

# Manual Testing Notes:

## Manual Testing Part: 1

### Topic: 1: Introduction and Syllabus

Manual Testing Syllabus	
<ul style="list-style-type: none"><li>● <b>Part-1</b><ul style="list-style-type: none"><li>➢ What is Software Testing?</li><li>➢ SQA (Software Quality Assurance)</li><li>➢ SDLC (Software Development Life Cycle)</li><li>➢ Models for Software Development<ul style="list-style-type: none"><li>○ Fish Model</li><li>○ Waterfall Model</li><li>○ V-Model</li><li>○ Agile Model</li></ul></li><li>➢ Agile Ceremonies (Meetings)</li><li>➢ Functional &amp; Non-Functional Testing and its types</li><li>➢ Interview Real Time Questions</li></ul></li><li>● <b>Part-2</b><ul style="list-style-type: none"><li>➢ Test Plan &amp; Test Methodology</li><li>➢ STLC (Software Testing Life Cycle)</li><li>➢ Defect/Bug Life Cycle</li><li>➢ Test Case Design</li><li>➢ Bug Tracking Tool<ul style="list-style-type: none"><li>○ JIRA</li><li>○ HP ALM</li></ul></li><li>➢ Accessibility Testing (NVDA Screen Reader)</li><li>➢ Performance Testing (JMeter Tool)</li><li>➢ Demo Lecture :- How Actual Sprint Work in Company</li><li>➢ Manual Part 1 &amp; 2 Real time Interview Questions</li></ul></li></ul>	

### What is Software?

- The software is a kind of programs that enable a user to perform some specific task or used to operate a computer.
- Software is collection of specialised program which takes input from customer & generate desired output.

### Types of Software:

**1) System Software:** System Software (a type of computer program) provides a platform to run computer's hardware and computer application to utilise system resources and solve their computation problem.

**Example:** The best-known example of system software is the **operating system (OS)**.

It responsible for manages all the other programs on a computer.

**2) Application Software:** Applications software is capable of dealing with user inputs and helps the user to complete the task. It is also called end-user programs or only an app. It resides above system software. First user deal with system software after that he/she deals with application software.

**Examples:** of Application Software are Word processing software, **Spread sheets** Software, Presentation, Graphics, **CAD/CAM**, Sending email etc.

**3) Programming Software:** Compiler, Text Editor (Note Pad, MS Office: Word, PPT, and Excel)

**4) Custom Software:** (Based on Client/Customer Requirements)

Ex: Hospital Management System, Banking Application, SAP Software

## What is Software Testing?

- **Software Testing** is the process of identifying the **correctness and quality** of software programs. The purpose is to check whether the software satisfies the specific requirements, needs, and expectations of the customer.
- In other words, testing is executing a system or application in order to find software **bugs, defects or errors**. The job of testing is to find out the reasons for application failures so that they can be corrected according to requirements.

**Example:** Car manufacturer tests the car **for maximum speed, fuel efficiency and safety from crash**. These test later become the part of advertising strategy for car sales.

## Classic Examples of Software Testing

The truth of the matter is that all companies are prone to software bugs. In fact, some of the most credible and well-respected companies have run into serious problems of their own. Take **NASA** as an example. The company's **Mars Climate Orbiteer** crashed because it went too low too fast. In the end, it was discovered that the agency had utilized **non-metric units**. Unfortunately, the software needed **metric units**. That simple lapse cost the **company \$125 million!**

## Why you select testing field?

1) **Its challenging** – Testing is not easy — there are constantly puzzles and problems to solve. The job will likely bring something new every day.

If you prefer a boring job where you don't have to think too much then don't pursue a software testing career.

But if you want a job that keeps you on your toes, anyone will tell you that testing is a really great choice.

2) **Its important** – Testers don't always get enough recognition for the work they do, but we'd be lost without them.

- ✓ As a tester, you're advocating the end user and making sure that they're being delivered a quality product.
- ✓ Without someone to find bugs before software is delivered,
- ✓ many businesses would be suffering from poor reputations and unloyal customer bases.

3) **Its creative** – You have to get a little innovative when testing.

The process isn't going to be spelled out for you — in fact, it takes a little detective work.

By acting as the end-user, you're the one who has to get creative when thinking of places there may be inconsistencies.

4) **There are many paths** – Every company that uses software needs software testers,

- ✓ Which is to say, pretty much everyone needs software testers.
- ✓ Testers are valuable in basically any industry, from healthcare to retail to video games.
- ✓ Additionally, you can choose whether you want to go into manual testing, automated testing, performance testing, etc.

5) Since the high demand for software testers, software testing will be a high-paying job.

And you will have more chances to grow your career.

## **Why do we need Software Testing?**

- 1) Helps in saving money
- 2) Security
- 3) Quality of the product
- 4) Satisfaction of the customer
- 5) Enhancing the development process
- 6) Determining the performance of the software

## **Topic: 2: Resources Involve in S/W Development**

- 1) Customer/End User/Client
  - Customer has no. of requirements
- 2) BA (Business Analyst)
  - Gathering Information/Requirements from Customer and made Requirement Document
- 3) Developer
  - It is person who develops Application as per customer requirements.
  - Developer only check positive scenario as per customer requirements
- 4) Tester
  - Checking application **completeness** and **correctness** as per customer requirement.
  - In which we do Positive as well as Negative Testing

**-Positive Testing:-** Use Valid data for testing

Ex. Mobile Number Field: we test field by using only 10 digit number

**-Negative Testing:-** Use Invalid data for testing

Ex. Mobile Number Field: We test field by entering digit less or more than 10, by entering any alphabets from A-Z, or any Symbol

5) After success full testing process Final Product will be delivering to customer

### **Topic: 3: SQA (Software Quality Assurance)**

- ✓ It is communication between Customer and Business Analyst
- ✓ SQA is done to Monitor & measure software development Factors

#### **Factors:**

##### **1) To meet the customer Requirement**

A) Which types of the application customer want?

-Banking Domain

-Telecom Domain

-Healthcare Domain

-E-Commerce Domain

##### **2) To Meet the Customer Expectations**

A) Privacy: privacy include security of any software

Ex. Banking Domain: Data gather from customer which very sensitive so data kept confidential

B) Performance: S/w should balance/sustain heavy load conditions

Ex. Amazon Application: Mobile Sell

##### **3) Costing of the Project:**

-Project costing for IT Companies is per Hour cost.

-Customer has to pay it.

-This payment depends upon resources utilization as well as time to complete the project

#### **4) Timing Delivery:**

- At the time of Information Gathering from customer Application deliver date decided.
- If company exceed the delivery date/ time then company have to pay penalty to customer called "**Escalation**".

#### **5) Maintenance:**

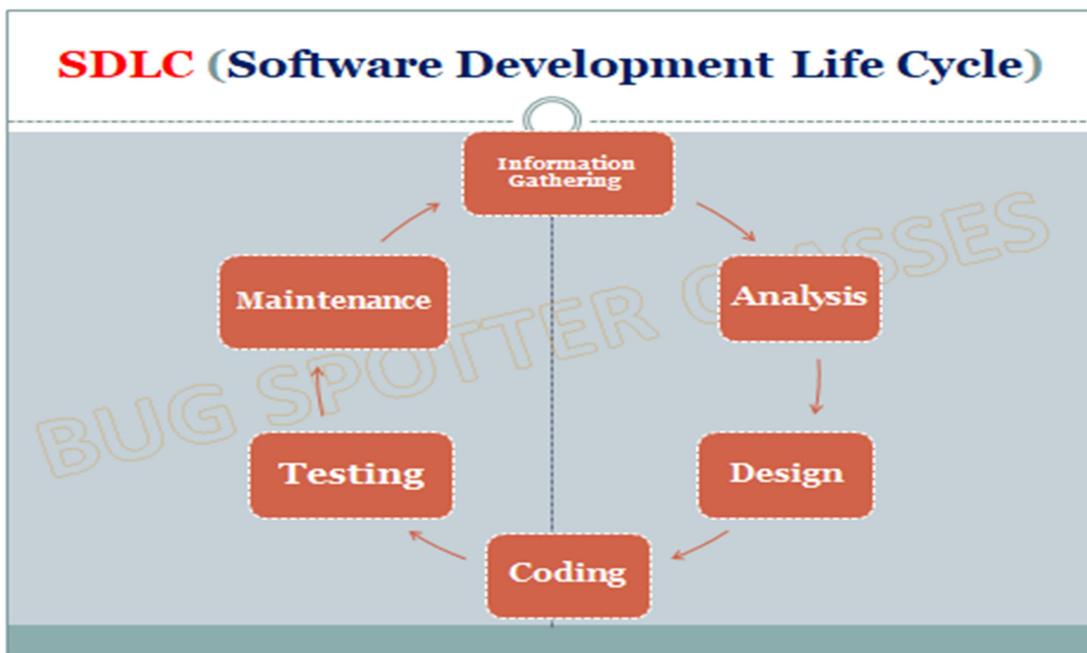
- maintenance is service provided after delivery of product/Application.
- If any problem occurs after delivery then company have to fix that issue.

### **Topic: 4: SDLC**

#### **SDLC: Software Development Life Cycle**

- Start End to end software development Process/Stages
- Means from **Information Gathering** to Product **deliver to customer**

#### **Stages of Software development life cycle (SDLC)**



- 1) Information/Requirement Gathering
- 2) Analysis
- 3) Design
- 4) Coding (Development)
- 5) Testing
- 6) Maintenance

## **1) Information Gathering/Requirement Gathering**

- Customer and Business Analyst (BA) involve in this
- Customer has no. of requirements
- BA gather those all requirements
- Create one document called BRS (Business Requirement Specification)

**Example:** Facebook: - 1) Signup 2) Log In 3) Home Page 4) Account Information

## **2) Analysis**

- In this stage **BRS document convert into SRS** (Software Requirement Specification)
- also called detailed documents
- This is work of Business Analyst.
- For **example**: Signup Page main Module:-
- Sub Module**: First Name, Last Name, EmailID/Mobile Number, Date of Birth, Gender, Signup button
- It is also known as **functional requirements specification**.

**SRS (Software Requirement Specification)** include following some parts:

**1) Functional Flow Diagram:**

**2) Functional Requirement**

**3) Use Cases**

**4) Snap shot**

**1) Functional Flow Diagram:**

-Functional flow diagram it is flow of your application.

-Relationship between the modules.

-This gives proper sequence of the task

-It is a Stepwise representation of software

**Example:** Facebook Application:

**Functional Flow diagram:** 1) Sign Up Page 2) Log IN Page 3) Home Page 4) Request Page...

**2) Functional Requirements:**

-Functional requirements means attributes which are required to complete specific functions.

**For Example: Facebook Sign Up Page**

**Requirement for Sign Up:** First Name, last name, EmailID/Mobile Number, New password, Confirm Password, Date of birth, Gender, Signup Button

**For Example: Facebook Log In page**

Requirement for Log In page: EmailID/Mobile Number, Password, Forgot Password, Create New Account, Log In button

### **For Example: Facebook First Name**

#### **Requirements for First Name:**

- 1) Name should be in Character
- 2) Name does not have numbers
- 3) Name does not have special Characters
- 4) It should not have spaces

### **For Example: Facebook Mobile Number Field**

#### **Requirements for Mobile Number:**

- 1) Only 10 digit should be accept
- 2) Mobile Number field should not accept Character
- 3) Mobile Number field should not accept space
- 4) Country code for India +91
- 5) It should not accept Special Character for ex. Symbols

### **3) Use Cases:**

-It is functionality in terms of **Input, Process and Output.**

For ex Facebook Application:

- ✓ Input: for Sign up page: All required field like First Name, Last Name, Mobile Number....added
- ✓ Process: Click on Sign up button then it start process to navigate on Log In page or Sign up account creation
- ✓ Output: 1) Account will be created successfully or 2) Navigate to Login Page

#### **4) Snap shot:**

-Snapshot are visualization of functionalities before development of product/application/module/software

-Snapshot created by **Business Analyst (BA)**

-BA create snapshot by using IRise software/tool.

-Snapshot gives ideas to developer that how s/w supposed to look like.

#### **3) Design:**

There are two Types of Design

##### **A) High Level Design**

For ex: Signup Page, Log In Page, Home Page

-It is **related to Main Module**

-It is developed/designed by **Design Architecture** or System Architecture

##### **B) Low Level Design**

-It is related to **Sub Modules**

-For ex: For Main Module Sign UP Page:- First Name, last Name, EmailID, Mobile Number.....etc.

-It is designed by **Front End Developer**

#### **4) Coding:**

- ✓ Coding nothing but the Programming
- ✓ One line is code
- ✓ Multiple lines of codes called Programming
- ✓ Set of Programs written by developer called software.
- ✓ Done by Developer

## **-Types of Developer:**

### **1) Front End developer**

- Work of Front End Development

-Create UI (User Interface) or GUI (Graphical User Interface) design

-Functional Flow

-Process

### **2) Back End Developer**

- Work of Back End Development

-Back End Develop/ Database of the company develop

-Data management

-Data Gathering

-Data Security

### **3) Full Stack Developer**

-Knowledge of both Front End and Back End

## **5) Testing:**

-Testing is the process of checking **correctness** and **completeness** of the application as per **customer requirements**.

### **Types of Testing:**

1) White Box Testing

2) Black Box Testing

3) Grey Box Testing

## **1) White Box Testing:**

- White box testing done by the developer.
- Also called code level testing
- Also called Unit Testing
- In this developer do Positive Testing only
- Testing do with only valid data
- Example:** Mobile Number: Field will be check by entering 10 digits only
- In which developer check correctness and completeness of the Program.

## **2) Black Box Testing:**

- BBT also known as System & Functional Testing
- It is done by Tester.
- Overall Functionality of application will be test by Tester.
- In this Tester check positive as well as negative scenario.
- Example:** Positive Scenario: Mobile Number Field: Field will be check by entering 10 digit only
- Example.** Negative Scenario: Mobile Number Field:
  - 1) More or less than 10 digits number
  - 2) Character should not be accept
  - 3) Special should not be accept
  - 4) Space should not be allow

### **3) Grey Box Tester:**

- It is **combination of WBT and BBT**
- Grey box tester has knowledge of Programming.
- Advantage of GBT: If tester found any defect in software then do not need to send back to developer instead of that he/she can solve the issue Because of coding knowledge.

## **6) Maintenance:**

-Maintenance is **service provided after delivered** of the product/Application.

-In this **Mainly Two Types:**

### **1) BPO- Business Process Outsourcing**

-It is **Non-Technical** Department

-**For Example:** Customer Care, HR, Finance and account team....

-**For Example:** If customer found any defect in application then he/she will contact to Customer care of the company.

### **2) KPO- Knowledge Process Outsourcing**

-It is **Technical** Department

**Examples:**

- ✓ Market Research Activities
- ✓ Data Analytics
- ✓ Business Research Services
- ✓ Solution for any issue

**Process of Solving Issue:** if Customer found any defect in Application:

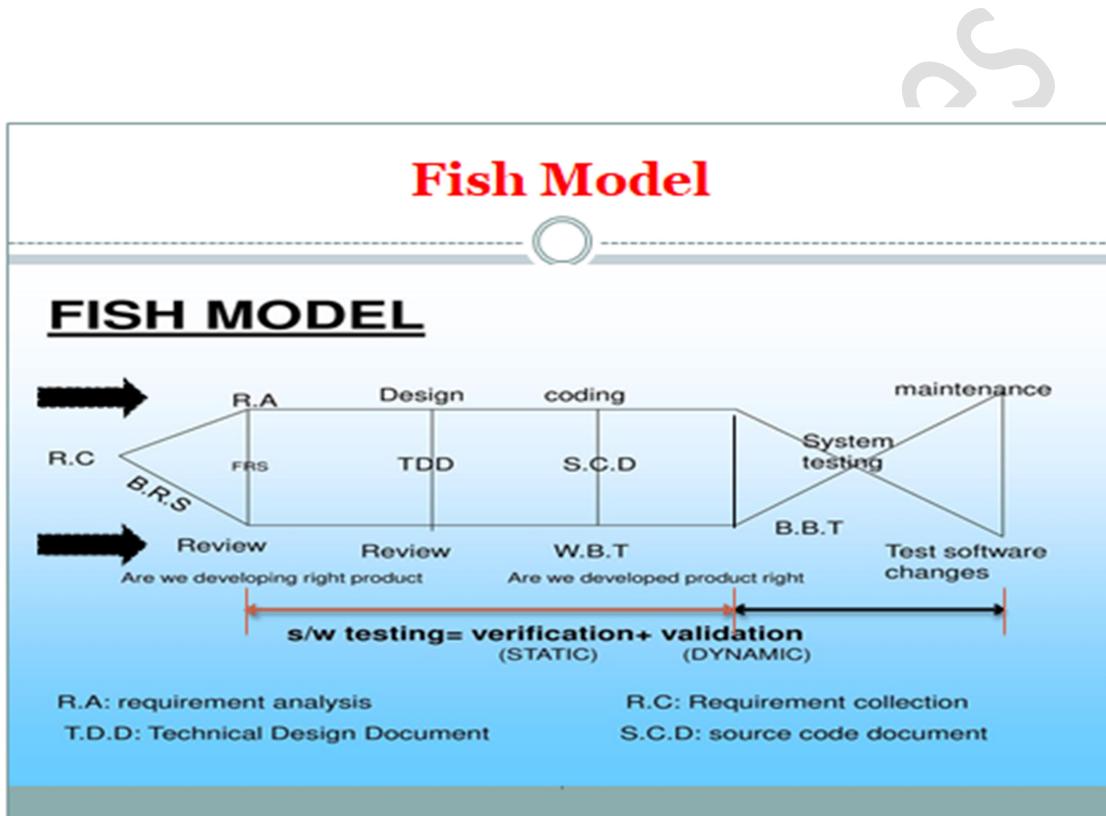
- 1) Customer will contact to customer Care team of the Company
- 2) Customer Care team don't have Technical Knowledge so they passed issue to Technical Team
- 3) Technical Team: Involve BA, Developer, Tester..
- 4) Business Analyst check issue in Requirements and assigned issue to Developer
- 5) Developer Analysis issue and find out the root cause of issue
- 6) If developer found root cause then he fixed/resolved the issue and send issue to Tester for Testing
- 7) Tester then check is issue resolved or not which is raised by Customer
- 8) If issue still there then Tester send issue again back to Developer
- 9) If issue not present then Tester passed for Production/Release
- 10) Upcoming release application delivered to Customer with Required fixes.

## Topic: 5: Fish Model of SDLC

### Fish Model of SDLC

-It is advance version of SDLC.

-In Fish Model just Review Process added in Verification this is only difference Compare to SDLC



-Fish Model Divided in Two Parts

#### 1) Verification

-In this A) Analysis or Requirement Analysis B) Design C) Coding

#### 2) Validation

A) Testing B) Maintenance

## **1) Verification**

-In this stage we just Verify Documents

-It is also called **Static Testing**

-It is also called **Quality Assurance**

### **A) Analysis or Requirement Analysis**

-In this stage **BA** analysis convert BRS Document into SRS Document

-SRS Document involves detail information of Functionality of application.

-After creating this document **BA check whether it document is correct or not** because all further process depends on SRS document.

-This Checking process called **Review Process**.

### **B) Design:**

-In this **HLD** and **LLD** are involve which done by **System Architecture or Design Architecture** and **Front End Developer** respectively.

-Combination of this document called **TDD (Technical Design Document)**.

-In this same System Architecture and Front End Developer check...are design as per SRS Document or not. This is **review Process**.

### **C) Coding:**

-It is done by **Developer**

-In this FED and BED is involved.

-After coding completed developer need to check all code, he need to compile code and check is there any error present or not

-And check whether all code as per SRS or not. This is **review Process**.

-It is also called WBT (**White box testing**)

-It is also known as **Code Level Testing** or **Unit Testing**.

## **2) Validation:**

-After coding completed we check application functionality and correctness and completeness as per SRS Document or not.

-It is also called **Dynamic Testing**

-It is also called **Quality Control**

### **A) Testing:**

-In this main focus on the quality of Application

-Whole Application functionality get checked in the Validation process.

-It is known as System and Functional Testing or Black box Testing.

### **B) Maintenance:**

-It is service provided after the delivery of the product to customer.

-If customer found any issue at the time using application then customer can raise issue to company customer care.

-Then Customer care will forward issue to Technical Team (Development Team, Testing Team, and BA)

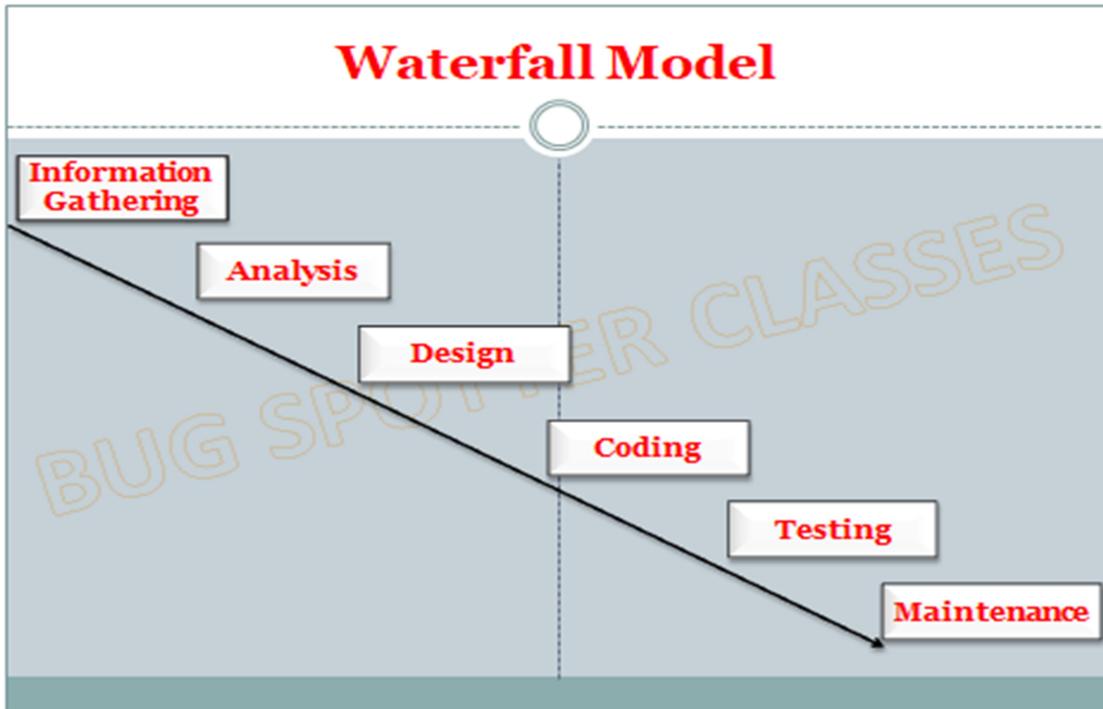
-Then BA assigns issue to Developer and then he solves this issue and send to tester for testing.

-After testing we check is issue present or not. If present then send back to developer or if not then passed to production/release again.

## **Topic: 6: Waterfall Model of SDLC**

>Waterfall Model is **step by step** implementation of SDLC

>This is also called **sequential process model**.



>In this model if **one stage complete then and then we can go to next stage**.

>In this if we one stage completed and moved in next stage then **we cannot return back to Previous stage**.

>**For Example.** We are in testing phase and if tester found defect then he/she cannot send back to the developer instead of doing that tester **log the defect, make report** and then this **issue will be fixed in next version** of the product.

>**Duration** of the Water fall model **3 Months**

>It is used in **Product based companies**.

>**For Example Product based Companies:** Samsung, Microsoft, Google, Facebook, and Adobe

## **When to use SDLC Waterfall Model?**

>**Waterfall Methodology can be used when:**

- 1) Requirements are not changing frequently
- 2) Application is not complicated and big
- 3) Project is short
- 4) Requirement is clear

### **Advantages of Waterfall Model**

- 1) Before the next phase of SDLC each phase must be completed
- 2) Suited for smaller projects where requirements are well defined
- 3) Well understood milestones (Predefined Goal)

### **Disadvantages of Waterfall Model**

- 1) It is not desirable for complex project where requirement changes frequently
- 2) Customer cannot request for change at the time SDLC

### **Companies involve in Product and Service also**

#### **1) Amazon:**

-**Product:** Amazon TV, AWS (Amazon Web Services Cloud based)

-**Service:** IT, marketing, logistics (FBA)

#### **2) IBM:**

-**Product:** lotus notes, baan, db2

-**Services:** Artificial intelligence, Block chain, Business operations, Cloud computing,

Data & Analytics, Hybrid cloud, IT infrastructure, Security, Supply chain

## Difference Between Product Based and Service Based Companies:

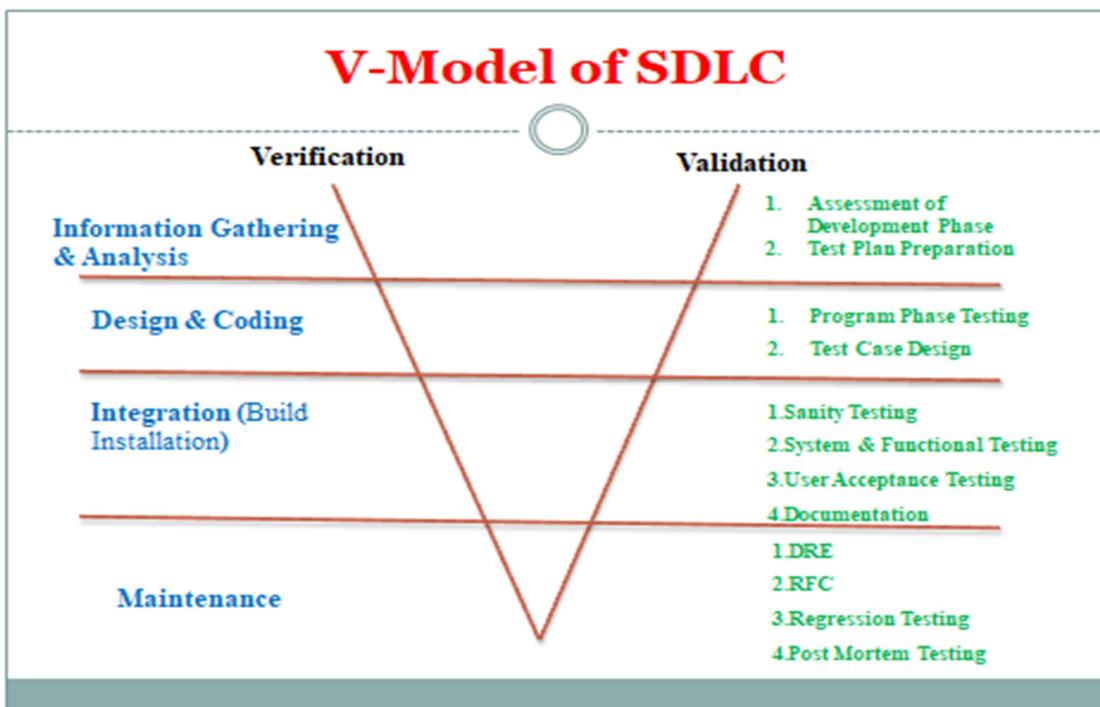
<b>Types of the Companies</b>	
<ul style="list-style-type: none"><li>● <b>Product Based</b></li></ul> <p>I. Own Product to Sell II. Provide Product to directly General People III. Mass Production IV. Made Product before demand of the People V. “Quality” of the product is “King” VI. <b>Example:</b> Google, Microsoft, Facebook, Paytm, Cisco, Adobe</p>	<ul style="list-style-type: none"><li>● <b>Service Based</b></li></ul> <p>I. Don’t have own product provide service to Product based companies II. Provide service as per Customer Requirement/Demand III. Limited Service IV. Provide service as per requirement V. “Customer” is “King” VI. <b>Example:</b> Infosys, Wipro, TCS, Cognizant</p>

Bug Spotter,

## Topic: 7: V- Model of SDLC

### Important Point:

- ✓ V Stands for **Verification**(Development) and **Validation**(Testing)
- ✓ In V-Model Verification and Validation **process perform Parallel**.
- ✓ In this **Development phases mapping** with **Testing Phases**
- ✓ **Advantage of V-Model:** In which customer **can request for changes** at any stage of SDLC but customer need **to pay some extra amount**.
- ✓ Means here we **can return back on any stage** of SDLC as per requirements.  
(Overcome Drawback of Waterfall Model)
- ✓ V- Model is used in **Big Organisation**
- ✓ **Duration** of Project development in V-Model in **3 Months.**



## **Verification:**

### **1) Information Gathering and Analysis**

-Here we can explain general **SDLC point Information Gathering and Analysis**

#### **Information Gathering**

-Customer and Business Analyst (BA) involve in this

-Customer has no. of requirements

-BA gather those all requirements

-Create one document called BRS (Business Requirement Specification)

#### **Analysis**

-In this stage **BRS document convert into SRS** (Software Requirement Specification)

-also called detailed documents

-This is work of Business Analyst.

**Note:** Once Left side point completed of verification then explains **Validation Point** given below.

## **Validation**

### **1) Assessment of Development phase:**

#### **A) Strategy** of the Testing of application will be decided

-Strategy will be decided by Project Manager

#### **B) Methodology** of Testing

-which methodology we are going to use for testing will be decided

#### **-A) Automation Testing B) Manual Testing**

-This is also done by Project Manager

**C) TRM** (Test Responsibility Matrix) will be finalised.

-**Definition:** TRM it is mapping **between testing factor and Development phase**  
(Information Gathering to Maintenance)

-It is also done by Project Manager.

**-Testing Factors:**

- 1) Authorization (This time we require Security Testing)
- 2) Access Control (This time we require Security Testing)
- 3) Audit trial (Data base testing require)
- 4) Correctness (Functional Testing)
- 5) Performance (Performance Testing)
- 6) Easy of use (Usability/Accessibility Testing)
- 7) Easy to operate (Installation Testing)
- 8) Portability (Compatibility Testing)

**Note:** At the time of explaining V-Model just tell definition of **TRM**, no need to tell example of Testing Factors.

**2) Test Plan Preparation:**

-Here **Implementation of TRM** will be done

**A) Resources Allocation:**

-Resources will be finalised.

-**Project Manager** Will prepared test team.

-**For example:** For Security Testing, Performance Testing, Data base testing, Functional Testing & Non -Functional Testing

-For above all types resources will be require and that will be finalised.

## **B) Test Estimation will be done**

- How much time required for complete testing that will be finalised.
- Here we give time estimation with our bandwidth Means some extra time. (because some time testing required more time then estimated.)

## **Verification:**

### **2) Design & Coding**

- Here first we explain same **Design** and **Coding** part we learned in SDLC then move to Validation part

#### **Design:**

There are two Types of Design

##### **A) High Level Design**

For ex: Signup Page, Log In Page, Home Page

- It is related to **Main Module**

- It is developed/designed by **Design Architecture** or System Architectures

##### **B) Low Level Design**

- It is related to **Sub Modules**

-For ex: For Main Module Sign Up Page:- First Name, last Name, EmailID, Mobile Number.....etc.

- It is designed by **Front End Developer**

## **Validation:**

### **A) Program Phase Testing**

- It is related to Developer
- It is similar to **White box Testing**

-Here Developer only check positive Scenario

## B) Test Case Design

### Important:

>>When SRS document created then BA send this document to Development and Testing Team on same time.

>>When Developer doing design, coding part then same time testing doing Test Case Design.

>>**Test Cases** Design means "**How to test**"

-Test cases means multiple steps involved while testing

-Test cases are **mapped** with customer requirements.

>>**For Example:** Scenario: 1 **Check Facebook application loaded or not**

### >>**Steps:**

1) Take the URL of the Facebook Application

2) Open any browser (chrome) and paste the URL in browser search box and press Enter Key

3) Then we check application loaded or not

>>**Test Scenario** Means "**What to Test**"

>We can write multiple test cases for one Scenario.

**For Example:** We have to check Facebook Application is loaded or not

>>In this positive Scenario and Negative Scenario Testing

>>This is work of Black Box Tester

## **Verification**

### **3) Integration (Build Installation)**

- It is process of **adding new module into old Application**. Or
- Developers develop number of module separately and adding in one application called **Integration**.
- In V-Model:
  - Duration of V-Model is **3 Month**
  - In this duration **5-6 module** will be develop.
  - Once development of all modules done then in at the last **integrate/combine** into one application/Software.
  - It is **work of Developer**. Once Developer work completed then he send build or module to testing (For that developer Provide Testing URL)

### **Validation:**

Once application came for testing then first testing we perform **Sanity Testing**.

### **Sanity Testing:**

- It is come under **Validation Process**.
- In Sanity Testing we check only **Core Functionality**.
- Basic Functionality** will be get checked in this Testing.
- In this testing we **only raise Critical Errors** (Defect which Blocked Functionality of the application)
- Means here we verify build is stable or not (**Build Verification**)
- For Example:** Facebook Application: Different Modules: Facebook Logo Spelling, Log IN Button, Create New Account button, Sign Up button.

-**Critical Issue:** Sign Up button not enable (Means not clickable), Facebook Logo Spelling mistake then its Blocker defect **so raise in Sanity Testing.**

## **System & Functional Testing:**

-When **Sanity Testing successfully completed** then and then we start **System & functional testing.**

-In this testing we **check all functionality** of the application **as per SRS Document.**

-In this **we raise small to large defect** will be documented.

-In this Positive and Negative testing will be perform.

-**Black box tester** is responsible for that.

-For **Example:** Facebook Logo color not proper, Signup button color not as per SRS,

-Spelling mistake for text present on page which are not useful for Customer.

-**System & functional testing fine then we passed application to UAT**

## **UAT (User Acceptance Testing)**

Before move to UAT we discuss Different Environment

### **>>Environment of Testing:**

**1) DIT (Development Integration Testing)**

-**Developer involved**

**2) SIT (System & Integration Testing)**

-**Tester Involved**

-When system and Integration testing successfully completed then product move to UAT.

**3) UAT (User Acceptance Testing)**

-**User and Tester involve in this testing.**

**4) Production** (Final product will be deliver to Customer)

## User Acceptance Testing:

-After **Successful completion of System and Functional Testing** product moved to UAT.

-**Tester and User Involve**

-In this **environment release version** will be updated

-**Example:** Suppose Previous Version Chrome is 90.0 and if some new features added in application or may be some issue will fixed then this changes need to pass Customer and for that need release updated version.

-**Release** has two Types mainly

### 1) Minor Release (with some changes or Bug Fixes)

>So if Minor Release for Chrome Browser then Version will update from **90.0 to 90.01** when came in UAT and

### 2) Major Release (with Lot of Changes and Bug Fixes)

> If Major Release for Chrome Browser then Version will update from **90.0 to 91.0** when came in UAT.

**Note:** You do not need to mention above two point related release while explaining UAT. It is just for your information.

-So in UAT Tester test application with User/customer. Some time customer have their test data so tester will test application with that data.

-In this environment we check UI, Design, color and Functionality of the application as per SRS Document. Less defect found in UAT.

-Mostly customer not request for change at the time UAT because of less Time bandwidth of Release date.(Release date already fixed)

-If customer request for change at the time UAT then BA take decision as per Time Bandwidth.

-When **user gives the permission** then Product will be **moved to Production**.

## **Documentation/Test Documentation**

- Test Documentation is report of Testing.
- Each tester has its own test report.
- Whenever tester did testing on Module then he/she create document of testing.

**Test Report involve following point.**

**1) Name of the Module**

Ex. Sign Up Module

**2) Test Cases designed count**

Suppose Designed: 20 Test Cases

**3) Test Cases Executed**

Suppose Executed: 20 Test Cases

**4) Passed**

Suppose Passed: 19 Test Cases

**5) Failed**

Suppose Failed: 1

-So we create bug ticket of failed test cases.

-We create this report in Excel Sheet and after **that send to Team Leader**

-Then Team Leader sent this report to **Project Manager/Test Manager**

-Project Manager sends this **document to Customer**.

## **Verification:**

### **Maintenance**

-It is service provided after delivery of application to customer.

-In this Mainly Two Types:

#### **1) BPO- Business Process Outsourcing**

-It is Non-Technical Department

-For Example: Customer Care, HR, Finance and account team.

#### **2) KPO- Knowledge Process Outsourcing**

-It is Technical Department

-for example: Market Research Activities, Data Analytics, Business Research Services, and Solution for any issue

## **Validation:**

### **1) DRE (Defect Removal Efficiency)**

-DRE is the process of calculating at which level tester testing did.

-In that Tester efficiency will be calculated.

-Tester Efficiency will be calculated by below formula,

**Formula for calculate DRE=A/A+B**

-A=Defect found by **Tester** (at the time of user acceptance testing)

-B=Defect found by **User/Customer** (at the of user acceptance testing)

**Suppose:** at the time of UAT

-Tester found 60 defects

-Customer found 30 defects

$$\text{DRE} = \frac{60}{60+30} = 0.6 \text{ (Avg. Testing)}$$

**If DRE**, then Remarks,

1) 0.8 to 1= **Good Testing**

2) 0.5 to 0.8= Avg. Testing

3) Below 0.5= bad testing

**Suppose,**

Tester found 30 defects and User found 50 defects

$$\text{DRE} = \frac{30}{30+50} = 0.375 \text{ (bad Testing)}$$

**Suppose,**

Tester found 90 defects and User found only 10 defects

$$\text{DRE} = \frac{90}{90+10} = 0.9 \text{ (Good Testing)}$$

## **2) RFC :( Request for Change)**

-In V-model **customer can request for change** at any stage of the SDLC

-If customer request changes then **BA takes those changes** and add in the **SRS Document**.

-SRS Document one Section "**CR**" Means **Change Request** in Red color

- It is also called as MR (**Modification Request**)
- But for this changes **customer need to pay some extra amount**
- For Example:** Facebook Sign Up page
  - If customer want to add Nick Name field below Name field then BA and took Customer request and add this in SRS Document in **Red Color** but for this changes customer **need to pay some extra amount.**

### 3) Regression Testing

-Regression testing is **process of checking is there any adverse effect** on application due to some **new changes added or Coding changes for fixing the issue.**

- 1) Means we performed Regression testing if some **new feature/module added** in old application
  - Means in above **example** of Facebook Sign up: Nick Name feature/module added in Sign up page. For that Developer added codes
    - Because of that coding is there any functionality braking of application we check through Regression Testing.
- 2) Some time we found defect and for resolving/fixing that defect **developer change the existing code** so due that might be chances brake functionality of application so that we check through Regression testing.
  - We performed this "**Regression Testing**" After **SIT** and **UAT** because changes will be added in this environment.

#### **4) Post Mortem Testing:**

- In this testing **Developer Involve**.
- At the time of Testing we **found product not delivering desired output** which customer want that time we create bug ticket and send to developer.
- Then **for finding out the root cause of defect** Developer need to check whole coding did by him. So **process of checking all codes** called **Post Mortem Testing**.
- It is **very time consuming** process.

**For Example:** Suppose we have to **buy Mobile from Amazon.com**

- 1) So searched Specific Mobile and added in Cart
- 2) Then clicked on Payment detail for paying amount
- 3) Added all required information Account Number, Mobile Number, Amount....etc. and Clicked on Place Ordered
- 4) But due to some reason order not placed
- 5) Then that time we raise this issue to developer and for find out to root cause of issue developer need to **Post Mortem Testing**.

## **Topic: 8: Agile Model of SDLC**

### **Agile Model/Agile Methodology**

-Now days **Agile methodology** is very famous in **Service based** company because of its number of advantages

- 1) Agile Methodology is **Module Driven** Methodology.
- 2) It is **not plan driven** methodology.
- 3) In Agile Methodology **Customer can request for change at any stage** of the SDLC.
- 4) For any new changes **customer do not need to pay extra amount.**
- 5) Changes in the requirement **do not effect on the development of other modules.**
- 6) Agile Methodology it is a **Value driven** methodology.
- 7) **Sprint Duration** of the Agile Methodology is **1 Month** (Change Company wise for 1 week, 2 week, 3 week or Month)
- 8) **Agile Methodology** used in **Service based Companies.**

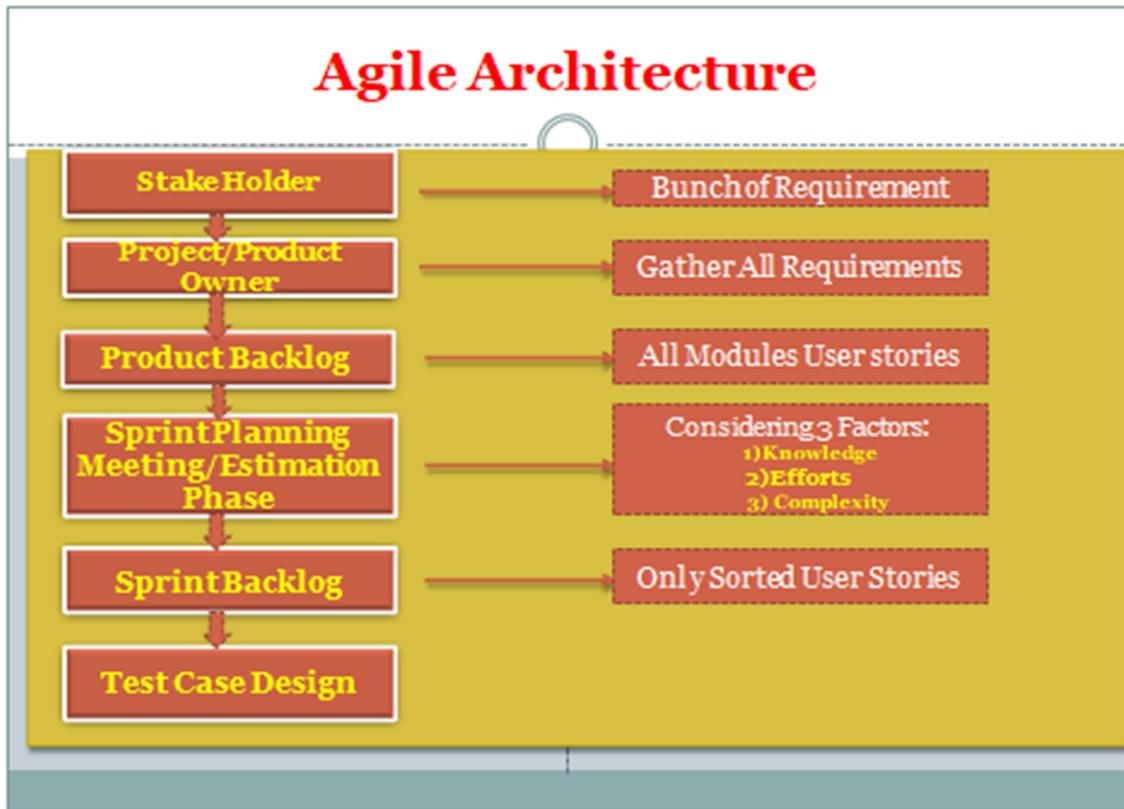
For **Example:** Suppose there is software project having 8 Modules

-Then in Agile Methodology project divided in group of modules means

-In Phase-1: 2 Module developed and released to customer

-Then Phase-2: other 2 Module developed and released and same for last four module.

# Agile Architecture



## 1) Stake Holder

- Stake holder nothing but **Customer/User**.
- In Agile Methodology Customer is **member of top most body of Company**.
- Stake Holder can request for change at any stage means Development, Testing, and Production.
- They have **bunch of Requirements** (Nothing but **User Stories**)

## 2) Project Owner/Product Owner

- Project Owner **gathers all requirements** from the Stake Holder
- Project owner is **member of Sprint planning meeting**.
- Project owner responsible for **creating product backlog**.

### **3) Product/Project Backlog:**

- Project backlog are the **total requirements** for **whole product/project**.
- It includes **requirements of all modules**.

### **4) Sprint Planning Meeting/Estimation Phase**

- In the agile methodology the focus is on **module wise release**.
- It is **process of sorting requirements** to development of module.
- Involvement:** Project Owner, Scrum Master, Development Team, Testing Team
- In this meeting only those user stories select **which will be developing in next sprint**.
- Selection of user stories depends on following factors:**

#### **1) Knowledge:**

- Whenever team formation done then each member of team should have domain knowledge of project is consider.

#### **2) Efforts:**

- In this Higher Management decide how much efforts require for project.
- How much **resources**, people (Developer and Tester) require for project is consider.

#### **3) Complexity:**

- For estimation of **time, cost and resources** first complexity of the Project will be consider.

### **5) Sprint Backlog**

- Created by Project Owner
- Sprint backlog contains sorted user stories at the time of Sprint planning meeting.
- Sprint backlog contains details information of requirements which are going to develop in next sprint.

## **6) User stories:**

- User stories nothing but **functional requirements** for the module development.
- User Stories** get decided at the time of Estimation/Sprint planning meeting.
- It consist “**Description**” and “**Acceptance Criteria**”
- Description** Means **what as user he wants to do** (Process) and **what it's desired output**.
- Acceptance Criteria** Means are the **Scenarios** when these **scenarios** are correct then system **generate correct output** otherwise **system show failure**.

## **7) Test case Design:**

- Once Sprint backlog ready then **Project owner** sends it to **Development Team** and **Testing Team**.
- Then **Developer** starts **their coding work** and same time **tester start test case design**.
- Test cases** are mapped with **User Stories**
- Tester is responsible** for **Test Cases Design**.

## **Topic: 9: Advantages of Agile Methodology:**

-There are number of advantages of agile methodology I will explain some from them,

### **1) Check Point:**

-**Example:** Consider there are four module developed M1, M2, M3, and M4...

-Then Check point will be added between the two module means Between M1 and M2, Between M2 and M3...same for M3 and M4

-If at the time production or after production error occurred in application or Module then **tester do not need to test all modules** instead of that tester can first just test **Check point provided between Module M1 and M2** if error not present then test other check point provided between **M2 and M3.**

-If tester found issue between M2 and M2 then tester will raise the same check point to Developer.

-Then developer just **finds out root cause for Check point** where issue occurred and fixed issue. But in **V-Model** Developer **need to check all modules** codes that called **Post Mortem Testing.** It is very time consuming Process.

-For **testing Check point** Tester Used "AVAS" Tool.

-Check point save time of both Tester and Developer.

### **2) Scrum Meeting:**

-It is also called **Daily Stand-up/Daily status Call Meeting.**

-**Time Duration:** 15-30Min

-Mostly happened between (10:30-11 or 11 -11:30 AM)

-**People Involved:** Development Team, Testing Team, Product/Project Owner, Scrum Master

-**Chair Person of Meeting:** Scrum Master

**Agenda of Meeting:** "What is progress of Meeting".

-**Three Main question** asked in meeting:

**1) What we did yesterday?**

-It is the report/status of previous work which was completed by team members whether they are Tester or Developer.

**2) What we are going to do today?**

-It is pending work which have to complete today or any new work going to start today.

**3) What are the road blocks or issues?**

-If Developer and Tester facing some issue

>If requirement not clear for developer or tester

>If tester found any issue

**3) Implementation of Automation:**

-We can implement Automation in Agile Methodology.

-Automation is **Time saving Testing technic.**

-**For Example:** Facebook Page

-If in Sprint-1 we completed Manual Testing for **Sign Up page** then in Next Sprint we will automate this Sign Up page.

-So next when Sign Up page came for testing with some new changes then we do not need to do Manual Testing for all Fields, that time we will just execute Automation Script and only new changes will test manually. We will **get direct pass, fail count of test cases through Automation Script.**

-So here our **Time is definitely save.**

-For Automation mostly used **Selenium** tool which free sources available in market.

### **Advantages of Automation:**

- 1) Less Resources Required
- 2) Less Resources means less cost
- 3) High Accuracy
- 4) Less Human Errors
- 5) Time saving testing Method

### **4) Sprint Wise Delivery:**

- In Agile Module wise delivery possible because in this we develop small module and developed module can send to Customer within Less time of duration (Means Sprint wise)
- Means suppose total 5-6 module we have to develop within sprint but we just developed 3 modules. Then **only developed module we can send** to customer as per their requirement that's why also called **Value driven Methodology**.

### **Types of Framework used in Agile Methodology:**

- 1) Scrum
- 2) X-treme Programming
- 3) LEAN
- 4) KANBAN
- 5) Dynamic system development method
- 6) Future driven development method

-From above list Agile Methodology mostly used **Scrum Framework**:

## **Scrum Framework:**

- It is framework used to **implementation** of the agile software development Methodology.
- Scrum is framework in which **team solves complex problems** and **comes with productive solution** and **Innovative solution**.
- Scrum is used** where **Project Requirement** continuously changing.
- In Scrum Framework **Project is divided into small no. of requirements** means **Sprint**, Means by using **Scrum Framework** sprint wise delivery possible.
- Scrum is **Module driven** not **plan driven** that's why we use this.

## **Topic: 10: Integration Testing & Sanity Testing**

### **10.1 Integration:**

- It is **process of adding new module into old Application**. Or
- Developers develop **number of module separately and adding in one application** called **Integration**.
- When white box testing over, developer have to do integration
- Integration is **process of mapping new module with old module/application**.
- Developer should have knowledge of **functionality, relation, dependency** of modules over each other that's why developer does Integration.
- In Software **one module output is input for other module**.
- When developer did integration then he also do Integration Testing.

## **10.2 Integration Testing:**

-It is **Developer Task**

-It is **process** of checking **correctness** and **completeness** of the flow of functionality whenever **integration of modules** completed.

**-Integration has two types:**

### **1) Front End Integration (FEI)**

-Related GUI (Graphical User Interface) or UI (user Interface)

-In **front end developer** connects **two modules** using "**Called**" Function.

-Ex: M1 and M2 connect by using called function.

### **2) Back End Integration (BEI)**

-Related to data bases.

-Back End Integration **connect two or more tables** in data bases using **JOIN Query** (Full join, Left join, Right Join)

**-Example:** T1 and T2 (Table) connect by using Join Query

-Whenever Integration of modules done then **Integration testing start by Developer for that** he use three approaches

## **10.3 Testing Approaches:**

1) Top Down Approach

2) Bottom Up Approach

3 Bi-Directional Approach/Hybrid Approach

### **1) Top Down Approach:**

-If Developer have to do integration testing & he developed module but don't have next Module which can check correctness of new Module.

- When we have **Main Module** (Sign Up Module) but **don't have Sub Module** (Next Module) in that condition developer use **Top Down Approach**.
  - In that condition developer developed/create **dummy module** called "**Stub**"
  - Stub is created by using **XML Language Programming**.
  - Stub is checked by using **SOAP UI** (Simple Object Access Protocol User Interface) tool.
  - URS (User Requirement Specification)** for Stub is **WSDL (Web Service Description Language)**
  - Request and Response in XML (**Extensible Markup Language**)

## **2) Bottom Up Approach:**

- When we have **sub module** (Next Module (Login Page)) but **don't have Main Module** (Sign Up Page) then use **Bottom Up Approach**.
  - In that condition developer creates one **dummy module** by using xml programming language called "**Driver**".
  - Developer creates **dummy Main Module** for testing.
  - This "**Driver**" programming in XML Language.

## **3) Bi Directional Approach:**

- It is combination of Top Down and Bottom up approach
- For Example:** If developer developed Login Page and he have to check functionality with Sign UP and Home page that condition developer create **dummy module of Sign up Page** by using **Driver** and **dummy module of Home page** by using **Stub**.
- It is also known as **Hybrid Approach**.
- After the Successful Integration Testing completed then Developer passed Application URL to Testing team and also provide some testing instructions.
- Once Application came to testing after integration then first testing tester do which is Sanity Testing.**

## **Topic: 11: Sanity Testing:**

### **Aim:**

- To check **Core/Basic Functionality**
- Also called **Tester Acceptance Testing.**
- Also **Build Verification Testing**
- Also called **Zero Level Testing.**
- In this testing only “**Critical Error**” get documented/raised to developer.

### **In Sanity Testing we do mainly testing for:**

- 1) Basic Core Functionality
- 2) Tab Validation
- 3) Link Validation
- 4) Page Validation
- 5) GUI Validation

#### **1) Basic Core Functionality:**

- In this tester test **buttons, icons**, from which user can proceed to next stage.
- Example:** Submit button, Sign UP button, Login Button. (We check it is clickable or not)

#### **2) Tab Validation:**

- Tabs are nothing but text boxes in which we enter the values.
- Example:** First Name, Last name, EmailID, Mobile number...field.

- In this we check Tabs are **enable or not**.
- Whenever we entered any value in text box through Keyboard then those char, symbols, number should be entered in Text Box.
- This Functionality we validate in **Tab Validation**.

### **3) Link Validation:**

- Link Validation means is this **process sequence of interlink pages get tested**.
- Example:** Flipkart Application:
  - There is link on Flipkart Home Page for Mobile Page: When **user clicked on that then Mobile** informative page should be open (That arrangement done by Developer).
  - So we need to check here **Mobile Link** is active or clickable.

### **4) Page Validation:**

- Page Navigation means Navigation Validation
- In this process when we **clicked on next and back button** then **should be navigate on front & back Page respectively**.
- This testing also called **Navigation Testing**.

### **5) GUI (Graphical User Interface) Testing:**

- This testing test the **interface with which user interact directly**.
- In this test, tester check whether the **pages displayed correctly** or not, **Image should be clear not blur**, **Page should be full loaded**.
- This is **Validation of visualisation**.

**Note:**

- 1) So whenever interviewer asked you **what is Sanity Testing** then Explain with **definition** and **above types in that**.
- 2) When Interviewer asked **what is difference between Sanity Testing and Smoke Testing** then
  - just tell them actually **there are no more differences in Smoke and Sanity Testing** and **our company we are using Sanity Testing** so I have good Idea about it.
  - If you asking about difference then I knew one difference...then Explain below definition and Explanation,

## **What is Smoke Testing?**

- We can say it is **similar to Sanity Testing or Advance version of Sanity Testing**.
- Actually **Sanity and Smoke Testing** term used by different **Organizations as per their Benefits**.
- It is **Combination of Sanity Testing + Package Validation**.

## **Package Validation:**

- It is a Combination of objects.
- Example:** Facebook: If in Sign up we entered all required information in Tabs/fields-
  - Then clicked on Sign up button then all filled information will get saved in data base and we will navigate to Log In page.
  - When we entered valid log In credentials in log In Page field and clicked on Log In button then get errors occurs.
  - Then **developer need to check from which package this issue happened**. That's called **Package Validation**.

## **Topic: 12: Functional Testing**

### **Functional Testing:**

- It also called **System and Functional Testing**.
- It is process of checking **correctness and completeness** of functionality of application.
- In this we check **internal functionality** depends on **external functionality**.
- Means whenever **client/user enters data in any field** it should be get store in data base.
- In this we execute test cases written by us.

In this we check internal Functionalities, means In this we check different Coverage's:

- 1) Behavioural Coverage
- 2) Input domain Coverage
- 3) Error Handling Coverage
- 4) Service Level Coverage
- 5) Calculation based Coverage
- 6) Back End Coverage

#### **12.1 Behavioural Coverage:**

- In this we Check Property & Behaviour of the object

#### **Example: 1: Text Box**

**Property:** It should accept user input

**Behaviour:** Focus and Non-Focus, Means when we clicked on Text Box then should focusable when remove focus it should un-focus.

## **Example: 2: Dropdown Box**

**Property:** It should hidden list

**Behaviour:** show/hide list

## **Example: 3: Check Box**

**Property:** do check when user click

**Behaviour:** checked/unchecked.

## **12.2 Input domain Coverage**

-It Check **Type** and **Size** of the input

-**Type** means **DataType** of input

-**Size** means if mobile number tab is there then size will be 10.

-In this Coverage we maintain **BVA** (Boundary Value Analysis) & **ECP** (Equivalence Class Partition)

-**BVA** check **Size of the input**

-**ECP** Check **type of the input**

### **Example: 1: Mobile Number Field:**

-**BVA:** Min=10, max=10, min-1 (not Accepted), min+1(not Accepted), max+1(not accepted), max-1(not accepted)

-**ECP:** Invalid and Valid input check:

>**Valid:** 0-9 number

>**Invalid:** a-z, A-Z, Special Symbol, (\_), Space

**Example: 2:** First name field should accept 4-6 char

-**BVA:** min=4, max=6, min-1(not accepted), min+1(Accepted), max+1(not accepted), max-1(Accepted)

-**ECP:**

-**Valid:** a-z, A-Z

-**Invalid:** 0-9 number, Special Symbol, Space

**Example: 3:** Example of online transaction using Debit Card (So find out BVA and ECP for Debit Card number and CVV Number)

### **12.3 Error Handling Coverage**

-In this we check whether system generate error message or not.

-If we entered invalid information in field then system should be generate error message.

**Example:** Mobile Number Field: If we entered less and more than 10 digit in mobile number field then system generated message

"Please enter Valid Mobile Number".

### **12.4 Service Level Coverage**

-In this we check working of system as per functional flow diagram or not.

-In this Coverage check sequentially of functional module.

-Ex: If you are filling online examination form

-Then, 1) Personal Information Detail 2) Contact Details 3) Academic Details 4) Banking/Payment details 5) Submit

-In above example once one stage completed then system should allow to next stage.

## **12.5 Calculation based coverage:**

-In this we check arithmetic operations.

-**Arithmetic operations:** Addition, subtraction, division, multiplication

**Example:** On Amazon.com If we added **Vivo mobile** to cart & its price **20K** then & again added new **Samsung Mobile & Samsung Air Conditioner** price **18K** and **30K** respectively. Then it will be added in to cart then total price is **68K** (It is done by **Addition**) and if we cancelled/removed **Vivo Mobile** from Cart then total is **48K** (It is done by **Subtraction**)....So here we check this operations.

## **12.6 Back End Coverage:**

-The backend of any software is **data base**.

-IN back end coverage we check whether entered data stored successfully in data base or not.

-We also check whether data get fetch from data base.

-By using **SQL Queries**.

**Example:** -If we filled online examination form by adding all required information then clicked on Submit button.

-Then all data save in data base. Then we check we are able log in or not by using **User Name and Password**.

-We can also fetch specific person/candidate information from data base.

## **Topic: 13: Non-Functional Testing**

### **Non-Functional Testing:**

- In this fast pace world of Technology development software companies are **not only focused on bug-free product** but also for an **excellent performing product**.
- In Functional testing **check the correctness of internal Functions** while in non-functional testing **check the ability to work in External Environment**.

#### **13.1 Recovery Testing:**

- It is process of checking whether system/Application is able to **recover from abnormal condition/Situation to normal condition/Situation**.
- Also called **Reliability Testing**.
- Recovery requirement are given by customer.
- Customer can give requirements that he wants that **application should recover from specific point or from start point**.

**Example:** Suppose we Downloading large music file and suddenly internet connection lost, then downloading paused, When Internet connection return back then System should start downloading of music from paused position. (**This is customer Requirement**)

- If Music downloading start from starting then there will customer data lost.

#### **13.2 Compatibility Testing:**

- It is process of checking **whether application/system compatible with user expected platform (Browsers, Operating System)**
- Browsers:** Chrome, Mozilla Firefox, IE, Safari
- Operating System:** Windows, Linux, Mac

-Also called **portability Testing**.

There are mainly two types of Compatibility Testing:

### **1) Forward Compatibility Testing**

-If **build/application is correct** but **browser/operating system do not working properly** then it is Forward Compatibility Testing.

-We found **less error** in this testing.

### **2) Backward Compatibility Testing:**

-If **browser/operating system is ok** but **build/application is not working properly** then it is Backward Compatibility Testing.

-We found **maximum error** in this test.

**\*Note:** In interview just explain below types, no need to explain above types.

-In Compatibility Testing I am involve in **Browser Compatibility Testing**.

**Browser Compatibility Testing have two types**

#### **A) Cross Browser Compatibility testing**

-In this process tester **check application on different browsers**.

**Example:** Chrome, Mozilla Firefox, opera mini, Edge, IE, Safari...

#### **B) Version Comparison Compatibility Testing:**

-In this process tester **check application on different version of same browser**.

**Example:** Chrome: 94.0, 94.01, 94.02.....or IE 11, IE 10, IE 8...

### **13.3 Inter System Testing:**

-It is process of checking whether **our application able to share resources with other application or not.**

-Means we check **our application share data with other application** or not.

**Example:** 1: Airtel Customer Recharge with Phone Pay Application

-We enter our mobile number in Phone Pay for plan search then clicked on Search button...Then you will get plans in result. so here **Airtel server share data with Phone Pay server.**

**Example:** 2: Withdrawal of money from other bank ATM.

-Suppose you are SBI Bank Holder and you have to withdraw money from SBI ATM. You went in SBI ATM but in **SBI ATM Cash not available.**

- Then you went in ICICI bank ATM and inserted ATM Card in machine and entered Password then **ICICI server fetches data from SBI Bank server** and proceeds for Money withdrawals.

### **13.4 Globalization Testing:**

-It is process of checking whether **application is support different languages or not.**

-also called multilingual feature checking.

**Types:**

#### **1) Localization Testing:**

-Check whether application **support local languages like**

**Examples:** Marathi, Telgu, Kannad, Gujrathi, Punjabi...

## **2) Internationalization Testing:**

-Check whether application **support official languages of Countries**.

**Examples:** Hindi, China, French, Germany.... (Country wise)

## **3) Globalization Testing**

-Check whether application supports **Global English Language**.

-Also called **G11N Testing** means **GlobalizatioN testing**

-Whenever user change language, language should be change **but Number should be in English** (It is user requirement)

## **13.5 Sanitation Testing:**

-It is process of checking is **there any extra features added by developer or not** (Which is not mentioned in Sprint Backlog) customer requirements.

**Example:** Mobile Number field

-User requirements was **just create 10 digit mobile number field** but **developer added +91 field front of Mobile number** as per his Previous Experience.

-So at the time testing **tester found this extra feature**. So will **create bug ticket** for that and assign to developer.

-Then on that extra feature **discussion will happen in Daily Stand-up meeting**.

-Then **PO will contact to Customer** and ask about this extra feature **if customer want this extra feature then Customer need to pay extra amount** for that.

-**if customers don't want** then developer **need to remove this extra feature**.

-It is also known as **Garbage Testing**.

## **Topic: 14: Miscellaneous Testing**

### **14.1 Usability Testing:**

- It is process of **checking user friendliness with our Application** or build.
- In usability we check **system should take less number of events to complete task** or easy validation.

**Example:** If user clicked on submit button then App should immediately open next page quickly.

**In Usability we do mostly two types of the testing.**

#### **A) GUI Testing (Graphical user Interface)**

- This testing tester test the interface that user interact directly.
- Ease of use
- Speed of Processing
- Test all the functionality of elements present on Web Page
- Means Pages displayed properly or not, image should not be **blur**, Page should be fully loaded...
- It validation of visualization.

#### **B) Accessibility Testing:**

- This testing performed for **Blind user perspective**.
- When user clicked on any button and navigated on any elements present on webpage then it system gives voice feedback.
- Tool used** in For Accessibility Testing in Company: 1) **NVDA** (Non Visible desktop access) Tester use this tool 2) **Jaws Reader**
- Developer use Ace Tool for Testing.** (This tool gives how many serious defects, how many critical defects, high defect)

## **14.2 Security Testing:**

-It is process of **checking privacy related to user operation**.

### **A) Authorization:**

>Process of checking whether person is authorized or not.

>Authorized person is registered person.

### **B) Access Control:**

-It is process of checking whether authorized person has permission to access Specific Operation/Application.

### **C) Encryption & Decryption**

>>Cryptography is used to secure and protect data during communication.

>>It is helpful to prevent unauthorized person or group of users from accessing any confidential data.

>>Encryption and decryption are the two essential functionalities of cryptography.

- 1) A message sent over the **network is transformed into an unrecognizable encrypted message** known as **data encryption**.
- 2) At the receiving end, the **received message is converted to its original form** known as **decryption**.

## **14.3 Performance Testing:**

-Performance testing is performed to evaluate the performance of the components of a particular system under a particular work load.

-Performance Testing Attributes: Speed, Stability, Reliability

-Performance testing it is process of checking speed of processing of our application/build.

-Tool: J-Meter, NewRelic, Load Runner

-We can say **actually I am not involve in performance testing** because our company have **separate team** for that.

#### **14.4 Re-testing:**

##### **What is Re-testing?**

**-Re-testing** is process of **re-executing same build/application with multiple test data**.

-We performed **retesting two times**:

- 1)** Suppose we found defect then before creating bug ticket we retest same defect with multiple data.
- 2)** After developer solved defect and assigned us. Then again we retest that application and check whether issue still present or not.

**Suppose** we have to log In into Facebook application but we don't remember password. Then that time we trying to login different password which you used previously. **same concept used here for retesting.**

##### **What is difference between Retesting and Regression Testing?**

Main difference between **Retesting and Regression testing** is in Retesting we just retest raised bug resolved or not (Means here we just check raised issue)

**But** in Regression testing we check **is there any adverse effect on other modules or not** due to bug fixing. (Means here we check all modules)

## **Topic: 15: UAT Testing**

### **UAT (User Acceptance Testing):**

It is **process of collecting feedback from customer.**

**People involved:** User/Customer and Tester

**Tool Used:**

- 1) **Q-Messenger** for Desktop Sharing
- 2) **AVAS & Bharat Mantri** for Search Session ID
- 3) **JIRA** for Raising bug ticket or for internal communications

-Also known as **End to End Testing**. (Because here we check **Front End and Back End** with **API as well**)

### **Important Points:**

- 1) UAT starts after successfully completion SIT.
- 2) In UAT version will be update.

**Example:** Facebook current version 10.0 After SIT when it's came in UAT then version will update as 10.01(Minor Release), 11.0(Major Release)

- 3) Here user will decide from total user stories, how many user stories he want to test from Tester.
- 4) Tester then checks all test cases related to that user stories.
- 5) Customer can change test data and it is on Customer/user whether to send application for production or not.

## **Process of UAT:**

- 1) Testing Team share Desktop to user by using **Q-Messenger**.
- 2) Tester Start the Test cases execution written by him/her.  
->Some time User provides data for testing then that time we use that data for testing.
- 3) After completion one test case execution clear cache, cookies, history and then move next test case execution.
- 4) When one test case execution completed then tools gives session ID in that contains all the action performed at that session.
- 5) When we get session ID then copy it and use "**Bharat Mantri**" or "**AVAS**" tool for **searching log file**.
- 6) Once we added **Session ID** in above tool we will get **log file**.  
-**Log file** is available in **.txt format**  
-**Log file** has **front end operation** and **back end operation**.
- 7) Tester sends this **log file** to development team by using **Q Messenger**.
- 8) Then **Front End Developer** checks whether **front end data match with back end or not**.
- 9) After that **Back End Developer** check database using **SQL Queries & fetch data to see really it get stored or not**.
- 10) When all process successfully completed then User give permission to for release the product/Application.

**In UAT have two Types:**

### **15.1 Alpha Testing**

-Alpha testing used in Service based companies.

-It is conducted in **control environment**. (Means here **tester** and **User** involved)

**Example:** HDFC, HSBC, IDBI, ICCI

### **15.2 Beta Testing**

-Beta testing is used in Product based companies.

-Beta Testing conducted in uncontrolled environment (**User directly not involved** in UAT)

**Example:** Microsoft, Rupay, Master Card, and Visa

## **Error, Defect, Bug, Issue**

**Error:** Mistake in Program is called error.

**Defect:** when tester found mistake (error) then it is called defect.

**Bug:** When developer accepts it is an actual defect then it is called as "Bug".

**Issue:** When developer faced/found difficulty to solve bug then it is called as "Issue"

## **Topic: 16: Production Issue**

### **Production Issue/Hot Fix**

-When after production customer found any defect in the product then that is called Production issue or Hot Fix.

**Tool:** JIRA for Communication & raising ticket

#### **Important Points: Procedure for solving Production Issue**

1) If customer **found any major or just defect** then **raised issue to Customer Care** of the company.

2) Then Customer care team **raised ticket to Technical team** because they **don't have technical knowledge**.

**-Technical Team:** Product/Project Owner, Development Team, Testing Team, Scrum Master

4) Product owner will **assign issue to developer**

5) Then developers find out the **root cause** and **then fix/solve issue**.

6) After fixing issue **developer sent to testing team** for testing.

7) Tester test the **issue...is present or not**. If issue still present then again raise to development team or reassign to development team & If issue **resolved then we pass for production**

## **Topic: 17: Priority & Severity**

### **17.1 Severity:**

- Severity is **always related to functionality of the application.**
- Severity means **how much system/application is getting affected because of the defect.**
- Severity level will be decided by Testing team because tester knew about the seriousness of defect on functionality of product/application.

### **Different Severity Levels:**

**Critical:** If a defect causes the termination or complete shut-down of the application, then it is "Critical".

**High:** If the defect results in **the termination of the system but there exist one or**

**More alternative methods to achieve the desired results** or use the system, then the defect is said to have the level "High".

**Medium:** The bug will be marked as "Medium" when the defect in the system **does not cause the program to terminate but produces results that are not correct or inconsistent.**

**Low:** A defect is marked as "Low" when the **usability or functionality of the system is not affected much** but must be fixed.

- The results are obtained by small corrections and there is no break-down of the system caused by the defect.
- Defects that are related to **the look and feel of the system** are given the severity "Low".

## **17.2 What is Priority?**

- Priority nothing but **which defect should be fix first.**
- Priority is considered from the **customer's point of view.**
- Priority **indicates how soon the defect needs to be fixed** by the developer.
- Priority is **set by the product Owner/customer** and it determines the **time frame given to the developer to fix the bug.**

### **Different Levels of Priority:**

**Low:** A defect that can be **deferred** or **fixed** in the later stages once the higher priority ones are fixed, as **it is not serious from the requirement point of view** is of **low** priority.

**Medium:** A defect that **needs to be fixed during the normal course of development activity** is given the status as "**Medium**".

-Such defects occur when a particular feature cannot be used the way it should be **because of some environmental issue, defect in the program, or some code that has to be added.** Usually, these **defects are fixed and delivered to the testing team** as a part of a new release.

**High:** Those defects that **need to be fixed as soon as possible so that the testing team can continue with the testing** are said to be of **high priority**. The core functionality fails as a result of such defects and the system cannot be tested or used until the defect is fixed.

### **Who decides the Severity and Priority of a Defect?**

The organization decides the standards regarding who sets the priority and severity of a defect.

- However, in most cases, **the severity type of a defect is set by the tester based on the product functionality and the written test cases.**
- The priority is decided by the product Owner** based on **customer requirements.**

### **-Understanding with examples**

Let us try to understand **severity and priority by considering an e-commerce application like amazon.com**

#### **1) Example of High Severity and Low Priority**

-Suppose the tester clicks on the “**Privacy Notice**” **hyperlink** at the bottom of the amazon.com homepage and the page is not displayed.

-This defect will be of high severity because the functionality is not working.

-The priority is low because people do not normally spend time reading the privacy notice.

#### **2) High Severity and High Priority**

-You logged in to your **amazon.com account, add items to the cart and click the “Proceed to Checkout” button.**

-You **make the payment and the system crashes**. This defect makes the **whole buying functionality unusable and so the severity is high**.

-The basic purpose of amazon.com is to buy and sell products and most of the customers are affected by this.

-So, **this defect is of high priority** which must be fixed immediately for the buying process to work.

#### **3) Low Severity and High Priority**

-Suppose, that in the amazon.com website, the logo is displayed as **”amazn.com”** with the letter “**o**” missing.

-This **defect does not affect the buying/selling or any other functionality** in any way.

-So, the **severity of this defect is low**. But, a mistake in the **company logo affects the brand identity and impacts the user experience**. So, the **defect is of high priority**.

### **Another Example:**

- Suppose if the **Flipkart logo is misspelled as Flipkrt**. That time it directly impacts the online business for Flipkart Company.
- People will think it's not a genuine product and they won't buy it.** Business impact is huge. **So it got a very high priority issue.**
- But for **developers fixing this issue not that difficult**. It is not even breaking any workflow also. **So severity is very low.**
- >**Severity means how the bug is impacting the applications.** It's staged like blocker/ Show stopper, critical major minor
- >**Priority means which bug needs to be fixed first.** Stages like urgent, high, medium, and low. It always **impacts the customer business.**
- Suppose Flipkart logo issue not fixed soon.** Flipkart Company will be under huge loss.so it got a **very high priority**.

### **4) Low Severity and Low Priority**

- Suppose the tester clicks on the “**Conditions of Use**” **hyperlink** at the bottom of the amazon.com homepage.
- If there is an **alignment issue in the text displayed** or if there is a **spelling mistake in the content displayed**, the defect is said to be of **low priority** because people rarely read this page and it **does not impact the user experience**.
- The **severity is also low** because the functionality of the application is not affected.

## **Topic: 18: Testing Terminologies:**

### **18.1 Monkey Testing:**

- When we have no. of test cases for execution but don't have sufficient time that condition we perform monkey testing.
- Also called Speed Testing
- Here we only execute High Priority test cases.
- This situation arise when developer not able to solve bug then that time developer takes extra time that bug is called blocker defect.
- Here we test basic core functionality with respect to customer requirements.
- So here we test High Priority test cases. If time left then we test Medium and Low Priority.
- Here we use random data for testing.

### **18.2 Exploratory Testing**

- When we are not aware about the application but we have test cases, test case data, Sprint Backlog...then that time we conduct Exploratory testing.
- Exploratory testing is all about exploring, discovery, investigation & learning.
- It is all about exploring, finding out about the software means what it does, what is doesn't do, what works & what doesn't.

### **18.3 Ad-hoc Testing**

- When we are aware about application, we have test cases as well but don't have test data...that time we do Ad-hoc testing.
- Doesn't have test data so testing carried out on the basis previous knowledge of the testers, tester test randomly without following the requirements.

-Hence **Success of this testing on capability or Experience of the tester...of specific application.**

-Adhoc Testing will be effective when tester have **throughout knowledge of System/Application.**

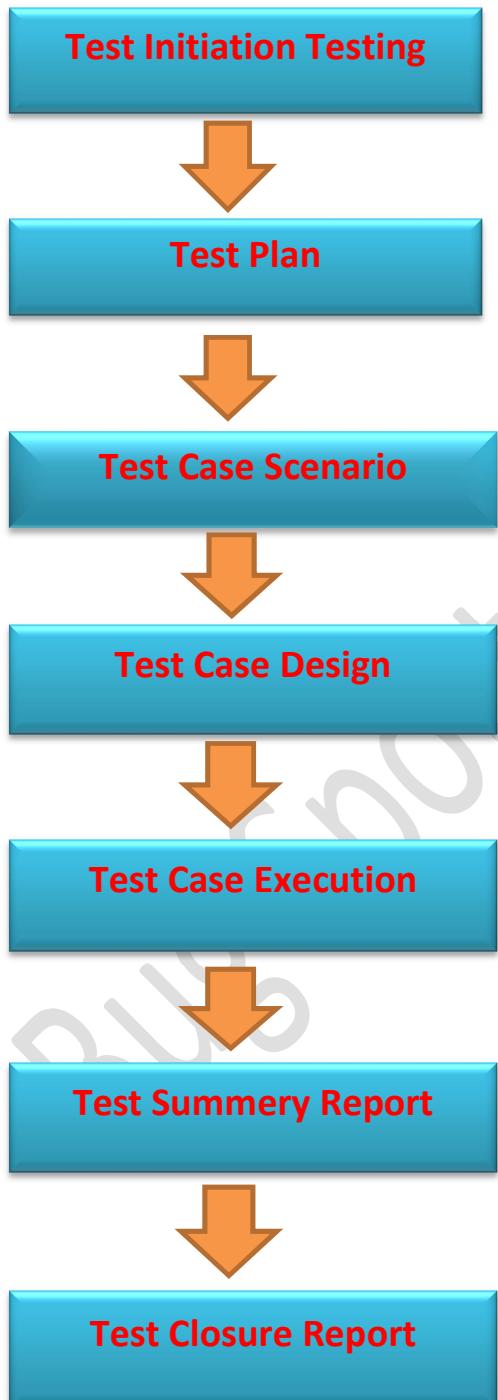
-It is also **experienced based testing technic.**

## **What are the Roles and Responsibility of Test Engineer?**

- 1) User Story Analysis
- 2) Test Scenario Identification
- 3) Test Case Design
- 4) Test Case Review
- 5) Test Case Execution (Perform Functional & non-Functional Testing)
- 6) Test Summary Report
- 7) Defect Logging (Create bug ticket) & reporting (Assign to specific developer)
- 8) Sometimes we have to do client interaction
- 9) Estimation of Testing (Means how much time require for testing we give in Story Point)
- 10) Carrying out regression testing & retesting after defect fix.

## **Manual Testing Part-2**

### **Topic: 19: Software Testing Life Cycle (STLC):**



## **STLC Stages:**

### **1) Test Initiation Testing:**

#### **A) Requirement of the Project:**

-Requirement of project means **domains of the project**

1) Banking Domain 2) Telecom Domain 3) E-Commerce 4) E-Educational 5) Health Care 6)  
Insurance Domain

#### **B) Scope of the Project:**

-**Strategy and Methodology** of the project will be decided

-**Methodology** means which type of testing methodology **Automation testing or Manual Testing**

-Project owner/Product Owner involve here

#### **C) Risk Involved in the Project**

-**Less Resources:** if less no. of resources involve then person need to do extra work/efforts

-**Less Test Data:** If there less data or no test data then we perform "**Adhoc Testing**"

-**Lack of knowledge:** KT (Knowledge Transfer) will be provided

### **2) Test Plan:**

#### **A) Resources Allocation**

-As per scope of the project test methodology selected and then as per that

-Work will allocate to specific tester (work of Test Team Leader)

#### **B) Estimation of Time** (Start date and End date of the project will be decided)

### **3) Test Scenario:**

- Means "**What to test**"
- Test Means "**Validate/Verify**" and Scenario Means "**User Journey**".
- It is also called "**Test Condition**" or "**Test Possibility**"

### **4) Test Case Design:**

- Test Cases are "**how to be tested**".
- Test Case Design means **Steps involve while testing**
- For this we refer either **Sprint Backlog Document or Description and Acceptance Criteria Mentioned in Development Ticket**
- In that **Positive and negative test case** are there. (But we just write down positive test cases but while testing we test positive and negative both scenario)

### **5) Test Case Execution:**

- Once developer local code merged in Master repository of the company then ticket ready for testing.
- Means Application came in **SIT Environment from DIT Environment**.
- Once **developer assigns ticket to testing** with application URL then we start "**Test Case Execution**".
- Here we start testing as per **Test case design steps**.
- If we **found any defect** then we **perform retesting and then raise the bug ticket**.
- Once developer resolved issue then again perform "**Retesting**" and "**Regression Testing**".

## 6) Test Summary Report/Test Documentation

- We “Tester” responsible for “Test Summery Report”.
- Total how many test case designs
- How many test cases execute
- How many test cases pass
- How many test cases fail
- We will create **bug ticket** for failed test cases.
- We create this report in excel format. Once this Test report created then we send to **Testing Team Leader**.

**For Example:**

	Total Test Case Design	Test Cases Executed	Test Cases Passed	Test Cases Failed
Daily	15	15	14	1
Monthly	300	280	265	15

## 7) Test Closure Report

- “Team Leader” is responsible for to make **Test Closure Report**
  - In this closure report he checks whether all process are correct or not.
- Activities:** 1) Analysis of Test Summary Report 2) Analysis of Bug Ticket/Report
- Once Test Closure Report created then **Team leader send it to Test Manager/Product Owner/Project owner.**

# **What is Test Case Review?**

-**Definition:** Recheck test cases after writing.

## **Types of Review:**

### **1) Self Review:**

-Review going to done by our self after creating/designing of test cases.

### **2) Peer Review:**

-Review going to be taken by your Colleague/team member/Senior Member.

### **3) Internal Review:**

- **Involvement:** [Tester + PO (Product Owner)] +Developer

### **4) External Review:**

-Perform by the client during UAT.

-**Involvement:** Development Team + Testing Team + PO + Client

-Also called **Walkthrough/Inspection**

## **How you are going to know about good test cases?**

- 1) Should be simple, easily understandable.
- 2) Complete all Functionalities (Means here no functionality miss from Testing Team)
- 3) Avoid test case repetition
- 4) Do not assume Functionality
- 5) Create test cases with end user in mind.
- 6) Test case must be identifiable (Test Case ID)

## **Requirement Traceability Matrix (RTM)**

- It is also known as **Traceability Matrix**
- It is **mapping between prepared test cases and business requirements.**
- RTM Document** is generally **used to ensure the test Coverage.**
- If all requirements are **mapped with the test cases** then we can **ensure test coverage.**
- This document **prepare** before the **test case execution.**
- Due to this testing team **will not miss any functionality** during testing phase from the requirement specification.

**There are mainly two types of RTM:**

### **1) Forward Traceability Matrix:**

- Actually it is mapping between **prepared test cases** and **business requirements.**

### **2) Backward Traceability Matrix:**

- Actually it is mapping between **Defect** and **business requirements.**

## **Topic: 20: What is a Test Case?**

- Test Cases are “**How to be Tested**” functionality?

**Test case** – is the smallest unit of the testing plan – which includes a **description of necessary actions and parameters to achieve and verify the expected behaviour of a particular function** or the part of the tested software.

So you need to understand **where to start testing**, which **general steps need to be executed** and **what the result should be**. And then this scenario is broken down into more detailed parts – test cases – to define all **positive, negative, localisation** and other behaviours of the software.

**Example:** testers need to test the **functionality of uploading photos**.

We create **first test scenario** as:

1. The user must be logged
2. Move to the “upload photos” page
3. Click the “upload” button
4. Select photos
5. Upload them

Now, this **scenario should be divided into detailed test cases**,

**Example:**

- Check the logged user possibility to go to the “upload photos” page
- Check the not logged user possibility to go to the “upload photos” page
- Check whether the user can click “upload” button
- Is it opens a form to select a photo and possibility to close it
- What happens if you do not select photos, choose another file format (for example video), choose photos of a maximum size and so on
- Check the possibility to upload photos
- Check if photo is saved
- Possibility to reload or delete photos
- What happens with photos in the case of the disappearance of the Internet or the device is switched off
- Are all buttons displayed correctly at another location or on different operating systems (if any difference)

And so on...

-The **number of test cases depends on the experience and imagination** of the tester.

-Therefore, the process of writing test cases starts from forming a test scenario or user story, and then it can be divided to check different occasions.

## **20.1: Test Case Design Techniques**

- 1) Static Test case Design Technic** (Related to Quality Assurance Developer involved)
- 2) Dynamic Test Case Design Technic** (Related to Quality Control Tester Involved)

**-Also called Black Box Testing Technic.**

**Types:**

- A) Boundary Value Analysis**
- B) Equivalence Class Partitioning**
- C) Decision Table**
- D) Cause And Effect Graph**
- E) State Transition Testing**

**Note:** from Above technics in our company we just follow “**Boundary Values Analysis**” and “**Equivalence Class Partitioning**”.

### **A) Boundary Value Analysis (BVA)**

-Boundary value analysis is based on **testing at the boundaries between partitions**. It includes **maximum, minimum, inside or outside boundaries, typical values and error values**.

-It is generally seen that a **large number of errors occur at the boundaries** of the defined input values rather than the centre. It is **also known as BVA** and **gives a selection of test cases which exercise bounding values**.

-This black box testing **technique complements equivalence partitioning**. This software testing technique base on the principle that, **if a system works well for these particular values then it will work perfectly well for all values which comes between the two boundary values**.

## **Guidelines for Boundary Value analysis**

- 1) If an input condition is **restricted between values x and y**, then the test cases should be **designed with values x and y as well as values which are above and below x and y**.
- 2) If an input condition is a large number of values, the test case should be developed which need to exercise the **minimum and maximum numbers**. Here, **values above and below the minimum and maximum values** are also tested.

-Means here we check **min, min-1, Min+1, max, max-1, max+1**

**Example:** Password field accepts **minimum 8 characters and maximum 12 characters**

**Allow 1 Capital Char, 1 Small Char, special Symbol & Number, Space not allowed**

**So here BVA Size is a**

**Min=8, Max=12, Min+1, Max-1 will be Accepted**

**Min-1, Max+1 will not be Accepted (Rejected)**

## **B) ECP (Equivalence Class Partition)**

**Equivalent Class Partitioning** allows you to **divide set of test condition into a partition** which should be considered the same. This software testing method divides the input domain of a program into classes of data from which test cases should be designed.

The concept behind this technique is that test case of a representative value of each class is equal to a test of any other value of the same class. It allows you to **identify valid as well as invalid equivalence classes**.

**Example:** Password field accepts **minimum 8 characters and maximum 12 characters**

**Allow 1 Capital Char, 1 Small Char, special Symbol & Number, Space not allowed**

So here Equivalence Partition is

**Valid ECP:** 0-9 Number, a-z Char, A-Z Chart, Special Symbols

**Invalid ECP:** (-) Space not allowed

## 20.2 Test Case Design Structure:

Steps	Test Case Field	Description
1	Test Case ID	Each test case should have a unique ID.
2	Test Priority	It is useful while executing the test. 1) Low 2) Medium 3) High
3	Test Designed By	Testers Name
4	Date of test designed	Date when test was designed
5	Test Executed by	Who executed the test(tester)
6	Date of the Test Execution	Date when test needs to be executed
7	Name or Test Title	1. The title is important because it's often the first or only thing you see when you are scanning a list of test cases. 2. Clear titles are the key to help testers to find quickly the right test cases.
8	Description/Summary of Test	A detailed description of the test case. In this section, you can also set up categories to organize your test cases into logical groups.
9	Pre-condition	Any requirement that needs to be done before execution of this test case
10	Test Steps	1. Test Steps section gives the tester a numbered list of the steps to perform in the system, which makes it easier to understand the test case. 2. 8 test steps per one test case. Too many steps make it difficult for developers and testers to reproduce the steps when a bug report is filed against the test case.
11	Test Data	You can enter test data directly in the test data field, or refer to a separate file that contains test data for one or more test cases.

<b>12</b>	Expected Results	1. Mention the expected result including error or message that should appear on the screen.  2. The tester needs to know the expected result in order to assess whether the test case is successful.
<b>13</b>	Post-Condition	What would be the state of the system after running the test case?
<b>14</b>	Status (Fail/Pass)	Mark this field as failed, if actual result is not the same as the expected result
<b>15</b>	Notes/Comments/Questions:	If there are some special conditions which is left in the above field
<b>16</b>	Requirements	List of the requirements for a particular test cycle.
<b>17</b>	Attachments/References	The files and documents that are attached to the test case, such as screen captures and other supporting material.
<b>18</b>	Automation? (Yes/No)	Fill “YES” when test cases are automated

**Note:**

-Above structure is standard structure of test case design but Organisation wise somehow it varies. In above structure given lot of test case field but not all field used in Organisation.

-Below we have given 2 test case design template of our companies for your reference.

# **Project Name:** Banking System

## **Test Case**

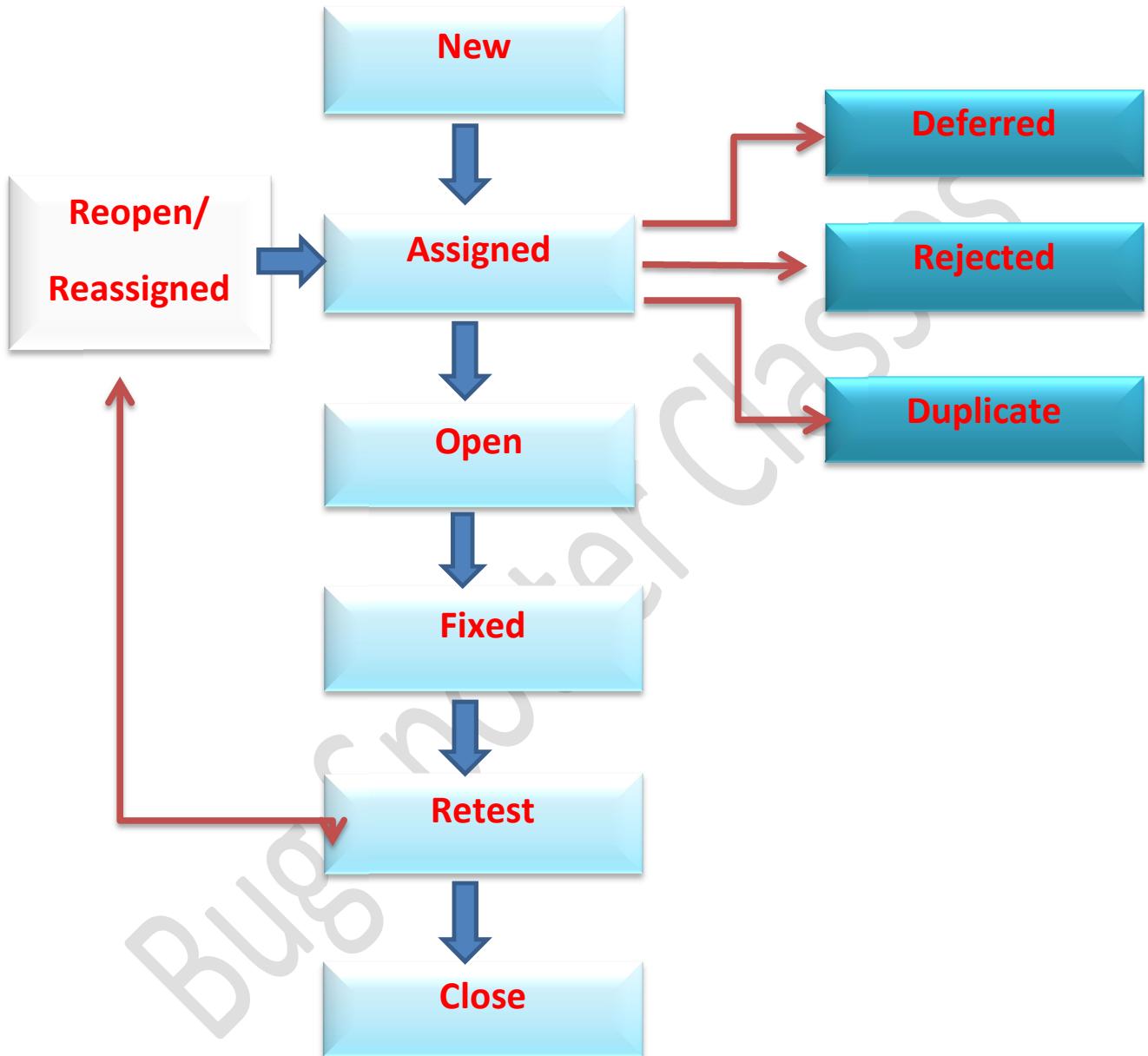
<b>Test Case ID:</b> BU_001	<b>Test Design By:</b> Govind Chavan
<b>Test Priority:</b> Medium	<b>Test Designed Date:</b> 25/08/2021
<b>Module Name:</b> Bank Login Screen	<b>Test Executed By:</b> Govind Chavan
<b>Test Title:</b> Test Login Functionality of Banking System	<b>Test Execution Date:</b> 26/08/2021
<b>Description:</b> Verify login with valid user name and password	

**Pre-Conditions:** User has valid user name and password

<b>Steps</b>	<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status (Pass/Fail)</b>	<b>Notes</b>
1	Navigate to Login Page		User should be able to Login	User is able to login	Pass	
2	Provide Valid User Name	User: Govind	Credentials can be entered	As Expected	Pass	
3	Provide Valid Password	Pass: Govind123	Credentials can be entered	As Expected	Pass	
4	Click On Login Button		User should be logged In	User Logged In Successfully	Pass	

**Post Condition:** User is Validated with Data base and successfully logged in into account. The Account session Details are stored in the Data base.

## Topic: 21: Defect/Bug Life Cycle:



## Bug Life Cycle/Defect Life Cycle

- 1) New:** When we create bug ticket first time then that time Status is "New"
- 2) Assigned:** Then we assign issue to specific Developer then status will be "Assigned"
- 3) Open:** When Developer doing Coding changes for fixing the issue
- 4) Fixed:** When Developer fixed or resolved issue through Code changes
- 5) Retest:** Once it fixed then again assigned to us for Retesting
- 6) Closed:** We verify issue if issue not present then we closed that ticket.
- 7) Reopen:** If issue not resolved after developer work then we again re-assign to Developer

**Note:** So 1-7 Point is Regular Bug life Cycle so in interview just explain this. If Interviewer asked specific about below Status point then explain below points.

**There are also some Bug ticket Status**

**Deferred:** Means it is known issue.

-But this issue is **Low Priority and Severity**.

-It's not make much **effect on functionality of application** (Means not Stopper or Blocker issue)

-Then **PO will decide** this issue **will fix in next Release or Sprint**.

**Rejected:** If we raised bug and **Developer not accept is as a Bug** then he/she will reject this bug.

-Then that we can **discuss with PO at the time Scrum Meeting** with All details information with **Screenshot/video of defect**.

-Then PO check Customer Requirement then took decision accordingly.

**Duplicate:** means if we found same issue twice and created ticket for that then that ticket will be "Duplicate"

-In that condition one ticket will be **cancelled** as Duplicate ticket.

## **Topic: 22: Bug Tracking/Defect Management Tool JIRA:-**

There are no. of Bug Tracking/Defect Management tools but now most of the companies followed **JIRA** and **HPALM**

-**In our company we are using JIRA Tutorial so we will learn in detail this tool.**

### **Topic will be Cover:**

- 1) Agile Meetings** (Backlog Grooming, Sprint Planning, Scrum Meeting, Sprint Retrospective Meeting)
- 2) How create Bug ticket, Task Ticket, Epic Ticket**
- 3) Priority and Severity**
- 4) Defect/Bug Life Cycle**

### **User Stories:**

- User Stories nothing but **Customer Requirements**
- Whenever Stakeholder give requirements (User stories) to Product owner then those requirements for whole Product called **Product backlog**
- In Estimation Phase/**Spring Planning Meeting** decide which module to develop in next sprint and that Modules Requirements
- Only those requirements sort and take for next sprint (For Ex: Sprint-1)

### **User Stories Consist:**

- 1) Description 2) Acceptance Criteria**

#### **1) Description:**

-It is description **As a User, what User Want to do** (Process) and **what it is desired Output**.

## **2) Acceptance Criteria:**

-These are the Scenarios when these scenarios true then system generate correct output otherwise System shows Failure.

For Example:

**Name of User Story:** Credit Card Payment

**ID:** US001

**US001:** Credit Card Payment

**Description:** As a "Customer", I want to pay with Credit Card (Process) so it can confirm my Order (Desired Output)

### **Acceptance Criteria:**

#### **1) Discover Card**

-When we enter card number then it should be display name of the Company of card (means Rupay, Master Card)

#### **2) Validate Credit Card Information (Name of holder)**

3) Then Validate Expiry date and CVV Number

4) Validate Address

5) Generate success or failure result (Means place order or fail order)

### **Agile Meetings Flow:**

**1) Backlog Grooming Meeting** (1Week before of next Sprint)

**2) Sprint Planning Meeting** (1-2 day before of next sprint)

**3) Daily Scrum Meeting** (Daily basis after sprint start)/Daily Stand-up meeting ( Here tester and developer gives their current work status)

**4) Sprint Retrospective Meeting** (After 1 day of Sprint End)

**>>Meeting 1<<**

**1) Backlog Grooming Meeting** (Duration: 1hr to 1.5hr)

**Involvement:**

--Scrum Master (**Chair Person of Meeting**)

--Product Owner

--Development Team

--Testing Team

**When Happened: 1 Week before start next sprint**

**Agenda:**

A) Create **Developer task ticket** for next sprint

B) Add **Description and Acceptance Criteria** to ticket also **discuss** same with **Developer and Tester**

C) Also **Discuss Bug ticket if Already there** and need to take in next sprint

D) Add **Story Point to ticket** (Developer point + Tester Point)

**>>Steps for Create ticket<<**

1) Click on **Create button** (Present on top of JIRA)

2) Select **Project from dropdown**

3) Select **Issue Type from Dropdown**: Task, Bug, Epic, Story (For us Important are **Bug, Task, Epic**)

**A) Task:** (Suppose Tester want to create **Automation ticket** then we create task) or **For Developer Task**

**B) Epic:** (is like a folder in which we can add Bug, and Task ticket of Specific project)

**C) Bug** (When we found defect then we create bug ticket)

**D) Story** (related developer work)

4) **Add Summary** (Means **Title of ticket** (Development Ticket or Bug Ticket) )

5) **Add Description and Acceptance Criteria & Priority & Label & Sprint** (work of PO or Scrum Master)

6) **Add Story Point** to ticket (Work PO or Scrum Master)

7) Attach screenshot of design

8) Add Severity (If Bug Ticket)

**Story Point:** Means **how much time or Effort requires** for completion of **specific task** or **testing time** required.

**Example:** 1, 2, 3, 5, 7 (SP)

-1 **Story Point means** 1 day require for Development of specific task or time for testing specific task

-1 story Point= 9hr or 1 day

**For Example:** 5 Story Point for Ticket= (3 SP for Developer + 2 SP for Tester)

9) Then **click on Create button** below (**Ticket will be created**) and then ticket ID will be generated.

## >>Create Bug Ticket<<

### Steps:

- 1) Click on Create button (Present on top of JIRA)
- 2) Select Project from dropdown
- 3) Select **Issue Type** from Dropdown: **Bug**
- 4) Add Summary (Title of Bug for. Ex.: **Create New Account Button not working**)
- 5) Add Description

### -Steps to reproduce Issue:

>URL of Application

>Detailed Steps to navigate to issue

### 6) Add **Expected and Actual Result**

### 7) Add **Severity** (Critical, High, Medium, and Low) Related to Functionality of application

-Severity will be **decided by Tester** only because he knew how much functionality affected.

### 8) Priority will be added by PO (High, Medium, Low) - which defect should be solved/fixed first from list of defect as per Customer Requirement

### 9) Add **Label** as Bug (which will helpful in future for applying filter to find out specific bug)

### 10) Add screenshot of issue

### 11) Then **Click on Create** so Bug ticket will be created.

### 12) Then Bug Ticket ID will be generated.

## >>Create Epic Ticket<<

**Epic:** (Like as Folder in which we can **add no. of bug and task ticket**)

-Tester only create Automation ticket Epic

-Developer Task and Bug Tickets Epic will be created by PO (Product Owner)

### Steps for Create Epic Ticket:

- 1) Click on Create button (Present on top of JIRA)
- 2) Select Project from dropdown
- 3) Select **Issue Type** from Dropdown: **Epic**
- 4) Add some Description
- 5) Add Priority
- 6) Click on Create button then Epic ticket will be created.

## >>Meeting 2<<

### 2) Sprint Planning Meeting (Duration 45 Min to 1hr)/Estimation Phase

--This meeting will be arranging **before 1 or 2 days of next sprint.**

#### Agenda:

- A) Select tickets** (Developer Task, Bug Ticket) from **Product Backlog** (Means from Backlog Tab) and **add into Sprint Backlog** (Means Sprint-1)
- B) Assignment of task ticket to Specific Developer**
- C) Add any Automation ticket** if tester want to take for next sprint

## How you receive Developer Task Ticket for Testing? Process

### Steps:

#### 1) When sprint Start

-Tickets in Developer "**To Do**" list

#### 2) When Developer Start working then

-Tickets "**In Progress**" (Means "**Developer**" doing **Coding part** for develop specific module)

-Same time "**Tester**" doing **Test case Design** Work

#### 3) Once **Developer** work complete then he/she send his/her code to Review (Means raise "**Pull Request**")

-Then that time ticket in "**AWAITING Code Review**"

-Developer team member review code and gives approval for Merge Developer local **Machine Code to Company main Repository** (GitHub, Bit bucket)

-Once Developer code "**Merged**" in main Repository then ticket will be ready for testing.

-Then Developer adds comment on ticket "**This is ready for testing**" and adds **Application Test URL** in Comment and assign to specific tester.

-Then that time ticket in "**AWAITING QA**"

#### 4) Then Tester start testing

-When Tester doing testing then that time status is remains "**AWAITING QA**"

-At the time testing we **mapped test cases with customer requirements** Means here we start "**Test Case Execution**".

-Some test cases failing then we create bug ticket.

(**If issue related developed module** then we **create Sub bug under same Developer Task ticket** OR otherwise we **create separate bug ticket**.)

- If everything working fine then we passed ticket for UAT
  - Then that time ticket status is "**AWAITING UAT Deployment**".
- 5) Then we again test Application on UAT Environment
- Status will be "**AWAITING UAT Testing**"
  - Once we passed from UAT then its goes to Release
  - Then ticket status will be change as "**AWAITING Release**"
  - And last status is **closed/ done**.

**>>Meeting-3<<**

### **3) Daily Scrum Meeting/Daily Stand-up Meeting**

**Agenda of Meeting:** Check Progress of Development Process

-Here Developer and Testing gives their work status

- 1) What you did yesterday?**
- 2) What are you going to do today?**
- 3) Is there any roadblocks/problem if facing**

**>>Meeting-4<<**

### **4) Sprint Retrospective Meeting**

-1 Day After completion of Sprint

**Involve:**

- |                              |                |
|------------------------------|----------------|
| -Scrum Master (Chair Person) | -Product Owner |
| -Development Team            | -Testing Team  |

**>>Agenda of Meeting:**

**1) What went well during sprint?**

**Example:**

-Development Team and Testing completed urgent mid-sprint requirement of customer within Sprint.

-Successful release of the Application

**2) Which things need to improvement?**

-Mid-sprint requirement added in sprint that's why tester needs to test application twice on SIT Environment.

-Release ticket should be available for testing before 5-6 days of release so tester can test application properly.

**3) Shout out (Appreciation to Team Member)**

**-Jalinder** done good work for Testing Facebook Application because **he tested application in very less time.**

**-Tejas** done good work for Developing Facebook Application.

## **Part-3: Manual Testing Interview Questions**

1. How does quality control differ from quality assurance?
2. What is Software Testing?
3. Why Software Testing Required?
4. What are the two main categories of software testing?
5. What is quality control?
6. What different types of manual testing are there?
7. Explain the difference between alpha testing and beta testing.
8. What are the different levels of manual testing?
9. What is a test bed in manual testing?
10. Explain the procedure for manual testing?
11. What is Manual Testing?
12. What is Regression Testing?
13. What is Automation Testing?
14. What is Re-Testing?
15. What is Smoke Testing?
16. What is Functional Testing?
17. What is Unit Testing?
18. What is User Acceptance Testing?
19. What is the F rule in Web Testing?
20. What is the life cycle of Testing?
21. What is Sanity testing?
22. What tool is used for Manual Testing?
23. What is Build?
24. What is Release?
25. What is Sprint in agile?
26. What is Sprint Backlog?
27. What is a Product backlog?
28. What is the time limit of a sprint in Agile?
29. What is Agile?
30. What is the User story in Manual Testing?
31. What is a Test case?
32. What is Bug in Manual Testing?

33. What is Epic?
34. What is a Change request?
35. What is an Enhancement Request?
36. What are the types of bugs?
37. What is static testing?
38. What is Black box testing?
39. What is white box testing?
40. What is Test Plan?
41. What are the contents of the Test Plan?
42. What are Verification and validation?
43. What are SDLC and STLC?
44. What is Equivalent Testing?
45. What is boundary testing?
46. What is Dynamic Testing?
47. What is the difference between static testing and dynamic testing?
48. What is Integration testing?
49. What is the top–bottom approach?
50. What is the Bottom – Top approach?
51. What is System Testing?
52. How system Testing differs from UAT?
53. What is Test Scenario?
54. What is Test Case?
55. What is Test Script?
56. What is a defect?
57. What is Grey Box Testing?
58. What is Acceptance Testing?
59. What is Non-Functional testing?
60. What is Alpha Testing?
61. What is Beta – Testing?
62. What is the Difference between Alpha Testing and Beta testing?
63. What is Boundary value?
64. What is Cause – Effect?
65. What is SDLC?
66. What is STLC?

67. What is the Advantage of Manual Testing?
68. What are the disadvantages of Manual testing?
69. What is Bug Leakage?
70. What are the key challenges of software testing?
71. What is the role of QA in project development?
72. Can you explain the V model in manual testing?
73. Can u explain the structure of the bug life cycle?
74. What is “bug leakage?” and what is “bug release?”
75. Can you explain the waterfall model in manual testing?
76. Can you explain to me the levels in the V model manual?
77. What is the value of a testing group? How do you justify your work and budget?
78. What is the difference between scenarios and test cases?
79. What is the Test case Life Cycle?
80. How I can do GUI testing, what is its important content, plz tell me all the property of GUI testing?
81. What part of the testing phase is the most important part of testing in the cycle?
82. What is unit testing?
83. What is the difference between smoke testing and sanity testing?
84. What is the difference between test scenarios and test strategy?
85. What is the difference between build and release?
86. What is the difference between usability testing and GUI?
87. What is the test strategy that will prepare that one? And what will be there in the test strategy?
88. What is test development?
89. What is the definition of the test life cycle?
90. What is the Test case Life Cycle?
91. What is a test bed?
92. What is unit testing in the manual?
93. What is a test plan who will prepare that one?
94. How do you develop a test plan and schedule? Describe bottom-up and top-down approaches?
95. What is system testing?
96. What are SRS and BRS in manual testing?

97. What is the Sanity Test, Adhoc Testing & Smoke Testing? When will you use the Above Tests?

98. What is the V model in manual testing?

99. What is the V model can u explain?

1. What is STLC how many phases are there can you explain them?
2. What is the Review?
3. What are alpha testing and beta testing?
4. What is SRS and BRS document? Can you explain them briefly?
5. What is stub and driver in manual testing?
6. What is stress testing?
7. What is the prototype model in manual testing?
8. What is performance testing?
9. What is mean by GUI testing? What is mean by the client/Server? What is meant by a web-based application?
10. What is integration testing?
11. What is a fish model can you explain?
12. What is the exact difference between Debugging & Testing?
13. What is compatibility testing?
14. What is BUG Lifecycle?
15. What is Black Box Testing?
16. How to do regression testing, and can give one or two examples on that in the same application?
17. Difference between bug, error, and defect?
18. How many modes of recording are there?
19. Give an exact and appropriate definition of testing.
20. How will you review the test case and how many types are there?
21. What are the types of software testing?
22. What is in mind while performing testing?
23. When do you start testing?
24. When do you end testing?
25. What are verification and validation?
26. Define retesting.

27. Define regression testing.
28. What is the difference between retesting and regression testing?
29. What is functional testing?
30. What is non-functional testing?
31. What is module testing?
32. What is system testing?
33. What is end-to-end testing?
34. What are the differences between system and end-to-end testing?
35. What are the testing techniques?
36. What are the testing methodologies?
37. What is the boundary value analysis (BVA)? Explain with example.
38. What is equivalence portioning (EA)? Explain with example.
39. What is the test case?
40. What is the requirement traceability matrix? (RTM)?
41. What is the first step to start the testing process?
42. What is a test plan?
43. What is the basic format of the test plan?
44. What is the basic format of writing a test case?
45. What is a bug?
46. What is a defect?
47. What is a fault?
48. The difference between bug, defect, and fault?
49. What are must present fields while raising a defect?
50. What are the steps to reproduce while logging a defect?
51. What is the defect life cycle?
52. What are defect tracking tools?
53. What is the severity and priority of the bug?
54. Example of high severity and high priority.
55. Example of low severity and low priority.
56. Example of high severity and low priority.
57. Example of low severity and high priority.
58. Who decides the priority of bugs?
59. Who decides the severity of the bug?
60. In which phase of SDLC, your works start?

61. What is monkey testing?
62. Explain the test driver and test stub with examples?
63. What is the use case?
64. What is a test strategy, and who prepares it?
65. What is a test execution plan?
66. What is compatibility testing?
67. What are localization and global testing?
68. What is mobile responsiveness testing?
69. What is static testing?
70. What is dynamic testing?
71. What is user acceptance testing?
72. What is a testing environment?
73. Why you joined as a Tester
74. Testing starts before coding or after coding
75. What are Hot Fix and Production Issue?
76. What is Scrum?
77. What you will do if Sanity Testing Fails?
78. Which are the different Ceremonies in Agile Methodologies?
79. Are you involved in cross browser compatibility testing? If yes then explain
80. Are you part of QA or QC?
81. What are the Drivers and Stubs?
82. How Many Actors in Agile? What are they?
83. What will you do if you found error in testing and developer not agrees with that?
84. What actually happened in Estimation phase in Agile?
85. How you get user Stories?
86. Explain the review Process? Peer, Formal, Informal
87. What is your involvement in release process?
88. What is entry and exit criteria for test case execution
89. How Many test cases can you write and execute per day?
90. What is test case? What are the field in that? Where you write test cases?
91. How do you know your test cases are completed?
92. How do you know your test cases are good or bad?
93. Do you written negative test cases?

94. Explain one scenario if have where you were not able to write test cases for given requirements.
95. What is difference between Preventive and Reactive approaches in testing?
96. Why are you in QA?
97. Have you written Test Plan?
98. Tell me some key points while writing/ creating bug ticket?
99. What are the key challenges of software testing you faced in your career?
100. How Many test cases did you write for last project? How much time require for write test cases?
101. How Many defects did you detect in your last project?
102. Did you face any problems during defect reporting and tracking?
103. What is difference between front end testing and back end testing?
104. What are the testing types you performed in your career?
105. What are the test case design technics that you used?
106. How you communicate with developer for resolving issue?
107. What is sprint in agile? What Sprint Backlog and Product backlog? What is Agile?
108. If you are getting bugs in every build then what will be your first approach?
109. If you require more time for testing to complete any task and given very less time then what will you do as a Tester?
110. Why testers are needed when developer can do same task?
111. How you test login feature of Web Application?
112. You have been given 100 test cases for testing, how will you start when you have not given time limit?
113. How you check if the build is testable?
114. What is Confirmation Testing?
115. What are your two Strengths that you will bring to your QA/ Testing Team?