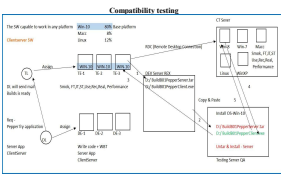
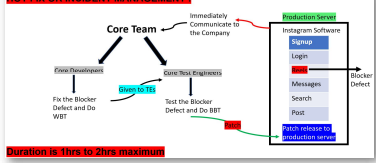


## BLACK BOX TESTING / CLOSED BOX TESTING / BEHAVIOURAL TESTING / FUNCTIONALITY TESTING

Difference	White Box testing	Functional testing	Integration testing	System testing	Acceptance testing	Smoke testing	Adhoc testing	Exploratory testing	Usability testing	Performance testing	Globalization testing		Compatibility testing	Regression testing	Reliability Testing	Recovery testing
What	Testing each and every line of the code is called White Box testing.	Testing each and every components of an application thoroughly/Thoroughly against Customer Requirement Specification is called as Functional Testing /Component Testing.	Testing the data flow or interface between two or more modules/Features is called as Integration Testing.	It is an End to End testing done by the Test Engineer in Testing Server/Environment which is similar to the Production Server Environment.	It is an End to End Testing done by the EndUsers Customers where they use Software for real time business for some particular period of time and check whether it can handle all real time situations and scenarios.	Testing the Basic and Critical Features of an application before doing Thorough/Regression Testing is called as Smoke Testing.	Testing the application or software randomly without looking into the requirement is called as Adhoc testing.	Understand the application, identify all possible scenarios, document the scenarios & test the application by referring the document is called Exploratory Testing.	Testing the User Friendliness of an application is called Usability Testing.	Testing the Stability and Response Time of an application by applying load is called Performance Testing.	IBSN Testing	LIJN Testing	Testing the functionality of an application /SW for different hardware & SW configurations is called as Compatibility testing.	Testing old or unchanged features to make sure that changes like adding a feature, removing a feature, modifying a feature and fixing defects is not introducing any defects in old or unchanged features is called Regression testing.	Testing the functionality of an application continuously on application continuously for a particular period of time is called Reliability testing.	Testing the functionality of an application to check how well the SW recovers from crashes or failure is called as Recovery testing.
											Testing the SW which is developed for different languages in called Globalization Testing.	Testing the sw / application if it is developed for country standards / country culture is called Localization testing.				
When	1) Requirement should be present. 2) Coding should be done.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional Testing should be done. 8)Integration Test Scenarios and Test cases should be ready.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional Testing should be done. 8)Integration Test Scenarios and Test cases should be ready.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)When minimum bunch of modules are ready 7)Smoke Testing should be done. 8)Functional Testing should be done. 9)Integration testing should be done. 10)Testing server similar to production server setup should be available. 11)When we start getting less number of requirement 12)Basic functionalities of all the modules should be working fine 13)Product/SW should be relatively stable - no blocker / Critical defect 14)System test Scenarios and Test cases should be ready.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional,Integration, System, Usability,Reliability, Recovery, Global, Performance testing should be done. 8)Once after the SW quality is found good after testing for number of cycles.	1)As soon as we get the build we(Te) should do Smoke Testing. 2)When Developer give the build to the Customer,there are three developer might need to copy few of the files as customer will do smoke testing to check whether files are copied properly and build is as installed properly or not. 3)Business Engineer /Build engineer will do smoke testings check whether Build is as installed properly into testing server and production server. 4)Developer will do Smoke Testing after doing WBT before giving build on to Test Engineer.	1)As soon as we get the build we should do smoke testing, since smoke is positive testing here we should not do testing/Adhoc. 2)While doing PT/IT/ST either in or at the end, if you have some time you can do Adhoc testing. Suppose if you don't have time document the adhoc scenarios and communicate to the TL. 3)Once after complete SW is build as per the requirement then we can do Adhoc testing (i.e PT / IT / ST) 4)Once after SW is tested for 10-15 days test cycles and if the product / SW is stable then we will start doing adhoc testing (i.e PT / IT / ST) 5)While doing PT / IT / ST if you come up with good adhoc scenarios stop doing PT / IT / ST and test for adhoc scenarios (i.e PT / IT / ST) but don't spend too much of time in doing adhoc testing immediately switch back to PT / IT / ST	1) When there is no requirement 2) There is requirement but TE doesn't have time to understand the req 3) Req is present but it's very complex to understand.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional,Integration, System testing should be done. 8)Usability test Scenarios and Test cases should be ready.	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional,Integration, System testing should be done. 8)If the software is developed for different languages	1)Requirement should be present. 2)Coding should be done. 3)WBT should be done. 4)sw/Build should be installed into server. 5)Resources should be available. 6)Smoke Testing should be done. 7)Functional,Integration, System testing should be done. 8)If the software is developed for country standards / country culture	Once after the SW is tested for the base platform the SW can be tested for Compatibility				
Why	To check if each and every line of code is working as expected or not	To check whether each and every individual component is working as per the customer requirement specification.	To check whether the data flow between one or more modules is working as per the customer requirement specification.	To check when we navigate through all the features & check whether if the end feature is working as expected or not.	1. To get confidence on the product/software. 2. To ensure that product meets the business requirement 3. To make sure that s/w company is not developing new features. 4.Chances are there because of business pressure company might miss some defects.	1. To check that whether s/w is build as install or not. 2. First day build while doing smoke testing should be done. 3. To ensure that product meets the business requirement 4. To make sure that s/w company is not developing new features. 5. When we are doing smoke testing indirectly, we are ensuring build is installed properly or not. 6. Developer is giving new build means he will be doing some code changes and this might affect old basic and critical features, so we should do adhoc testing. 7. This is a negative testing, so adhoc testing has to be done. 8. Requirement are not followed, so we should adhoc testing. 9. The intention is to break the SW as we should do adhoc testing. (Find max number of defects)	1.Chances are there customers or end users use the SW randomly and they might find defects in order to avoid that adhoc testing should be done. 2. Developer & Te will look into the req and develop & test the SW, chances of finding the defects are less in this case, so TE will come up with creative scenarios to find more defects. 3. This is a negative testing, so adhoc testing has to be done. 4. Requirement are not followed, so we should adhoc testing. 5. The intention is to break the SW as we should do adhoc testing. (Find max number of defects)	If we do exploratory testing when there is no requirement we can find all blocker & critical defects and we might miss some minor defects and when software is released to the customer. Our customer can use the SW and run the business without facing any blocker problem. So when there is no requirement it is always best to do ET instead of releasing the SW to the customer without testing	To check whether SW or application is user friendly.	To check, how is the response time and stability of an application when load is applied.	1. To check whether right language is displayed 2. To check whether right content is displayed in right place.	1. To check whether right date format is displayed 2. To check whether right currency format is displayed 3. To check whether right Postal code is displayed 4. To check whether right phone format is displayed (National flag code)	1. TE will have tested the SW in one platform, chances are there the Users will use the SW in different platforms & the SW might not work. When will spread load more & customer usage will go down, in order to avoid this we should do Compatibility testing 2. To ensure all the features is consistently working for all the platforms we will do Compatibility testing			
How	1. By writing flow chart 2. By using 'Test & Take' condition in program 3. By using any programming tool 4. By using automatic tool like a highlighted with dotted line 5. By comparing data memory 6. By improving performance of codes	Ex. Mr. Excel 1. When user enter click on build, test should be displayed in build 2. To check that when user click on a cell and click on copy the cell should be highlighted with dotted line. 3. To check that when user select two cells and click on merge and centerize cell should be displayed.	Ex. Gmail 1. When user select Emails and click on delete and click on trash bin should be displayed. 2. When user click on compose and mail to 2nd user and click on sent, deleted mail should be displayed. 3. When user click on compose a mail to 2nd user and click on back button, new click on drafts mail should be displayed.	Ex. EPFO 1. When user enters URL:EPFO.GOV.IN and welcome page should be displayed and login page should display, enter (UAN) and Password and login home should be displayed.	1) Acc to the Case/SPs, Test engineers of company moved to the Customer place and did Acceptance Testing/ demand of customer. 2) To ensure that product meets the business requirement 3. To make sure that s/w company is not developing new features. 4.Chances are there because of business pressure company might miss some defects.	Ex. SBIN 1. To check that when user enter welcome page should be displayed. 2. To check that when user click on copy, input page should display, enter (UAN) and Password and login home should be displayed.	1. I will understand the application and I will try to enter all possible inputs to each & every components and I will do IT 2. I will understand the application & check data then how the module by doing ET 3. I will explore the app & I will understand how every feature works and I will test all possible and to end scenarios by doing exploratory testing.	Ex. ESC.GOV.IN 1) Mobile number field is a mandatory field but there is no asterisk. 2) When user enters today date & DOB Test field not displaying proper error message. 3) When button is enabled without entering any data in any components.	Using performance testing tool (VU Gen) 1. Click on start recording 2. The recording will be started (it will capture each & every actions we perform. 3) Open the browser, enter url, UIN & Pin and click on login button. 4. Click on stop recording. 5. The tool will automatically write the code for actions, enter the url and number of users into the tool and click	1. Right language 2. Right content in right place 3. Alignment (Subtotal) / Bold/italic / Text tip	1. Currency format 2. Date format 3. Pin code format 4. Image format	Ref. Diagram below	Testing old or unchanged features to make sure that changes like adding a feature, removing a feature, modifying a feature and fixing defects is not introducing any defects in old or unchanged features is called Regression testing			
Types	1) Path testing 2) Cyclomatic Complexity 3) Conditional testing 4) Predicate test 5) Loop testing 6) GUI testing 7) Testing from memory point of view 8) Testing from performance point of view	1)Over testing 2) Shadow Test 3) Optimistic testing 4) VE Testing 5) Non incremental Integration testing 6) Bug hang theory	1) Incremental Integration testing a) Top down incremental Integration testing b) Bottom up incremental Integration testing 2) Non incremental Integration testing 3) Bug hang theory	1) End to End testing 2) Under testing 3) Usability testing 4) Performance testing 5) Compatibility testing 6) Adhoc testing 7) Regression testing 8) Smoke testing	1) User Acceptance testing 2) Operational Acceptance testing 3) Contact Acceptance testing 4) Compliance Acceptance testing 5) Beta testing	1) Formal Smoke testing 2) Informal Smoke testing	1) Buddy testing - Dev & TE 2) Pair testing - TE & TE 3) Monkey testing - TE like monkey	1) Yellow Box testing 2) GUI testing 3) Graphical User Interface 4) Accessibility Testing 5) American Disability act/ADA	1) Usability testing 2) Stress testing 3) Scalability testing 4) Volume testing 5) SaaS testing 6) Endurance Testing	1) Internationalization testing (IBSN Testing) 2) Localization testing (LIJN Testing)			1) Hardware compatibility testing 2) Software compatibility testing 3) Full Regression Testing	1) User Regression testing 2) Regional Regression testing 3) Full Regression Testing		
Alternative names	1. Open box testing 2. Glass box testing 3. Clear Box testing 4. Transparent testing 5. Validation testing 6. Structural testing 7. Unit Testing	Component Testing			* Hot Fix/Incident or Defect Management * Short release/Interim release	1) Sanity Testing 2) Dryrun Testing 3) Skin Testing 4) Build verification Testing 5) Health check of a product Testing 6) Confident Testing 7) Source Testing	1) Monkey Testing 2) Gorilla Testing 3) Out of the box Testing 4) Negative Testing		1) Baseline testing 2) Break point testing 3) Spike testing 4) Bottle neck testing 5) Threshold testing	Localization testing (LIJN Testings-Format testing)	Possibility Testing	Release Candidate Testing				



### HOT FIX OR INCIDENT MANAGEMENT



How can application will be implemented and tested in a User Architecture.

1. User Architecture

