**Velocity Corporate Training Center, Pune**

**Problem Solving Ability**

**Logical Thinking**

**Instructions:**

1. Let’s keep class interactive.

2. Please raise your hand.

3. Laptop Strictly

4. Pen and Notebook/ Rough book 🡪 Points

5. Notes 🡪 Handwritten

6. Homework

7. Mute/Unmute

8. Video on off

9. Question 🡪 Unmute 🡪 Ans

10. Time 7.30 to 9.30 🡪 7.25 am

11. Daily Study

Non-working 🡪 7-8 hrs (8hrs + 2hrs +2hrs)

Working 🡪 4 hrs (9hrs+2hrs+2hrs+4hrs)

12. Camera Condition

……………………………………………………………………………………………………………………………….

**Software Testing**

Automation Testing

* Java Scripting Language
* Selenium Tool

Manual Testing

* Why manual testing?
* What is software?
* What is testing?
* Project Team Details
* SDLC
* Methodologies
* Types of testing
* Testing Terminology
* 7 Principles of testing
* STLC
* Bug/Defect life cycle
* Documentation
* Mis.

Database Testing

* SQL

API Testing

* SOAP UI
* Postman

Project Management Tool

* Jira
* ADO

Test Management Tool

In demand skill – Performance / Mobile Testing etc.

Project Sessions 2

Resume

Resume Session

HR Session

Job Applications 🡪 Naukri, Indeed, LinkedIn 🡪 Applications

……………………………………………………………………………………………………………………………….

1. Daily Mock

4-5 Groups

A

Group Study – Google Meet

Daily Study / Previous Revision / Interview Practise

English

Update – Attendance

…………………………………………………………………………………………….

4 to 5 People 🡪 A Varun Meghana Tushar Noopur Niketan

Mobile Nos.

Contact

Google Meet

5 people together

Daily Mock Time 🡪 Decide

Mutual Agreement

……………………………………………….

1. Today’s Portion

2. Previously done topic 🡪 Systematic revision plan

3. Q and A

………………………………………………..

English

2. External Mock Interviews

External Experts 🡪 Working As a tester

Saturday / Sunday

Feedback

20 Marks

Issue 🡪 Google 🡪 Stackoverflow / Google 🡪 Solutions Try 🡪 Coll. Friend 🡪 Senior Coll. 🡪 TL

Issue 🡪 Google 🡪 Stackoverflow / Google/ ChatGPT 🡪 Solution Try 🡪 Mock Members 🡪 Batch Senior 🡪 Staff.

……………………………………………………………………………………………………………..

**What is Software?**

Collection of specialised programs which takes user input and generates desired output.

Whatsapp 🡪 Java, C, C++, .net, Python, React etc.

Phonepay 🡪 Ashish 🡪 1234@ybl 🡪 UPI Id 🡪 Type 🡪 Verify 🡪 Amount 10,000 🡪 UPI Pin 🡪 Debit / Credit

Whatsapp Message 🡪 Hi

Open

Login

Contact 1234 Save

Search 🡪 click 🡪 TextBox 🡪 Cursor Blink 🡪 Hi…!!! 🡪 Send.

**What is Software Testing?**

**Testing is a process of checking whether the given software / application is generating desired output.**

e-commerce 🡪 Clothes 🡪 Listed 🡪 Buy

Whatsapp

1. Msg Sending
2. Voice call
3. Video Call
4. Everyone 🡪 Delete
5. Multimedia (docx,pdf,png, images etc)
6. Location
7. Etc

Paytm 🡪 Recharge Icon 🡪 Radio Buttons Prepaid 🡪 Enter Mobile BSNL 9421300899 🡪 Operator Jio 🡪 Circle 🡪 Plan 🡪 Payment 🡪 Success

BSNL / JIO 🡪 Recharge -> Fail / Success

…………………………………………………………………………………………………………………………………………………………….

**Process of checking the completeness and correctness of software with respect to client expectation.**

Recharge

Icon

Radio Button Prepaid / Postpaid

Mob No

Operator

Circle

Plan

Payment

Success

…………………………………………………………………………………………………………..

**Questions:**

1. **What is software?**
2. **What is software testing?**
3. **What do you mean by Completeness?**

Completeness is the availability of all features/attributes listed in the requirements in order to complete that functionality.

1. **What do you mean by Correctness?**

**Correctness** means the system/application/functionality behaves exactly as expected/intended.

In other words, functionality is working correct or not.

…………………………………………………………………………………………………………..

**Teams in Project**

1. Project Team

New Feature development

* 1. Delivery Manager (1)

Business related discussions } Project duration / Pricing / Human Resources Planning (dev/tester)

Project Delivery } Delivery Deadlines/Dates

Very senior person 🡪 15 to 20 yrs IT exp (Dev/Tester)

Non-technical job

Management

* 1. Project Manager (1-2)

Project related work (Design 🡪 Development / Coding 🡪 Testing 🡪 Support)

12+ yrs (dev/testing)

Semi technical

* 1. Business Analyst (3-4)

Discussion with client about requirements.

Zomato 🡪Mr. Goyal

Requirements Gather

Doubts 🡪 Clear

Bridge between client and our team

Domain Expert ( Hotel Industry )

SBI 🡪 Banking Software 🡪

d. UI Designer (2)

Graphic Designer

Colours, Fonts , Images , Images size etc.

e. Developer (10-12)

Coding of the application

f. Testing (3-4)

Tester / Software test engineer / Test analyst / Quality Analysit / Quality Engineer QA Engi.

………………………………

Mobile Software Updates:

Bug fixes

Enhanced security

Performance boost

Stability etc.

New feature updation

……………………………….

2. Support Team

Bug

New Update

New feature

KT -> Knowledge Transition / Training

a. Developer (3-4)

b. Tester (1-2)

c. Manager (1)

……………………………………………………………………………………………………………………………………………………………

**Questions:**

1. What is your team size?

2. Who all are involved in your project?

3. How many testers were there in your team?

4. How many devs were there in your team?

5. What is the role of business analyst?

…………………………………………………………………………………………………………………………………………………………….

Developers

1. Frontend Developer

2. Backend Developer

FB / Insta

Signup 🡪 Registration

Username Password 🡪 Database 🡪 Store

UI and DB connection

3. Full stack developer

FE+BE

……………………………………………………………………

IT Companies

TCS

Infy

Wipro

Capg

TechM

Quick Heal

Mastercard

Visa

WhatsApp

1. Product based 🡪 Own product 🡪 Sell

Quick Heal

WhatsApp

Mastercard

Java + selenium

Profit High

2. Services based

Client 🡪 XYZ Software 🡪 Requirements

TCS Wipro

Mercedes 🡪 java + selenium 🡪 3 yrs

BMW 🡪 phython + selenium

Multiple 🡪 100%

2%

Increments 🡪

***CID 🡪 Consistency***

***Intention***

***Dedication / Discipline/Determination***

Intention 🡪 No job

Job 🡪 Work – Life Balance / Salaries

Spouse / Husband 🡪 Better Life

Child 🡪 Future Secure

Criticise

Prove

Proud Feel

Consistency 🡪 4 Months 🡪 100+ 120 days 🡪

20+

Dedication / Discipline / Determination 🡪

99%

…………………………………………………………………………………………………………………………………………………………….

**SDLC**

**Software Development Life Cycle**

Life Cycle : Start to End

Concept – Disposal

Human Life Cycle : Birth (First breath) – Death (Last breath)

Stages: 1. Infant

2. Child

3. Teen Age (13-19)

4. Youth

5. Mid Age

6. Senior Citizen

7. Death

Tea Dev L C 🡪 Desire 🡪 Ingredients 🡪 Water / tea powder/ sugar/ Milk 🡪 Ready

SDLC :

1. Requirements Gathering

2. Analysis

3. Design

4. Development

5. Testing

6. Deploy / handover

7. Maintenance

Story 🡪 Zomato 🡪 Mr. Depender Goyal

Paytm 🡪Recharge

…………………………………………………………………………………………………………………………………………………………….

**A. Requirements Gathering**

* Client 🡪 Business Analyst 🡪 Discuss
* Laptop 🡪 Croma/ Vijay Sales 🡪 Laptop i5 11th 16 GB 1TB SSD 14.5” Black HP/Dell 🡪 Sales Person
* Discussion about the requirements which client is giving to BA.
* B.A. 🡪 Respective Domain 🡪 Knowledge
* Documentation
* Not very detailed document
* BRS – Business Requirement Specification
* This doc is not very detailed 🡪 High Level
* BRS Doc Contains--
  + Sign Up
  + Login
  + List of Restaurants
  + Menu Screen 🡪 Name/Price / Photo / Rating
  + Cart
  + Payment
  + Profile 🡪 Pre. Orders / Photo
* FB
  + Sign Up
  + Login
  + Home Feed / Wall
  + Settings
  + Profile
  + Messenger

**B. Analysis & feasibility check**

* BA
* Study
* Can we do this project?
* Yes 🡪 Do we have the human resources? Yes
  + NO🡪 Hiring
* Detailed Document – SRS
* **Software requirement specification**
  + **Sign Up screen** 
    - First Name
    - Last Name
    - Mobile No
    - Email Id
    - DoB
    - Password
    - Gender
    - Policies and Terms and Conditions
    - SignUp button
  + **Login Page** 
    - Logo of FB
    - Tagline
    - Username/Mail textbox
    - Password textbox
    - Login Button
    - Fogot Password – Hyperlink
    - Create New account button
  + Fav App / Fav screen 🡪 HW

SRS – Software Requirement Specification

1. Functional Requirement

2. Functional Flow Diagram

3. Use Cases

4. Snapshot / Screenshots/ Wireframe

1. Functional Requirement

Attributes which are required to finish a particular function/Functionality / feature.

Feature/ Functionality

Sign Up Page

FN 🡪 Alphabets

LN 🡪 Alphabets

MN 🡪 Numbers 🡪 9 NOOO / 11 NOOO 🡪 Stiff 10 Digits “IN”

EI 🡪 [address@domainname.com](mailto:address@domainname.com)

[dhananjay@gmail.com](mailto:dhananjay@gmail.com)

DOB 🡪

Gender

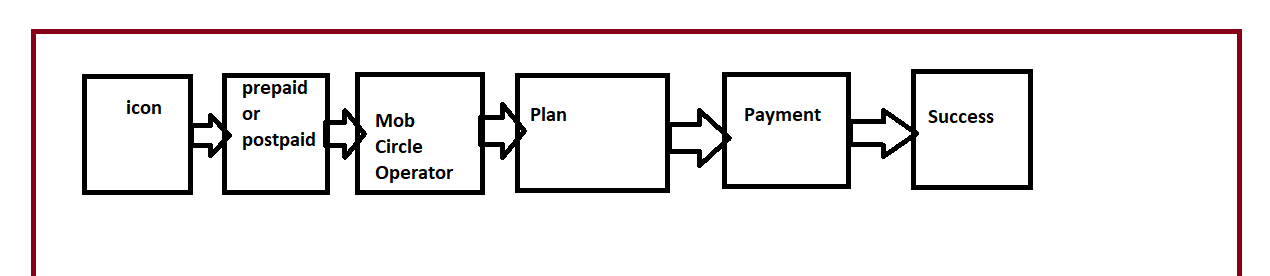
2. Functional Flow Diagram

Diagram which shows flow of our task/ input / functionality.

Shows the relationship between tasks

Proper sequence of task

Dependencies and co-relation of function is seen.

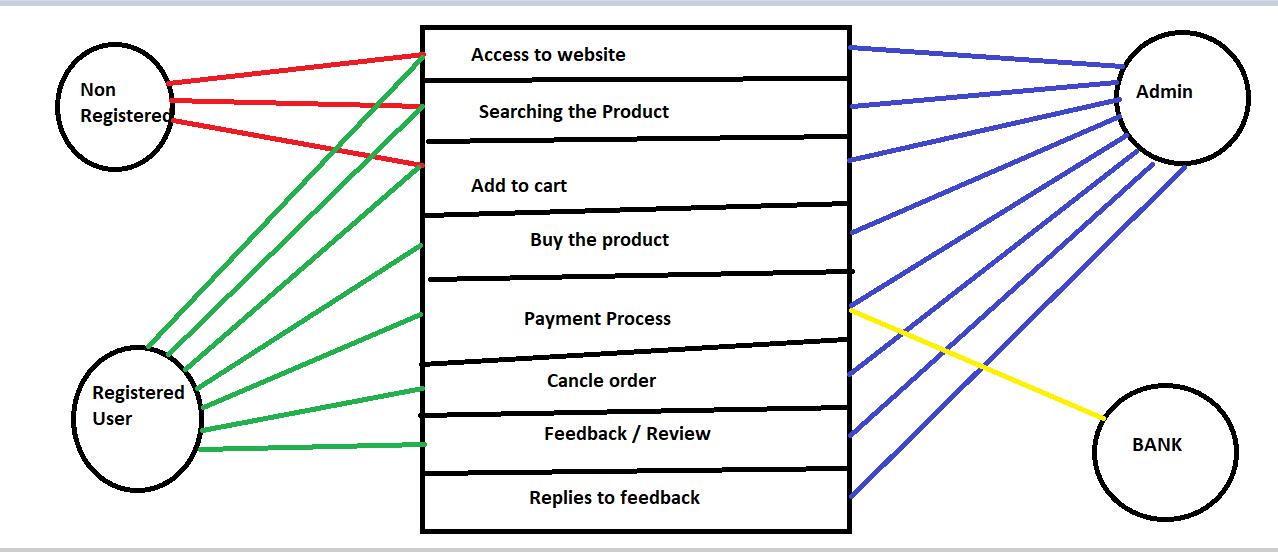


3. Use Cases

Possible usage scenarios of any functionality or application.

It is the functionality in terms of input and output.

INS Vikrant 🡪 Air Craft Carrier 🡪 Defence



9421300899@ybl 🡪 Notification 🡪 Flipcart has requested 1,30,000Rs 🡪 Pay and Decline

4. Snapshot / Wireframe

Visualization of the functionality before development.

Pictorial representation of my functionality

We just create images using some tools to understand how that application / screen / system/ software will look.

Tools : Iris , Canva , Photoshop

BA 🡪 Business Analyst

**C. Design**

1. UI Design 🡪 User Interface / GUI 🡪 Graphical User Interface

The screen which is being used to interact with system by user.

Application Frontend; Colour;Images;Banner etc Font

Look and Feel 🡪 Enrich

UI Designer

Graphic Designer

Design Engineer

2. Arch. Design 🡪 Design and placement of Modules, Internal Arrangement, Connection Database etc.

Software Architecture

Min 8 – 10 yrs 🡪 DEV

|  |  |
| --- | --- |
| **High Level** | **Low Level Design** |
| Working of Module, DB (Not very detailed) | Submodules (very detailed) |
| Relation and dependencies between modules | Static Logic of every submodule |
| Design for Modules | Design for modules as well as submodules |
| Sign Up, Recharge, Log In , Home Page | Recharge   * Icon * Prepaid / Postpaid * Mob, Operator, Plan * Payment * SMS, Mail 🡪 Client information system |

First name 🡪 SIgnUp 🡪 INPUT 🡪 Feed, Profile, settings , Photo – friends, FN

Code 🡪 Logic 🡪 Sub module Value 🡪

Photo Upload 🡪 1 Time 🡪

Dynamic and Static 🡪

FAQs

…………………………………………………………………………………………………………………………………………………………….

Petrol 🡪 Volatile 🡪 Spillage

1 Month

Syllabus 🡪 Volatile 🡪 Read – Revise – Repeat 🡪 Complicated

………………………………………………………………………………………………………….

D. Development

* Developer
* Backend & Frontend
* Actual Coding
* Stages
  + Analysis 🡪 ~~Study~~
  + Design – Page layout (!)
  + Program / Coding
  + Unit Testing 🡪 50 Lines
  + Integration
    - Paytm 🡪 Fastag 🡪 50 lines for fastag 🡪 Main Application – Paytm (3000)
    - Source Code + Newly created code = ADD
  + Integration Test
  + Tester handover 🡪 Deployment to test it further
* Program

Set of instructions to perform a particular task.

Multiple lines of Code

* Code

Psvm{}

………………………………………………………………

**E. Testing**

* Check the completeness and correctness
* Tester / QA / Test Analyst / Software test engineer / SDET
* Types of testing
  + White Box Testing
  + Black Box testing
  + Grey Box Testing

|  |  |  |
| --- | --- | --- |
| White Box | Black | Grey |
| Unit Testing , Static Testing | Conducted by Tester | Mix of black box + White box |
| Developer | Tester / QA/ TA/ SDET | Tester with knowledge of Development / coding |
| Code level analysis | Everything Test | Everything test |
| Positive Scenarios | Positive + Negative | Positive and Negative |
| Bug 🡪 Immediately Fix | Bug 🡪 Raise | Bug 🡪 Grey box tester 🡪 Fix |
| Code Access | No code access | Code access |
|  |  |  |

POSITIVE & NEGATIVE SCENARIOS:

9421300899

* Mobile No. 🡪 text box 🡪
* Which type of values ?
  + Numeric 🡪 ~~Decimal, Binary , Hexadecimal, Fraction~~ , integers
  + Only 🡪 10 integers
  + It shoudn’t start with zero
* Positive Scenario
  + Correct Mob No.
  + It should Work
  + Dealing with it in a correct way
* Negative Scenario
  + Alphabets
  + Special Character
  + Start with 0
  + Decimal
  + Fraction
  + More than 10
  + Less than 10
  + Spaces
  + Num + Alphabets
  + Num + Special Chars
  + Alpha + Special chars
  + Toll Free 1800
  + Blank
* Positive Scenario 🡪 Correct Inputs 🡪 Working Fine 🡪 Pass 🡪 No Bug
* Positive Scenario 🡪 Correct Inputs 🡪 NOT Working Fine 🡪 Fail 🡪 Bug
* Negative Scenario 🡪 Incorrect inputs 🡪Not Working Fine 🡪 Pass 🡪 No Bug
* Negative Scenario 🡪 Incorrect inputs 🡪 working fine 🡪 Fail 🡪 Bug

We will release the functionality to the client.

We will deploy it on Client environment.

We can make code live / Website 🡪 Live

**F. Maintenance**

**Service or Warranty**

Service Period Agreement 🡪 Duration 🡪 Warranty Period

1. Bug Fixes

2. Updations (Minor)

3. Knowledge Transfer (KT)/ Training 🡪 How to use that software / feature?

……………………………………………………………

Service Period Agreement 🡪Lapse 🡪 Maintenance 🡪 Charge

Support 🡪 Charge / Bill/ Money

…………………………………………………………..

**Types of Applications**

**1. Web Based Application**

* Websites
* Flipkart , Amazon , Zerodha
* URL + Browser + Internet

**2. Desktop / Windows / Standalone**

* Vice City
* IGI
* Notepad
* Paint etc.

**3. Mobile App**

**……………………………………………………….**

Software Development starts at Requirements Gathering – ends at Maintenance

Travel : Pune to Goa

Trains

Air

Road

Walk

Water

…………………………………………………………

Tea Development

Desire 🡪 Water 🡪 Tea Powder 🡪 Sugar 🡪 Spices 🡪 Milk 🡪 Boil 🡪 Strain 🡪 Enjoy

LPG Gas 🡪

Induction 🡪 Too fast 🡪 Not good for health

Chulha 🡪 Slow Cooking 🡪 Pollution / Breathing

Kettle

Small Heater

…………………………………………………………

SDLC / S/w Dev

Req Gat

Analysis FC

Design

Dev

Testing

Deploy

Maintenance

…………………………………………………………

**AGILE Methodology**

* Ask for Menu Card
* Starter
* Main Course
  + Sabji Bread
  + Rice
* Dessert
* Order, Special Request, Serve 🡪 Waiter
* Preparation 🡪 Chefs
* Starter, Main Course , Dessert 🡪 At One Time
* Orders 🡪 ONE GO
* Phased Manner
* DEMAND 🡪 Dessert 🡪 Suggestion Waiter
* Change request / Special Request 🡪 Accept Happily
  + We feel Nice
  + Value Addition 🡪 Relation
* Compliment 🡪 NICE / BAD 🡪 Review
  + Examination / Assessment / Evaluation

………………………………………………………………………………………………………………………

Paytm – 2013

* Recharge
* Wallet
* Paytm Mall
* Ticket Booking
* Hotel Booking
* Utility Bill Payment 🡪 Electricity / Water / Education Fees/ LPG etc.
* Movie ticket
* Payments Bank
* Stock Market -🡪 2022

9 yrs

……………………………………………………………………………………………………………….

**Agile Methodology**

* Meaning 🡪 Able to move fast, quickly.
* Agile Methodology is a way to manage a software development project by breaking it up into several phases.
* It is a process of continuous development and testing of software.
* It involves constant collaboration with stakeholders and continuous improvement at every stage.
* Once the work begins then team goes through a cycle of planning, execution and evaluation.
* Value Driven approach 🡪 Priority and Importance 🡪 Client
  + New Changes 🡪 CHANGE REQUEST 🡪 Those will be accepted at any stage of development.
  + No additional charge for change request.
* Iterative and Incremental Approach
  + Every Sprint 🡪 Planning, Development, Testing , Client Handover
    - Starter, Main Course and Dessert
  + After every successcive sprint our application/ software gets more features added, and it starts growing in terms of features/ fun.
    - Stomach 🡪 Starter 30 Main Course 50 Dessert 20
* Sprint 🡪 A fixed time duration in which team has to work on some specified requirements.

………………………………………………………..

Big Project

* 1 Year
* Weeks 🡪 52 Weeks
* Sprint 🡪 Short Run 🡪 100 200 500 m 🡪 Usain Bolt
* Marathon 🡪 Long Run 🡪 10 kms 20 kms 50 kms
* Sprint 🡪 A fixed time duration in which team has to work on some specified requirements.
  + 2 weeks
  + 4 weeks
* Big 🡪 Phased / Breakdown
* Sprint in a year 🡪 26 sprint
  + 1st Jan Start Date 🡪 Sprint 1 🡪 2 weeks 🡪 end date 14th Jan
  + 2ns Sprint 🡪 15th Jan 🡪 end date 28th Jan
  + 3rd Sprint
  + 4th
  + 5th

………………………………………………………………………………………………………………………………..

**Agile Terms**

**1. Stake Holder 🡪 Client**

* Requirements

**2. Product Owner 🡪 Business Analyst**

* Requirements Gathering
* Documentation
* Pass on to Team 🡪 Doubts / Questions / Clarification
* Bridge between client and our team

**3. Scrum Master 🡪 Manager**

* Boss
* Team Manage
* Requirement specify , Work allocation , Tracking , Assessment / examine / evaluate , Agile meeting / Ceremonies 🡪 Host , Facilitator

**4. Designer**

**5. Developer**

**6. Tester**

**7. Backlog – Requirements**

* **Sprint Backlog** 🡪 30 , 40, 50 : Prioritized items/ requirement on which team has to work on in a particular sprint.
* **Project / Product Backlog** 🡪 Paytm 🡪 12,000 Requirements : All the requirements for whole project.
  + **12000 Requirement 🡪 1 yr**
  + **Agile 🡪 Phased / Breakdown several phases**
  + **2 weeks 🡪 1 yr 🡪 26 Sprint**

**Agile Team :**

Scrum Master 🡪 1

PO 🡪 1

Devs 🡪 6

QAs 🡪 3

Total Members : 11 people working

Devs vs QA = 2:1 🡪 Normal Projects

3:1 🡪 Not so critical project

1:1 🡪 NASA / SBI / RBI

…………………………………………………………………………………………………………………………………………………………….

**Agile Meetings / Ceremonies (D1 – D14)**

1. Grooming / Backlog refinement

* To make ready 🡪 Team 🡪 Sprint
* Requirements / Backlog Discussion
* PO 🡪 Knowledge about requirement 🡪 Doubts / Questions
* 1 Hr
* Scrum Master, PO, Developers, Tester (whole team )
* 1st May – Sprint Start D1 } 2 Days before the sprint starting } 29th April
* (!) Mid

2. Sprint Planning meeting

* Plan the sprint
* Work 🡪 Requirements 🡪 Plan
  + What to do ? 🡪 Requirements on which we have to work on
  + When to do ? 🡪 Sprint 1 Duration
  + Priority 1 task 🡪 Requirement 1
  + Who will do which task ? 🡪 Requirements 🡪 Allocated 🡪 Varun / Pallavi / Mahesh
  + How much time should be allocated ? 🡪
* Requirements (RE01, RE02, RE03, RE04)
* Resource Allocation (RE01 Login 🡪 Varun (Dev), Mahesh (tester))
* Estimate ( RE01 = Dev 10hrs , Testing 5hrs = 15 hrs)
* 1 HR
* Scrum master , PO, Dev, Tester (whole team )
* D1 – Sprint Planning Meeting
* 1st May 🡪 sprint start 🡪 10 am to 11 am 🡪 Sprint Planning Meeting 🡪 Sprint Work starts

3. Daily stand up

* Daily
* 15 mins (D1 – D14)
* 3 Questions
  + What I did yesterday?
  + What I am going to do today?
  + Is there any road blocker?
* Whole Team
* Morning 10am 🡪 11am

4. Sprint Review

* Demo to the client
* D14 🡪 Last day of sprint
* 1 hr
* Whole Team + Client
* Tester / Developer 🡪 Demo
* Virtually
* If client feels everything as expected 🡪 Good
* Not ok 🡪 Changes 🡪 Change Request
* Client ABSENT 🡪 Demo WHO 🡪 PO

5. Sprint Retrospective

* D14
* Last Day of sprint
* 3 Questions
  + What went good??
  + What went bad??
  + Is there anything that needs to be changed?
* Team / Individual
* 1 hr
* In this meeting Scrum Master, Developer, Testers, Designer, PO in short our whole team takes participation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting** | **Purpose** | **Involvement** | **Duration** |
| Grooming  (Before sprint/ before sprint planning) | To clear the doubts about User Stories | Dev, tester, PO, Scrum Master, Designer | 1 Hr |
| Sprint Planning Meeting  D1 | Sprint plan from the product backlog, estimation for user story, story points. | Whole Team | 1 hr |
| Daily standup/ Scrum meeting  D2-D13 | Daily work status  1. What I did yesterday?  2. What I am going to do today?  3. Any blocker ? | Whole Team | 15 mins |
| Sprint Review  D14 | Demo | Whole team + Client | 1hr |
| Sprint Retrospective meeting | What went well, what went bad, what needs to be improved? | Whole team | 1 hr |

…………………………………………………………………………………………………………………………………………………………

**Estimation:** Time required to complete a particular feature/ functionality / User Story

Estimation is being given by devs as well as Testers.

3 factors that are considered

**a. Knowledge**

Team should have knowledge of the domain/ functionality in order to develop or to test the task. If team members are having knowledge then in that case less time will be required.

**b. Efforts**

Team can decide how much efforts are required to complete the given task for a project / feature / functionality / user story. Team has to decide how many resources/ people are going to work

**c. Complexity**

We need to check the difficulty of the task / user story and that factor needs to be considered while doing the estimation.

Disha Test Case question

Nilesh

…………………………………………………………………………………………………………………………………………………………….

**User Story**

* User story is nothing but 🡪 Requirement
* Stakeholder 🡪 PO 🡪 Product Backlog (All req)
* Sprint Planning 🡪 Prioritization Task 🡪 Sprint Backlog (10 Req/ User Story ) 🡪 Assign 🡪 Estimation.
* User Story 🡪 2 Parts
  + A. Description
  + B. Acceptance Criteria
* Description
  + What User wants to do (input) and what will be the output.
  + Sign Up Page Development
  + As a < user type> I want to <process / input> so that <output>. 🡪 Template Based Format
  + User Type 🡪
    - Ecommerce 🡪 Registered Users, Non, Admin etc
    - Hospital 🡪 Receptionist , Doctor , Medical Store , Paramedical staff , Lab staff
    - Hotel Management 🡪 Cash Collection, Waiters, Captain, Chef
  + As a New User, I want to provide FirstName, LastName, Mobile or MailId, New Password, DoB and Gender so that my details will be stored in database and account will be created.
  + As a registered user I want to provide UserName, Password and click on login button so that user is able to login into the system and home page is seen.

…………………………………………………………………………………………………………………………………………………………….

User Story 🡪 Sign up page development

* Description
  + - For new user registration
    - FirstName Textbox: One textbox to accumulate the first name of new User
    - Lastname Textbox : One textbox to accumulate the Last name of new User
    - Mobile no Textbox :One textbox which accepts mob. No. of new user
    - DoB : Date Picker
    - Gender : Radio Buttons
    - All details should be saved in database
* Acceptance Criteria
  + - Mob No : TextBox 🡪 Only Integers , 10 digits
    - Firstname : Only alphabets ( 30 – 40 chars)
    - Lastname : Only alphabets
    - DoB : Atleast 16 yrs should be completed for user
    - Gener Radio buttons : Selection of on only

…………………………………………………………………………………………………………………………………………………………

Ecommerce 🡪 Laptop Buy 🡪 Credit Card / Debit Card

As a registered user I want to use my credit card / debit card to pay the amount of my order so that my order will get placed.

Acceptance Criteria :

* Discovery of the type of card (visa/mastercard/Rupay)
* Validate card information 🡪 Bank
* CVV and Expiry Date 🡪 Validate
* OTP 🡪 Validate
* Payment Successful 🡪 Response 🡪 confirm / fail

…………………………………………………………………………………………………………………………………………………………….

**Explain Your Daily Routine**

Daily start 🡪 9 am 10 am

First Thing 🡪 Mails / Replies 🡪 OUTLOOK

Stand Up 🡪 11 am – 11.15 am

Get back to our work 🡪 Testing

Lunch 1 – 1.30 pm

Collegues Discussion / meeting

3.30 pm 🡪 All testers call 🡪 Challenge

6.30 pm Lead 🡪 Update

7.00 Log out

…………………………………………………………………………………………………………………………………………………………….

**Story Point**

Solid Sub 🡪 Measure 🡪 Kg/ gms/ mgs

Liq 🡪 Lit. ml etc

Distance 🡪 mtr/ km etc

**Story Point 🡪 It is measure of complexity and size of the user story.**

Story Point 🡪 Estimation

Maggie for 1 person 🡪 5 mins Complexity

Puranpoli / Mutton for 1 person 🡪 1 hr

Maggie 🡪 1000 person 🡪 2 hr 🡪 Size / Quantity

Scale 🡪 1 to 10

7 6 8 9 10

1 2 3 4 5 etc.

IT 🡪

Sprint Planning 8 hrs 1 Story Point

Story Point 🡪 1 🡪 8 hrs

Story Point 🡪 5 🡪 40 hrs

………………………………………………………………………………………………………………………………………………………….

Sprint Duration = 14 Days

Start Date 1st May

End Date 14th May

Week Off 🡪 4 days ( sat sun sat sun)

Actual working Day 🡪 10 days

**RedBus**

**2000 User Story 🡪 Product Backlog**

**Stakeholder 🡪 PO 🡪 Scrum Master & Team = 20 User Story**

Bus Booking = 4 US

Flight Booking = 5 US

Rail Booking = 5 US

Car Pool = 6 US

**Grooming / Backlog Refinement** 🡪 20 User Stories 🡪 Knowledge / Doubt Clear 🡪 Ready 🡪 Sprint 🡪 PO 🡪 28th April

**Week 1**

Day -1 1st May

* Sprint Planning
  + Work Assign / Allocation 🡪 (US01, US02 Varun Dev / Divya Tester )
  + Estimation 🡪 How many hours you will require to complete this? Story Point
  + Start Sprint
  + Get back to the work
  + US 01 – Varun – 16 HRS tester Divya 8 hrs
  + Varun 4 hrs work complete

Day – 2 2nd May

* Daily Standup meeting / Scrum
  + What I did yesterday?
  + What I am going to do today?
  + Any blocker ?
  + 15 mins
  + Varun – 8 hrs

Day – 3 3rd May

* Daily standup meeting
  + 4 hrs 🡪 US01 🡪 Complete
  + Divya 🡪 4 hrs

Day – 4 4th May

* Daily Standup meeting

.

.

.

.

Day 11 11th May

* Grooming / Backlog Refinement

.

.

Day – 14 14th May

* Daily Standu Up meeting
* Demo Preparation Call
* Sprint Review Meeting
* Sprint Retrospective meeting

……………………………………………………………………………………………………………………………………………………………

Questions:

1. What is s/w testing ?
2. What is SDLC?
3. What is difference SDLC and STLC?
4. What is BRS ?
5. What is SRS and what it contains?
6. When tester will start TCD?
   * 1. SRS, Doubts then we will start the test case design.
7. What is WBT?
8. Advantages of WBT ?
9. What is BBT?
10. Difference between WBT & BBT
11. What is Static and what dynamic?
12. Difference between Validation and verification.
13. What is your team size ? 10 dev 3 tester 1 designer BA 1 1 PM 1 DM
14. How many developers ?
15. How many testers?
16. What is Waterfall model ?
17. What is agile methodology?
18. Difference between agile and waterfall.
19. Advantages of Agile
20. Disadvantages of Agile
21. Disadvantages of Waterfall
22. Where do we use waterfall and where do we use Agile?
23. Who decides the methodology?
24. What is agile?
25. What is EPIC ?
26. What is Burn up, burn down and velocity ?
27. Explain all agile ceremonies. Grooming , Sprint Planning, Daily scrum/daily stand up, Sprint review, Sprint Retro
28. What is story point?
29. What you have discussed in your last retro?
30. Dev – Late – So testing delayed- Dev team should stick to deadlines and estimate.
31. How agile is performed in your org ?
32. What you will do if any of the task is not getting completed in this sprint?
33. What is sprint zero ?
34. What is user story ?

……………………………………………………………………………………………………………………………………………….

Responsibility 🡪 Google Sheet 🡪 Questions Type / Manual Java

………………..

Manual Testing 🡪 Concepts 🡪 Today Point / Previous Topic Revision / Q and A

……………………………………………………………………………………………………………………………………………

Agile Advantages

* Flexibility 🡪 Change Request / Development
* Less Documentation
* Fast and Continuous Development
* Customer satisfaction / CR money
* Constant collaboration with Stakeholders
* Quick Delivery of product
* Good Communication in team
* Faster feedback from customer 🡪 Review
* On time delivery
* ~~Higher development cost --~~
* Quick identification and elimination of errors in product
* Every cycle is tested
* Transparency – jira / Standup
* Less confusion on product
* Cust. Gets overview of software due to sprint wise releases
* Checkpoint \*
* Incremental
* Efficient utilization of Human Resources \* (100)

…………………………………………………………………………

Disadvantages

* Fragmented output
* KT – Knowledge Transfer / Transition
* Not applicable in small projects
* No clear end
* Deviation from the requirement
* Cost is more (Skillfull and Experiences)
* Time bound and strict deadlines (5%, 10%)
* Clear understanding US
* Bug possibility 🡪 High 🡪 due to continuous development

………………………………………………………………………………………………………………………………………………

Checkpoint

* Roads
* Railway
* Flights
* Water ways

…………………………………………………………………………………………………

**Waterfall Model**

It can be defined as sequential model for development and testing of software/application.

Sequential – After completion of 1 stage, we proceed to next stage.

Oneplus 11R

In-house testing team 2 bugs

1 million 🡪 Use -🡪 100 bugs

Freelancers Testers 🡪 20 bugs

1 month

Total bugs 122

2-3 months software update

Software Update 🡪 122 bugs fix

Advantages

1. Product definition stable

2. Project duration is less

3. Used for small projects

4. This model is very easy to understand and implement due to its rigidity.

Disadvantage

1. Back track is not possible.

2. Not for complex or big project where requirements are changing

3. Delivery is delayed and can take more than 3 months too ( usually 2 to 3 months)

4. If we find any bug then we can not send it back, but we will have to create a report (document it) and then it will get fixed in next release.

Small Projects 6 months

100

60% Requirement 🡪 Analysis 🡪 Design 🡪 Dev 🡪 testing (bugs/defect) 🡪 delivery 1st May – 31st July

40 % + Bugs from 1st Release Requirement 🡪 Anlysis 🡪 design 🡪 dev 🡪 testing 🡪 Delivery 1st August – 31st Oct

Where ?

Product based 🡪 Bank Management 🡪 Sell

Small Project

………………………………………………………………………………………………………………………………

Who decides the methodology?

HSBC 🡪 Bank 🡪 In house IT team 🡪 Waterfall/Agile 🡪 Client

Bikaner Sweet 🡪 Our Team (requirement)

Clothes Showroom 🡪 IT Company 🡪 Client

……………………………………………………………..

Logical Thinking

Problem Solving Ability

……………………………………………………………

Problem 🡪 Google Baba / Stackoverflow 🡪 Colleagues 🡪 Senior 🡪 TL

…………………………………………………….

**Verification and Validation**

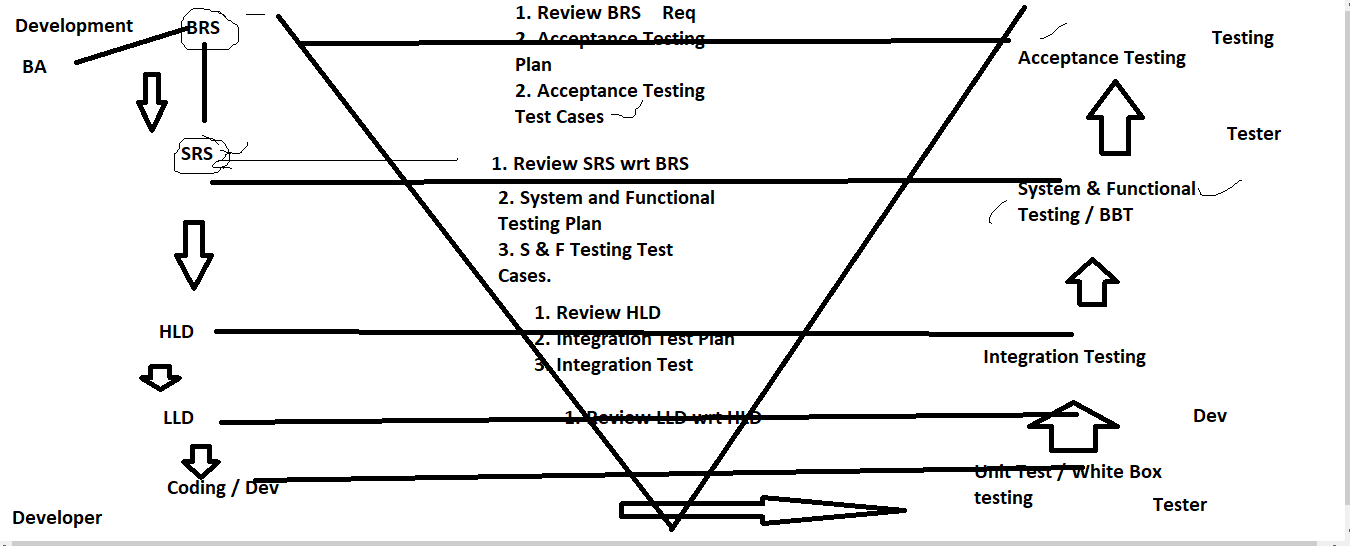
**Soil Vassal Mfg**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| We check the we are developing the right product or not. | We check whether the developed product is right or not. |
| Verification is also known as static testing | Dynamic testing |
| Quality Assurance | Quality Control |
| Review / Inspection/ Walkthrough (demo) | System and Functional / Acceptance testing |
| Execution of code - NO | Execution of code - YES |
| Bugs – Early idenfication | Bugs escaped/ miss from verification process |
| Verification is done before validation | After Verification |

…………………………………………………………………………………………………………………………………

**V-Model**

**Verification and Validation Model**



This model came up in order to overcome the drawback of waterfall model – here testing starts from the requirement gathering phase.

The V model is shown in the figure in the next page.

1) In the first stage, the client gives requirement doc to BA and BA sends the BRS both to developers and testers.

The testers do the following tests on BRS,

1. Review BRS

a. conflicts in the requirements

b. missing requirements

c. wrong requirements

2. Write Acceptance Test plan

3. Write Acceptance Test cases

The testing team reviews the BRS and identifies mistakes and defects and send it to the development team for correcting the bugs. The development updates the CRS and continues developing SRS simultaneously.

2 ) In the next stage, the SRS is sent to the testing team for review and the architects and senior developers start building the HLD of the product. The testers do the following tests on SRS,

1. Review SRS against BRS

a. every BRS is converted to SRS

b. BRS not converted properly to SRS

2. Write System and functional Test plan

3. Write System and functional Test case

The testing team reviews every detail of the SRS if the BRS has been converted properly to SRS.

3 ) In the next stage, the architecs and developers start building the LLD of the product. The testers do the following tests on HLD,

1. Review HLD

2. Write Integration test plan

3. Integration test (dev)

4 ) In the next stage, the developers start with the coding of the product. The testing team carries out the following tasks,

1. Review LLD

After coding, the developers themselves carry out unit testing or also known as white box testing. Here the developers check each and every line of code and if the code is correct. After white-box testing, the s/w product is sent to the testing team which tests the s/w product and carries out functional testing & system testing and acceptance testing and finally deliver the product to the client.

How to handle requirement changes in V&V :-

Whenever there is change in requirement, the same procedure continues and the documents will be

updated.

**Advantages of V&V model**

1) Testing starts in very early stages of product development which helps in finding defects early

2) Testing is involved in every stage of product development

3) Development and testing stages are in parallel/simultaneous or mapped together – as developers are building SRS, testers are testing BRS and also writing ATP and ATC and so on. Thus as the developers give the finished product to testing team, the testing team is ready with all the test plans and test cases and thus the project is completed fast.

**Disadvantages**

1) Initial investment is more – because right from the beginning testing team is hired

2) More documentation work – because of the test plans and test cases and all other documents

Applications of V model

**When to apply**

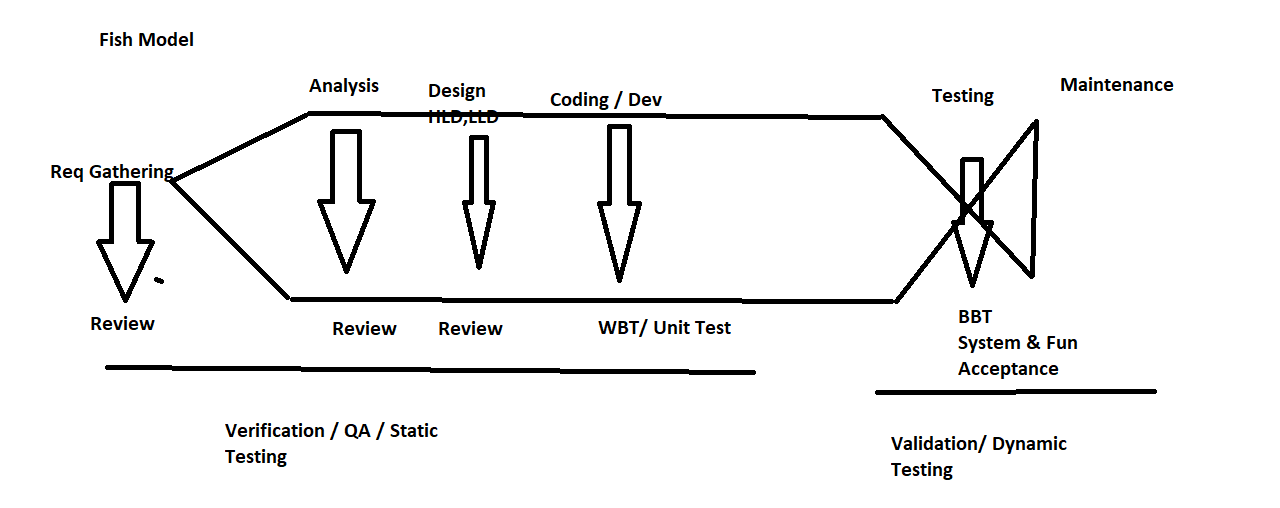
1) for long term projects

2) for complex applications/softwares

3) when customer is expecting a very high quality product within time frame because every stage is tested and developers & testing team are working in parallel

……………………………………………………………………………………………………………………………………………….

**Fish Model 🡪 Advanced SDLC**



**SQA – Software Quality Assurance**

Practice of monitoring the s/w development process (SDLC) and methods in a project to esure proper quality of the s/w.

Monitoring and measuring the factors in development of software.

BA / PO

**Factors**

1. Fullfillment of customer requirement

What type of project ? – Banking / E-commerce

Purpose ?

2. Customer Expectation

a. Performance 🡪 High

b. Security 🡪 Privacy of all cutomer 🡪 Data to be protected.

3. Cost of Project

Human Resources Allocation

4. Timely Delivery

Penalty 🡪 5%, 10% (Fine)

5. Risk

Prajakt 🡪 3 yrs 🡪 Resign (Dependent Resource)

6. Maintenance

……………………………………………………………………………………………………………………………………………………

**Test Scenarios & Test Cases**

**Test Scenarios**

Any functionality that / which can be tested is called as Test Scenario.

Eg. Laptop Buy

Checking the hinges ( Functionality that can be tested)

Verification Recharge Functionality 🡪 Scenario

**Test Cases**

Set of actions that needs to be executed to test/verify a particular feature and functionality of s/w.

Checking the hinges

Test Cases –

1. Open the box

2. Verify the hinges externally

3. Open the flip

4. Close the flip

5. Repeat step no. 3 and 4.

……………………………………………………………………………………………………………….

Verification of Recharge Functionality –

Test Cases

1. Open the browser

2. Hit the url <https://paytm.com>

3. Click on recharge icon

4. Select Prepaid or Postpaid radio button

5. Enter valid mob no

6. Select valid Operator from dropdown

7. Select Circle from dropdown

8. Select proper plan

9. Click on Proceed Recharge

10. Complete payment

11. Success

……………………………………………………………………………

Test Scenario : Verification of Login Functionality

Test Cases : Positive

1. Open browser 🡪 Browser should open

2. Hit URL <https://fb.com/login> 🡪 Login page should open/load

3. Enter valid Username 🡪 Username should be

4.Enter valid Password 🡪 Password should be accepted

4. Click on Login button 🡪 Login should be successful and he should be taken to homepage

…………………………………………………………………………………

……………………………………

|  |  |
| --- | --- |
| Username Invalid | Password valid |
| Username Invalid | Password Invalid |
| Username Valid | Password Invalid |
| Username Blank | Password Valid |
| Username blank | Password invalid |
| Username valid | Password blank |
| Username invalid | Password blank |
| Username blank | Password blank |

………………………………………………………..

Test Case :

1. Open browser 🡪 Browser should open

2. hit URL <https://fb.com/login> --> Login page should open

3. Enter invalid username 🡪 Username should be accepted

4. Enter valid password 🡪 Password should be accepted

5. Click on Login button 🡪 Login should fail / User should not get logged into the system.

…………………………………………………….

Test Scenarios: Verification of writing feature of Pen

Test Cases

1. Buy the Pen

2. Open the cap 🡪 Cap should get open properly without breaking the point.

3. Close the cap 🡪 Point should be covered by the cap and no ink should be disbursed by the pen.

4. Open the cap 🡪 Cap should get open properly without breaking the point.

5. Write on the Paper –“ Hello World” 🡪 Writing should be smooth and ink spreading should be consistent.

…………………………………………………………………………………………………………………….

PhonePay 🡪 UPI 🡪 9421300899@ybl

1. Click phonepay 🡪

2. Enter password to open app

3. Click “To Bank / UPI ID”

4. Click UPI ID

5. Add valid UPI ID

6. Verify UPI ID

7. UPI Save

8. Enter amount 🡪 50,000 /-

9. Click on Proceed to pay

10. UPI Pin

11. Success Payment

………………………………………………………………..

Expected Result

Actual Result

Expected Result = Actual Result 🡪 NO BUG 🡪 Test Case / Test / Testing 🡪 PASS

Expected is not equals to actual 🡪 BUG 🡪 Test case / test 🡪 FAIL

Pass / Fail / WIP / Blocked / Incomplete -🡪 Status

………………………………………

Test Case 🡪 100 Steps 🡪 67 Step fail 🡪

Overall Test Case 🡪 FAIL

……………………………………….

Install 🡪 1st 🡪 SignUp / Registration 🡪 Login

Login 🡪 Signup

……………………………………………….

Password :

Min 8 Char

Min 1 Upper Case

Min 1 Lower Case

Min 1 Number

Min 1 Special Character

Velocity@123 🡪 Valid

velocity@123 🡪 Invalid

VELOCITY@123

1234567890

!@#$%^&\*()

ASDFGI@%$^466

ASDFGHJK123456

………………………………………………

SignUp Page 🡪 Positive Scenarios

Signup Page 🡪 Negative Scenarios

Whatsapp / Insta / google meet / Zomato / Swiggy / Amazon / Myntra / Ajio / nykaa / Misho / Phonepay / gpay / flipcart / paytm / snapchat etc

Individually / Mock 🡪 Zomato

Signup

Login

……………………………………………

**Environment**

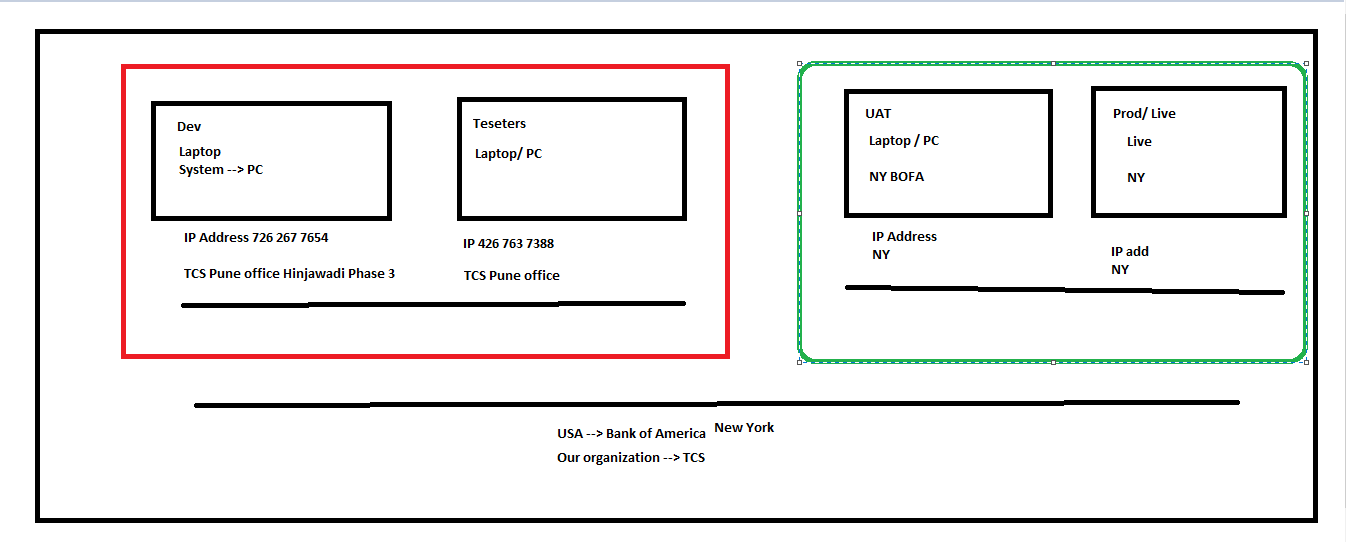
Atmosphere 🡪 Surrounding

1. Dev

2. Testers

3. UAT User Acceptance Testing

4. Live / Production



**1. Developers Environment**

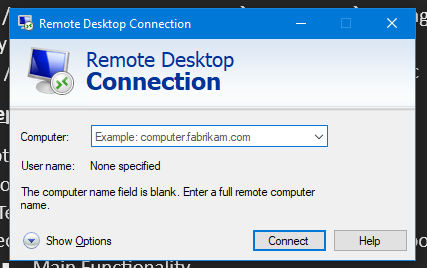
* Developer Coding / development
* Unit Testing / White box testing 🡪 Positive Scenario
* Integration
* Integration Testing
* Mail / JIRA / Communication Software ping – inform Tester 🡪 I have completed the development process along with the unit testing and integration testing. Ready for Testing.
* Docs / URL / Login credentials / Test data / unit testing results etc

**2. SIT – System Integration Testing Environment**

* Developer will deploy build (code) to SIT
* Smoke Test
* Sanity Test
* Functional
* Non – functional
* Security Testing
* Performance testing
* Regression Testing
* Bug 🡪 Raise / Report 🡪 JIRA 🡪 Revert to dev
* Fix the bug 🡪 retesting
* Retest 🡪 Bug resolved 🡪 happy
* Testing Done
* Mail / JIRA / Communication soft ping 🡪 UAT Team 🡪 Testing has been completed successfully on SIT. You can initiate UAT.
* Docs / URL / Login credentials / Test data / testing results etc

**3. UAT – User Acceptance Testing environment**

* User Acceptance Testing
* Testing is done in front of client
* Testing 🡪 Test Cases
  + Selective Test Cases (2000 Total 🡪 100 test cases choose UAT)
    - Main Functionality
    - High Priority
* UAT Team 🡪 TESTERES
  + Client Side
  + Our Testers
* If there is no separate UAT team in that case , testers who worked in SIT, can conduct UAT.
* UAT 🡪 On client environment
* Use 🡪 RDC Remote Desktop Connection



* Were you involved in UAT ??
  + Yes. I was involved in UAT Team sometimes.
* Bug 🡪 Immediate Fixing (Highest Priority)
* UAT Done 🡪 Dev Team Mail / Jira / Communication 🡪 UAT is done. You can make this live/ You can deploy this on production. 🡪 Dev team will deploy this on Prod/Live.
* SIT 🡪 UAT 🡪 UAT Team 🡪 SIT Test case 5000 TC 🡪 Put in front of client 🡪 Choose / Hand pick 🡪 Execute on app in front of client.

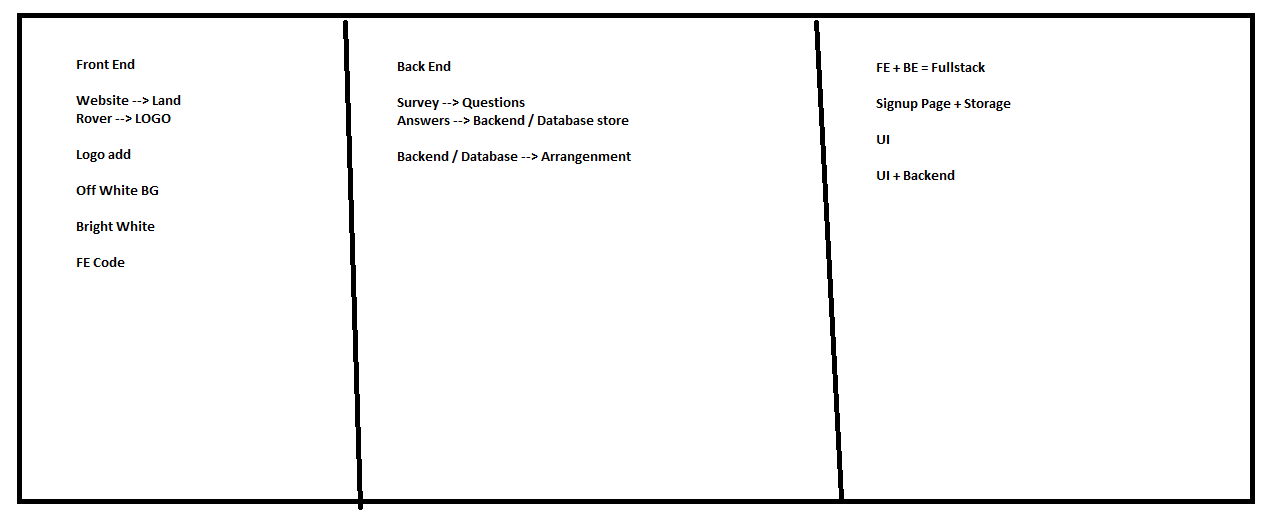
**4. Live / Production**

* Live
* End Users 🡪 Start using website

……………………………………………………………………………………………………………………………………………….

1**. Developer’s Environment**

* **Unit Testing** 
  + Developer
  + Positive Scenarios
  + Code level analysis
  + Only the piece of code which he has developed
  + If any defect/bug arises then dev fixes it immediately.
* **Integration Testing** 
  + Integration 🡪 Dev will combine all submodules with main module
  + Integration Testing 🡪 Dev will test the main module which is created by combining many submodules.
  + Integration is of 2 types
    - Frontend
    - Backend / Database Integration



* Starting Module 🡪 Later modules not ready
* Starting module not ready 🡪 Later modules ready
* Starting No / End NO 🡪 Intermediate Ready /Mid
* 3 Approaches for testing

a**. Top-down approach**

We don’t have submodules ready (later modules/ sub-modules are not ready)

Temporary Program Create

STUB

**b. Bottom Up**

We don’t have starting modules or main modules but we have later modules

Temporary programme

DRIVER

**c. Bi-directional / Sandwich**

Start No – End No } We do have middle modules

Stub and Driver ……………………………………………………………………………………………………………………………………

**2. SIT Environment**

System and Integration Testing Environment

Tester 🡪 WE

Test Environment

* Smoke Test
* Sanity Test
* Functional
* Non – functional Testing
* Bug Raise 🡪 Dev 🡪 JIRA
* Fix
* Retest
* Regression Test
* **SANITY TESTING**
  + ~~Developer 🡪 Development Task~~
  + ~~Build Deploy 🡪 SIT 🡪 Testers Environment~~
  + ~~Sanity 🡪 Basic Build Verification~~
  + ~~Sanity 🡪 Basic Stability of build~~
  + ~~Different Validations:~~ 
    - ~~GUI / UI 🡪 User Interface / Graphical User Interface~~
    - ~~Links – Hyperlinks~~
    - ~~Tabs/Pages~~
    - ~~Navigation 🡪 Reload, Back, Forward~~
    - ~~Core functionality~~
  + ~~First Testing / Level Zero / Zero Level~~
  + ~~Tester’s Acceptance Test~~
  + ~~Build Verification Test~~
  + ~~Defect / Bug~~
  + ~~I CAN’T TEST THIS BUILD~~
  + ~~Developer 🡪 Inform~~
  + ~~Developer will rework on this 🡪 Updated build~~
  + ~~Paytm Build 🡪 Recharge V2.0 🡪 Recharge V2.1~~
  + ~~Test Cases 🡪~~ **~~NOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO~~**
  + ~~Bugs 🡪 System Hangout / Runout / UI Glitches / Popup , links , tabs , pages load etc / Core fun not working~~
  + ~~Time 2 – 3 hrs~~

……………………………………………………………………………………………………………………………………………….

* **Smoke Test**
  + ~~Sanity + Troubleshooting~~
  + Developer 🡪 Development Task
  + Build Deploy 🡪 SIT 🡪 Testers Environment
  + ~~Sanity~~ 🡪 Basic Build Verification
  + ~~Sanity~~ 🡪 Basic Stability of build
  + Different Validations:
    - GUI / UI 🡪 User Interface / Graphical User Interface
    - Links – Hyperlinks
    - Tabs/Pages
    - Navigation 🡪 Reload, Back, Forward
    - Core functionality
  + First Testing / Level Zero / Zero Level
  + Tester’s Acceptance Test
  + Build Verification Test
  + Defect / Bug
  + I CAN’T TEST THIS BUILD
  + Developer 🡪 Inform
  + Developer will rework on this 🡪 Updated build
  + Paytm Build 🡪 Recharge V2.0 🡪 Recharge V2.1
  + Extra sake of Sanity
  + Bug / Defect 🡪 Root Cause find out 🡪 Troubleshooting -🡪 RCA
    - Requirements Misunderstanding
    - Functionality Not understood
    - Ambiguity from Client
    - Not understanding the flow of Functionality
    - Problem in code logic
  + Tester + Developer
  + Sanity + Troubleshooting (tester + Package Validation – developer)
  + Test Cases 🡪 NOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
  + 2 – 3 hrs
  + **IN MY PROJECT 🡪 SMOKE TESTING**

…………………………………………………………………………………………………………………………..

3. System and Function Test

We will test all system and software functioning/ functionalities.

a. Usability Testing

Usability 🡪 Which is usable 🡪 User Friendly

Tester will validate the USER Friendliness of the screen / GUI / Functionality.

i. GUI / UI / Frontend

Look and Feel of UI

Wireframe , Designer 🡪 UI Specification Document (font , colour, sections, Header , footer etc)

Ease of application / functionality

Speed of Interface

ii. Manual Support Testing

Sensitiveness of screen

Validation of manual input values

b. Function Testing

Validation of internal and external features of application.

i. Functionality Testing / Functional 🡪 (BIE BSC) Validation of all internal features.

Ex. All functionalities in an application, Recharge in phonepay etc.

ii. Non – functional Testing 🡪Validation of all the external features.

Ex. Google Chrome, Safari 🡪 Mac OS

i. Functionality Testing / Functional Testing (BIE BSC)

**a. Behavioural Coverage**

Validation of behaviour of the object which is present on UI.

Behaviour and Property of Object

|  |  |
| --- | --- |
| **Object** | **Property** |
| Text Box | Click 🡪 Cursor Blinking, Focus & Unfocus |
| Radio Buttons | On & Off / Selection of one / Deselect |
| Check Boxes | Multiple Select, Check/Uncheck |
| Buttons | Clickable – Not clickable / Enable or disable |
| Drop Down | Populate all values and wraps it up after another click |
| Hyperlink | Clickable and open up the new URL |
|  |  |

**b. Input Domain Coverage**

Validation of input which we are passing into objects.

Checking size/ length and data type of input into the object.

**1) BVA – Boundary Value Analysis** 🡪 Input Size and Length

Validation of size and length

Mob No: 10 Integers

Deposit Min. Amount 1 RS – 10,00,00,000 RS (1 to 9)

|  |  |  |
| --- | --- | --- |
| Example | Pass | Fail |
| Mobile No | 10 Digits  Min = 10  Max = 10 | Min -1 =10-1 = 9  Min + 1 = 10+1= 11  Max-1 = 10-1=9  Max+1 = 11 |
| Bank Deposit 1 Rs Max 10,00,00,000 | Min = 1  Max = 10,00,00,000  Max-1= 9,99,99,999  Min + 1 = 2 | Min – 1= 1-1 =0  Max + 1 = 10,00,00,001 |
| Bike Parking  (1000 Bikes )  Min =0  Max = 1000 | Min  Max  Max-1  Min +1 | Max+1= 1001  Min-1 = -1 |
| Cinema Hall  (Max 500)  5 People show start | Max 500  Min 5  Min +1  Max-1 | Min -1  Max+1 |
| Indigo Airlines  Seats 260  Min 50 Passengers | Max = 260  Min 50  Min +1  Max-1 | Min -1  Max+1 |

**2) Equivalence Class Partition** 🡪 Data type

Validation of input data type

Int / float / Boolean/ double / char / string etc

|  |  |  |
| --- | --- | --- |
| Example | Pass | Fail |
| Mobile No. | Integer | String / Char / float / Boolean etc decimal hexadecimal etc |
| Password  (Alphabet/ Spe Char / Number) | Int, String, Char, Spl Char | Boolean, Blank , decimal etc |

**3) Decision Table Technique**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Objects | Combi 1 | Combi 2 | Combi 3 | Combi 4 | Combi 5 |
| Username | Valid | Valid | Invalid | Invalid | Blank |
| Password | Valid | Invalid | Valid | Invalid | Blank |
| Login Button | Click | Click | Click | Click | Click |
| Result / Output | Success | Fail 🡪 Error Message | Fail 🡪 Error Message | Fail 🡪 Error Message | Fail 🡪 Error Message |

**c. Error Handling Coverage**

Validation of different type of error messages which are generated by application / software.

When we pass invalid data or when we do some invalid operation.

**d. Calculation based coverage**

Validation of all arithmetic / mathematical operation

**e. Backend Coverage / Database Testing**

Signup form 🡪 Details Fill out 🡪 Save / Proceed 🡪 Data Store 🡪 Database

~~Laptop word File 🡪 Save 🡪 Hard Disk Save~~

Database 🡪 Structured Query Language SQL

SQL 🡪 Close Proximity with English

FirstName Mobile No.

**f. Service Based Coverage**

Validation of the sequentially Functionality of the application

Signup 🡪Login Page open 🡪 Valid Username 🡪 Valid Password 🡪 Click login 🡪 Success

Recharge 🡪 Open 🡪 login 🡪 Recharge 🡪 Mob No 🡪 Prepaid 🡪 Operator 🡪 Circle 🡪 Plan 🡪 Payment 🡪 Success

c. Security Testing

d. Performance Testing

ii. Non-Functional Testing (RCC IISPG)

a. Recovery Testing

b. Compatibility Testing

c. Configuration Testing

d. Installation Testing

e. Intersystem Testing

f. Sanitization Testing

g. Parallelization Testing

h. Globalization Testing

**a. Recovery Testing**

Validating the response of the application for abnormal situations.

We are going to check the reaction of application.

Whenever situation is normal, then application should be back on track and working fine.

Ex. Google Search 🡪 Manual Testing 🡪 Data Off 🡪 Dino Game 🡪 Internet is back 🡪 Page Reload / It will resume the 9th Page.

Ex. HDFC Bank 🡪 Internet Banking Login 🡪 Balance 🡪 Fund transfer page 🡪 Net off 🡪 Net is back 🡪 LOGIN AGAIN 🡪 It will resume the app again from START.

**b. Compatibility Testing**

Suitable / adjustable

Build or software is compatible/suitable on user expected platform or not.

Operating System 🡪 Windows, Linux, Unix, Ubantu, Fedora , IOS, Android etc.

Browser 🡪 Chrome. Firefox, brave , safari, edge etc.

**1. Operating System Compatibility**

Our build or s/w is compatible on user expected Operating System.

**2. Browser Compatibility**

Our build or s/w is compatible on user expected browser.

**i. Cross Browser Compatibility**

We test it on different – different browsers

Chrome Edge Opera FF Safari

**ii. Version Control Compatibility / Browser Version Control Compatibility**

We test the same application on a browser but on different different version

Chrome Current Version 🡪 2.04

Chrome 🡪 2.03 / 2.02 / 2.01

Exe files 🡪 Installation 🡪 internet

…………………………………………………………………………………………………………………………………………………..

1. Forward Compatibility Test

Build is OK but OS / Browser is not OK.

Less bugs expected than BCT.

2. Backward Compatibility test

Build is not OK , OS / Browser is OK

More Bugs expected

…………………………………………………………………………………………………………………………………..

**C. Configuration Testing**

Hardware Compatibility Testing

Testing of our application is supporting the hardware devices or not.

MMT 🡪 Ticket 🡪 Print Ticket Button 🡪 Dialogue Box – Print Details 🡪 Print Button

**d. Sanitization Testing**

~~Swacchh Bharat Abhiyan~~

Testing extra feature developer has added but not present in customer’s requirements.

Extra features in application 🡪 Tester will test it 🡪 Raise it as defect/bug

Mobile No.

9421300899

+91

9421300899

Mobile No.

Garbage Testing

…………………………………………………………………………………………………………………………..

**e. Globalization Testing**

Checking whether our application is supporting different – different lang.

1. Local Languages Testing 🡪 Eknath Shinde 🡪 SEW 🡪 Application 🡪 Maharashtra 🡪 Marathi

Nagaland State Gov 🡪 Naga local lang.

2. International Language Testing 🡪 Kim 🡪 Korian 🡪 Application

Israel 🡪 Hibrew 🡪 Application

Countries Official languages

3. Global / Standard Lang. testing 🡪 English

Google Translator

**f. Installation Testing**

NOT DONE IN SIT

WE ARE NOT INVOLVED

Process of checking the installation of our build into existing software or on user expected platform.

1. Setup program execution

Check the exe

Checking diff. packages

Drivers

2. Easy Interface

Easy Installation process

3. Disc Space

250 GB

To check that system has sufficient disc size or not

4. Uninstallation

**g. Intersystem Testing**

Paytm

IRCTC

Inox/PVR/Esquare

Whether our application is able to share the resources / data/ info with other systems/ apps.

**h. Parallelization Testing**

We check our product with similar other product.

Qulcom snapdragon 630, 8 gb ram, 256 GB 🡪 Samsung (our product)

Qulcom snapdragon 630, 8 gb ram, 256 GB 🡪 Oppo / Vivo

Comparison Testing

Eg. Phone pay (our) and Google Pay

Not done in SIT Environment.

…………………………………………………………………………………………………………………………………………………

Address

Mob

DoB 🡪 HIPPA Healthcare

Gender 🡪

Server 🡪 Store 🡪 Auditing 🡪 Penetration Testing 🡪 Hacker

Western 🡪 Sundar Pichai US / UK / European Union ( Countries )

Data without permission / Data leak / Data breach 🡪 Annual Turnover \* 20 Times

1 LCR = 20 LCR

Chinese 🡪 Service Product 🡪 INDIAN Partner

MI Phone 🡪 TATA / CEO

Servers 🡪 India

……………………………

**Security Testing**

Sensitive Information

Privacy

Testing of all the processes in application which are related to Privacy and sensitive information of the user.

1. Authorization

FB 🡪 Login 🡪 Authorized User 🡪 User Registration (SignUp)

Infosys 🡪 ID Card Access card / System Credentials

Healthcare 🡪 Doctor

Cust/ Chef / Waiter / Manager / Owner / Housekeeping

2. Access Control

Process of checking whether the **authorized person** has permission to access a specific function.

Kitchen 🡪 Restricted

~~Cust~~/ Chef / Waiter / Manager / Owner / ~~Housekeeping~~

Whastapp Group

ADMIN Rights

Add

Remove

Chats Control

Flipcart 🡪 END USERS

ADMIN

3. Encryption and Decryption

Raazi / IB72 /

--..--..--……-

Court 🡪 Shorthand 🡪

Here is danger 🡪 Parrots

Apple Logo

Turing machine 🡪 encode

Hi Team 🡪 w3e45rt7yuhjbvcfdr56t7y8uijknbhvgcdre5r6t7y8uihjbhvgct

URI 🡪

Money Heist

Berlin Hi Team 🡪eertyghbvcdrtyuhijknmbvggyuhjk

Profeessor Positive

UserName Password

Password 🡪 Database

Database

FB Employee Access

Password Table 🡪 Database 🡪 Dhananjay#1 🡪 23456789oiuytrewsxcghjkmnbvcxer5tyujnbvcdertyhbvcfrtyujmnbvcdr5678iokmjnbfr567ujnbvfrtyujm

Pass@123 🡪 ed5r6t78uioklmknmbcxdsertyuiolkjhtredfghyuioiuytrewqasdfghjkmnbvcdert

………………………………………………………………………………………………………………………………………..

9.59 am 🡪 1 Lack 2 Lack

VCTCPUNE 🡪 1 Lack

10 am AC

11 am SC 10 CR

Server 🡪 Capacity 🡪 IRCTC 50 Lack Users

Load Balancing

Load Balancing Servers

Main Server Exact Copy

50 L +

…………………………….

VCTCPune 🡪 5 L Gradually

5 L Sudden

5 L 🡪 6 hrs

1 10 1 10

……………………………….

Tools

Jmeter 🡪 Thread Group 🡪 Thread 🡪 User

5 L Threads

Multithreading

…………………………………….

Performance Testing?

E-Marketing

Whole World

PT ??

VCTC Pune 🡪 App 🡪 Exclusive 🡪 Staff 🡪 100 + 200 = 300 / 400 = 1000

………………………………………….

……………………………………………………………………………………………………………………………………………….

Retesting

Re+Test

Re+Turn

Re+Check

* User Story testing is going on
* Bug (Exp Act not matching)
* You will report / raise 🡪 JIRA
* Dev FIX
* Again he will give you a modified build 🡪 SIT
* RETEST
* Re 🡪 Again we have to do that activity
* Recharge 🡪 Bug
* Retest is always done by passing multiple test data.
* Test Data : As a tester 🡪 Data input
  + OLD Project – We will be having the test data on databases. (fb)
  + New Project –1 🡪 Tester and Developers will create the data.

Small Quantity 🡪 Tester

Huge Quantity 🡪 Developer

* + New Project –2 🡪 Amazon 🡪 Payment Getaway 🡪 Phonepay
    - * + Test data has some dependency over other project / app/third party

Client 🡪 Phonepay

……………………………………………………………………………………………………………………………………………….

**Regression Testing**

Testing 🡪 Bug 🡪 Raise 🡪 Dev will fix it 🡪 New modified build 🡪 RETESTING (Multiple Data)

Bug Fix 🡪 Coding / Changes in code / MODIFICATIONS 🡪 class A (method) / class C 🡪 Application IMPACT

SignUp 🡪 First Name / LastName / Username / Password 🡪 Profile Page / Post / Comment / Home Page / Login Page

A. Signup / Profile/ Login Page 🡪 Bug 🡪 Login Page Regression

B. Signup 🡪 Login Page 🡪 Impact Check Signup Test

Login 🡪 Varun 🡪 Dev Work 🡪 Tester 🡪 Test Cases

Signup Regression 🡪 Signup TEST CASES 🡪 Signup dev TC

Usually, We do not prepare test cases for regression, we use already prepared test cases.

Regression is nothing but “Re-execution of tests/ test cases on modified build”

Test execution in order to analyse / ensure the impact.

2 Places / Types

1. SIT 🡪 Build 🡪 Smoke 🡪 Fun & NFun 🡪 Bug 🡪 Raise 🡪 Dev 🡪 Fix 🡪 Modified Build 🡪 Retest with multiple test data 🡪 if there is impact on other modules 🡪 Regression

2. SIT 🡪 Build 🡪 Smoke 🡪 Fun &NFun 🡪 Testing Passed 🡪 Impact is on another module 🡪 Regression

3. **Final Regression** 🡪 When our application is moving from one environment to another environment then there will be final regression 🡪 Whole Application

SIT 🡪 UAT 🡪 Prod. 🡪 Final Regression to ensure that nothing is breaking due to change in environment.

HOW?

Test Cases 🡪 Yes

Suit 🡪 Collection / Bunch

Test Suit 🡪 Collection of Test Cases.

Regression Test Suit / Regression Suit 🡪 Collection of regression related test cases.

~~5000 Test Cases 🡪 No~~

a. Failed Test Cases

b. Highest Priority Test cases 🡪 Core & critical functionality

c. Extra feature / function test cases

d. If time permits, we can add more (high / med / low)

Regression Suit 🡪 How many test cases ? 🡪 200 to 300

Duration: 2 – 3 days Estimate

When Final Regression 🡪 Once in a month before the client release.

……………………………………………………………………………………………………………………………………………

End to End Testing

SIT Environment 🡪 Tester Envi.

* We will check the application / functionality before going to UAT, this type of testing is called as end to end testing
* Integration Testing
* Yes, After completion of module.

……………………………………………………………………………………………………………………………………………..

**UAT USER ACCEPTANCE TESTING**

SIT 🡪 Done 🡪 Mail / Jira / MS Teams Ping 🡪 SIT done 🡪 UAT Start 🡪 Test Result / docs etc

UAT 🡪 Diff envi

Client Environment

RDC

2000 Test Cases

Hand pick 🡪 Core Fun & Critical Fun.

UAT Separate UAT Team / SIT Testers

YES

2 Types UAT

1. Alpha Testing

2. Beta Testing

|  |  |
| --- | --- |
| Alpha | Beta |
| Controlled environment | Uncontrolled Environment |
| Devs and Testers involvement | No Devs No Tester |
| Client interaction is more | End user interaction is more |
| Bug 🡪 Immediate Fix | Bug 🡪 Next Release / Version |
| ~~Service Based~~ | ~~Product~~ |
| Small amount | Beta testing – large amount data |
| This can be tested by limited no. of users | This is tested by n no. of end users |

…………………………………………………………………………………………………………………………………………..

Unit Testing

Integration Testing

Smoke Testing

Functional

Non functional

Security

Performance

Retest

Regression

End to end testing

Final Regression

UAT 🡪 Alpha or Beta

Final Regression

**Production issue / Live Issue / Hotfix**

Production bug 🡪 Immediately fix

The fix which we provide to client (code modification) 🡪 Hotfix

1. Production 🡪 Bug 🡪 Mail 🡪 Tester will confirm the bug on SIT / He will try to recreate on SIT 🡪 VALID 🡪 Yes there is bug 🡪 Dev notify 🡪 imm. Fix 🡪 Retest 🡪 Deploy on Production.
2. Production 🡪Bug 🡪 Mail 🡪 Tester will test on SIT 🡪 Not Found 🡪 INVALID 🡪 NO bug 🡪 Configurations 🡪 Dev notify 🡪 Configuration / Environment / Deployment 🡪 Issue with Configuration / envi / deploy 🡪 Fix developer 🡪 Fix
3. Missed requirement also causes the bug. 🡪 CHANGE REQUEST will be made by client.
   1. Impact is less 🡪 Can be catered in very less time 🡪 Immediately Done (Current Sprint)
   2. Impact is more 🡪 Taking lot of time to be developed 🡪 Inform client / CR will be considered in next sprint.

……………………………………………………………………………………………………………………………………………….

**Error**: Something wrong in code

**Defect**: Whenever we are testing something and we found out that error in code that is called as defect.

**Bug**: We will send this defect to dev, and if he accepts it then it is called as Bug.

**Issue**: Critical bug, very difficult to resolve / fix 🡪 Issue / Dev is not able to fix.

……………………………………………………………………………………………………………………………………………….

**Failure:** Not showingas expected behaviour

……………………………………………………………………………………………………………………………………………….

**DRE – Defect Removal Efficiency**

Output / Input = Efficiency = 0-1

(O/I)100= % Efficiency = 0 to 100

How effectively we have done the testing?

0 to 1

Meghana 🡪 SIT 🡪 100 Bug

~~10 Bugs Cancel (Invalid)~~

90 Bugs 🡪 Bugs found out in SIT = A

Amol 🡪 UAT 🡪 10 bugs Valid bugs = B

DRE = Bugs found by Tester on SIT / Total bugs (SIT bugs + UAT Bugs)

= A / (A+B)

90/ (90+10)

90/100

0.9

% eff = 90%

0.8 – 1 = Good

0.5 – 0.8 = Avg

Below 0.5 = Below avg

………………………………………………………………………………………………………………………………………………

**Testing Terminologies**

**1. Monkey Testing**

* 200 test cases execute before release
* 12 hrs
* 30 Test cases
* **Randomly**
  + Core Func./ Critical
  + High Priority Test cases
* Bug 🡪 Immediately 🡪 Dev info 🡪 Fix.
* 30 + defects 🡪 Extra Work 🡪 Over Time or Sat Sun
* **We will carry forward in next sprint**
* **Carried over bugs/ Spill over bugs / Spillage.**

**2. Exploratory Testing**

* We don’t have knowledge of the application but we do have test cases and test data.
* Colleague 🡪 Absent 🡪 Paytm Payments bank 🡪 Test Cases --- Testing
* New Join – HealthCare 🡪 Test Cases
* We will get to know about the functionality

**3. Ad-Hoc Testing**

* We do have knowledge of application / module / req. but we don’t have sufficient test cases and test data.
* Previous Knowledge 🡪 Testing
* Amazon 🡪 Flipkart 🡪 Switch
* And if time permits then we can write the test cases as well using the knowledge which we possess.
* Info 🡪 Shortfall

………………………………………………………………………………………………………………………………………………………….

**Priority and Severity**

To define the Impact of defect

**Priority: Impact of client Business**

Earliest fixing of bug

High Priority

Impacting Business Logic/ client business

How soon I want to fix this defect?

Urgency

Most Important

Preference

Highest/ High / Medium / Low /Lowest

1/2/3/4/5

**Severity**: Impact on application/ Software

Tech

Critical /High / Medium / Low

…………………………………………………………………….

1. High Priority and High Severity

Core functionality is not working

Phonepay 🡪 Scanning / UPI Money Transfer

FB 🡪 Not able to log in 🡪 C P/ C U

2. High Priority and Low Severity

LOGO not visible on homepage

TATA 🡪 Trust / Brand / Quality / CSR

PNG/JPG 🡪 Image

3. Low priority and high severity

Rare used functionalities not work

Invoice downloading on e- commerce

Promo Code not working

Terms and Condition / Privacy Policy Hyperlink Not working

4. Low priority and low severity

Spelling Mistakes

Colour Change

Black – 220

Black 210

…………………………………………………………………………………………………………………………………………………….

HW 🡪 Examples

…………………………………………………………………………………………………………………………………………………………..

**Bucket Testing**

A/B

Split Testing

Chrome 🡪 V23.07

Chrome 🡪 V23.06

Whatsapp EMOJIS / Msgs EMOJIS react V22.1

Whatsapp Payments V22.2

Which performs better

In which we do the testing on 2 different versions of a application/software to check which one works/ performs better.

……………………………………………………………………………………………………………………

**Bug Slippage**

**Latent Defect**

**Bug Leakage**

…………………………………………………………………………………………………………………….

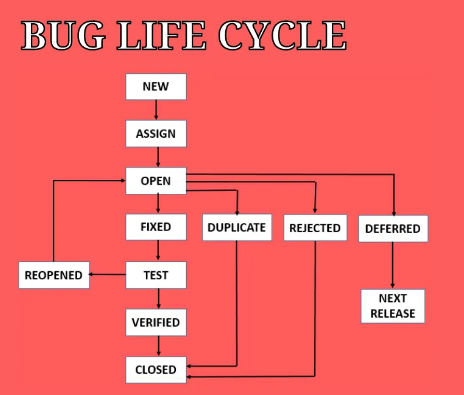
**Bug Life Cycle / Defect Life Cycle**

Life cycle 🡪 Concept to delivery

Start : Find 🡪 Raise 🡪 Defect Management Tool 🡪 JIRA

End : Retest 🡪 Verify 🡪 Regression 🡪 Happy 🡪 Closed/ Done

* **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
* **Assigned:** Once the bug is posted by the tester and assigns the bug to the developer team
* **Open**: The developer starts analyzing and works on the defect fix
* **Fixed**: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
* **Test/Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* **Verified**: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
* **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.
* **Closed**: If the bug is no longer exists then tester assigns the status “Closed.”
* **Duplicate**: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* **Deferred**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs



…………………………………………………………………………………………………………………………………………….

**Triage Call**

Tri 🡪 3 Entities

Defect Mgt Team (PO+ Scrum Master+ Dev lead + Test Lead) 🡪 1

Dev Team 🡪 2

Test Team 🡪 3

Whole team

Acceptance or rejection of bug

Validity of Bug

No 🡪 Rejected 🡪 Close

Yes 🡪 When to fix ?

Current sprint

Deferred.

………………………………………………………………………………………………………………………………………..

**7 Principles of Testing**

Principle

SDLC 🡪 Req Gath/ Analysis / Design / Dev / test / Main.

**1. Early Testing**

Testing 🡪 Deploy SIT

As a tester 🡪 Efforts 🡪 Req Gat.

Testing Activities 🡪 Req Gath

Good understanding and knowledge about functionality

Familiar 🡪 Spot Bug

Guess Bugs

Mob No Text Box 🡪 11 / 9 / Alphabets

Doubts 🡪 Clarify

Test 🡪 Cut down the time for testing 🡪 Fine Testing

**2. Testing shows the presence of defect / bug**

Testing 🡪 SIT

UAT

Testing 🡪 Bug ? 🡪

**3. Absence of error 🡪 Fallacy**

NASA / Tesla Elon Musk / RSA / ISRO

100%

99.999999999%

1000 Functionalities 🡪 1000 Scenarios

**4. Exhaustive testing is not possible**

Bank 🡪 Deposite 🡪 text box 1 rs 1000 CR

**5. Pesticide Paradox**

6.

80% output is depended upon 20 % input

1

2

3

4

5

6

7

8

9

10

80-20

……………………………………………………………………………………….

100 Modules 🡪 20 Modules 🡪 80% Bugs responsible

Identify Modules 🡪 Test Cases/ Testing 🡺 TLC Tender Love Care

………………………………………………………………………………………………………………………………………..

**STLC – Software Testing Life Cycle**

STLC is part of SDLC

* Req Gathering
* Analysis
* Design
* Dev
* Testing 🡪 STLC Detail
* Maintenance

Stages of STLC :

* Requirement Gathering (Passive)
* Analysis
* Test Plan Preparation
  + **Test Lead**
  + All testing related activities 🡪 Document
    - How many tester required ?
    - Which resource on which module ?
    - Which tools ? 🡪 Defect Raise – JIRA/ Bugzilla etc / Excel/ Google sheet/ Test management
    - How many days tester will take to complete the testing ?
    - Automation or Manual ?
    - Manual 🡪 Performance / Fun / Non-Fun / Security etc
* US01 🡪 Bhupesh—dev (24h) / Nikita –Test
* Study and analysis of that requirement / User Story – Tester – Testing Perspective
* T C D
  + Identification of scenarios
  + Write The test cases ( TS,TCID,TC,Steps etc)
  + Review
    - Self-Review
    - Peer review 🡪 Team member
    - Senior
  + Comments on test case
  + Fixing the comments received in review
  + Approval 🡪 Senior QA / Test Lead
* Requirement Traceability Matrix 🡪 Forward T M
* Test Cases Execution (Actual Testing)
* Documentation / Reports 🡪 As and when needed.

…………………………………………………………………………………………………………………………………………

Test Lead 🡪 Team 🡪 Tester Manage 🡪 Work Manage Allocation / Review / Blocker / Self Tasks

…………………………………………………………………………………………………………………………………………

1. Test Summary Report

Test Case Execution Summary Report

No. of test cases executed

**Details of Test Cases + Execution Status** 🡪 Actual Testing Result

Test Case execution status 🡪 Pass / Fail / Blocked /Incomplete / Work In Progress / Skip

|  |  |
| --- | --- |
| Pass | Expected = Actual |
| Fail | Expected is not equals actual |
| Blocked | You are not able to test it |
| Incomplete | Feature Incomplete / Incomplete Test Case |
| WIP |  |

……………………………………………………………………….

Project Name

Module Name

Test summary report created for Sprint 11

Test summary report created for Payment Getway

Epic

User Story –

Prepared by

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Test Case ID** | **Test Case Name** | **Test Case prepared by** | **Test Case reviewed by** | **Test Case executed by** | **Execution date and Time** | **Status** | **Defect ID** | **Remarks/ Comment** |
| 1. | VCT01 | Verification of payment getway | Varun | Rohit | Bhagyashree | 28/06/23 09:00 am | Pass | - |  |
| 2. | VCT02 | Verification form fun. | Pallavi | Komal | Pallavi | 28/06/23 09:00 am | Fail | VCT784 | TC is failing while doing final submission of the form. |

………………………………………………………………………………………………………………………………………………………….

2. Defect Summary Report

No. of defect

Who raised

Status

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No | Defect Id | Defect Title | Defect Description | Raised by | Status | Assigned To | Linked Test Case |
| 1 | VCT567 | Mute button not working | * + - * Details | Mahesh | In progress | Nilam | VCT 1 |
|  |  |  |  |  | To Do | -- | VCt2 |
|  |  |  |  |  | Deferred | -- | VCT3 |
|  |  |  |  |  | Done | Amit | VCT4 |
|  |  |  |  |  | Invalid | Rohit | VCT5 |
|  |  |  |  |  |  |  |  |

………………………………………………………………………………………………………………………………………………………

Traceability Matrix

Requirement Traceability Matrix

RTM

* We will be having track on all the requirements whether they are being tested or not.
* Each and every requirement should get tested.

**A. Forward Traceability Matrix**

* + Before execution test cases.
  + Each and every requirement should get mapped with atleast 1 test case.

REQ01 – Signup page

REQ02 – Login Page

REQ03 – Profile Page

REQ04 – Settings Page

TC01 – Veification Signup page

TC02 – Verification Profile

TC03 – Verification of Settings page

TC04 – Login page

TC05 – end to end

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ReqID | Req Desc | TC01 | TC02 | TC03 | TC04 | TC05 |
| REQ1 | Signup page | ✓ |  |  |  | ✓ |
| REQ2 | Login Page |  |  |  | ✓ | ✓ |
| REQ3 | Profile |  | ✓ |  |  | ✓ |
| REQ4 | Settings |  |  | ✓ |  | ✓ |

B. Backward Traceability Matrix

* After execution of test execution
* Executed test cases are mapped with requirements.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name/ Desc | REQ1 | REQ2 | REQ3 | REQ4 | Status | DefectID |
| TC01 | Verification of signup | ✓ |  |  |  | Pass |  |
| TC02 | Profile |  |  | ✓ |  | Pass |  |
| TC03 | Setting |  |  |  | ✓ | Fail | VCT01 |
| TC04 | Login |  | ✓ |  |  | Pass |  |
| TC05 | End | ✓ | ✓ | ✓ | ✓ | Fail | VCT01 |

C. Bidirectional (Forward + Backword)

………………………………………………………………………………………………………………………………………………………………..

Test Documentation:

**1. Test Policy**

* Company Level
* Director/ Head / CTO / Testing Head
* Defines the objective of Project ( Moto/ Purpose/ Benefit / Outcome / Expectation from this project)
  + Reliable software prepare
  + Client satisfaction
  + Revenue

**2. Test Strategy**

* Strategy / Approach to full fill the objectives of the project
* Project 🡪 Manually and Automation
* Manual 🡪 Fun / Non Fun / Security / Performance
* Automation 🡪 Tool / Lang.

3. Test Methodology

Types of Testing

TRM 🡪 Test Responsibility Matrix 🡪 What test factors we will be using in our project 🡪 Types of testing at different stages of SDLC

RG AFC D Dev Test Main

Development stages are being mapped with types of testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ~~RG&AFC~~ | ~~Design~~ | Dev | Testing | Maintenance |
| Functional | ~~NO~~ | ~~NO~~ | Yes | Yes | Yes |
| Non Functional | ~~NO~~ | ~~NO~~ | Yes | Yes | Yes |
| Security | ~~NO~~ | ~~NO~~ | Yes | Yes | Yes |
| Performance | ~~NO~~ | ~~NO~~ | NO | Yes | Yes |

**4. Test Plan**

**Test Lead**

a. Job Allocation (What to test, What not to test, When to test)

b. Resource Allocation (Who will test)

c. Estimation (How much time will be required to test)

a. Test Plan ID (PTM 01 – Test Plan)(

b. Testing Modules 🡪 Which modules we have to test?

c. Features to be tested 🡪 Features / Functionalities to be tested?

d. Features not be tested

e. Pass and Fail Criteria

Pass 🡪 Expected = Actual

Fail 🡪 Expected is not equals actual

f. Entry and Exit Criteria

Entry and Exit Criteria 🡪 User Story

ENTRY

* Unit Testing
* Integration Testing
* Smoke Testing

EXIT

* Expected is matching with actual result
* Acceptance Criteria

g. Environments

* DEV/ SIT / UAT /Prod /Live
* IP Address / URL

h. Road blockers

Possible challenges / abnormal situation that can occur

i. Test Deliverables

Element of Output

Policeman 🡪 Law and Order maintain

Doctor 🡪 Treat to patients

Teacher 🡪

Students 🡪

Test Lead 🡪 Test plan preparation / Lead the team / Blocker / Voice

Manual Tester 🡪

Automation

Senior Tester 🡪 Review The Test Cases

j. Roles and Responsibilities

Varun 🡪 Review / Performance

Pallavi 🡪 Payment Module

Komal 🡪 Automation

Meghana 🡪 DB testing

Pranali 🡪 XYZ

k. Trainings

Manual

Automation

API

Database

1. Mandatory 🡪 Data privacy and data security / POSH

2. Required for project 🡪 API testing BHUSHAN API TESTING

l. Risks

Attrition 🡪 Company Leaving

Cost Estimation

Project Deviation

Delivery Dates

m. Approvals

PM / TH / Director

……………………

1. Stop Learning

2. Financial Growth Stop

3. Toxic Work culture