**Manual Testing**

1. What is Software Testing?

Software Testing is a part of our software development process and it used for identify the defects in the software and the testing main objective Is to release a quality product to the client.

1. Software Quality?

The quality is defined a justification for all the requirements of a customer in a product quality is not defined a product it will defined the customers mind.

Ex: Bug free, delivered on-time , budget, meet requirements as per expectations, maintainable

1. Product Vs Project?

**Project**: A software application developed for a specific customer based on the requirements is called Project.

**Product**: A software application developed for multiple customers based on the market requirements is called Product.

1. Why do we need Testing?

* Bug free
* The system meets software specifications as per customer requirements
* It meets a end user expectations
* Fixing the bugs before release the software

Ex: error, Defect/Bug, failure

1. Why the software has bugs normally?

* Miscommunication or no Communication
* Software complexity
* Programming errors ( compile errors)
* Changing requirements
* Lack of skilled testers

1. Software Development Life Cycle (SDLC)?

The software development life cycle is a process to follow the software industries the process shown below :

1. Requirements Analysis
2. Design
3. Development
4. Testing
5. Maintenance

7. Waterfall Model (Advantages & Dis Advantages) ?

**Advantages:**

* Quality of product is good
* If the client change a previous requirements in the software application is not allowed
* Chances of finding the bugs is less
* Investment is less on initial stage later the tester will be need
* Recommended for small projects

**Dis Advantages:**

* Requirement changes are not allowed
* If there is a defect in requirement that will continued in next phases
* Time investment is more it leads to high investment
* Testing will starts after development (coding).

8.Why Testing is Necessary?

the software applications is using any level of testing will cannot declare that there is no defect in the product

**Development:**

* The developer will always work as per client requirements
* But each level of software building (including system level) to access whether it is really works or not.
* Integration issues will caused when different units do not work together, even working independently
* To bring a individual units together and make a final product, and some defects may be possible in the source code when developers are working with different places.

**Testing Process:**

* Testing is a process of demonstrating that errors are not present in the product this approach is used in acceptance testing where if the application meets acceptance criteria, then it must be accepted by the customer.
* Testing is gives number of detects present which indirectly gives a measurement of software quality.
* More number of defects indicate bad software and bad processes of development

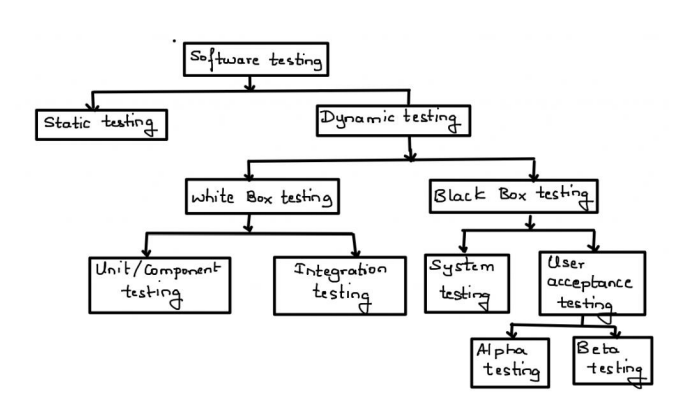
**Software Testing:**

* it is process to test an application to find out error in it
* checking the software is ok.
* The goal of software tester to find bug
* Verifying and validating that a software or application is bug free

**Testing Types:**

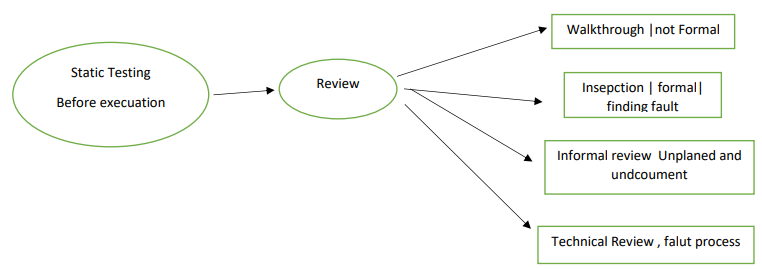
* **Manual Testing:** Manual testing is testing a software manually without using as tool or any script
* **Automation Testing:** When the tester writes scripts and uses another software to test the product. This process involves automation of a manual process.

Software testing basically two types static testing and dynamic testing



**Static Testing Techniques:**

* Analysis of a program carried out without executing the program
* Done during verification process before development
* As we know 85% errors found in design phase
* is code is tested in static testing ? “No” || the documentation is tested
* software development starts , continue and ends with documentation
* early documentation is used to define the software to be built
* later documentation covers the software training, installation, and operation (user guides).
* The primary goal of static testing is reduce defect by reducing defects in the documentation from which the software is developed



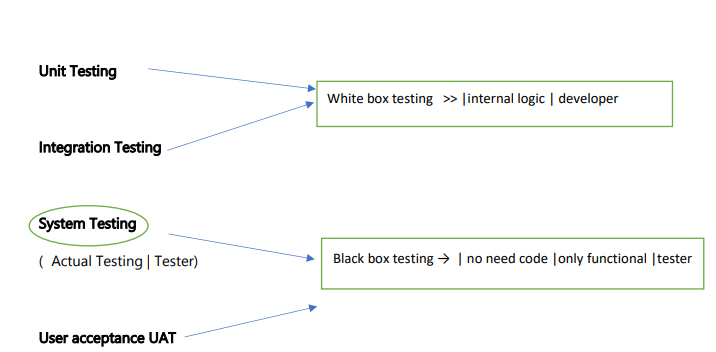
**Dynamic Testing Techniques:**

* **Unit Testing:** is unit testing individual component of software tested. The purpose of this testing is each module is working properly it focus on the smallest unit of software design

Ex: in a program we are checking if loop, method or function is working fine

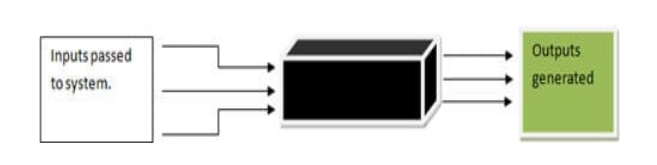
* **Integration Testing**: in integration testing individual units are combined and tested as group (developer) like (i) Top-down, (ii) Bottom-up (iii) Sandwich (iv) Big-Bang the main purpose of integration testing to check modules are communicating each other as DFD Data Flow Diagram which is specified in TDD (Technical Document Diagram).
* **System Testing**: In this testing we can test whole application (complete/integrated software is tested) done by tester.
* **Acceptance Testing**: a level of software Testing in which software is tested for user acceptance UAT done at client location where software is actually used **Alpha Testing:** done by tester in company in presence of customer

**Beta Testing**: done by customer to check software is ok, satisfy requirement



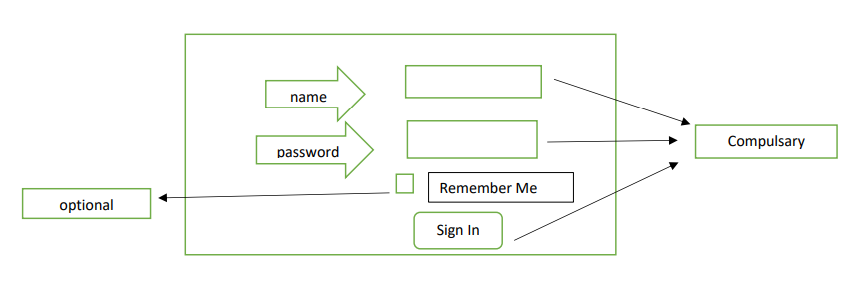
**Testing Types:**

* **Functional Testing:** is the process through which QA’s determine if a piece of software is acting in accordance with pre-determined requirements. It is uses black-box testing techniques, in which the tester has no knowledge of the internal system logic. **Functional testing is only concerned with validating if a system works as intends** Functional testing is a type of black box testing “does this actually work?” The ultimate of functional testing is to ensure that software works according to specifications and user expectations like input values , test cases, compare actual and expected output

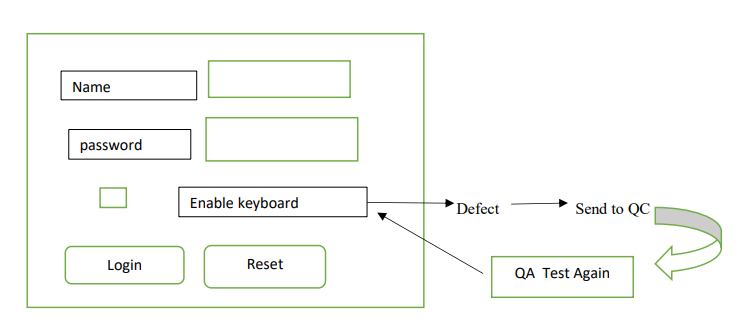


Ex. Login functionality, registration functionality

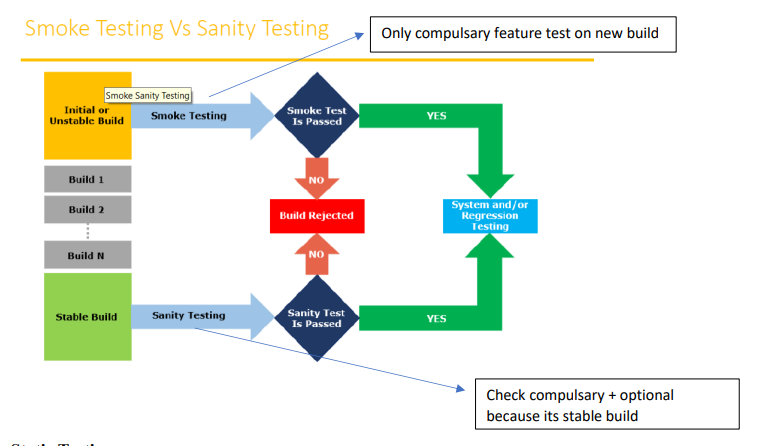
* **Non Functional Testing:** Load testing, Reliability, The readiness of a system, usability testing Ex: checking how many people can simultaneous basket
* **Black box testing:** Monitoring internal structure check internal logic. Done by developer
* **Smoke testing:** testing on newly released build compulsory requirement it is first testing on newly released build (Build Verification Testi check the deployment software build is stable or not

****

* **Sanity Testing:** (Testing on newly released build then check compulsory + optional) in the above fig all field are compulsory then it is sanity testing
* **Retesting:** Testing functionality once again

****

* **Regression Testing (Re-running if code changes):** it is overall testing whenever new change is occurred re-running functional + nonfunctional test code change , does not impact existing functions after changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

****

**Static Testing:** The basic process of testing is Review, Inspection, Walkthrough Test software without executing software (just test documents) This test done for avoid bugs in early stage (review testing ) look and feel why static testing **early defect detection and correction**, **to get fewer defect at a later state of testing**

**Static Technique has 3 types:**

**Review:** review before development i.e simple document read the document s/b correct or not document s/b correct and complete requirement review , design review , Test plan review, review testing review testing can do anybody, manger | developer | tester |coworker etc.

**Walkthrough:** it is informal , anytime, not planned, done when ever required author of document will explain to their team

**Inspection:** Most formal , 3-8 people in meeting Proper meeting, Schedule which is intimated by mail

**Dynamic Testing:** The main purpose of dynamic testing is to test software behavior with dynamic variables dynamic testing requires code to be executed. Static testing -> just analyze the code, no need execution

* **Alpha Testing :** Its is final testing in development Advantage : immediate solution is possible
* **Beta Testing :** it is 1st testing in client side . it is also called user user acceptance testing UAT Disadvantage : no immediate solution if defect is found
* **Installation Testing :** providing required resources at client location It is type of testing in which test engineer check deployment process is successful as per user guideline Deployment document /user manual : it is document prepared by project manager
* **Usability Testing :** checking application for user friendliness
* **Monkey Testing** : used for game testing, used for random input To check the application or system will crash
* **Portability Testing** : Developed application Should support multiple environment
* **Forced error Testing :** to check valid error message will display
* **Exploratory Testing** : When test engineer does not have idea of functional testing then he is learning through exploring application
* **End to End Testing** : We can check all internal component for successful response Internal component like Client , Network, Server Database etc are working fine Means Testing internal component
* **Security Testing** : Checking Security of application
* **Reliability Testing** : The Developed application Should Support Longer Duration i.e. Stability
* **Audit** : it is independent evolution of software .
* **Inspection** : it is formal evolution of software
* **Concurrency Testing** : multiuser Testing
* **Debugging** : executing program line by line for finding errors.

**Some of the most popular SDLC models are:** no matter which module is used each has same phases

* Waterfall Model
* V- Shape Model
* Incremental Life Cycle Model
* Spiral Model

SDLC : Software development life cycle it is process used by software company to develop, test software

1. Requirement analysis

2. Design (blue print)

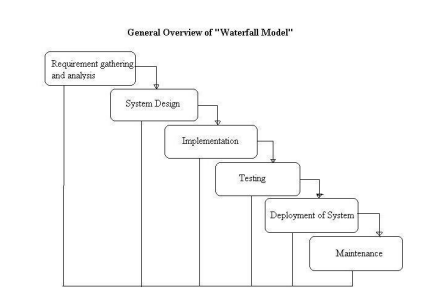
3. Coding or development

4. Testing

5. Maintenance

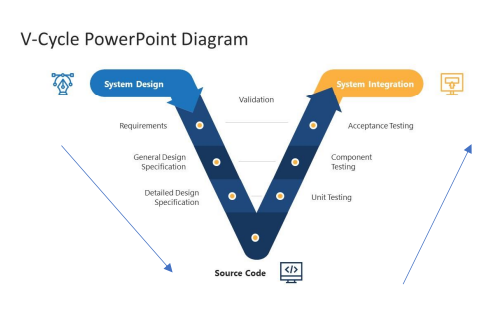
**Waterfall Module:** it is old traditional model it is linear model i.e steps 1 after another Each phase must complete to start new phase (i.e called one after another)

* In waterfall module quality of product is food because every phase has clear Documentation
* SRS (System requirement specification) not changed hence no bug
* Initial investment is less because no tester involved
* No change in middle
* Testing will start after coding



**V shaped Model:** Verification and validation verification -> done before development check we are doing correct -> verifying document -> because no software ready -> verification =before s/w = static -> review ->walkthrough -> inspection-> static testing = verifying doc

Validation -> testing software -> done after software ready -> done right ->validation = after s/w = dynamic -> dynamic testing = unit -> integration - > system -> user acceptance



System design = before development = verification = static testing = verify documents

System integration=after devep = validation=dynamic testing = unit |integration|system|UAT in V model = testing is involved every phase

Disadvantage : more documents

**Spiral Model:**

* Spiral Model is iterative model.
* Spiral Model overcome drawbacks of Waterfall model
* We follow spiral model whenever there is dependency on the modules
* In every cycle new software will be released to customer
* Software will be released in multiple versions so it is also called version control model.



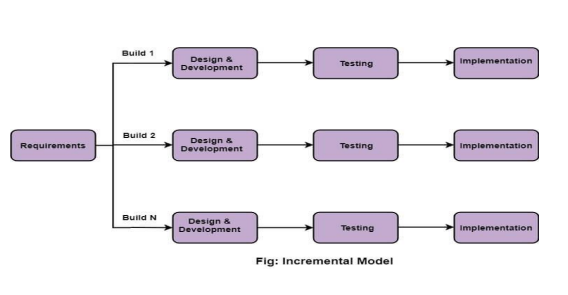
Advantages:

* Testing is done in every cycle , before going to the next cycle.
* Customer will get to use the software for every module.
* Requirement changes are allowed after every cycle before going to the next cycle

Disadvantages:

* Requirement changes are NOT allowed in between the cycle
* Every cycle of spiral model looks like waterfall model.
* There is no testing in requirements & design phase.

**Increment Model**: requirement are divided into multiple module each module goes through SDLC phase i.e analysis, design ,coding, testing, maintenance Requirement -> module 1 + module 2 +…module n



When use incremental model: A Project has lengthy development schedule. When the requirement are superior.

Disadvantages: Need for good planning, Total cost is high

**System Testing (Actual Testing):** the types of system testing are mentioned in the below

* GUI Testing
* Usability Testing
* Functional Testing
* Non Functional

**GUI Testing:**

* Testing GUI application , user interface testing
* Such as menus, check boxes, icon ,images
* Not functional , just look and feel
* Check size and position of element
* Image quality, spelling check, alignment
* Fonta are understanding or not

**Usability Testing:**

* Check the easiness of application
* Helping messages are display it user confuse
* Check user friendly application or not

**Functional Testing:**

* Check behavior of application
* Check database testing (work with database, ok?)
* Error handling, display error message ok?
* Calculation and manipulation
* Check text box disable or enable as user requirement
* Check database operation DML table, column, records etc
* Checking database operation Black box testing + white box testing = Gray box Testing

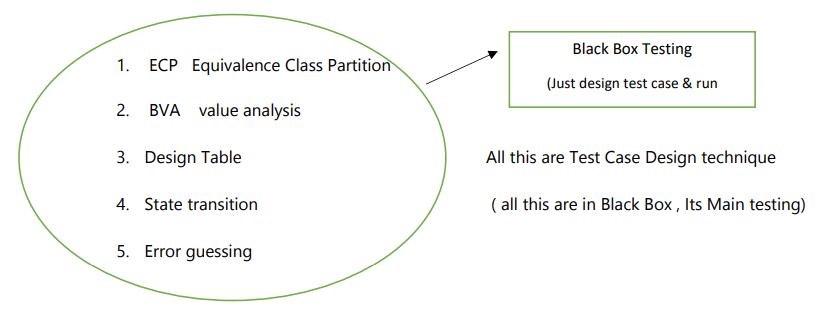
****

**Non Functional Testing:**

* Once functional testing done i.e. s/w work user requirement then do nonfunctional testing
* Performance testing Load testing – gradually increase the load
* Performance testing load testing – gradually increase the load , stress testing – suddenly increase the load (Eg. Online filling form), volume testing – how much data handle
* Security of software
* Recovery of application
* Compatibility testing – work with all platform
* Security of software
* Recovery of application
* Compatibility testing – work with all platform

**End to End Testing:** testing overall application after including all module Ex: login -> add customer -> delete and edit customer -> logout Testing all functions i.e. add delete edit and logout

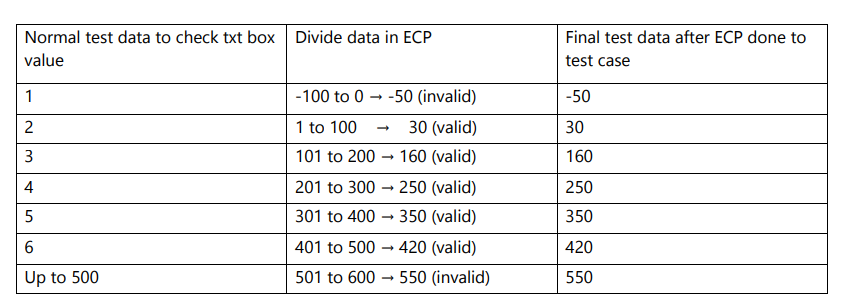
**Test Case Design Technique:** it helps better design and reduce the number of test case to be executed reduce data and more coverage

****

**ECP -> Equivalence Class Partition:**

* Value check
* Classify , divide , partition data in -> multiple classes to save testing time

Ex: Enter number \_\_\_\_\_\_\_\_ \* allow digit from 1 – 500



**Equivalence class Partition (ECP):**

**A..Z - > (Valid)** Test Data using ECP (XYZ)

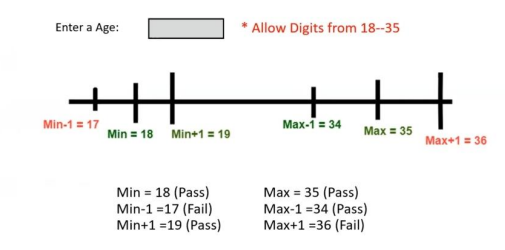
a..z -> (Valid) zyz

Special Characters -> (Invalid) @#$%

Spaces - > 250 (Invalid) Xy z

Numbers -> 320 (Invalid) 1234

**Boundary Value Analysis (BVA):**

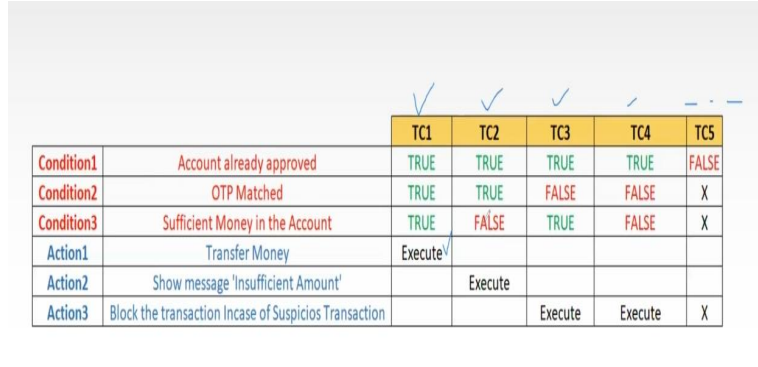
****

**Design table technique:** This technique is used if we have more condition and based on condition, we have to perform the action

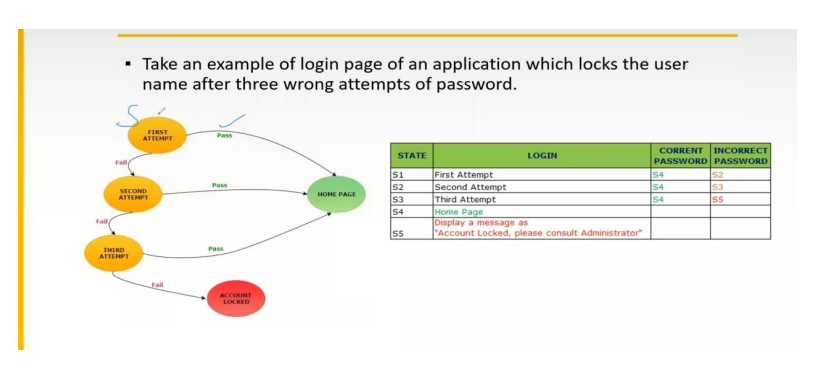
**Ex:** transfer money from account to account

**Condition:** 1. Account no has to approve , 2. OTP Matched , 3. Sufficiency money in action

**Action:** 1. Transfer money , 2.Show message insufficient money (if any), 3. Block if any suspicious activity

****

**State Transition Technique:** Take action depends on state

****

**Error Guessing Technique:**

* No any specific rule
* This test based on tester skill Ex Submit form empty and guess error.

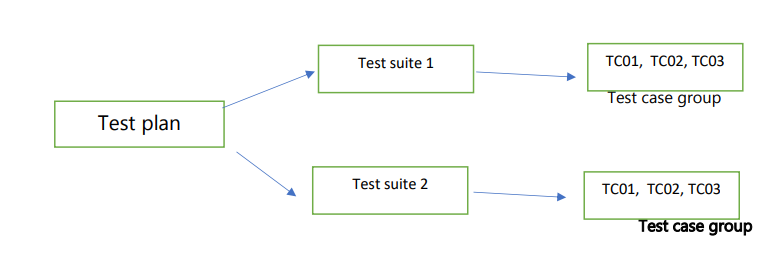
Test Case Scenario: Simply the name of test what to test (name of Test)

Test Case -> how to test i.e. step Group of steps that is to be executed to check functionality

Ex: Test scenario = check functionality of login button

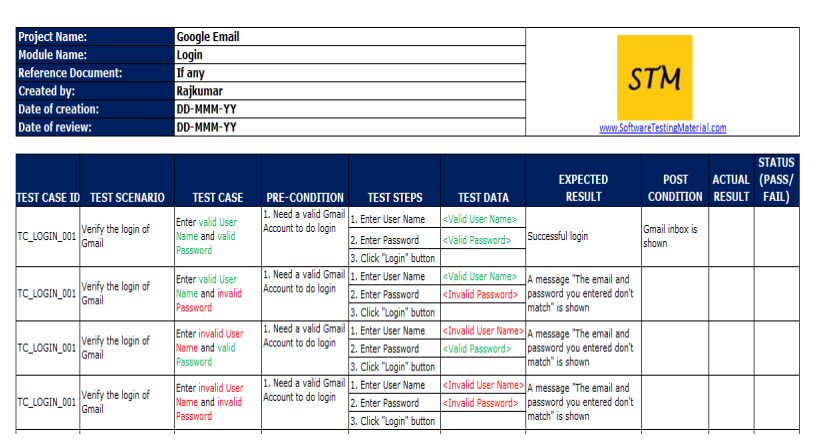
Test case = TCO1 , TC02 , TC03, … etc

Test suite -> group of test case



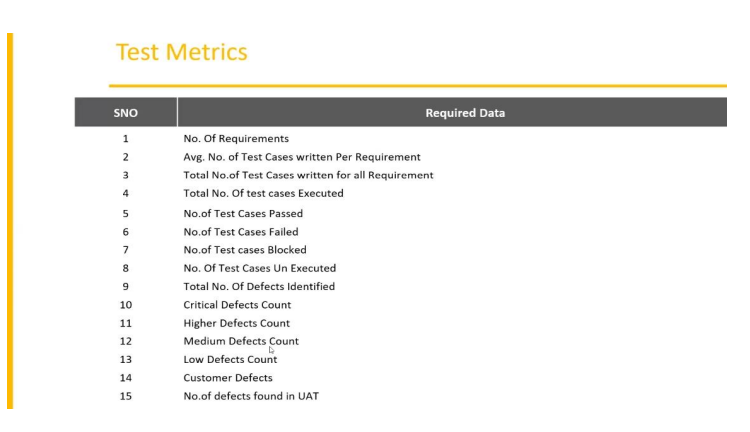
**Test case document:**

* Test case id
* Test case Title
* Description
* Precondition
* Priority
* Request id
* Steps/Action
* Excepted result
* Actual result
* Test Data

****

**Requirement Traceability matrix : (RTM):**

* Trace how many Test case are executed or covered
* In simple keep track of test cases

****

**Test Case Execution:**

* Executing test case based on test plan
* Mark status Pass | Fail | Blocked
* Reports defects in bug report

**Defect Reporting Tool:**

* ClearQuest : only bug report
* Devtrack : only bug report
* Jira -> test management tool (track each activity)
* Bugzilla -> test management tool (tack each activity)

**Defect Report Details:**

* Defect id
* Defect Version
* Step: details of step along with developer what to do
* Date
* Detected by
* Status
* Fixed by in process | fixed
* Severity – impact
* Priority – high | medium | low

**Severity of defect:**

* Blocker | Critical | Major | Minor
* Testing engineer decides the severity level of the defect.
* Blocker: this defect show application not processed
* Critical: main function not working
* Major: some undesirable behavior e.g. Email sent but msg not display
* Minor: look and feel

**Priority of defect:**

* High| Medium | Low
* Importance of defect on which priority defect will be solved or fixed
* P0 – High fixed immediately in same version
* P1 – Medium fixed in next release
* P2- Low next version

**Manual Testing Project : E commerce**

* Project introduction
* Understanding and explore the functionality
* Test Plan
* Writing test scenario
* Writing test cases
* Environment setup and build and development
* Test Execution
* Bug reporting and tracking
* Sanity testing, smoke testing, regression testing
* Test sign off

**E commerce Project:**

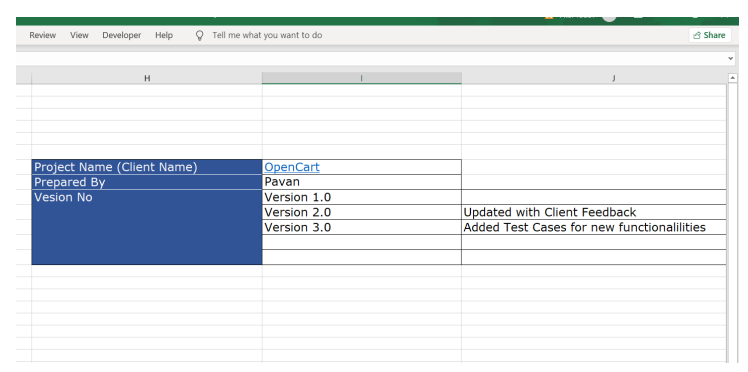
* Login
* Search for product and item
* Add them to cart
* Do payment
* Product will be delivered
* Return the product
* Etc.

**SRS document:** A Software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform Refer SRS Document

**E commerce Project:**

* Project information
* Understanding the functionality of project
* Test Plan -> a detailed document of testing activity
* Writing test scenario
* Test case and review
* Environment setup and deployment for testing application
* Test Execution
* Bug reporting and tracking
* Sanity and regression testing
* Test sign off

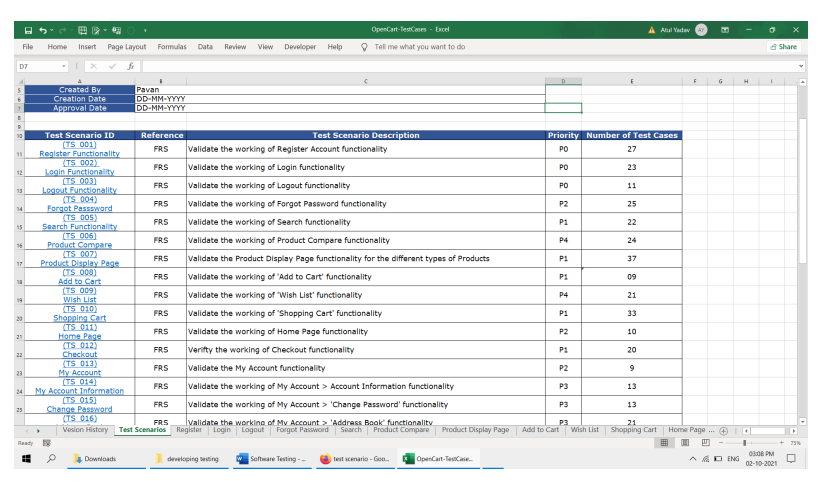
**Version Page:**



Opencart.com project

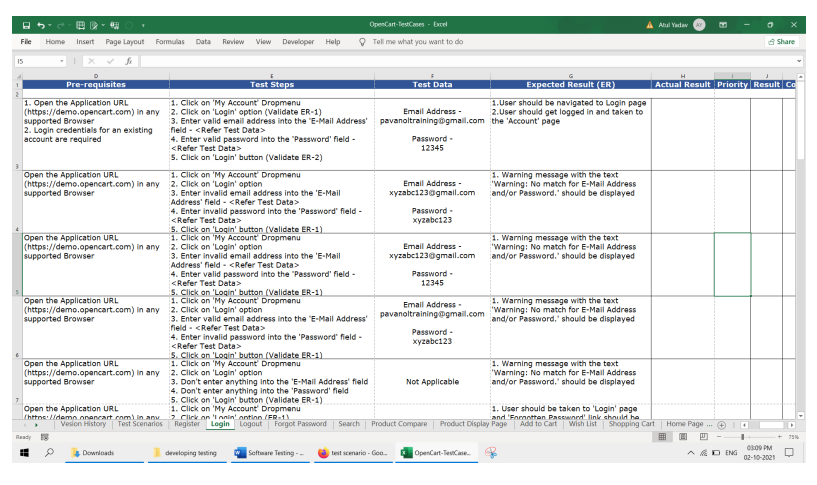
1. FRS Document -> How Software works An FRS, or Functional requirement specification is the document that describes all the functions that software or product has to perform
2. Test plan Document -> a detailed document of testing activity
3. Test Scenario Document -> Anything that can be tested is a Test Scenario -> Simply name of test case

Test Scenario Document



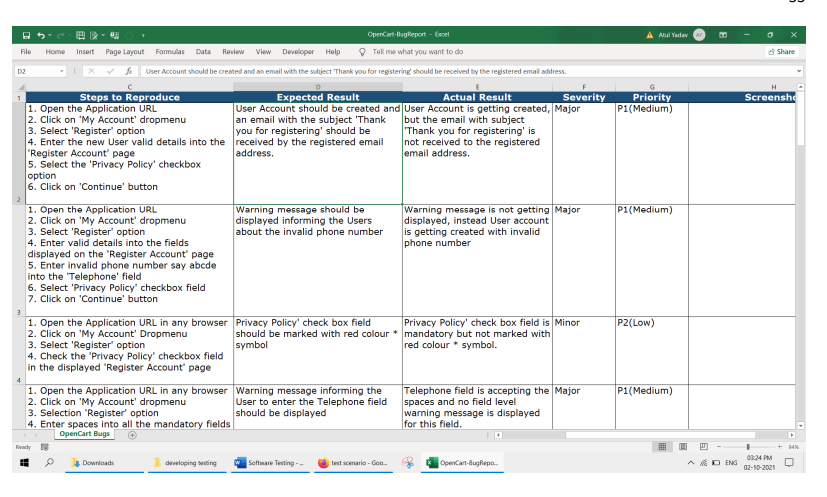
1. Test Case Design: While writing test case refer the test Scenario and FRS Document

Test case template:

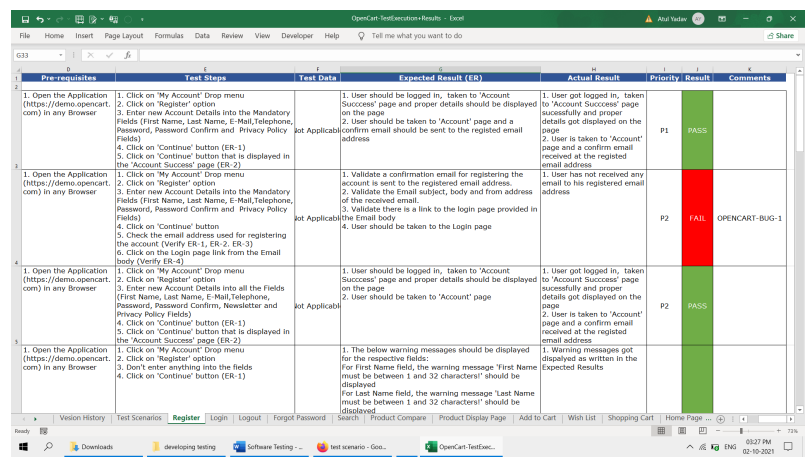


All above are sample test case After executing above steps i.e. test case update the field Actual result result-> pass or fail Priority After executing test case maintain or update sheep RTM e.g. No test executing or blocked etc.

If the bug found in testing report it in bug tracking file

Eg. 

Test case with result:



Big Bang Testing Approach

Big bang’ approach involve testing software system after development work is completed. This is also termed “system testing” or final testing done before releasing software to the customer for acceptance testing.

Big Bang == System Testing == Final Testing == Before Release

This is last part Of Software development as per waterfall methodology.

**Testing Vs Debugging**

**Testing:** Testing is done to Find bugs

**Debugging**: Debugging I an art of fixing bugs

* **Black box testing**: Mainly perform by testers
* **White box testing**: Mainly perform by developers
* **Unit testing:** Part of white box testing
* **Acceptance testing:** This is the final testing done by customer based on the agreements Load/ stress / performance testing . Testing an application load capacity
* **Usability testing:** Testing to determine the user friendly ness of the application
* **Install/ uninstall testing:** Testing of full, partial, or upgrade install/ uninstall processes.
* **Recovery / failover testing:** Testing to determine how well a system recovers from crashes , failures, or other major problems.
* **Incremental integration testing:** Continues testing of an application as new functionality is added
* **Ad-hoc testing:** conducting testing without requirements.
* **Comparison testing:** Comparing software weaknesses and strengths to competing products.
* **Alpha testing:** Part of UAT
* **Beta testing:** Part of UAT
* **Integration testing:** validating combined modules of an application.
* **Functional testing**: part of black box testing
* **System testing:** part of black box testing and validating the system requirements
* **End to End testing:** like system testing
* **Sanity testing or smoke testing** : An initial validation of a New build or release
* **Regression testing** : validating the existing functionality of the application once new fixes added
* **Compatibility testing**: Testing an application in different environments.

**A bug is a issue or error in code or any environmental issue.**

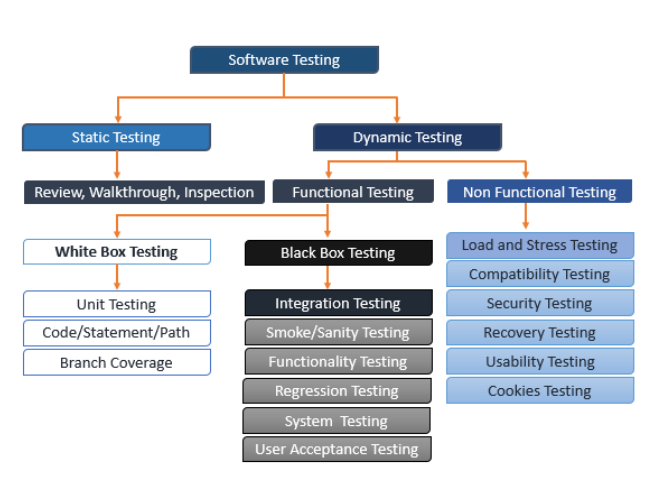
**What is Test Case?**

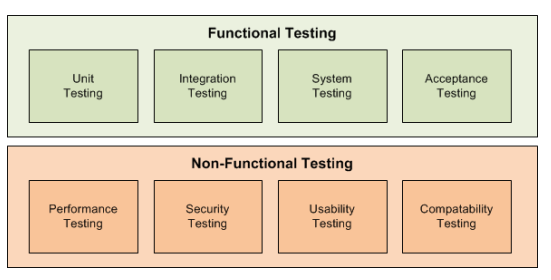
A Test case is a detailed explanation of as scenario. Test case is a document which describes pre-condition post-condition, test data, actors and navigation of the particular functionality. Each test case should have unique test name and test ID.

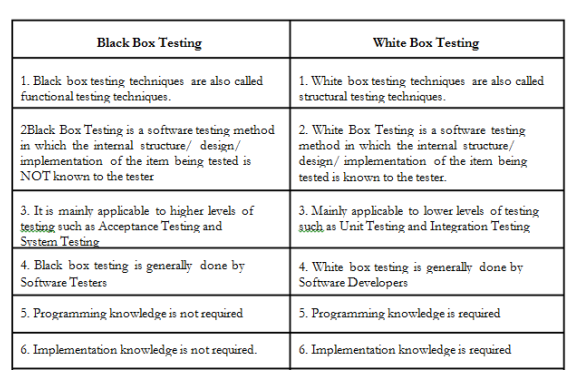
**What is Test Plan?**

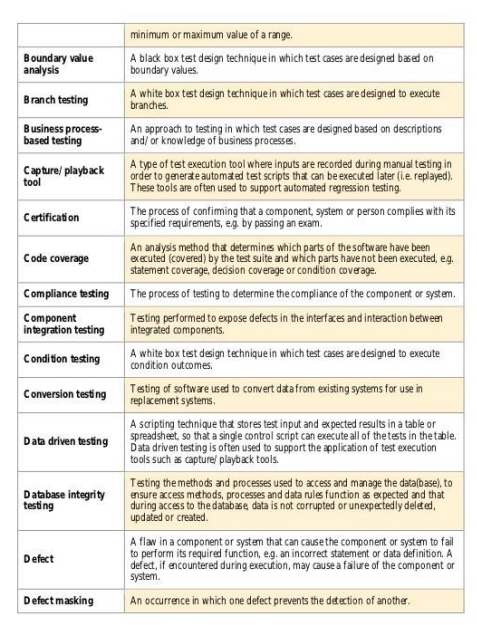
Test plan is a document that explains what to test, when to test, where to test and when to complete

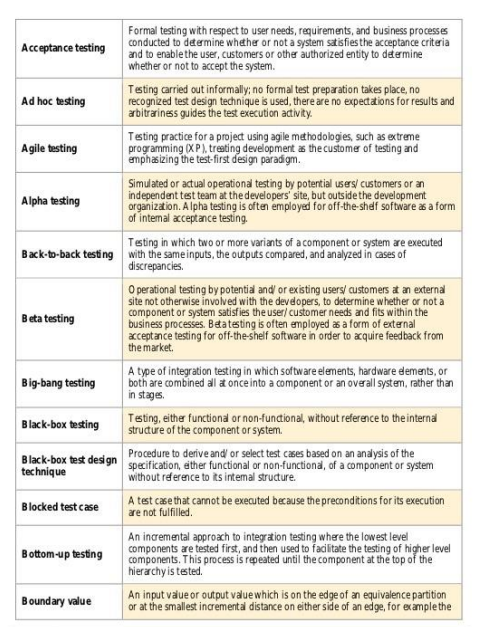
**3- Retesting is only done for failed Test cases while Regression is done for passed test cases**

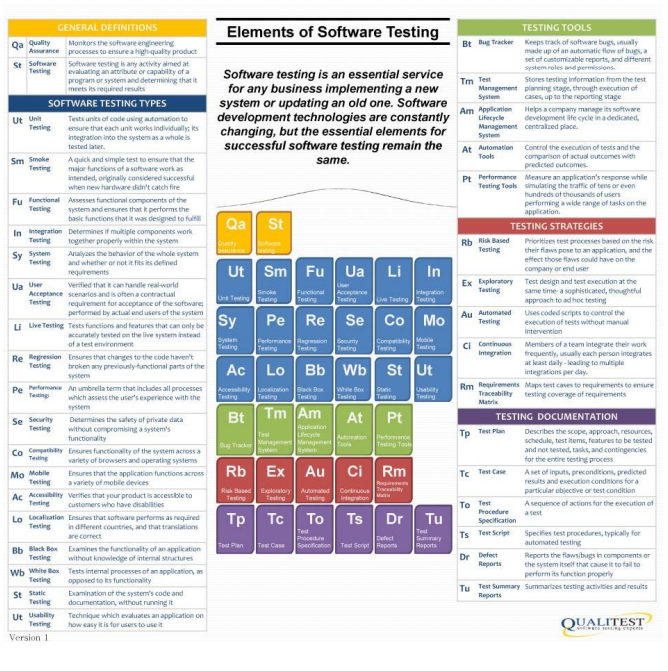


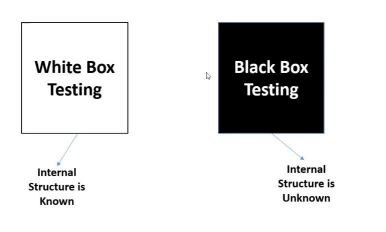


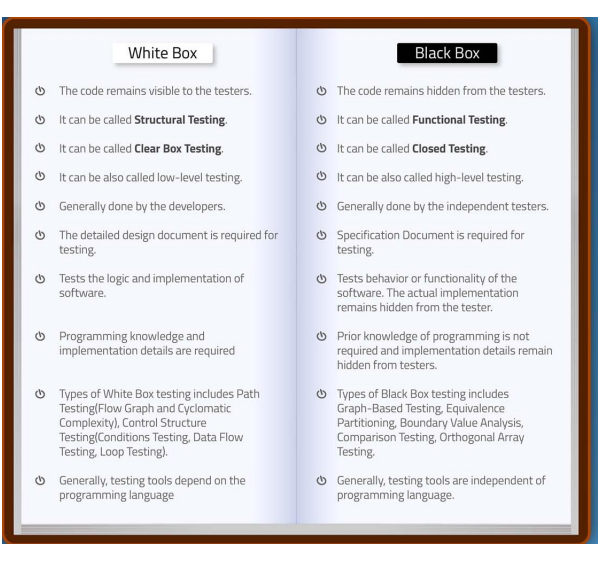


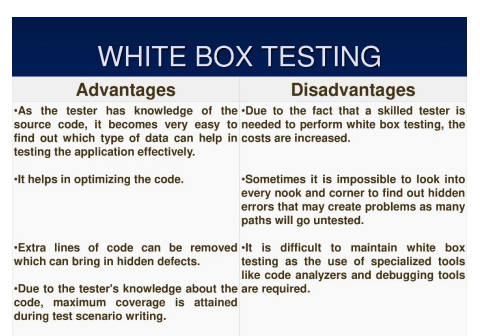


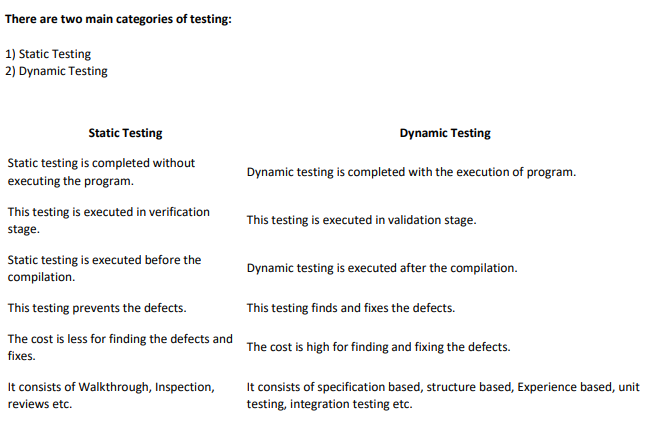


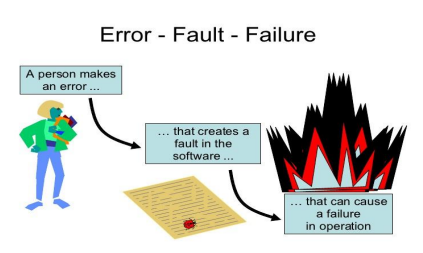


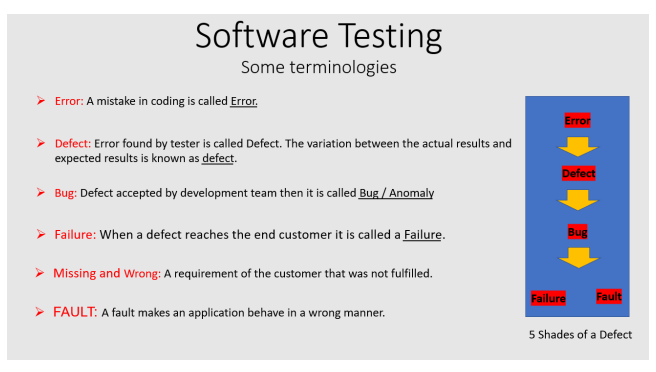


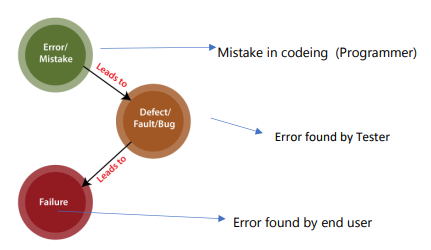


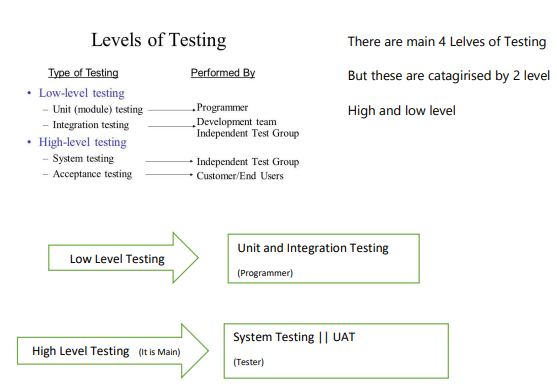


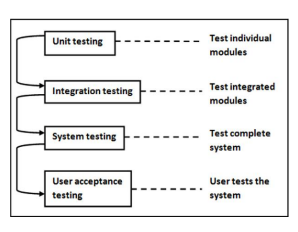




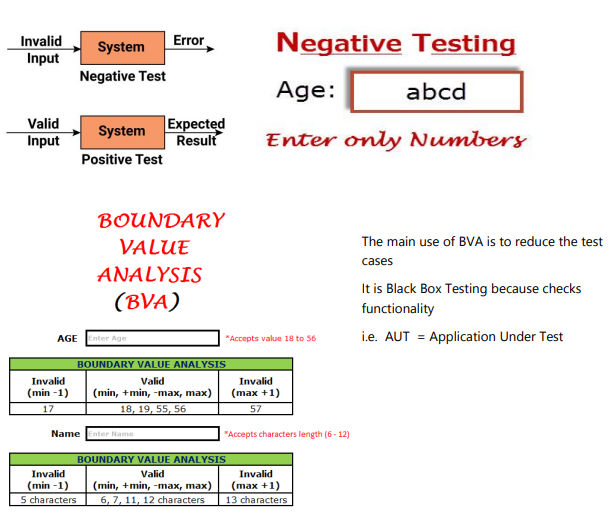


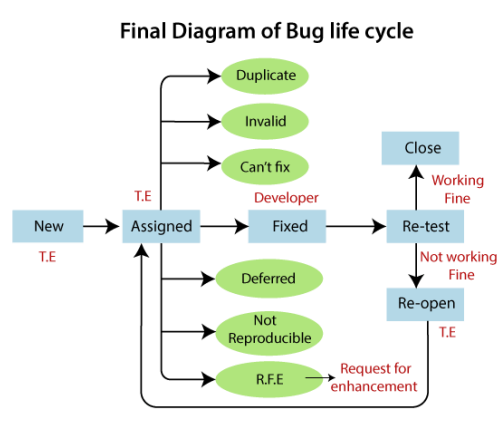


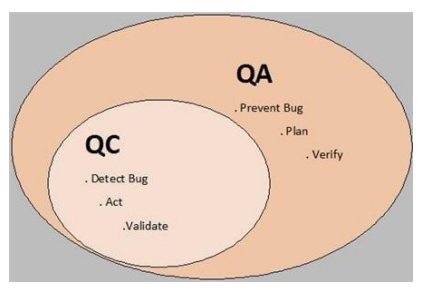




Test Planning : A document that describe how to perform testing on entire application Quality lead or QA Manger >> Prepare Test Plan -> What to test ? -> When to test? -> How to test?



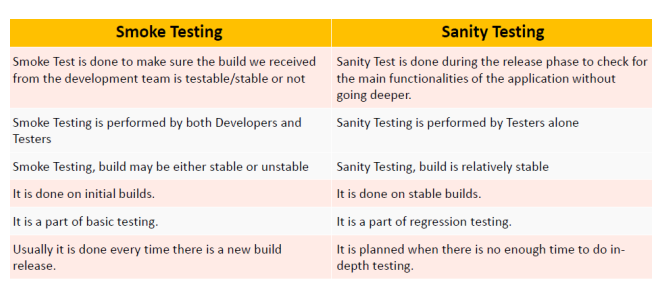




* Verification is the process confirming that something – software – meets its specifications
* Validation is the process confirming that it meets the user’s requirements
* Verification meets specification as document
* Validation meets user requirement (because software is ready hence main tested for User)
* Verification : tis static process of analyzing the document s, not actual product.
* Validation : it involve Dynamic Testing (unit, integration, system testing)







**Exploratory Testing:**

* We have to explore the application, understand completely and test it.
* Understand the application, identify all possible scenarios, document if then use it for testing.
* We do exploratory testing when the application ready but there is no requirement
* Test Engineer will do exploratory testing when there is no requirements.
* **Drawbacks**: You might misunderstand any feature as a bug (or) any bug as a feature since you do not have requirement
* Time consuming
* If there is any bug in application, you will never know about it.

**Ad hoc Testing:**

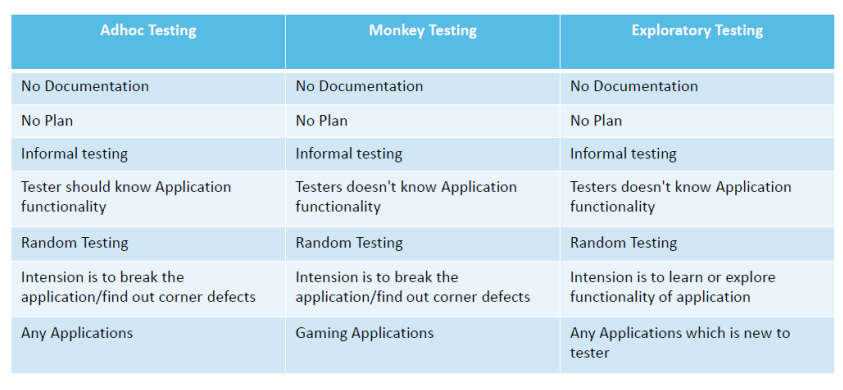
* Testing application randomly without any test cases or any business requirement document.
* Ad hoc testing is an informal testing type with an aim to break the system.
* Tester should have knowledge of application even though he doesn’t have requirements/test cases.
* This testing is usually an unplanned activity.



**Monkey/Gorilla Testing:**

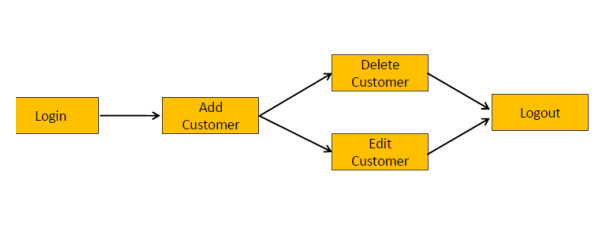
* Testing application randomly without any test cases or any business requirement document.
* Ad hoc t3esting is an informal testing type with an aim to break the system.
* Tester do not have knowledge of application.
* Suitable for gaming application.

**Difference b/w Ad hoc Testing Vs Monkey Testing Vs Exploratory Testing**

****

**END-To-END Testing:** Testing the overall functionalities of the system including the data integration among all the module is call3ed end-to-end testing.

* Login
* ADD New Customer
* Edit Customer
* Delete Customer
* Logout



**Globalization and Localization Testing:**

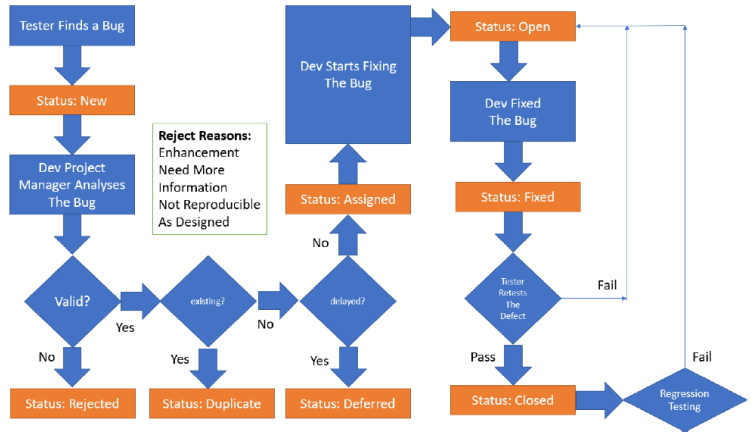
**Globalization Testing:**

* Performed to ensure the system or software application can run in any cultural local environment
* Different aspects o the software application are tested to ensure that it support every language and different attributes
* It tests the different currency formats, mobile number formats and address format and address format are supported by the application
* For example Facebook.com supports many of the language an it can be accessed by people of different countries Hence it is a globalized product.

**Localization Testing:**

* Performed to ensure the system or software application can run in any cultural or environmental
* Different aspects of the software application are tested to ensure that it supports every language and different attributes.
* It tests the specific currency formats; mobile number format and address format is working properly or not
* For example, Baidu.com supports only the Chinese language and can be accessed only by people of few countries, Hence it is a localized product.

**Bug Life Cycle**

****

**Black Box Testing:**

* It is way of software testing in which the internal structure of the program or the code is hidden and nothing is known about it.
* It is mostly done by software testers.
* No knowledge of implementation is needed.
* It can be referred as outer or external software testing.
* It is functional test of the software (Black box)
* This testing can be initiated based on requirement specification document.
* No knowledge of programming is required.
* It is the behavior testing of the software.
* It is applicable to the higher levels of testing of software.
* It is also called closed testing
* It is least time consuming
* It is not suitable or preferred for algorithm testing.
* Can be done by trial and error ways and methods

**Example:** search something on google by using keywords.

**White Box Testing:**

* It is way of testing the software in which the testers has knowledge about the internal structure or the code or the program of the software.
* It is mostly done by software developers.
* Knowledge of implementation is required.
* It is the inner or the internal software testing.
* It is structural test of the software.
* This type of testing of software is started after detail design document
* It is mandatory to have knowledge of programming
* It is the logic testing of the software.
* It is generally applicable to the lower levels of software testing.
* It is also called as clear box testing.
* It is most time consuming
* It is suitable for algorithm testing
* Data domains along with inner or internal boundaries can be better tested.

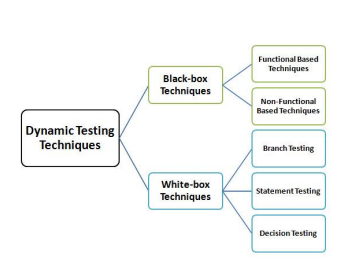
**Example:** by input to check and verify loops

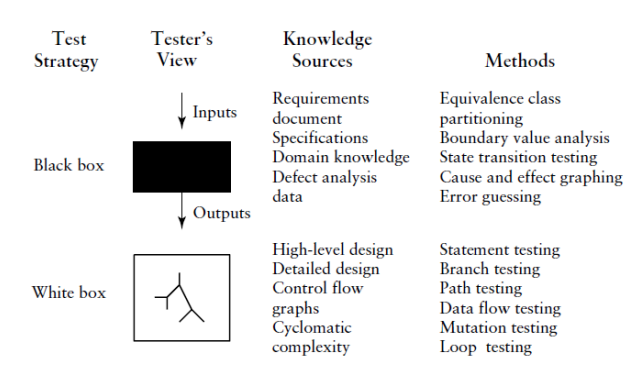
**Black Box Testing:** Types of Black Box Testing

* A. Functional Testing
* B. Non-Functional Testing
* C. Regression Testing

**White Box Testing:** Types of white box testing

* A. Path Testing
* B. Loop Testing
* C. Condition Testing

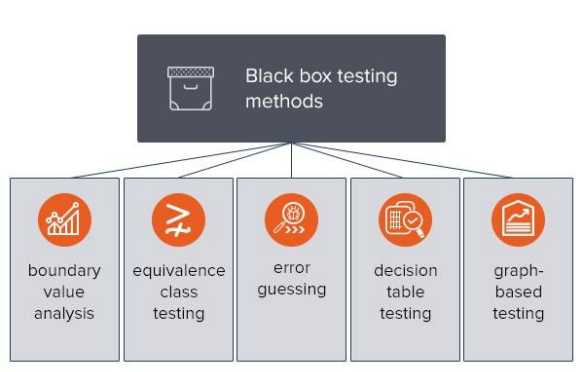




The two basic testing strategies.

Black box testing == data driven testing = input/output testing

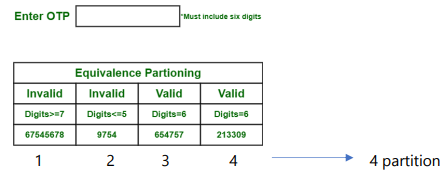
White box testing = = logic driven testing = Logic Coverage Testing



**BVA –** Boundary value analysis based on verification of only extreme boundary values e.g. maximum, minimum and typical (E.g. lower boundary 18 higher 56)

****

**ECP =** Equivalence class testing based on checking one value from each partition

****

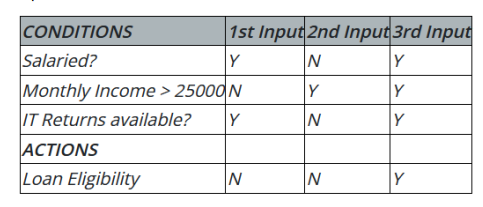
**Error Guessing :** based on the previous experience of a QA engineer also called experience based testing.

Error guessing is a testing technique that makes use of a tester’s skill, intuition and experience in testing similar application to identify defects that may not be easy to capture by the more formal techniques. It is usually done after more formal techniques are completed.

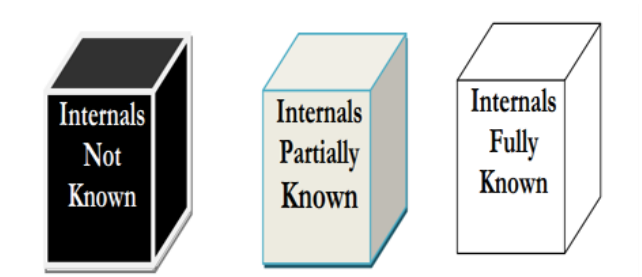
Examples:

* Divide by zero
* Entering blank spaces in the text fields
* Pressing submit button without entering values.
* Uploading files exceeding maximum limits.

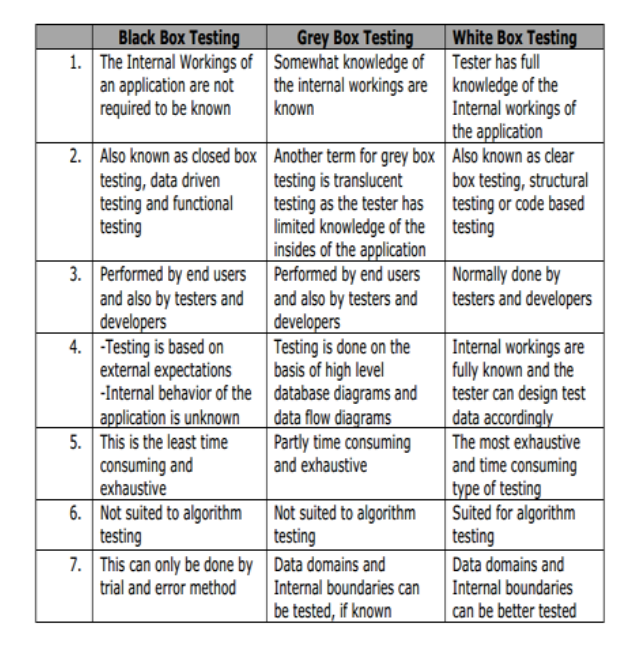
Decision table: testing based on a tabular representation of combinations of inputs and correspondent system behavior depends on condition



**Graph based**: Testing Where a test case is written for each graph that represents the object



**Comparison between the Three Testing Types**:



**White box testing :** is an approach that allows testers to inspect and verify the inner workings of a software system its code, infrastructure , and integration with external systems

**White box Testing Techniques:** They are many ways you can analyze software with white box testing. Most testers will use a process called code coverage analysis to eliminate gaps in the testing of the code. There are a variety of techniques you can use to accomplish this, including:

**Statement coverage:** This technique ensures that each line in the code is tested at least once to find faulty code more easily.

**Branch coverage:** Using this technique , each possible path or decision point of a software application is checked for accuracy

**Condition coverage:** All individual condition are checked.

**Multiple condition coverage :** All imaginable combination of all the conceivable condition outcomes are tested at least once.

**Basic path testing:** Control graphs are created from either flowcharts or code Cyclomatic complexity is then calculated to define the number of independent paths so that the minimum number of test cases can be designed for each path

**Flow chart notation:** This technique uses a directed graph made up of nodes and edges, where each node represents a decision point or sequence of statements.

**Cyclomatic complexity:** This is the measure of a software’s logical and cyclomatic complexity. It is used to define how many independent paths are present.

**Loop Testing:** Loops are commonly used in white box testing and are fundamental to many algorithms. Problems are often found at the beginning or the end of a loop. Loops testing can be divided into simple loops, nested loops and concatenated loops.