , 10 and -10 will be tested.



Negative Testing
InValid Values

Positive Testing
Valid Values

-10

0

10

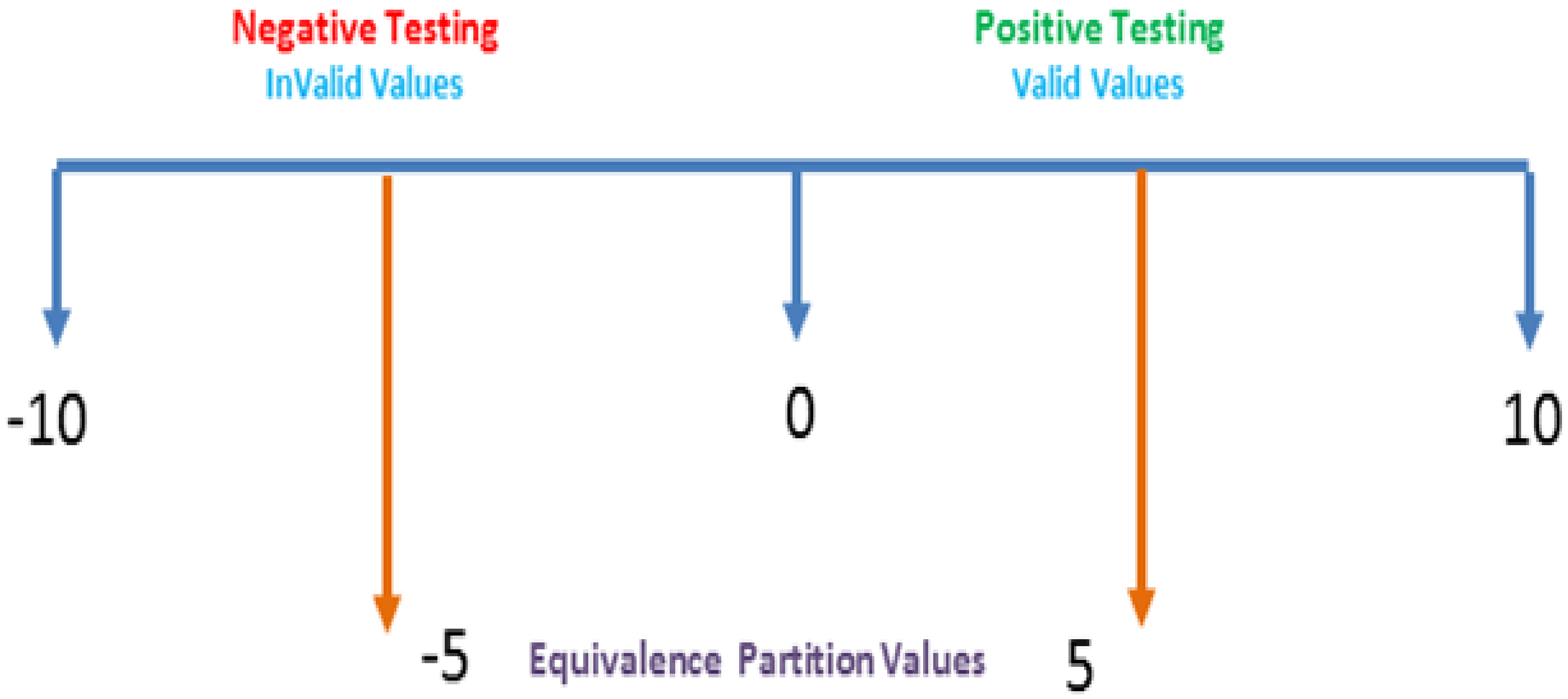
Boundary Values

Equivalence Partitioning

- This is a software testing technique which divides the input data into many partitions.
- Values from each partition must be tested at least once.
- Partitions with valid values are used for Positive Testing.
- While partitions with invalid values are used for negative testing.

Example

- Numeric values Zero to ten can be divided to two(or three)partition.
- In our case we have two partitions -10 to -1 and 0 to 10.
- Sample values (5 and -5) can be taken from each part to test the scenarios.



Difference between positive and negative testing

- For positive testing if all documented requirements and test conditions are covered, then coverage can be considered to be 100 percent.
- If the specifications are very clear, then coverage can be achieved.
- In contrast, there is no end to negative testing and 100 percent coverage in negative testing is impractical.
- Negative testing requires a high degree of creativity among the testers to cover as many “unknowns” as possible to avoid failure at a customer site.

Top Distinction between Positive and Negative Testing

Positive Testing (Valid)	**Negative Testing (Invalid)**
1. Positive Testing means testing the application or system by giving valid data.	1. Negative Testing means testing the application or system by giving invalid data.
2. In this testing tester always check for only valid set of values.	2. In this testing tester always check for only invalid set of values.
3. Positive Testing is done by keeping positive point of view for example checking the mobile number field by giving numbers only like 9999999999.	3. Negative Testing is done by keeping negative point of view for example checking the mobile number field by giving numbers and alphabets like 99999abcde.
4. It is always done to verify the known set of **Test Conditions**.	4. It is always done to break the project and product with unknown set of Test Conditions.
5. This Testing checks how the product and project behave by providing valid set of data.	5. This Testing covers those scenarios for which the product is not designed and coded by providing invalid set of data.
6. Main aim means purpose of this Testing is to prove that the project and product works as per the requirements and specifications.	6. Main aim means purpose of this Testing is try to break the application or system by providing **invalid set of data**.
7. This type of Testing always tries to prove that a given product and project always meets the requirements and specifications of a client and customer.	7. Negative Testing is that in which tester attempts to prove that the given product and project does, which is not said in the client and customer requirements.