

**Exploratory Testing:**

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

**Adhoc testing:**

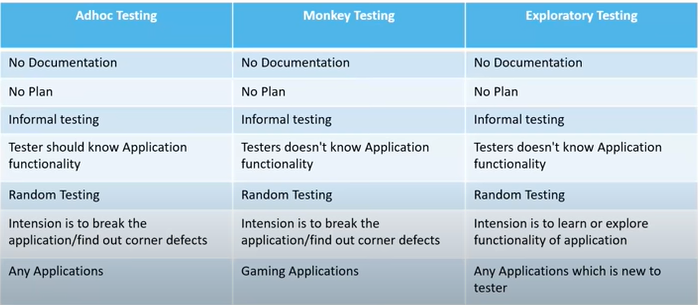
* Testing application randomly without any test cases or business requirement document.
* Adhoc testing is an informal testing type with an aim to break the system.
* Tester should have knowledge of the application even though he doesn’t have test cases/ requirements.
* This testing is usually an unplanned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



**Monkey/ Gorilla testing:**

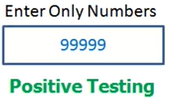
* Testing the applications randomly without any test cases or any business requirements.
* It is also an informal testing type with an to break the system.
* Tester do not have knowledge of the application.
* Suitable for gaming application.

**Adhoc testing vs monkey testing vs exploratory testing:**

****

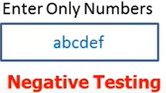
**Positive testing:**

* Testing the application with valid inputs is called as positive testing.
* It checks whether the application behaves as expected with positive inputs.

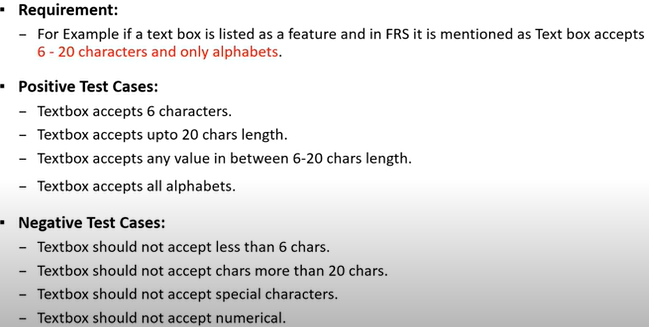


**Negative testing:**

* testing the application with invalid inputs is called negative testing.
* It checks whether an application behaves as expected with negative inputs.

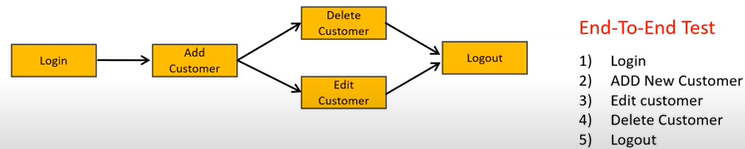
****

**Example:**

****

**END – to – END testing:**

Testing all overall functionalities of the system including the data integration among all the modules is called end-to-end testing.



**Globalization and localization testing:**

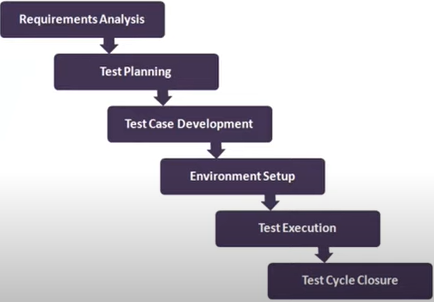
* Performed to ensure the system or s/w application can run in any cultural or local environment.
* This testing is performed on the application which supports globally like in many countries such as facebook it is accessed by people in different countries.
* It tests that it supports every language or not , different currency formats, different number format, address format by the application.

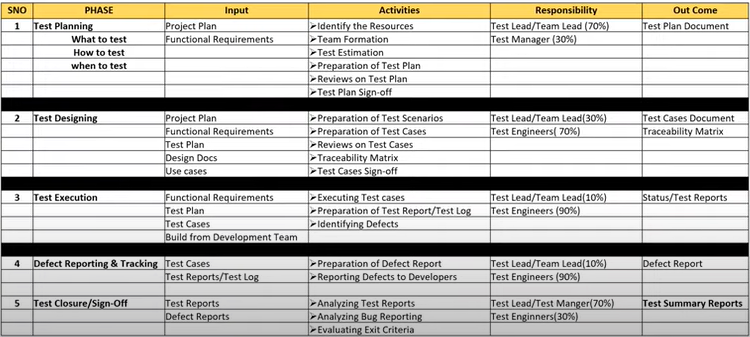
**Localization testing:**

* Performed to check system or s/w application for a specific geographical and cultural environment.
* Localized product only supports the specific kind of language and is usable only in specific region.
* It tests the specific currency format, mobile number format, 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 localized product.

**STLC (Software testing life cycle):**

1. Requirement analysis
2. Test planning
3. Test designing
4. Test execution
5. Defect or bug reporting or tracking
6. Test closure

****

****

**Test plan contents:**

A test plan is a document that describes the test scope, test strategy, objectives, schedules, deliverables, and resources required to perform testing for the s/w product.



**Use case, test scenario and test case:**

**Use case:**

* Use case describes the requirement.
* Use case contain three items.

**Actor,** which is the user, which can be a single person or a group of people, interacting with a process.

**Action,** which is to reach the final outcome

**Goal/ outcome,** which is successful user outcome.

**Test Scenario:**

A possible area to be tested. (what to test)

**Test Case:**

* Step by step action to be performed to validate the functionality of AUT. (how to test)
* Test case contain test steps, expected result and actual result.

**Use case vs test case:**

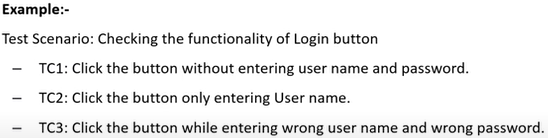
**Use case:** describes functional requirement, prepared by business analyst

**Test case:** describes test steps/ procedures, prepared by test engineer.

A test case is a set of actions executed to validate particular feature or functionality of your s/w application.

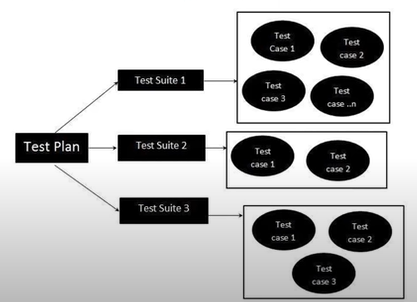
**Test scenario vs test case:**

**Test scenario** is what is to be tested **and test case**  is how to be tested.



**Test suite:**

Test suit is group of test cases which belongs to same category.

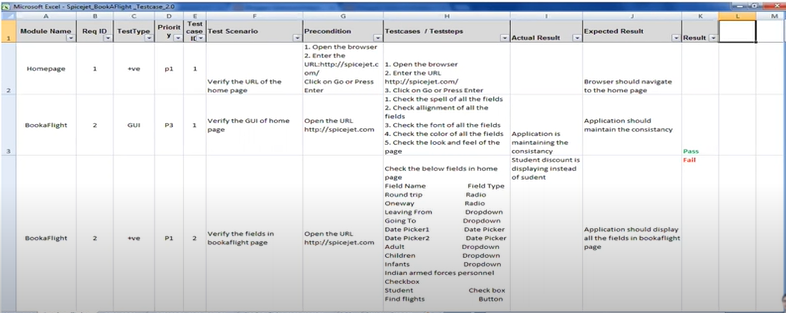


**Test case contents:**

1. test case ID 2) test case title 3) Description 4) pre condition

**5)**priority ( p0,p1,p2,p3) – order 6) requirement id 7) steps/action

8) expected result 9) actual result 10) test data



**Requirement traceability matrix: (RTM)**

RTM describes the mapping of requirements with the test cases.

The main purpose of RTM is to see that all test cases are covered so that no functionality is miss while doing the software testing.

Requirements traceability matrix – parameters include

* requirement ID
* requirement description
* test case ID

**Test environment:**

* test environment is a platform specially build for test case execution on the s/w product.
* It is created by integrating the required s/w and h/w along with proper network configuration.
* Test environment simulates production/ real time environment
* Another name of test environment is **test bed.**

**Test execution:**

During this phase test team will carry out the testing based on the test plans and the test cases are prepared.

**Entry criteria:** test cases, test data, test plan

**Activities:**

* Test cases are executed based on test planning
* Status of test cases are marked like, failed, passed, block, run, and others.
* Documentation of the test results and log defects for failed cases is done.
* All the blocked and failed test cases are assigned bug id’s.
* Retesting once the defects are fixed.
* Defects are tracked till closure.

**Deliverables:**

Provides defects and test case execution report with completed results.

**Guidelines for test case execution:**

* The build being deployed to the QA environment Is the most important part of the test execution cycle.
* Test execution is done in QA environment
* Test execution happened in multiple cycles
* Test execution phase consists of executing the test case + test scripts

**Defects / Bugs:**

* Any mismatch functionality found in an application is called as defect/bug/issue.
* During test execution test engineer are reporting mismatches as defects to the developers through templates or using tools.
* **Defect reporting tools:**

Clear quest

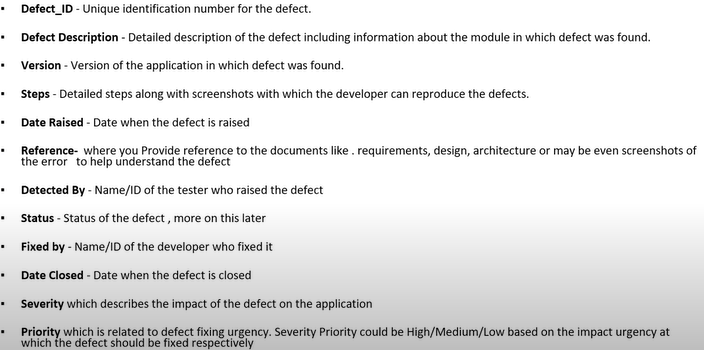
DevTracker

Jira

Quality centre

Bug Jilla

**Defect report contains:**

****

****

**Defect severity:**

Severity describes the seriousness of defect and how much impact on Business workflow.

**It can be categorized into four classes:**

**Blocked (show stopper):** this defect indicates that nothing can proceed further.

EX: application crashed or login not working

**Critical:** the main/ basic functionality is not working. Customer business workflow is broken.

They cannot proceed further.

EX: fund transfer is not working in net banking.

**Major:** it cause some undesirable behaviour, but the feature/ application is still functional.

EX: after sending mail there is no confirm message.

**Minor:** it won’t cause any major breakdown of the system.

Ex: look and feel issues, spelling mistakes, alignment.

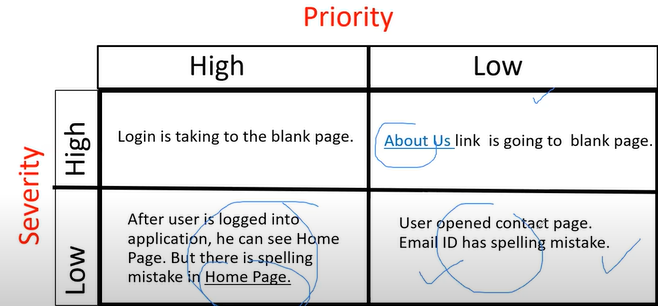
**Defect Priority:**

Priority describes the importance of the defect.

Defect priority states the order in which the defect should be fixed

**Defect priority can be categorized into three classes**

* P0 (High) : the defect must be resolved immediately as it affects the system severely and it cannot be used until it fixed.
* P1 ( Medium) : it can wait until new version/ build is created
* P2 (Low) : developer can release it in later releases

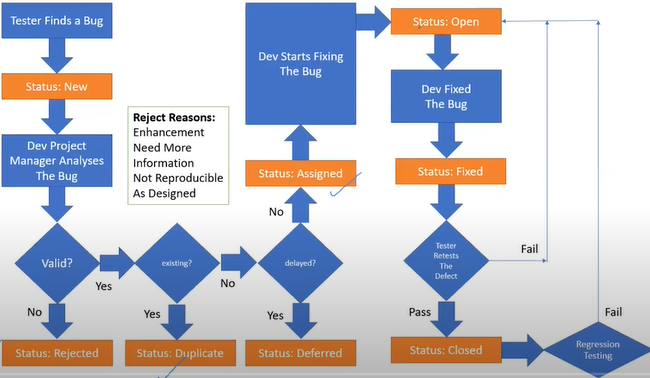


**Defect Resolution:**

after receiving the defect from the testing team, development team will conduct a review to fix the defect. Then they the resolution type to testing team for further communication.

**Resolution types:**

* Accept \*Reject
* Duplicate -- > that already bug has raised by testing team
* Enhancement --- > stating that it is new feature will be implemented in next versions
* Need more info \* Fixed
* Not reproducible -- > not getting error in dev environment
* As designed -- > stating not bug and working as per designed

**Bug Life Cycle:**

**Test cycle closure:**

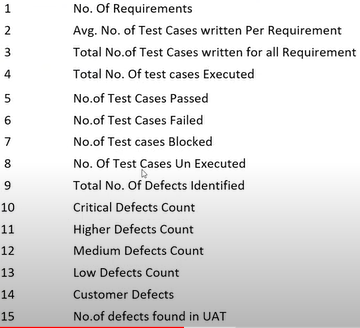
**Activities:**

* Evaluate cycle completion criteria based on time, test coverage, cost, s/w, critical business objectives, quality.
* Prepare test metrics based on above parameters.
* Document the learning out of the project
* Prepare test summary report
* Qualitative and quantitative reporting of quality of the work product to the customer.
* Test result analysis to find out the defect distribution by type and severity

**Deliverables:**

* Test closure report
* Test metrics

**Test metric required data:**

****

**% of test case executed:**

(No. of test cases executed/ total no. of test cases written) \* 100

**% of test case NOT executed:**

(No. of test cases NOT executed/ total no. of test cases written) \* 100

**% of test cases passed:**

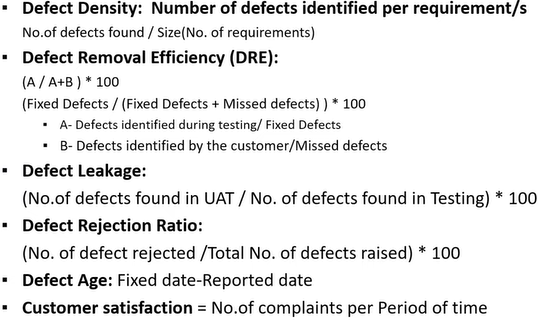
(No. of test cases passed/ total no. of test cases executed) \* 100

**% of test cases failed:**

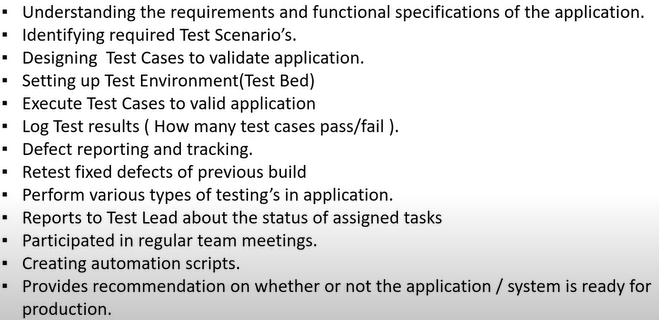
(No. of test cases failed/ total no. of test cases executed) \* 100

**% of test cases blocked:**

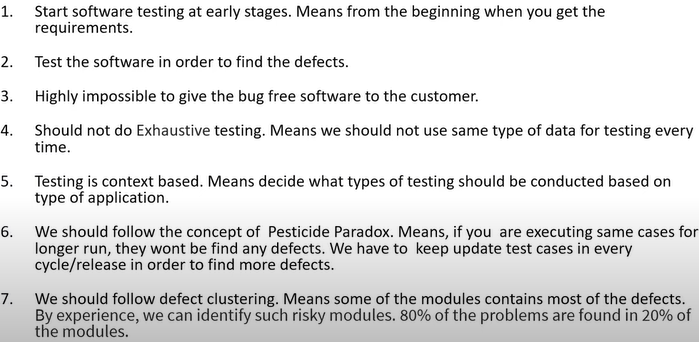
(No. of test cases blocked/ total no. of test cases executed) \* 100



**QA testing activities:**

****

**Principles of software testing:**

****