Software Testing

Software testing is a process where the software products match the requirements and ensure that the software product. If there are any bugs or errors in the software, they can be identified early and can be solved before delivery of the software product. The testing process involves evaluating the features of the software product for requirements in terms of any missing requirements, bugs or errors, security, reliability, and performance. The software testing process approaches are-

- White Box Testing - In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.
- Black Box Testing- This method is used in which testers evaluate the functionality of the software under test without looking at the internal code structure.
- Grey Box Testing - This helps to create better test cases in this process.

The software testing levels are given below –

- Acceptance Testing- It's used to sign the software and also for payment purposes.

- System testing- Ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experiences with the application.

- Integration testing-It is the process of testing the connectivity or data transfer between a couple of units tested modules. It is divided into the Top-Down Approach, Bottom-Up Approach, and Sandwich Approach (Combination of Top-Down and Bottom-Up).

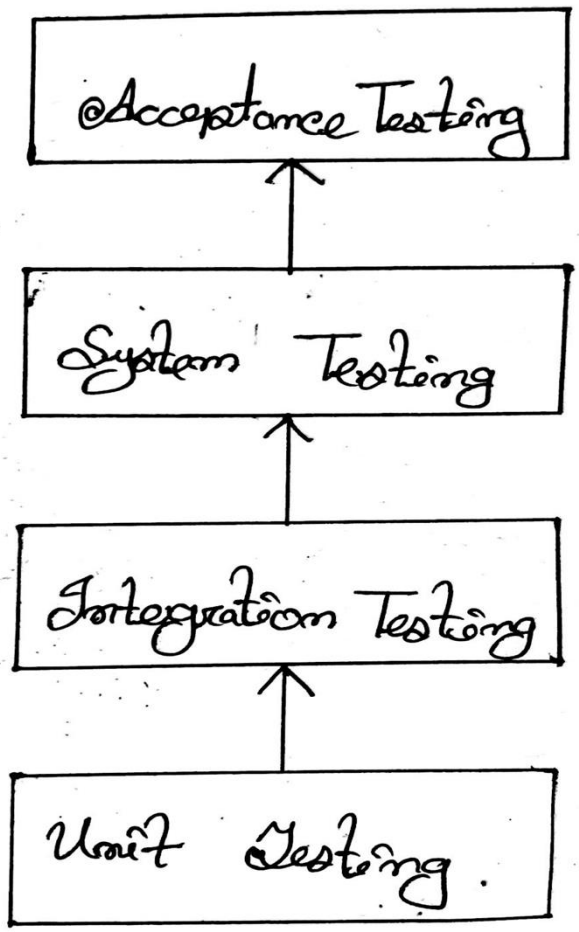
- Unit testing-Unit Testing is a software testing technique by means of which individual units of software i.e., a group of computer program modules, usage procedures, and operating procedures, are tested to determine whether they are suitable for use or not. In software programming, a unit may be an individual program, function, procedure, etc.

- ❖ Unit testing is a testing method using which every independent module are tested to determine if there is any issue by the developer himself.
- ❖ Unit Testing is defined as a type of software testing where individual components of the software are tested.
- ❖ Unit Testing of the software product is carried out during the development of an application. An individual component may be either an individual function or a procedure.
- ❖ Unit Testing is performed by the developer.

➢ Main Objective of Unit Testing:

1. isolate a section of code.
2. verify the correctness of the code.
3. test every function and procedure.
4. fix bugs early in the development cycle and save costs.
5. Help the developers to understand the code base and enable them to make changes quickly.
6. Help for code reuse.

Types of Unit testing:
1.Manual testing
2.Automated testing



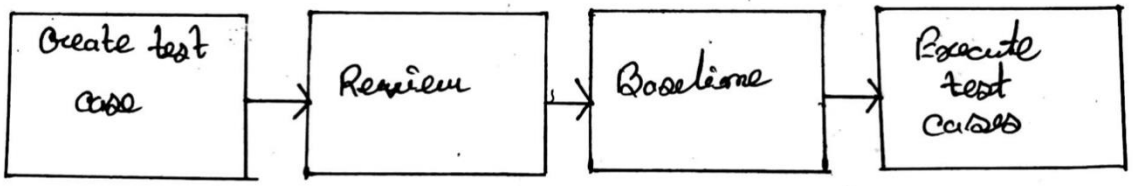


Fig: Work flow of Unit Testing

The payment method of Online Computer Shop's requirements are as following:

*Payment Method

— From account number (FACN)→ Text Box

— FACN —> accept only 4 digit

—To account no (TACN) —>Text Box

— TACN—>Accept only 4 digits

—Amount—>Text Box

— Amount —> Accept maximum 4 digits

— Transfer—> Button

—Transfer —> Enabled

— Cancel—> Button

— Cancel→ Enabled

The following applications is given by the customers :

-URL —> login Page

-Username/password/OK —> home page
-->While performing unit testing, we should follow some rules
-->To start unit testing, at least we should have one module.
-->Test for positive values
-->Test for negative values
-->No over testing
-->No assumption required
-->When the maximum test coverage is achieved, we will stop the testing.
-->start performing the unit testing on the different components such as
-From account number(FACN)

-To account number(TACN)
-Amount
-Transfer
-Cancel

From account number components
-Values Description
777 | accept
-123 | Error message —> account valid or not
-blank | Error message—> enter some values

-5 digit/ 3 digit | Error message—> accept only 4 digit
-Alphanumeric | Error message —> accept the only digit
- Blocked account number | Error message
- Copy and paste the value | Error message—> type the value

-Same as From account number and to account number | Error message
* To Account Number component:
- Provide the values just like we did in From account number (FACN) components
*For Amount component
-Provide the values just like we did in FACN and TACN components.
*For Transfer component:
-Enter valid From Account Number value
-Enter valid To Account Number value
-Enter the correct value of Amount
-Click on the Transfer button —> amount transfer successfully (confirmation message)
*For Cancel Component :
-Enter the values of From Account Number, To Account Number, and Amount.
- Click on the Cancel button —> all data entered should be cleared