Manual Testing Notes By SURESH

Mr. Suresh – Profile

- Currently working as a Consultant with one of the Top MNC and has 13 + years of genuine experience in Software Testing.
- Hands-on experience with Selenium since 2009 and has 6 years of in-house Corporate Training experience.
- Successfully trained 90+ batches on Selenium
- Expert in Selenium Automation using Selenium RC and WebDriver. Heavy Programming using Java, TestNG and worked with QTP and Ruby with Watir as well.
- Proficient in Planning, Designing, Building and maintaining complex automation frameworks (Keyword, Modular, Hybrid and POM).
- Pro-active Team Leader and Manager with strong focus on documentation and process.
- Rich industry experience as worked with **Symphony Services, Accenture, IGATE and Capgemini.**

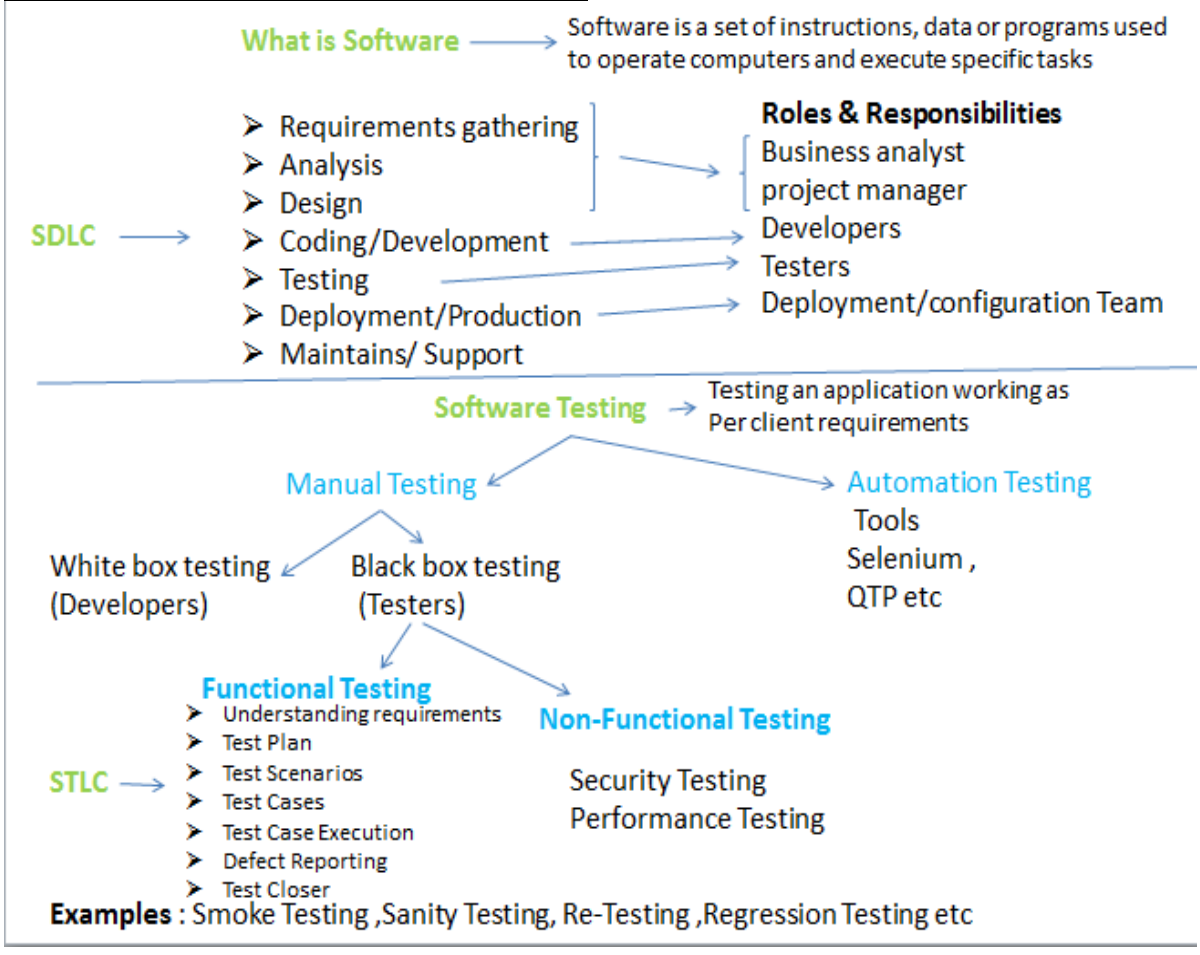
Specializations:

- Expertise in creating Test Automation Frameworks for open source automation tools.
- Competence in open source automation APIs like Selenium RC and Web Driver.
- Proficiency in conducting Classroom, Corporate and Online Training on Selenium.

Testing Course Agenda

Module 1 : Testing Fundamentals
Module 2 : Testing Roles and Responsibilities
Module 3 : SDLC & STLC Models
Module 4 : Test Management Team
Module 5 : Test Engineer Responsibilities
Module 6 : Defect Management
Module 7 : Status Reports
Module 8 : Automation

Overview of Software Testing



Why Testing Required

- To identify the defects in development phases
- To ensure Quality of the product
- Saves Money as defect identified in earlier stages
- To build customer confidence and business

Why Testing Job

- Software Testers Are Made for Challenging Work Environments
- You Can Enjoy Every Day of Work
- Flexible and Fun Work Environment
- It's Creative
- It Is a Secure Career Path
- There Is Attractive Remuneration and Room for Growth
- An Academic Background Isn't a Necessity

Testing Roles & Responsibilities

- Software Test Engineer(STE) – (0-4yrs)
- Sr. Software Test Engineer(SSTE) – (4-6yrs)
- Test Lead (TL) – (6-8yrs)
- Test Manager(TM) – (8-10yrs)
- Sr.Test Manager(STM) – (10+yrs)
- Assoc. Dir. – Software Testing
- VP – Software Testing

Software Test Engineer Responsibilities

- Understating the requirements of the application
- Identifying required Test Scenarios of the project
- Designing and preparing Test cases to validate application
- Execute test cases to validate application
- Logs Test Results(How many Test cases passes/failed)
- Defect Report and Tracking
- Retest fixed defects of previous builds
- Performed various Types of testing assigned by Test Lead(Sanity ,Functionality , Usability, Compatibility , etc)
- Preparing and Sending of status Reports to Lead on assigned tasks
- Participated in regular meetings , team meetings by lead & Manager
- Creating automation scripts for Regression testing

Sr. Software Test Engineer Responsibilities

- Same as test engineer responsibilities

+

- Participates in Review of Test Scenarios ,Test cases and defects
- Some Times involved in Test Plan preparation also.
- If required Leading the team when Team Lead is on Vacation

Test Lead Responsibilities

- Task Preparation and allocation to Team members
- Training Team members
- Team Management
- Test Scenarios & Test Cases Reviews
- Bug Reviews
- Preparation of Build summary report
- Conducting meetings with Team members
- MOM Preparation
- Test Plan preparation

Test Manager Responsibilities

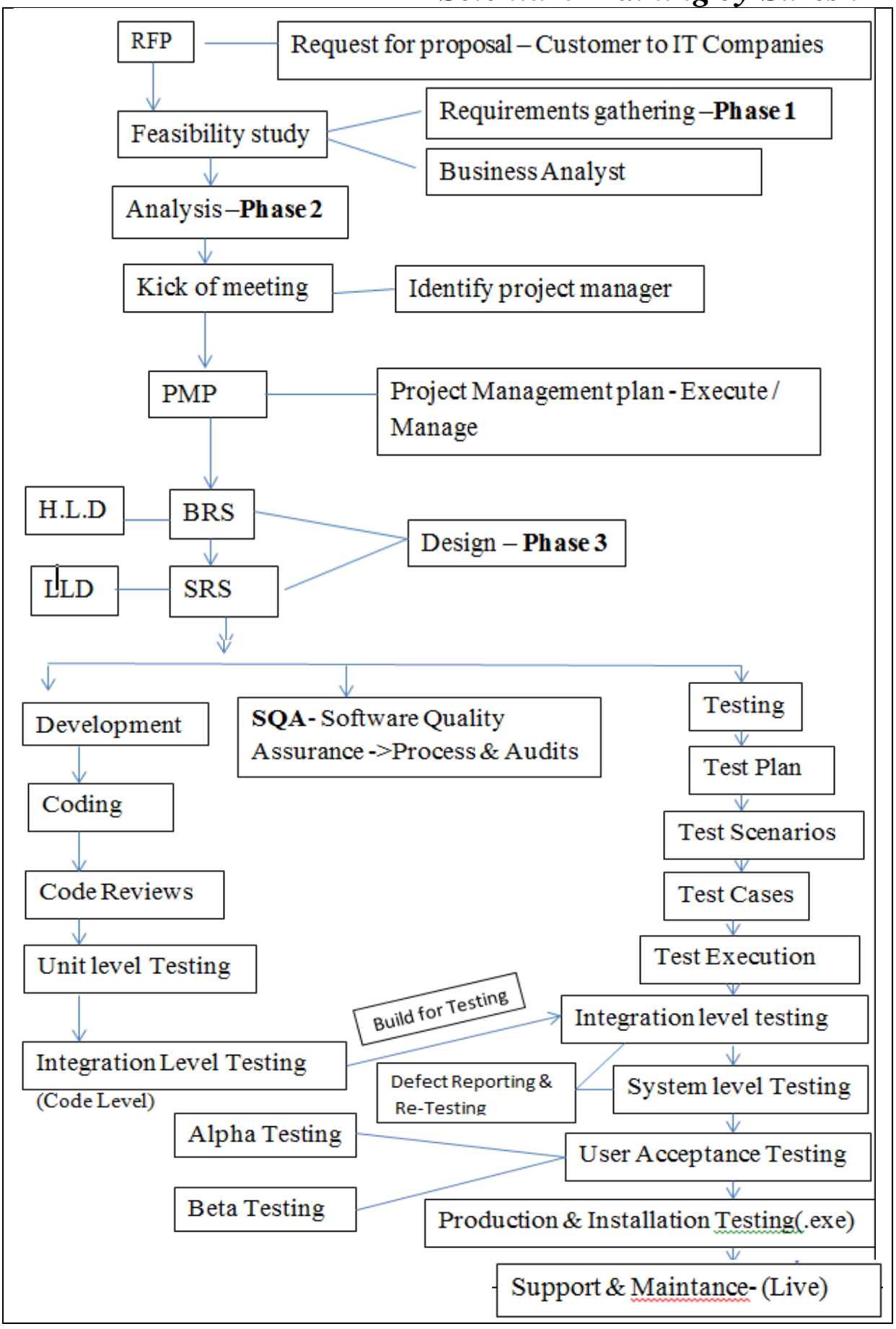
- Project plan and Review of Test Plan
- Effort estimates
- Project Management
- Training Plan – Identify training need based on Resource skills
- Preparing monthly reports
- Client Communications
- MOM
- Provide regular status updates to core team
- Scheduling meetings with Development and Testing Team

SDLC–Software Development Life Cycle

SDLC is a process of developing various quality Software.

Phases in SDLC :

- Requirements gathering
- Analysis
- Design
- Coding/Development
- Testing
- Deployment/Production
- Maintains/Support

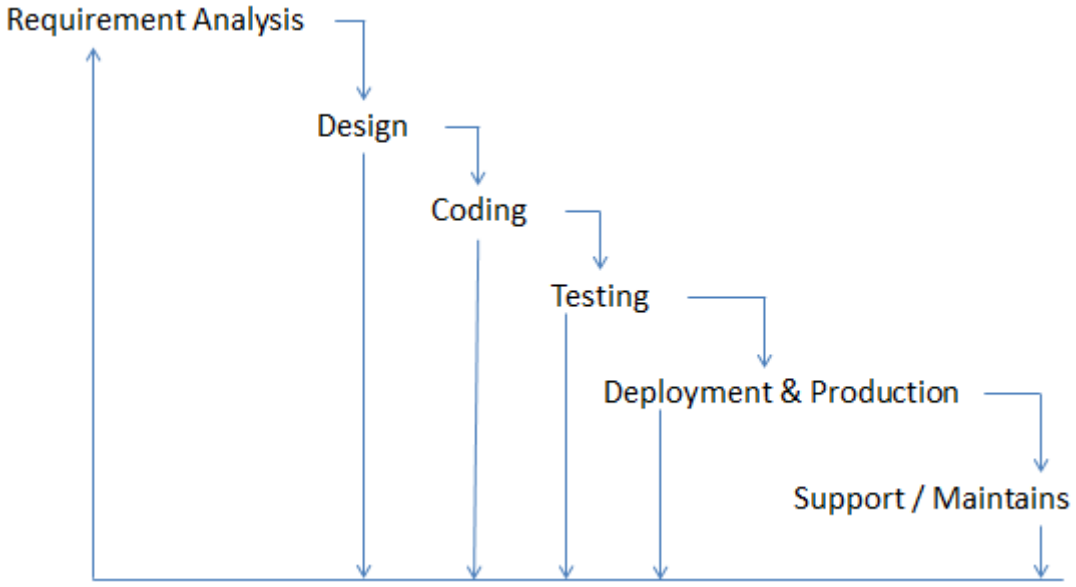

RFP
Request for proposal – Customer to IT Companies
Feasibility study
Requirements gathering –Phase 1
Business Analyst
Analysis –Phase 2
Kick of meeting
Identify project manager
PMP
Project Management plan - Execute / Manage
H.L.D
BRS
Design – Phase 3
LLD
SRS
Development
SQA- Software Quality Assurance ->Process & Audits
Testing
Test Plan
Coding
Test Scenarios
Code Reviews
Test Cases
Unit level Testing
Test Execution
Build for Testing
Integration level testing
Integration Level Testing
(Code Level)
Defect Reporting & Re-Testing
System level Testing
Alpha Testing
User Acceptance Testing
Beta Testing
Production & Installation Testing(.exe)
Support & Maintance- (Live)

Types of SDLC Models

- Waterfall model **or** Life Cycle Model **or** Linear Sequential Model
- V- Model **or** Verification & Validation Model
- Agile …etc

Waterfall model

- This model suggests a systematic and sequential approach to software development that begins at requirements analysis and progress through all life cycle phases sequentially
- Suitable for projects where requirements are clearly defined
- Small and medium term duration
- Having Domain knowledge



Advantages :

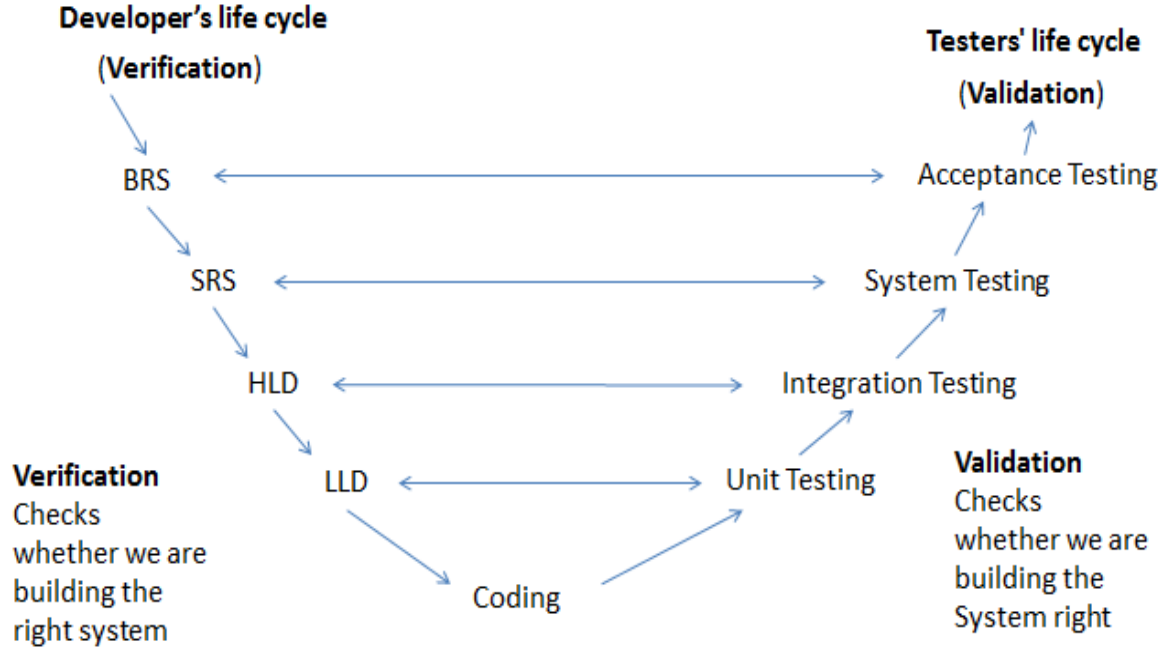
- Project under Control
- Pre-defined outputs at every phase
- Tracking changes is easy

Disadvantages:

- Not suitable for requirements changes
- Does not support going back to previous phase
- If any defect found need to go back to the originating phase

V - model

- V model means verification and validation model, the V shaped life cycle is a sequential path of execution process.



Advantages :

- Simple and easy to use
- Testing activities like planning and test design will be done before coding
- Testing planned parallel to development
- Bug detection in early phases

Disadvantages:

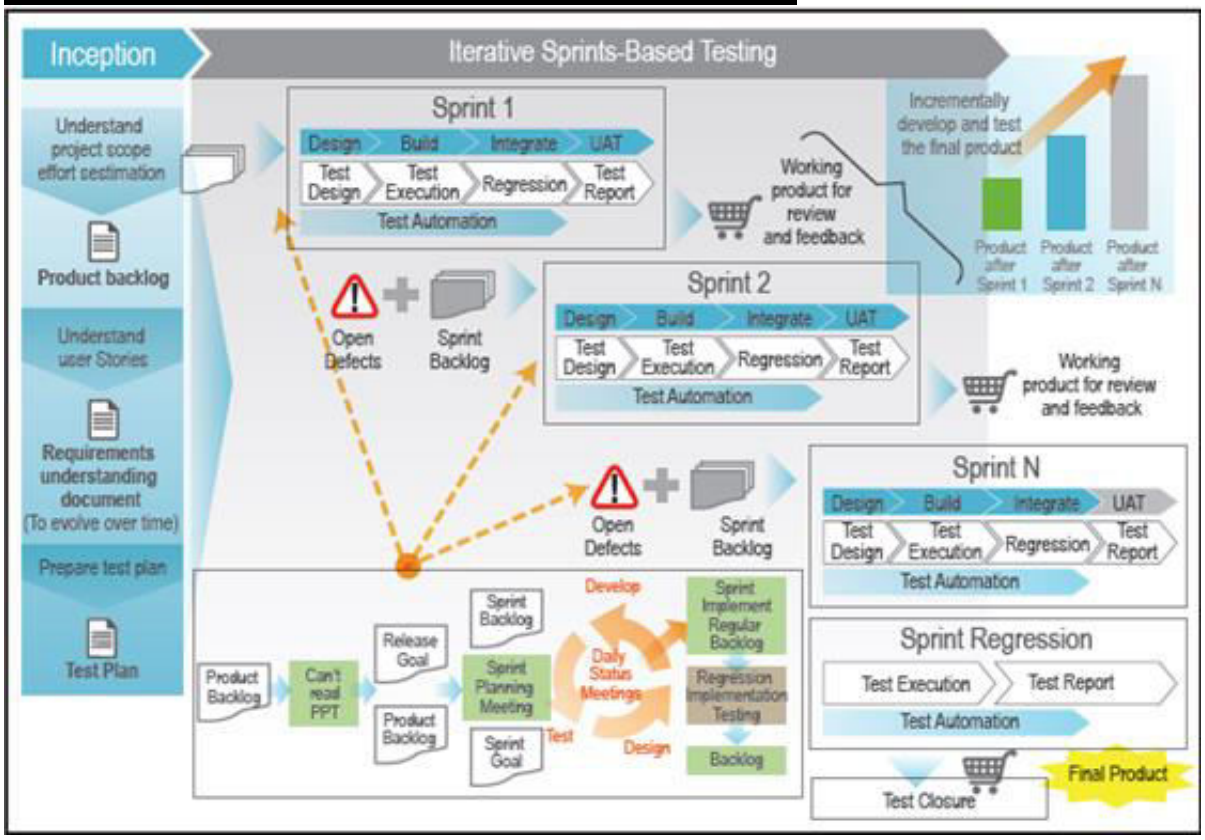
- If any changes happen in midway, then the test document along with requirements documents has to be updated.
- Not Suitable for large and complex projects
- The client sees the only final project ,not intermediate modules

Agile - Process

Overview of Agile Process

- Customer satisfaction by rapid delivery of useful software.
- Welcome changes in requirements ,even late in development
- Working software is delivered frequently (weeks rather than months)
- Close ,daily cooperation between business professionals and developers
- Continuous attention to technical excellence and good design

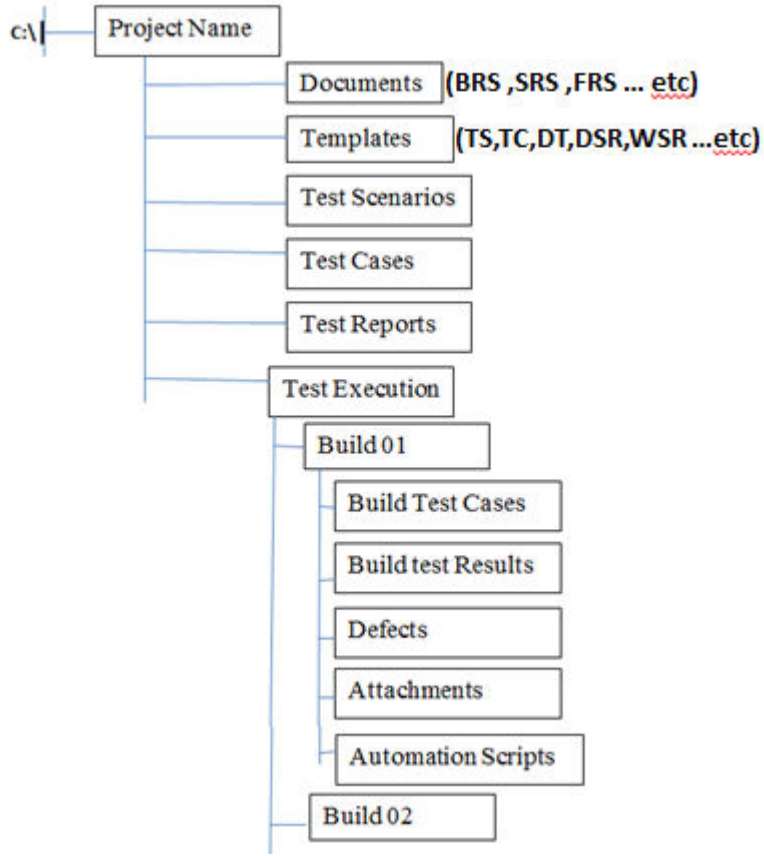
Typical QA Approach for Agile Projects



Agile - Terms

- **User Story** – A shorthand requirements document.
- **Product Backlog** -- A prioritized list of stories that are waiting to be worked on.
- **Product Owner** -- person whom holds the vision for the product.
- **Scrum Master Role** The Scrum Master is a facilitator for the team and product owner.
- **Sprint** -- A development process
- **Stand-up Meeting**– a short (15 minutes or less) daily meeting during which team members report on what they have accomplished since the last meeting, what they plan to accomplish today and report any impediments or blockers to making progress.
- **Scrum Meeting** – To discuss on Sprint planning, development and Review.

Working Folder Structure



STLC – Software Testing Life Cycle

- Requirements understanding (BRS, SRS, FRS, Mock-ups)
- Test Plan
- Test Scenarios
- Test Cases
- Test Execution
- Bug Reporting & Re-Testing
- Test Closer

Test Plan

Test Plan – A document describing the scope, approach, resources & schedule of testing activities. It identifies test items ,the features to be tested, the testing tasks who will be doing and any risks …etc

Entry Criteria to prepare Test Plan :

- Approved PMP
- Approved SRS
- Test Plan guidelines
- Test Plan template

Exit Criteria of preparing Test Plan :

- Test Plan should be reviewed and approved.

Table Contents in Test Plan

16. Approvals
17. Entry / Exit for each Testing activity
18. Test Suspension
19. Test Resumption
20. Test completion

Test Scenarios

Test Scenarios: Identify all possible areas to be tested **or** What is to be tested.

Entry Criteria to Identify Test Scenarios :

- Approved Test plan
- Approved SRS
- Test Scenarios Guidelines
- Test Scenario template

Exit Criteria for Test Scenarios :

- Test Scenarios should be reviewed & approved(mapping test scenarios with requirements)

Test Scenarios Template

Company Logo			Proiect ID		Reviewed By	
			Project Name		Date Reviewed	
			Identified by		Approved By	
			Date Identified		Date Approved	
TS#	Req#	Main Functionality	TestScenario Description	Test Case Name/TC ID	Document ID/Reference	Comments

Sample Example :
Company Logo : Logo of the company
Project ID : ID of the Project
Project Name : Name of the Project
Identified by : Name of the tester
Date Identified : Day of the started writing the Test Scenarios
Reviewed By : Test Lead Name
Date Reviewed : Future Date
Approved By : Test Lead Name
Date Approved : Future Date
TS# : Test Scenario ID
Req:# : Requirement ID
Main Functionality : Page Name
Test Scenario Description : What need to be Tested
Test casename /ID : Test Case ID for corresponding Test Scenario

Document ID / Reference : Which document we are refering to design the Scenarios

Comments : In case of any spefic description need to be mention.

TS#	Req#	Main Functionality	TestScenario Description	Test Case Name/TC ID	Document ID/Reference	Comments
TS_001	3.2	Login Page	Verify Login functionality		SRS	
TS_002	3.2	Login Page	Verify forget password functionality		SRS	
TS_003	3.2	Login Page	Verify new user register functionality		SRS	

Test Case

Test Case: A test case is a document which contain set of input values, pre-conditions, expected result … etc **or** How to be tested.

Entry Criteria to Identify Test Cases :

- Approved Test plan
- Approved SRS
- Approved FRS
- Approved Test Scenarios
- Test Case Guidelines
- Test Case Template

Exit Criteria for Test Cases :

- Test Cases should be reviewed & approved(mapping test scenarios with requirements)

Test Case Template

Company Logo		Project History				Test Cases				
		Project ID				Total No. of TCs				
		Project Name				Passed				
						Failed				
		Test Case History				Not Executable				
		Created By		Date Created		Defects Reported				
		Reviewed By		Date Reviewed						
		Approved By		Date Approved						
		Test Information								
		Test Executed By		Date Executed						
		Version		Build						
TC #	TS#	Test Design/ Steps	Input Data	Expected Result	Actual Result	Pass	Fail	Not Executable	Comments	Defect Id

--Sample Example :

Company Logo : Logo of the company

Project ID : ID of the project

Project Name : Name of the Project

Created By : Tester name

Review By : Test Lead name

Approved By : Test Lead name

Date Created : Day of the Started writing Test Cases

Date Reviewed : Future Date
Date Approved : Future Date
Test Executed By : Tester name
Version : Version of the Test Case Document
Date Executed : Form Which Date Test Case Execution need to start
Build : build number
Total No of Test Cases : The Total number of Test Cases Created for whole application.
Passed: Total number of Test Cases passed
Failed: Total number of Test Case Failed
Not Executable: not executed test case count
Defects Reported: Bugs reports for the Whole application.
TC# : Test Case ID
TS# : Test Scenario ID mentions in Test Scenario Document
Test Design Steps : Steps To be Followed for Testing particular Functionality
Input Date : Test Data need to be used to test particular Functionality
Expected Result : Result Excepting Based on Requirement Document
Actual Result : Result displayed in Application
PASS : Expected Result & Actual Result both are matching consider as PASS
FAIL : Expected Result&Actual Result both are not matching consider as FAIL
Not Executable : Test Case Which are not executed for that current Release
Comments : In case of any spefic description need to be mention.
Defect ID : Bug ID for failed test Cases

TC #	TS#	Test Design/ Steps	Input Data	Expected Result	Actual Result	Pass	Fail	Not Executable	Comments	Defect Id
TC_001	TS_001	1.Launch browser 2.Enter URL 3.Enter Valid UserName 4.Enter Valid Password 5.Click on Login button	URL : Username : Password :	1.Login should be successful 2.Search hotel page need to display						
TC_002	TS_001	1.Launch browser 2.Enter URL 3.Enter Invalid UserName 4.Enter Valid Password 5.Click on Login button	URL : Username : Password :	Error message need to displayed as "Invalid UserName"						

Test Case Design Techniques

Test Case: Design techniques can broadly split in to 2 categories.
Black box techniques
White box techniques

Black box techniques

- Equivalence Class Partitioning(ECP)
- Boundary Value Analysis(BVA)
- State Transition
- Decision Table / Cause Effect Table

White box techniques

- Statement Testing
- Branch/Decision testing
- Data flow Testing
- Branch condition testing

Equivalence Class Partitioning (ECP) :

Divide a set of test conditions into groups which is having similar behavior to reduce number of test cases.

Age [ ] **(accepts 1 to 60)**

Invalid	Valid	Invalid
0	1 to 60	>=61

Boundary Value Analysis(BVA):

BVA is based on testing the boundaries of condition

Formula : Minimum ,maximum ,Min-1 ,Max+1

Valid Boundaries : Minimum , Maximum

Invalid Boundaries : Min-1 ,Max-1

Name [ ] **(accepts 5 to 10 characters)**

Invalid (Min-1)	Valid (Min,Max)	Invalid (Min+1)
4	5 , 10	11

State Transition Table:

Application provides different output for same input based on previous stage

Username	Password	Result
Correct	Wrong	Invalid Password
Correct	Wrong	Invalid Password
Correct	Wrong	Invalid Password
Correct	Wrong	Account Locked

Decision Table :

Testing with different combination of inputs which produce different results.

Username	Password	Result
Correct	Correct	Login Successful
Correct	Wrong	Invalid Password
Wrong	Correct	Invalid Username
Wrong	Wrong	Invalid Username

Defect Management

Defect Status : New ,Open ,Fixed ,closed ,Re-Open ,Not a bug, Duplicate ,Need More information, Can't reproduce .
Severity: Importance of defect with respective to functional point of view…means criticalness of defect with respective to application.
Severity classification could be : CRITICAL , HIGH ,MEDIUM ,LOW
Priority: Importance of defect with respective to client point of view … means how soon it should be fix.
Priority Classification could be : P1-Urgent ,P2-High ,P3-Medium ,P4-Low

Difference between Defect /Bug/Error/Failure

Defect : Problem which is identified on Developer machine at development phase
Bug : Problem which is identified on Testers machine at testing phase
Error: Problem which is related to coding.
Failure : Problem which is identified By end users at production phase

Bug / Defect Template

Defect ID		Assigned To				
Status		Browser				
Severity		Found in Version				
Priority		Found in Build				
Module		Fixed in version				
Reported By		Fixed in build				
Title						

Description

Steps To Re-Produce :

1.

2.

3.

Expected Result :

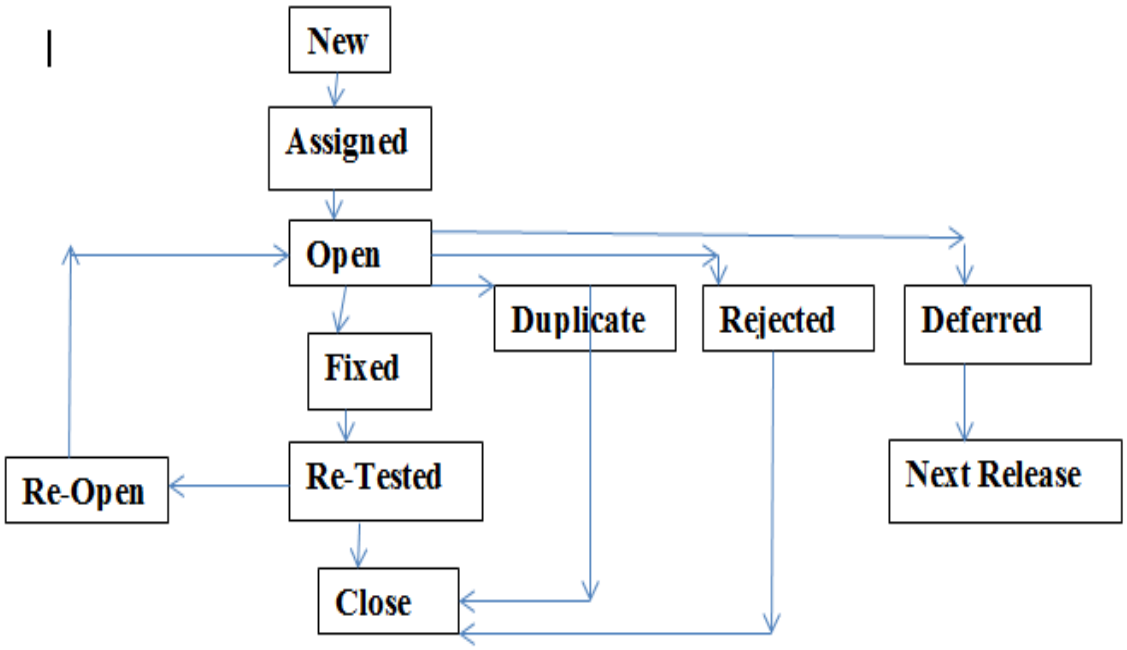
Actual Result :

Sample Example :
Defect ID : ID of the new Defect
Status : Status of the Bug
Severity : Need to provide based on Functionality
Priority : Need to provide based on Client Expectation
Module : Page Name
Reported By : Tester Name
Assigned to : Developer or Test Lead Name
Browser Name : Name of the Browser
Found in Build: in which build bug had found
Found in Version: number of the version
Fixed in Version : Developer will provide this
Fixed in Build : Developer will provide this

Bug / Defect Template			
Defect ID	DEF_001	**Assigned To**	DEV
Status	NEW	**Browser**	Chrome
Severity		**Found in Version**	1
Priority		**Found in Build**	1
Module	Search Hotel Page	**Fixed in version**	
Reported By		**Fixed in build**	
Title	Validation is not working for check in date (accepting previous date)		

Description : Steps To Re-Produce :
1.Launch Browser .2.Enter URL
3.Enter valid Username
4.Enter valid password
5.Click on login
6.Select all required feilds
7.Enter check in date as previous date
8.Click on search
Expected Result :
Error message should be displayed as "check in date should be current date or feature date"
Actual Result :
Search hotel page is displayed

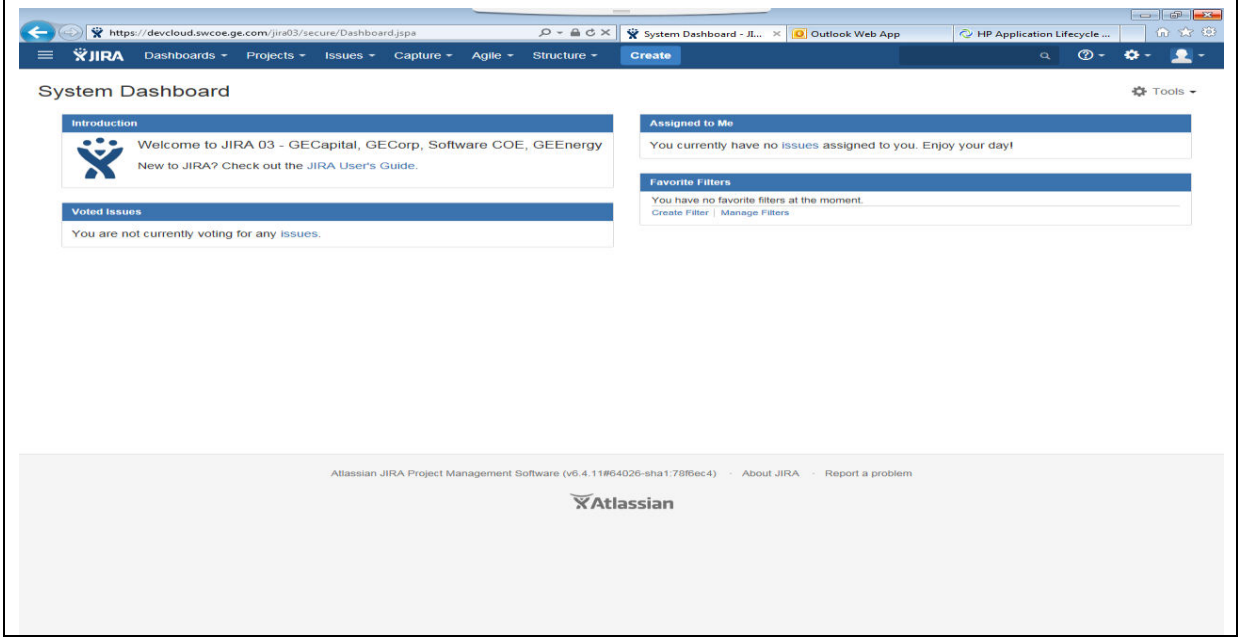
Bug or Defect Life Cycle



JIRA Tool:

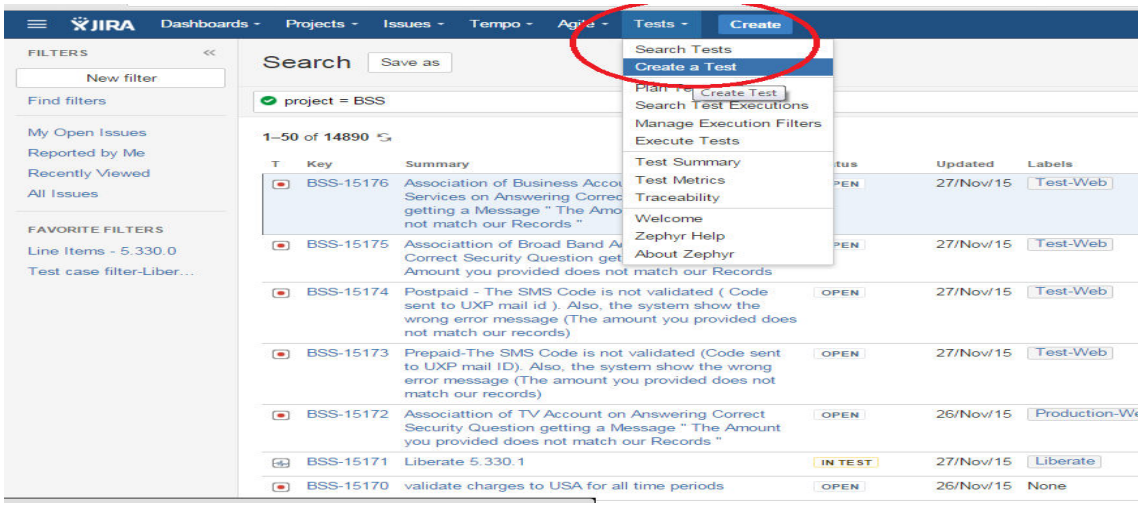
Jira Login Process:

- Open JIRA URL in the browser:
- Enter valid credentials (User Name and Password)
- JIRA Home Page shown as below:

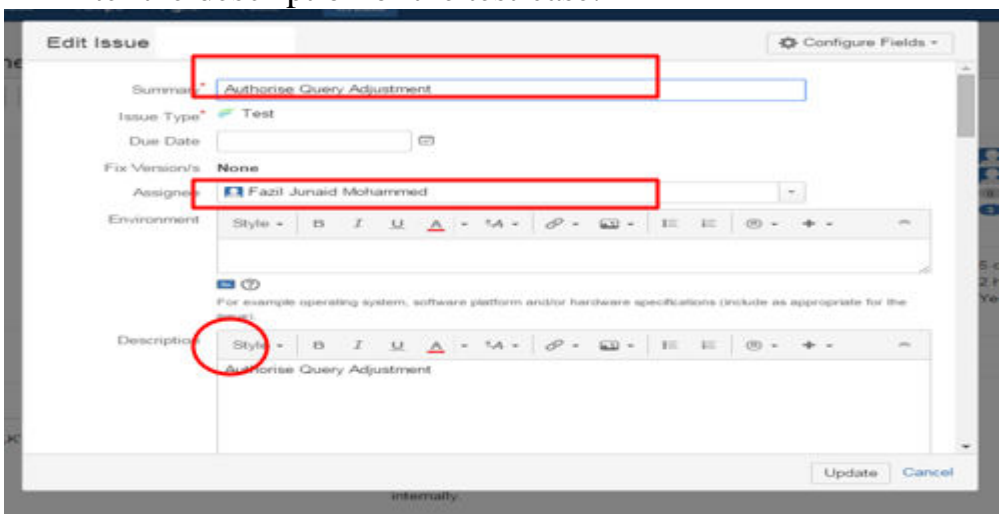


Steps to Create Test Cases in JIRA :

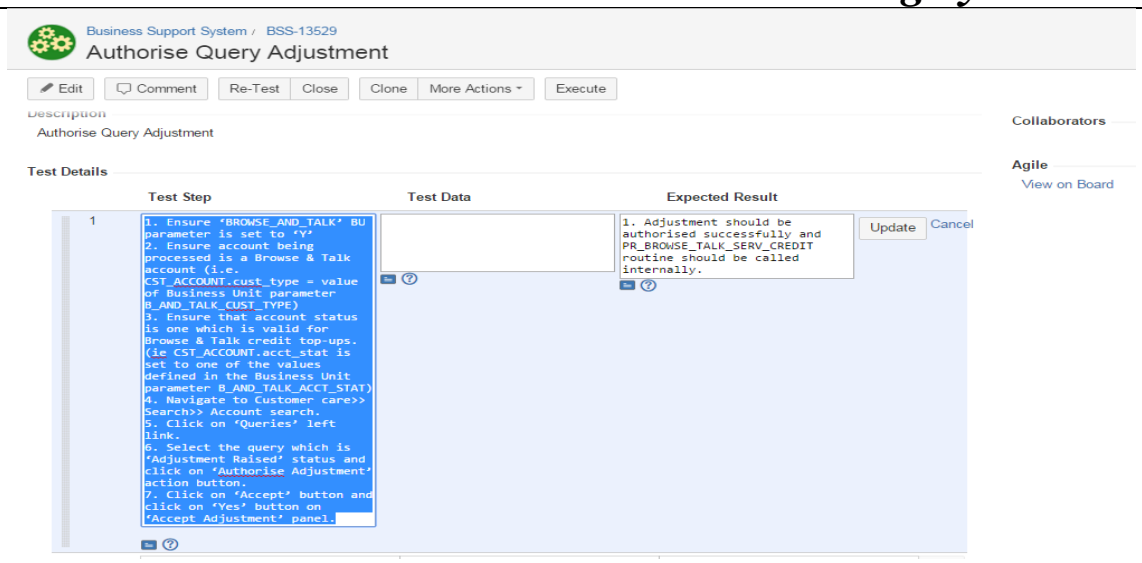
- In JIRA, Test case, Bug, Epic and Story will be called as Issues.
- To create a JIRA issue, you need the Create Issue project permission for the issue's relevant project. If you do not have this permission, please contact your JIRA administrator.
- To create a new JIRA issue (Test case):
- Click Create at the top of the screen to open the Create Issue dialog box.
- Select the relevant Project and Issue Type on the Create Issue dialog box.
- Under the tests tab, click on create a test.
- Please see the below screenshots for reference:



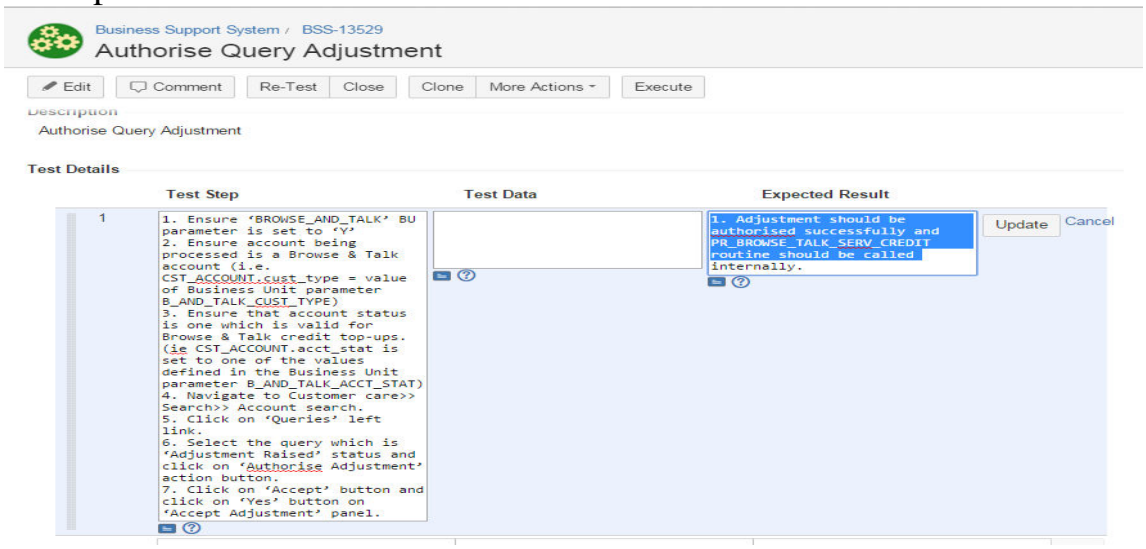
- Enter the test case summary in the summary filed.
- Assign the test case to anyone by typing their name.
- Enter the description of the test case.



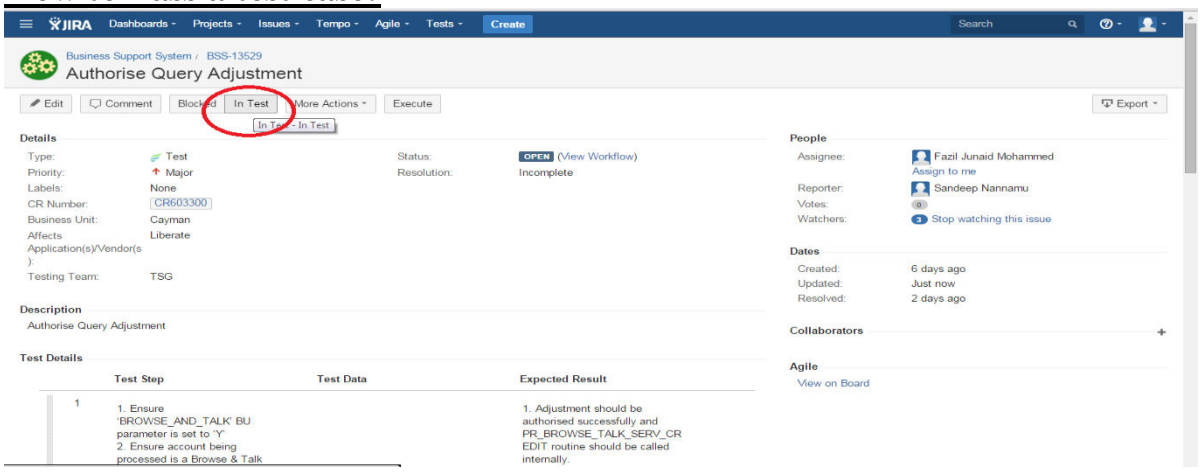
- Test case will be created and an ID will be assigned to it . Once the test case is created you can add test steps and expected results as shown in next page.
- Click under the test steps and enter the steps. Click under the Expected results and enter the results and click on update.
- Test steps



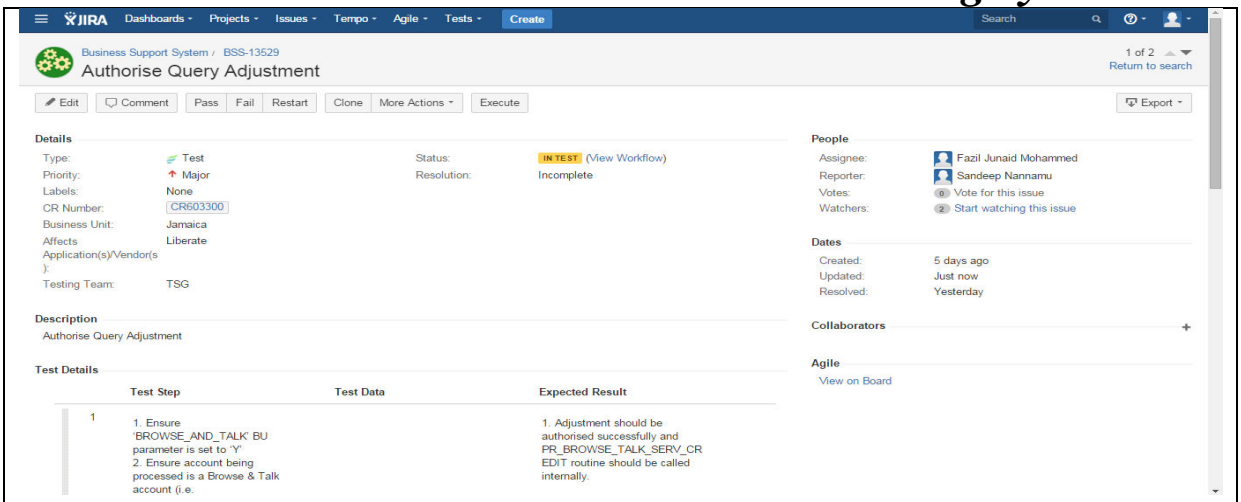
- Expected Result



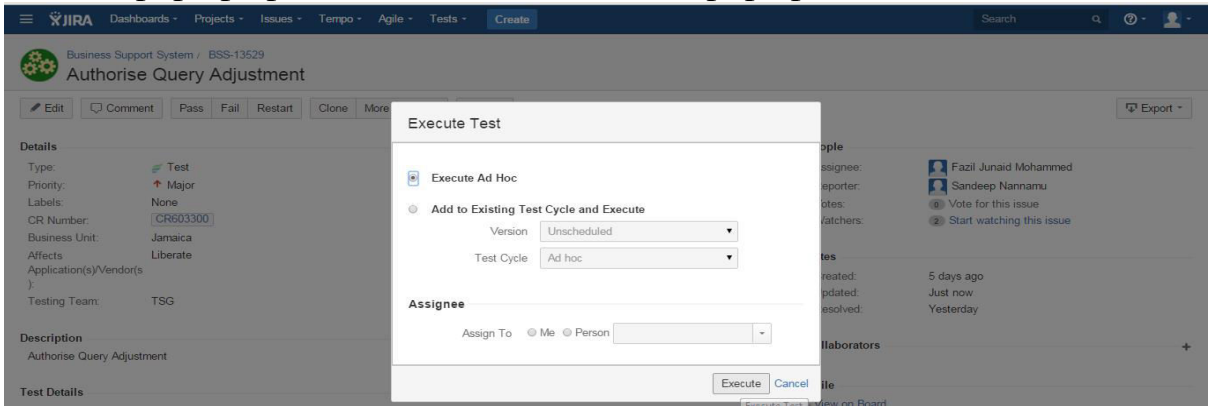
How to Pass a test case:



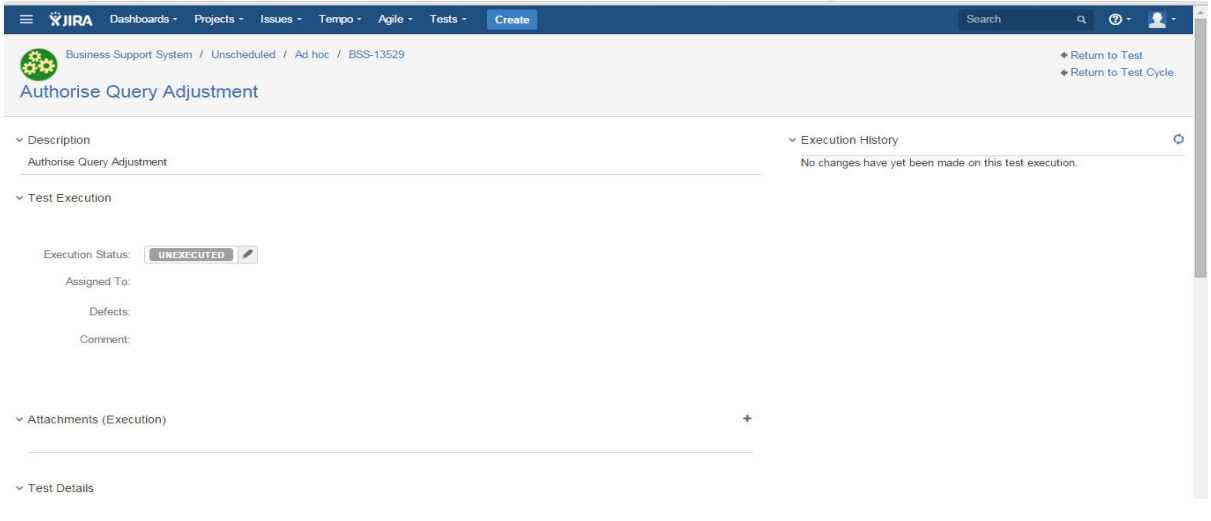
- The status of the test case is changed from "Open" to "In test"

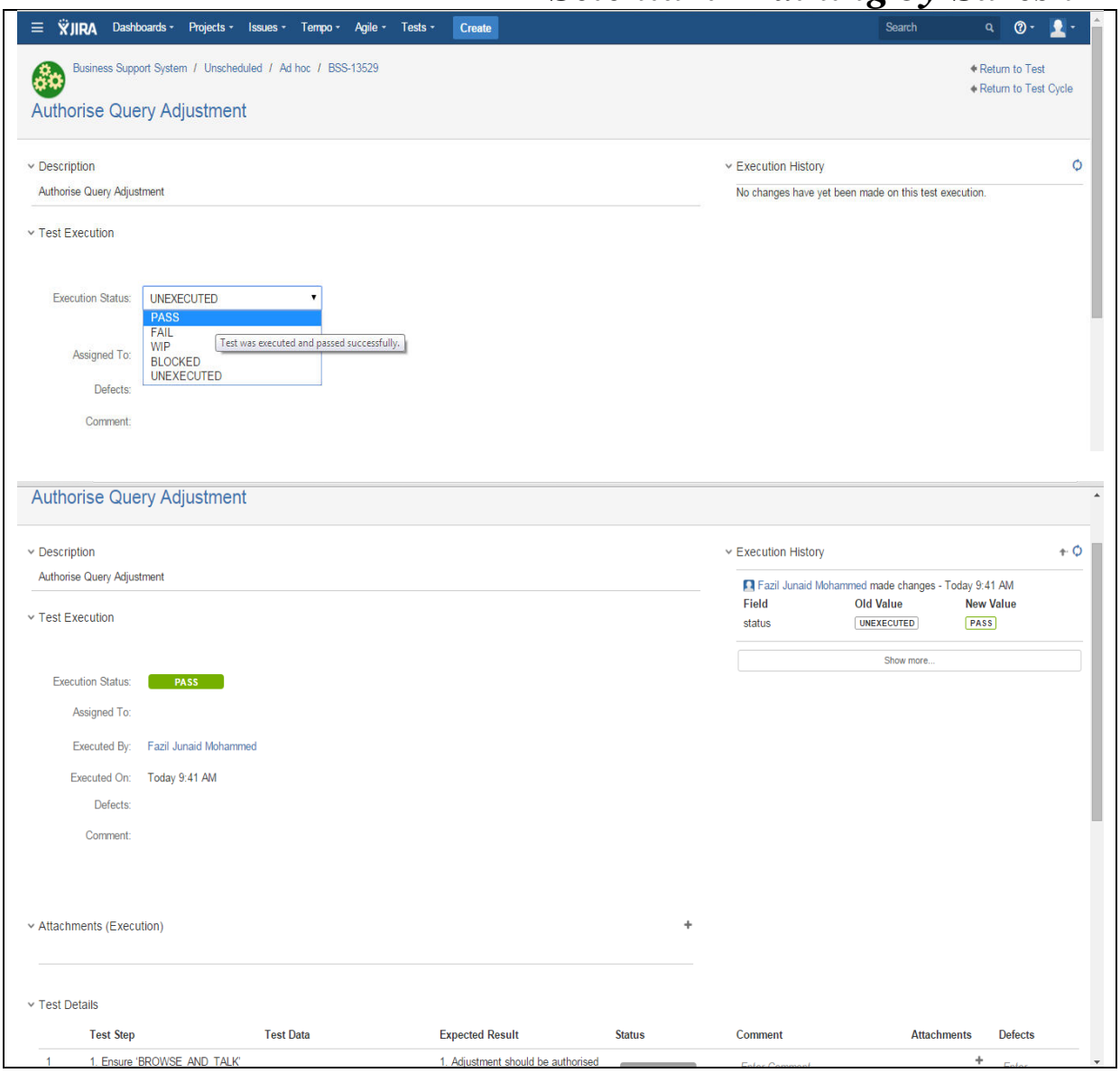


- Click on " Execute "
- A pop up opens. Click on execute in the pop up.



- Change the execution status from unexecuted to pass




JIRA
Dashboards
Projects
Issues
Tempo
Agile
Tests
Create
Search
Business Support System / Unscheduled / Ad hoc / BSS-13529
Return to Test
Return to Test Cycle
Authorise Query Adjustment
Description
Authorise Query Adjustment
Execution History
No changes have yet been made on this test execution.
Test Execution
Execution Status:
UNEXECUTED
PASS
FAIL
WIP
BLOCKED
UNEXECUTED
Test was executed and passed successfully.
Assigned To:
Defects:
Comment:
Authorise Query Adjustment
Description
Authorise Query Adjustment
Execution History
Fazil Junaid Mohammed made changes - Today 9:41 AM
Field
Old Value
New Value
status
UNEXECUTED
PASS
Show more...
Test Execution
Execution Status:
PASS
Assigned To:
Executed By:
Fazil Junaid Mohammed
Executed On:
Today 9:41 AM
Defects:
Comment:
Attachments (Execution)
Test Details
Test Step
Test Data
Expected Result
Status
Comment
Attachments
Defects
1
1. Ensure 'BROWSE_AND_TALK'
1. Adjustment should be authorised

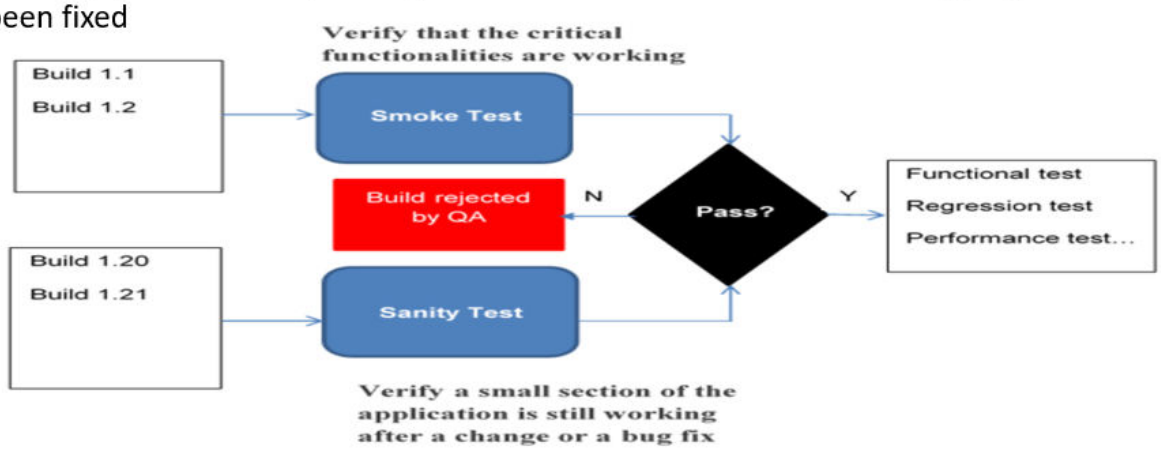
Types of Testing

- **Smoke Testing** or **BVT-Build Verification Testing** or **TAT – Testers Acceptance Testing**

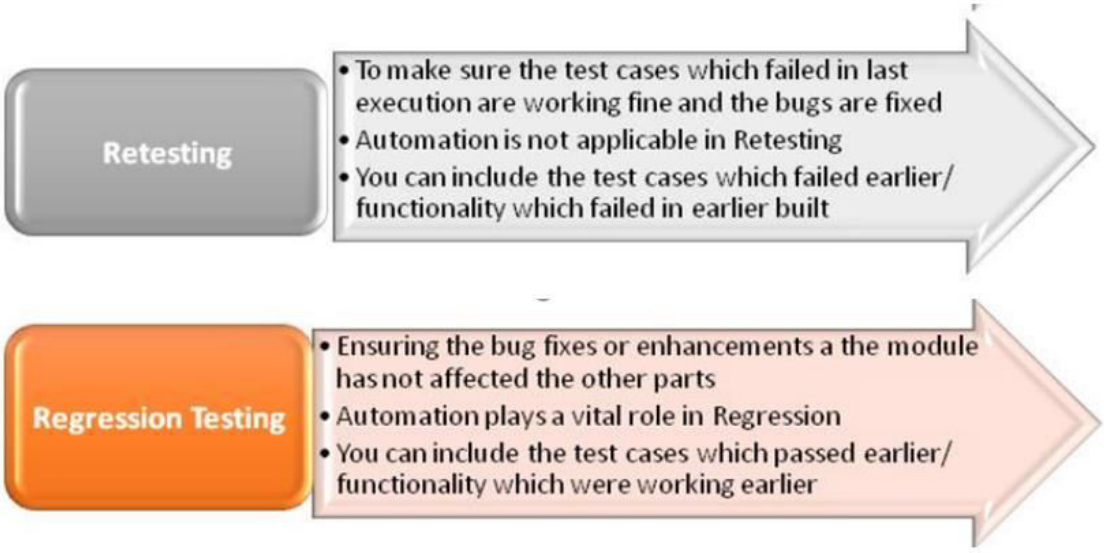
It is first level of testing on any newly released build to check main functionality of the application.

- **Sanity Testing :**

Sanity Testing is done during the release phase to check for the main functionalities of the application.Sanity Testing is done to check the new functionality/bugs have been fixed



- **Re-Testing :** Testing defects were fixed or not in the current build
- **Regression Testing:** To check existing functionality is un affected whenever the new change is added



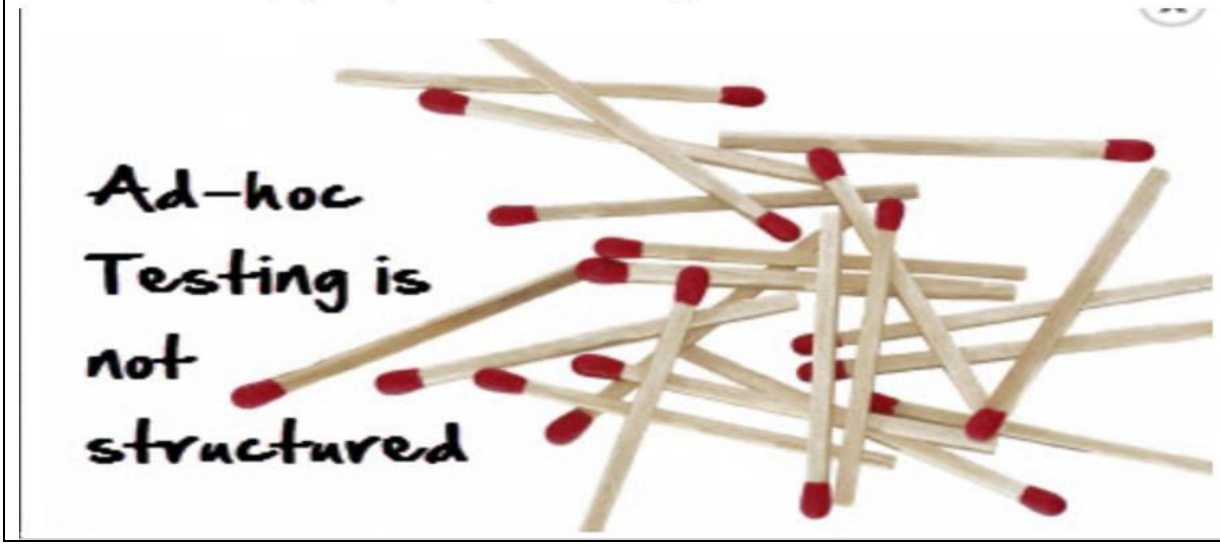
- **Static Testing:** Testing an application with out performing any action.

Eg: GUI Testing , colors ,spelling ,alignment … etc.

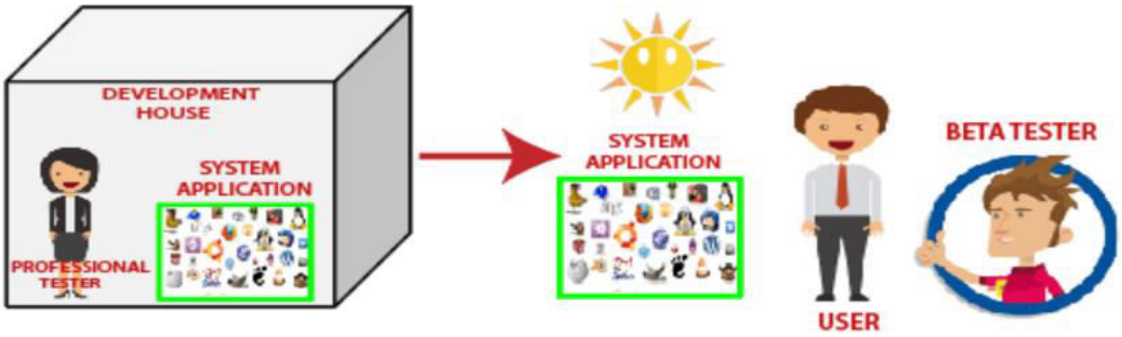
- **Dynamic Testing:** Testing an application by performing required actions.

Eg: Functionality Testing ,Textbox, button … etc

- **Ad-hoc Testing:** Testing the application with out any proper planning



- **Alpha Testing :**Final Testing on the application doing with in the development company
- **Beta Testing:** Testing doing in customer environment ,This testing will be done by the customer or third party test engineers.



BETA TESTING OF THE PRODUCT IN REAL WORLD ENVIRONMENT

Functionality Testing - Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing etc.

Usability testing - Usability testing is nothing but the User-friendliness check. In Usability testing, the application flow is tested so that a new user can understand the application easily. Basically, system navigation is checked in Usability testing.

Compatibility testing - Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.

Database Testing – Database connection and user entered Data in application is saving into respective database tables

Interface testing-Three areas to be tested here are - Application, Web and Database Server

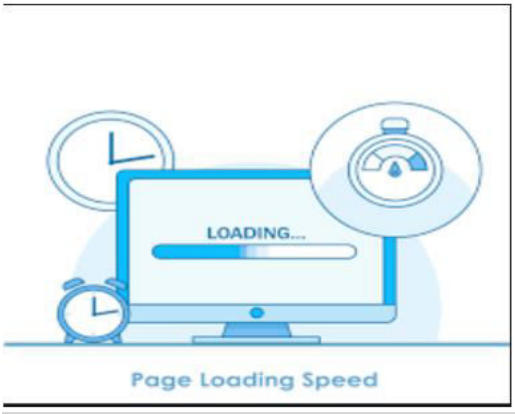
01. Application: Test requests are sent correctly to the Database and output at the client side is displayed correctly.

02. Web Server: Test Web server is handling all application requests without any service denial.

03. Database Server: Make sure queries sent to the database give expected results.

Performance testing – Testing page ,data, images load time

Security testing -Security Testing involves the test to identify any flaws and gaps from a security point of view.



When to stop Testing?

This can be difficult to determine. Most modern software applications are so complex, and run in such an interdependent environment, that complete testing can never be done. Common factors in deciding when to stop are:

- Deadlines (release deadlines, testing deadlines, etc.)
- Test cases completed with certain percentage passed
- Test budget depleted
- Coverage of code/functionality/requirements reaches a specified point
- Bug rate falls below a certain level
- Beta or alpha testing period ends