**🡺WHITE BOX TESTING: Whatever testing is done in code level is called White Box Testing. Unit Testing and Integration Testing. Programming skills is required.**

**🡺BLACK BOX TESTING: Testing conducts on functionality of the application whether it is working according to customer requirements or not. System Testing and UAT.**

**🡺INTEGRATION TESTING: Performed between 2 or more modules. Focuses on checking data communication between multiple modules. Incremental and Non Incremental Integration Testing.**

**🡺SYSTEM TESTING (end to end testing or after completion of integration testing): Actual area where tester works. We don’t understand the code because we are getting the complete application. We just need to understand the requirement whether it is satisfied or not according to the customer. Testing overall functionality of the application with respective client requirements. System Testing focuses on:**

* **User Interface Testing (Alpha Testing & Beta Testing). After completion of system testing**
* **Functional Testing and Non-Functional Testing**
* **Usability Testing**

**🡺 UAT TESTING (Alpha Testing & Beta Testing). After completion of system testing UAT team conducts acceptance testing in two levels.**

* **Alpha Testing: The users and customer will do the testing in develop/testing environment. They will comeback to the company where software is developed and do some testing.**
* **Beta Testing: After installation of software customer will do some testing in their environment.**

**🡺 GUI Testing: Testing the user interface of an application such as menus, checkbox, buttons, colors, fonts, sizes, icons, content and alignment images, text box, spelling check, font is readable or not, color of the hyperlink.**

**🡺USABILITY Testing: Suppose customer buy a mobile and with the mobile phone there is a user guideline how to operate the phone. This is important for the customer. So, as a tester I need to check the all of user guideline.**

**🡺FUNCTIONAL Testing: Is nothing but testing the behavior of the application. How application feature should work. Concentrates on user REQUIREMENT/EXPECTATION**

**🡺NON FUNCTIONAL Testing: Once the application is stable then we do Non-Functional testing.**

|  |  |
| --- | --- |
| **Object Properties Testing**  **Database Testing**  **Error Handling**  **Calculations/Manipulation Testing**  **Link Existence & Link Execution**  **Cookies & Sessions** | **Performance Testing**  **Security Testing**  **Recovery Testing**  **Compatibility Testing**  **Configuration Testing**  **Installation/Sanitation/Garbage** |

**🡺 SMOKE TESTING VS SANITY TESTING: Come into the picture after build release.**

|  |  |
| --- | --- |
| **Is done to make sure the build we received from the development team is testable/stable or not. Performed by both Dev & Tester.** | **Is done during the release phase to check for the main functionalities of the application without going deeper. Done by Tester alone.** |
| **Build may be either stable or unstable. It is done on initial builds. It’s a part of basic testing. It is done every time there is a new build release.** | **Build is comparatively stable. It is done on stable builds. It is part of regression testing. It is planned when there is no enough time to do in depth testing.** |

**🡺ADHOC—No documentation, No Plan, No Test Case. Informal/Random Testing.**

**Tester should know the functionality of the application. For any Application.**

**🡺 TEST DESIGN TECHINQUE: Reduce the number of test cases to be executed.**

* **Equivalence Class Partitioning-Black box testing-**
* **Boundary Value Analysis**
* **Decision Table Based-If we have more conditions**
* **State Transition**
* **Error Guessing—No specific rules. Depends on Tester analytics skills**

**🡺TEST PLAN: In test plan we are going to specify everything. What we are going to test, what we are not going to test. What is the scope, what is not in our scope? What are the different strategy we are going to use, what are different testing type we are going to use, when we have to conduct the test that means schedule, who will conduct the testing, in which environment we will conduct the testing like hardware environment? So, all we have to specify in the document as called Test Plan document.**

**🡺 Overview, Scope, Test Strategy, Defect Reporting Procedure, Roles/ Responsibilities, Test Schedule, Test Deliverables, Pricing, Entry and Exit Criteria, Suspension and Resumption, Tools, Risks and Mitigations, Approvals.**

**🡺TEST STRATEGY: What type of testing we are going to conduct? Like smoke, sanity, functional, regression or like automation or manual these are strategy.**

**🡺TEST DELIVERABLES: Each and every phase of a testing we are going to deliver a set of documents. Like Test Plan documents is one deliverable. Similarly Test Case, Defect Report, Test Execution Report is also deliverable. So, after completion of every step we are going to deliver a small document, this should be mentioned in Test Deliverables.**

**🡺Entry and Exit Criteria: Its prerequisite conditions that must be completed before start testing. Similarly, we need to full fill some conditions before conclude testing.**

**🡺 SUSPENSION AND RESUMPTION: Sometimes, suppose immediately will stop testing. Something is not working; something is broken in apps. Okay we will stop testing and then immediately after recovery, after getting new build, after getting the fix we will resume our testing. Exactly in which cases we need to stop the testing and which case we need to resume the testing that is specified in the Test Plan.**

**🡺USE CASE: Describe the functional requirements. Prepared by BA.**

**🡺TEST SCENARIO: A possible area to be tested (what to test).**

**🡺TEST CASE: is a set of action to validate the functionality of AUT. Test case contains: Test Case ID, Test Case Title, Description, Pre-condition, Priority, Requirement ID, Steps/Actions, Actual Result, Test Data.**

**🡺TEST ENVIRONMENT/TEST BED: is a platform specially build for test case execution on the software product.**

**🡺 TEST EXECUTION: During this phase test team will carry out the testing based on the test plans and test cases prepared. Entry Criteria: Test Case, Test Plan, Test Data.**

* **Test cases are executed based on the test planning.**
* **Status of the test cases are marked like passed, failed, blocked**
* **Documentation of test results and log defects for failed cases is done.**
* **Retesting once the defect is fixed and defects are tracked till closure.**

**🡺TEST EXECUTION GUIDELINE: Build must be run in QA environment. Test execution happen in multiple cycles. Test execution consists execution the Test Cases + Test Scripts for automation.**

**🡺 REQUIREMENT TRACEABILITY MATRIX: All the test cases are covered or not as per requirements. REQUIREMENT\_ID, REQUIREMENT DESCRIPTION, TEST CASE ID. STATUS.**

**🡺TEST METRICS: No. of Requirements, Avg. No. of Test Case Written Per Requirement, Total No. of Test Cases Written for all Requirement, Total No. of Test Case Executed, No. of Test Case Passed/Failed/Blocked/Unexecuted, Total No. of Defects Identified, Critical/Higher/Medium/Low Defects Counts, Customer Defects, No. of defects found in UAT.**

**🡺 QA/TESTING ACTIVITIES:**

* **Understanding the requirements and identify the Test Scenario’s. Designing Test Cases to validate the app.**
* **Setting up Testing Env and execute Test Cases to valid app. Create Log Test Results (How many pass/failed)**
* **Defect Reporting & Tracking. Retest fixed defects of previous build. Perform various types of testing’s and Report to Test Lead about the status of the assigned tasks. Create automation scripts.**

**🡺PRINCIPAL OF SOFTWARE: Test the software in order to find the defects.**

* **Highly impossible to give the bug free software to the customer. We should not use same type of data for every testing. Testing is context based. Fixed what type of testing should be conducted based on type of application.**
* **We should follow the concept of Pesticide Paradox. Means, if you are executing same cases for longer run, they won’t be finding any defects. We have to keep updated test cases in every cycle/release in order to find more defects.**
* **We should follow the defect clustering. Means some of the modules contains most of the defects. By experience, we can identify such risky modules. 80% of the problems are found in 20% of the modules.**

**🡺VERIFCATION: Checks whether we are building the RIGHT PRODUCT? Focus on documentation. Verification typically involves –Review---Walkthrough---Inspections.**

**🡺VALIDATION: Checks whether we are building the PRODUCT RIGHT? After completion of verifications, we start validation. Focus on SOFTWARE. Validation typically involves with actual testing like UNIT, Integration, System, UAT.**

**🡺DEFECTS/BUGS/ISSUES: Mismatch found in a functionality in an application. Bug Jilla, Quality Center, Clear Quest.**

**🡺SEVERITY: Impact of the defect of the application. How much it will impact on client business?**

* **BLOCKER/STOPPER: This defect indicates nothing can proceed further. Login not worked.**
* **CRITICAL: The main/base functionality is not working. Customer business workflow is broken.**
* **Fund transfer is not working, ordering product is not working in ecommerce website.**
* **MAJOR: Arise some unexpected behavior. After booking cab there is no confirmation**
* **MINOR: Spelling mistakes, alignments.**

**🡺PRIORITY: Describes the importance of the defect. How soon it should be fixed. High, Medium, Low**

* **P0(High): The defect must be resolve immediately as it affects the system severely and can’t be used until fixed.**

***HIGH SEVERETY HIGH PRIORITY*: Login is taking to the blank page.**

***HIGH SEVERETY LOW PRIORITY:* About us link is going to the blank page.**

**LOW SEVERETY HIGH PRIORITY: After user is logged into application, he can see HOME page. But there is spelling mistake in HOME PAGE.**

***LOW SEVERETY LOW PRIORITY:* user opened contact page. Email id is spelling mistake.**

**RESOLUTION TYPES: Accept, Reject, Duplicate, Enhancement, Need More Information, Not Reproduceable, As Designed.**

**DEFECT REPORT: Defect\_ID, Defect Description, Version, Steps, Date Raised, Reference, Detected By, Status, Fixed By, Date Closed, Severity, Priority.**