|  |  |
| --- | --- |
| **Manual ( 8 to 9)** | **Automation ( 9 to 10)** |
| Manual Part 1 | JAVA |
| Manual Part 2 | OOPS |
| SQL (Data Base ) | Selenium |
| API ( Web base Testing) | TestNG |
| Project I ( LIVE ) |  |
| Project II ( LIVE) |  |
| JIRA ( Project MNG Tool) – two types |  |
| Nuakri Profile |  |
| Interview Preparation |  |

1. Lecture – 8 to 10
2. Daily Mock – 2 Hours 🡪 We are preparing group of Student 🡪
3. Weekly Mock – 4 hours 🡪 12 No.

**Manual Content :**

**Part 1 – Understand culture / process / team position/ Nomenclature /**

1. **Methodology –**
2. SDLC Process
3. Waterfall Process
4. V Module / Process
5. **Agile Process**
6. **Different Testing –**
7. Sanity / Smoke Testing
8. BBT ( System & Function Testing)
9. Retesting
10. Regression Testing

**Part 2 – Actual Work of Tester**

1. Write down Test Cases
2. Execute Test cases
3. Prepared Report
4. Defect Info

**SQL / Database Testing –** How to fetch Database using Command / Queries

**API / Backend Testing** - SOAP & Rest service

**Live Project**

**Manual Testing :**

**Def.-** Which is performed **by manual** to check functions of application

**Team Size:**

**In IT Company – two Types of Team – Project & Support Team**

**Team Size –( 14 to 17 ) –** Implement new feature / new module / new functionality in Existing Application

Paytm ( Owner ) 🡪 Delivery Mng ( TCS) 🡪 Project Mng. (TCS) 🡪 Team Form ( Project Team) 🡪Design (Designer ) 🡪 Requirement (small Part ) 🡪 Developer work on Small Requirement (Story)

Project Handover ( Delivery Mng).

**Project Team: ( Developer : tester = 4:1)**

1. **Delivery Mng. (1) – He / She Work with Client .+ Time Control**
2. **Project Mng. ( 1) – Project Development + Resource Requirement + Testing Work Properly + Assign Work**
3. **Business Analyst (BA) (1) – Monitor Work + work on Blocker +**
4. **Designer (1) – Design Preparation (Front End + Backend Design)**
5. **Developer (8 to 10 ) – Coding**
6. **Tester (2 to 3 ) – Test ( Feature/ Working + Defect Raised+ Issue ) like end User**

**Support Team (5 to 8) –** Working on Existing Application/Module& work on Ticket which is raised by users

1. **Project Mng / Customer care Mng. (1) . –** Analyses Ticket + Assign to Developer + Track ticket
2. **Developer (3 to 4 ) –** Working on Existing application issue/ticket which raised by users & ticket resolve / close
3. **Tester (1 to 3) –** Working on Existing application + testing of resolve ticket + Report send it to PM

Client Project (PayTM) 🡪 Wipro 🡪 Project & Support team

YES Bank project 🡪 TCS 🡪 Project & Support team

In my Project 🡪 both team are working ( Project & Support team)

**I have worked with Project team**

**Process - Requirement 🡪Development 🡪 Testing 🡪 Delivery**

1. SDLC ( Software Development life Cycle)
2. Water Fall Process
3. V Module
4. **Agile Process**

Which process should be follow for project , It will decided by client (IT Team) –

Client (PayTM) 🡪 Wipro ( Project Team ) 🡪 Persistence ( Support Team )

**SQA ( Software Quality Assurance):**

Client

BA (Mng)

Developer

Tester

SQA Process Focus:

1. Customer Requirement Fulfil – (Developer + Tester +BA)
2. Customer Execution – ( Security + Performance) – Done by Tester
3. Cost / Budgets of Project
4. Delivery / Deployment of module to Client
5. Risk involved in Project – ( Is it possible + Any alternate ( Mitigation) solution + Completion Time)

**SDLC ( Software Development Life Cycle ) It is Standard Process–**

1. Information Gathering
2. Analysis
3. Design
4. Coding / Developer
5. Testing
6. Support/ Maintenance

**Information Gathering –**

1. Responsibility - BA
2. BA will collect all Information related to module from Client
3. From that info. He/ she Prepare BRS ( Business Requirement Specification ) Document.
4. BRS Define business related Info. Against application / Software
5. EX. Recharge Module ( Wipro) 🡪 From Client ( Paytm) 🡪 from that info. Prepared BRS
6. BRS document is only limited to BA & Mng.

**Analysis –**

1. Responsibility - BA
2. BA will Prepare SRS( Software requirement Specification) Documents from BRS
3. SRS is also called FRS ( Functional requirement Specification ) / CRS (Customer requirement Specification)
4. SRS Define **Functional related Requirement** of Software (Its Technical Requirement)
5. **SRS** Document Send to All developer & Testers & Design in the form of Story
6. **SRS Document Contain**

**Functional Requirement** – ( Total requirements with

Function )

**Functional Flow Diagram** – End to End Flow ( Ex. Prepared / Post-paid 🡪Mobile No. 🡪 Operator 🡪 Circle 🡪 Amount 🡪 Proceed Button)

**Use Case 1**. **Description** - Detail Description of

Requirement

Ex. For Mobile No. – It Accept valid mobile no. of 10 digit.

Initially field disable but after click in web element its

allow to enter mobile no.

Mainly for Developer

2. **Acceptance Criteria** – Does & Don’t about

Requirement

Ex. Mobile No. – It accept only no. which is of 10 digit

Delete pop up button only enable after enter mobile no.

Mainly for Tester

**Screen Shot / Snap Shot/ Block Diagram** – Image /

Structure / Visualisation of Module with no Function.

1. If from SRS dev. & Tester not understand the exact requirement that in such case **arrange meeting** with BA

**Design –**

1. From SRS Document he/ She prepare HLD (High Level Design ) & LLD ( Low Level Design)

Ex. Product purchase in Amazon

(Select Product 🡪 Add to Card 🡪 Bay Now 🡪 Address Selection 🡪 Payment)

HLD – Flow / Navigation Design

LLD – In Each Navigation give additional Function / Feature

**Coding / Develop –**

1. In this stage Developer will work
2. Developer prepare code for LLD (Low Level Design)
3. LLD contain small functions, Suggestion , Error, etc.

**Testing –**

1. When Development completed then will start testing, Developer sent module / Build to Tester
2. When tester got the build then they start with TCD ( Test case Design) & TCE ( Test case Execution)
3. While execution if we found any bug / defect /error – assign that bug to developer
4. Also tester doing different types of testing .

**Support / Maintenance –**

1. After Completion of Testing 🡪 module sent to client
2. After completion warranty Period ( Support from Project team ) 🡪 sent to Maintenance team
3. Support Team working on Ticket / Indi dual user queries/ issue of existing application

**Waterfall Module-**

Information Gathering

**Analysis**

**Design**

Coding ( Build / Module )

Testing ( TCD / TCE)

Support

1. **Def.** Sequence of Stage which start with Information Gathering to the Support
2. **Sequence Modul –** After Completion of First stage then second stage will start working , Completion Second Stage then third Stage will start working
3. **In Waterfall process Time Duration already Decide**

Disadvantages :

1. Error / bug/ issue / wrong Design no option to reverse back ( Backtrack)
2. Time Consuming (More then 3 month )
3. Repeat Work consuming more time

**V Module – (Verification & Validation)**

1. **Why V – Verification & Validation**
2. **Parallel Development & Testing**
3. **Time Duration to deliver project / Module to client – 3 Months**
4. **V Module is Also Called as Plan Driven Process.**

**LCD (Life Cycle Development) LCT ( L C testing)**

Info. Gathering (BRS) Prepare Test Plan

Analysis ( SRS/ FRS/ CRS) Requirement of Testing

Assignment of Dev Plan

Design ( HLD , LLD) Design Phase Testing

Coding – Dev. LLD ( Set Up) Program Phase Testing (WBT)

Test Case Design (+ve & -ve)

BBT ( System & Function Testing)

Install Build (Merging of Requirement) UAT ( User Acceptance Testing)

KT ( Knowledge Transfer)

DRE ( Defect Removal Efficiency)

Maintenance / Support CR ( Change Request )/

Post Morten Testing

1. **CR / RFC** – Change Request / Request for Change – Come from Client

Ex. Thank you massage - Thank you to use PayTM

According to Change Request 🡪 we will take extra charge for Development ( V Module)

1. **DRE** ( Defect Removal Efficiency) –

Def – How Much test application thoroughly we have test it.

**DRE =** (A / (A + B )) = 50 / (50+5) = 0.9 ( Ideal Value of DRE = 1)

Good DRE – 0.81 to 1

Satisfactory DRE – 0.61 to 0.8

Poor DRE – less than 0.6

A – No of Defect found in BBT / QA

B – No of defect found in UAT

1. **Post Morten Testing –** Done before delivery build / Module to client .

We are testing complete application with variety of Input / Varity of Flow / Verity of negative Scenario.

**Dev. 🡪 QA / SIT ( 50 ) 🡪 UAT ( Live User ) (5) 🡪 Production**

**Disadvantages of V Module –**

1. Time Limit of 3 Month
2. CR – Extra Charges
3. If requirement not clear from Client
4. High Initial Investment
5. Difficult to manage Document/ report

**Agile Module**

1. **Introduction**

* Def. – Continuous process of development & testing
* At any point of time CR come from client, We will accept that CR and not taking any extra charges for that

1. If CR affecting on Current Function 🡪 CR develop n next Sprint
2. If CR not affecting on current function 🡪 we will modify in same Sprint

* Agile is also called Value Driven Process ( Client Priority )
* Agile Delivery / Deployment duration 🡪 2 week / 3 Week
* Different types of Agile process

1. XP ( extreme Program) – Only Coding no Testing
2. Lean – For support team- ( Bunch of Ticket (10 No. / 10 Bugs)🡪
3. Kanban – For Support / SRE ( Handle Deployment)
4. **Scrum – Sprint ( Bunch of Requirement) wise delivery 🡪 2 /3 weeks**

**Ex. 100 Requirement (20 Bunch 🡪 first Sprint) 🡪 ( 21 to 40 Bunch 🡪 second sprint ) 🡪**

1. FDD – Future Driver Development – Assumption

**In my Project we follow Scrum Agile Methodology**

1. **Architecture of Agile**

|  |  |
| --- | --- |
| **SDLC** | **Agile** |
| Information Gathering (BRS ) | Product Backlog |
| Analysis ( SRS ) | Sprint Backlog – 2 week delivery |
| Use Cases ( Description & Acceptance) | User Story (US) - ( Description & Acceptance) |
| Designer – HLD , LLD | Designer – HLD , LLD |
| Coding ( LLD) | Coding ( LLD ) Against US |
| Testing ( TCD & TCE) | Testing ( TCD & TCE) Against US |
| Delivery Mng. | Solution Master |
| Business Analyst | Product owner |
| Project mng. | Scrum Master |
|  |  |

1. **Ceremonies of Agile / Meeting of Agile in each Sprint ( 2 Week)**

* **Grooming Meeting**
* **Sprint Planning Meeting**
* **Scum / Daily Stand Up Meeting**
* **Sprint Review Meeting**
* **Sprint Retrospective Meeting**

|  |  |  |
| --- | --- | --- |
| **Meeting** | **Purpose** | **Involve** |
| **Grooming Meeting**  Before Start of Sprint / 1st day Sprint | Requirements 🡪 Understanding 🡪 Doubt Clearing | Organiser - **Product Owner**  30 Min. / 1 Hrs.  Designer , developer , Tester, Scum master ( Optional) |
| **Sprint Planning**  1st Day / First meeting of Sprint | * How many User Story add in Sprint , that will decided   Ex. Sprint A – 18 US   * Estimation 🡪 Taken from Developer & Tester   Story Point – (1,2,3,5,8,13 ) | Organiser – **Scrum master –**  Invitation – PO, Developer , Tester , Designer |
| **Scrum call / Daily Stand up**  (Daily at the start – 10 :00 – 10:30 ) | * What you have done yesterday * What are you doing Today * Any road Block / Appendment / Issue | Organiser - **Scrum Master** , Designer , Dev. Ester , PO |
| **Sprint Review Meeting**  Last Day of Sprint  1 hrs | Demo of Work / Story by Tester | Organiser – PO, Designer , Dev. Teaster,  Scrum Master ( Optional) |
| **Sprint Retrospective meeting**  Last day of Sprint  -Prepare Document | * What are Good things * What are improvement | Organiser – Scrum master Designer , Dev. Teaster,  PO |

1. **Agile Daily Wise Plan**

* Agile Sprint Size - 2 Week (5 \* 2 = 10)
* Project Team –( Developer : tester = 10 : 4) = 14 + 1+1+1+1= 18
* Team 1 – working on Recharge Module ( 3 : 1)
* Team 2 – Electricity Module (3:1)
* **Team 3 – Invest stock (4 :2 Me)**
* Sprint 1= 4 US 🡪 2 US getting to each tester 🡪 3 Story Point + 3 story Point🡪 Analysis🡪 TCD ( + ve & -ve Scinario) 🡪 Testing ( TCE) 🡪 BUG 🡪fix by Dev. 🡪 Review with PO 🡪 Retrospective

**Sprint Start on Monday**

1. Planning Meeting ( 30 Min) ( 9:30 to 10:00) – Leaves + Estimation of US
2. First US 🡪 Analysis ( Understanding )

**Tuesday**

1. Daily Stand-up ( 9:30 to 10:00) - What you have done yesterday

* What are you doing Today

Any road Block / Appendment / Issue

1. Planning to do TCD for First US (7 to 8 hrs)

Wednesdays

1. Daily Stand-up ( 9:30 to 10:00)
2. TCD complete for First US + Analysis of Second Story (7 to 8 hrs)

Thursday

1. Daily Stand-up ( 9:30 to 10:00)
2. Completed Analysis of Second US + start working on TCD of Second Story (7 to 8 hrs)

Friday

1. Daily Stand-up ( 9:30 to 10:00)
2. Completed TCD of Second Story

Monday

1. Daily Stand-up ( 9:30 to 10:00)
2. Start Testing ( TCE) & prepare Report 🡪35 TC

Tuesday

1. Daily Stand-up ( 9:30 to 10:00)
2. TCE of first US 🡪 15 TCE 🡪 Found BUG 🡪 Raise same dev.

Wednesday

1. Daily Stand-up ( 9:30 to 10:00) + Road Block
2. Bug Fix 🡪 TCE 🡪 Bug fix or not + TCE for Second US ( 20 TC)

Thursday

1. Daily Stand-up ( 9:30 to 10:00)
2. TCE second US (30 TC) 🡪 BUG found 🡪 Raised to Dev.

Friday

1. Daily Stand-up ( 9:30 to 10:00)
2. Review to PO – (Giving Demo of Our Testing to PO 🡪 If Accept then US move to Completed State

If Not 🡪 PO Suggest some Modification in Testing 🡪 After completion of Suggested Testing Again Review then move done.

**Retrospective Meeting (30 min) –**

Good Things & Improvement area

1. **Agile Imp. Term**

* **Burn Down Chart –** It is Graph which shows, How many US are pending w.r.t. time.
* **Burn up Chart -** It is Graph which shows, How many US are completed w.r.t. time.
* **Velocity Chart –** How much time / Estimation we are giving , how much time required to complete sprint.

**Ex.** Sprint 1 contain 20 story (3+5+3+8+..) complete (PO 🡪 60 Sp)🡪 Actually ( Indi dual 🡪80 SP)

* **Epic - Main US of Module**

**EX . - Recharge or Pay Mobile Bill 🡪 Radio Button, Mobile no. Field , operator, Area Selection , Amount**

* **Estimation –** Time Span Developer & Tester to complete 1 US
* Estimation done in the form Story Point ( 1,2,3,5,8,13)
* PO Roughly Estimate Story point / Actual Story point given by Dev . Tester
* In Sprint Planning we are Estimating the Story point for US.

1. How much Knowledge we have against US
2. How much Complex US
3. Effort Required
4. **Dependency**

**Planning SM 🡪 Story Point 🡪 Daily**

**Advantages :**

* CR Accept at any stage of Development and no extra charges required.
* In Agile, Delivery / Deployment duration to client is of 2 /3 weeks and its deploy in the form of small story/build.
* All ceremony are help to start & track the development (Developer & Tester)
* In Depth Testing perform in Agile
* Automation possible in agile
* Check points are present in every sub module

1. **Disadvantages**

* **If Requirements are not Clear**
* **If development block , Development stop**
* **New Team Member , Knowledge sharing required & Difficult**
* **Poor Resources Planning.**
* **Budget is more compare to other process**

Thank you Massage

Payment Tab

Promo Code

Browser Plan

Mobile no, Circle etc

Recharge Icon

**1 US 3 US 3 US 2 US 5 US 1 US**

**CR ( Browser Plan)**

**Budget – 80000, 6 month – 10000 /month**

**CR – 7 month 🡪 70000**

**Question**

1. **What is your team Size / How many Del. & Tester working with you?**
2. **What is SDLC ?**
3. **What is SRS Document? / What SRS Document Contain? / How you get SRS /**
4. **What is V module**
5. **What is CR , How you deal with CR**
6. **What is Agile?**
7. **What are Agile Ceremony**
8. **What is SQA?**
9. **What is WBT & BBT / Difference between WBT & BBT?**
10. **What is Epic**
11. **What is Kanban , lean , Scrum**
12. **What is Estimation**
13. **How many US cover in one Sprint by you.**

Fish Methodology

Info. Design

Gathering Analysis (HLD, LLD) Coding Testing Support

Review (BA) Review (BA) Review (Designer) Review (WBT) BBT

Static Testing / Verification / Quality Control Dynamic Testing / Validation

/ Quality Assurance

|  |  |
| --- | --- |
| **WBT ( White Box Testing)** | **BBT ( Black Box Testing )** |
| WBT is perform by Developers | BBT is perform by Tester |
| Types :   1. Unit Testing 2. Integration Testing | Types:   1. Saniy & Smoke 2. Functional 3. Non Functional 4. Regression 5. Retesting |
| WBT 🡪 Check Function/ Check code / check loop statements / check condition | BBT 🡪 coverage of Function / All functionality of module / input domain / Cross browser / error handling |
| WBT also called Code base testing | BBT also called System & Functional Testing |

|  |  |
| --- | --- |
| **Static Testing / Verification / Quality Control** | **Dynamic Testing / Validation/ Quality Assurance** |
| In Static Testing 🡪BA, Designer , Developer | In Dynamic Testing 🡪 only Tester |
| Also Called Verification because they are verifying own work | Also called Validation because review / testing is perform on developer |
| It also called **In Process testing** | It is also called **End Process Testing** |
| In Static testing will control Quality of Module /Application | In Dynamic Testing we will assured the Quality of Software / Module |

**Project Technology**

**Different Technology:**

1. Front End 🡪 Dot Net Programming / HTML
2. API / Service End 🡪 Java Programming
3. Data Base 🡪 SQL Server

Request

API / Service End

(SOAP / REST)

(Manual)

Front End

(Manual & Automation)

(Manual

Data Base

(Manual )

Response

Store SQL

Project Environment :

* In my Project , 4 Types Environment are present
* Dev Env. , SIT Env., UAT Env. , Production / Live Env.
* In Dev Env. 🡪 Developer are Working in this Env. 🡪 WBT , Unit Testing
* Dev Env. 🡪 URL - 192.168.1.12: 8080/paytm.com
* SIT Env. (System Integration Testing Env. ) 🡪Testers are working
* SIT Env. Testing 🡪 BBT , Sanity / Smoke , System & Function , Regression
* SIT Env. 🡪 URL – 192.168.2.12:8080/paytm.com 🡪 Test data 🡪 Sample
* In SIT Env. 🡪 TCD & TCE
* After completion of Development 🡪 Developer send Email to tester ( CC – SM, PO ) +Jira Comment & attach artifact in same story , Bug Raise.
* UAT Env ( User Acceptance testing Env. ) 🡪 Test Data – Live 🡪 Client Side
* UAT Env. 🡪 URL – 192.168.3.12:8080/Paytm.com
* In UAT 🡪 Alpha Testing & Beta Testing
* After completion of SIT Testing 🡪 Jira Comment & attach artifact in same story , Bug Raise 🡪 PO Review 🡪 US deploy in UAT.
* Production Env. / Live Env. 🡪 Hot Fix ( High Priority Issue)

SIT

Dev

UAT / Pre Live

Prod / Live

Pune (Wipro-Project Team) Client (USA)

**Dev Env. :**

* In this Env. Developer are working
* Developer Prepare code on LLD + Testing
* Testing – WBT ( White Box Testing )
* WBT –1. Unit Testing 2. Integration Testing

**Unit Testing – (User Story level Testing)**

* Unit Testing will Perform after completion of Coding
* In Unit testing Developer perform testing on US
* Generally unit testing perform on Sub Module
* Developer Prepare unit testing document ( Function , Function flow / Screen Shot / Table etc )
* Ex . paytm Recharge Module 🡪 US1 ( Mobile No. Field )

**Integration Testing :**

* Integration Testing Perform by Developer
* After completion of Module , developer start Integration testing
* After completion of all Sub Module , developer start Integration testing
* Ex . Recharge module of PayTm 🡪 After complete development Integration testing Perform

Recharge Module

Mobile, Operator, Amount

Plan Selection

Promo Code

Payment Tab

Integration Test Different Approaches:

1. Top Down Approach:

* If we don’t have next Sub Module , Developer Prepare Demo/Temporary Module for Integration Test
* That Demo Module is called **STUB**

1. Bottum Up Approach

* If we don’t have previous module, Developer prepare Demo / Temporary Module for Integration test
* Demo module is called **Driver**

1. Sandwich Approach

* If Top module & bottom Module not prepare by Developer, Developer has In-between Module then Use Sandwich Approach
* In Sandwich Approach **STUB & DRIVER** both will be prepare by Developer

Input & Output are without Program / Hard Coded

Input Output

Recharge (20 US) 🡪US1 dev 🡪 Testing🡪UAT 🡪

🡪US2 dev 🡪 Testing🡪UAT 🡪

🡪US20dev 🡪 Testing🡪UAT 🡪

Recharge ( Integration testing ) 🡪 Regression Testing 🡪UAT 🡪 Prod.

**SIT Env. :**

* In SIT Testers are working
* When Developer complete Unit Testing of US, then they will send that US / Build to SIT Env.
* When we get New US for Testing we will test stability of that Build also check Unit testing Document.
* Developer 🡪Code + Unit Testing 🡪 Deploy in SIT 🡪 Inform in mail / In Comment Jira board/ update in Daily Update
* Developer Build complete 🡪Push Thair code in Github Repository 🡪 Jenkin Pipeline
* Tester Automation 🡪 Push Thair code in Github Repository 🡪 Jenkin Pipeline

1. Sanity Testing:

* When Developer complete Unit Testing of US, then they will send that US / Build to SIT Env. In SIT we will start testing of build with Sanity
* In Sanity Testing 🡪 We are checking stability of Build
* Sanity testing is also called Zero level testing / Build Verification / tester acceptance testing
* In sanity to check stability below point are cover:

1. Validation / checking core function of Build.
2. Validation / checking UI / GUI of Application
3. Validation / checking tab / button / link / page of build
4. Validation / checking Navigation of Build

* While sanity testing 🡪 if we found defect 🡪 tester rejecting build with proof / artifact
* Sanity Testing we are not writing TCD
* Sanity Testing 🡪 approximately (2 to 3 hrs)
* Reason for Defect 🡪 Core functions are not working / navigation issue / System Hang / Not lunching on browser / Links are not working / Run time error
* If Tester Rejecting build 🡪 inform to Dev + PO + SM + TL 🡪 JIRA Board
* Old version (v91.4.1) 🡪 New version (V92.1.0)🡪 Core function / Navigation / link 🡪 Stability

**Smoke Testing:**

* **It is advance version of Sanity**
* **In Smoke testing we are validating stability of Build / Module**
* **For Stability :**
* Validation / checking core function of Build.
* Validation / checking UI / GUI of Application
* Validation / checking tab / button / link / page of build
* Validation / checking Navigation of Build
* **If Build found Unstable , We are rejecting build with Troubleshoot report ( Root Cause of Un-stability)**
* **In Smoke Testing = Sanity Testing + Troubleshoot report**
* **In Smoke Testing = Sanity testing + Package Verification (Developer) 🡪 What is root cause**
* **For Smoke testing , we required 2 to 3 hrs.**
* **In my project we are performing smoke testing when we are getting new build from developer**

**Question:**

* 1. **What are different env. In your Project**
  2. **What is Unit testing & Integration testing**
  3. **How you got your Build 🡪 Developer Deploy build in SIT Env.**
  4. **What is difference between Dev URL & QA URL**

**Dev URL 🡪**  [**https://dev.paytm.com/recharge**](%20https://dev.paytm.com/recharge)

**SIT Url 🡪**  [**https://qa.paytm.com/recharge**](%20https://qa.paytm.com/recharge)

**UAT URL 🡪 https://uat .paytm.com/recharge**

**Prod. 🡪 https://paytm.com/recharge**

* 1. **What is difference between Sanity & Smoke**
  2. **Which defect normally found in Sanity & Smoke**
  3. **What are different technology used in your project ( Front End , Back end (Rest Assured / API testing) , Data base)**

**System & Functional**

* It also called Black Box Testing ( BBT)
* After completion of Sanity / Smoke Testing, we perform System & function testing.
* It will perform once we got a stable build
* Types:

1. **Usability Testing**
2. **Functional Testing –( Functional & Non Functional)**
3. Performance testing – Jmeter tool / Load Runner Tool
4. Security Testing - Jmeter tool

**Usability Testing**

* **In Usability Testing , we are validating User friendless of UI of Application**
* **Type: 1. GUI / UI Testing 2. Manual Support Testing**

1. **GUI / UI Testing**

* We are validating look & feel of Application
* Validating Ease of use
* Validating speed of application

1. **Manual Support Testing**

* We are validating sensitivity of input field support on application

**Functional Testing:**

* We are validating internal & external features of application
* Type: 1. Functional (Internal ) 2. Non Functional ( External)
* Functional – **BIEBSC**

1. **B**ehavioural Coverage testing
2. **I**nput Domain Coverage Testing
3. **E**rror Handling Coverage Testing
4. **B**acken Coverage Testing
5. **S**ervice level Coverage Testing
6. **C**alculation Coverage Testing
7. **B**ehavioural Coverage testing

* Validating **B**ehaviour of the Web Element / Object
* We are validating properties / Function of Web Element

|  |  |
| --- | --- |
| **Web Element** | **Behaviour / Property** |
| Radio Button | On & Off alternate |
| Text Field | Focus & Un-focus |
| Drop Down List | Combination of Element out of only one can select |
| Button | Enable & Disable |
| View Box |  |
| Check Box |  |
| calendar |  |
|  |  |

1. Input Domain Coverage Testing / Techniques of Writing Test Cases

* Validating Input data size/Length , Input Type of Data
  + 1. BVA (Boundary Value Analysis)
    2. ECP ( Equivalent Class Partition)
    3. Decision Table

1. BVA –

Validating input data size / Length of Input Web Element

EX. Paytm 🡪 Recharge module 🡪 Mobile No(10 Digit ), Amount ( 8 Digit)

1. ECP –

Validating data type of Input Web Element / Object

EX . Mobile No(int / Number), Amount ( 8 int / Number), Email ( Int/Special

char/ char)

1. Decision Table

Validating different input combination of data pass in Input Web Elements

EX. Login Module

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Web Elements** | **Rule no. 1** | **Rule no2** | **Rule No.3** |  |  |  |
| Email Id | Valid | Valid | Invalid |  |  |  |
| Password | Valid | Valid | Valid |  |  |  |
| Capcha | Valid | Invalid | Valid |  |  |  |
|  | Pass | Fail | fail |  |  |  |

Ex. Cycle parking Stand with capacity of 100

BVA – Min 🡪 0, max 🡪100

ECP – pass –-> Cycle , Fail Bike , Car , 3 Wheeler etc.

Technical Ex. Mobile No. Field

**BVA** – min 🡪0 , max 🡪 10 digit (TC Happy flow -Valid 10 digit mobile no.,

Negative flow 🡪 Invalid no. -1 digit , 9 digit , Null value

**ECP** – data type – Int. Happy flow

Negative Scenario – (int + char +spec. char ), ( int +char), (char) , (char+spec. char ), (Spec. Char)

Repeat TC – Validate invalid mobile no. with 2 digit

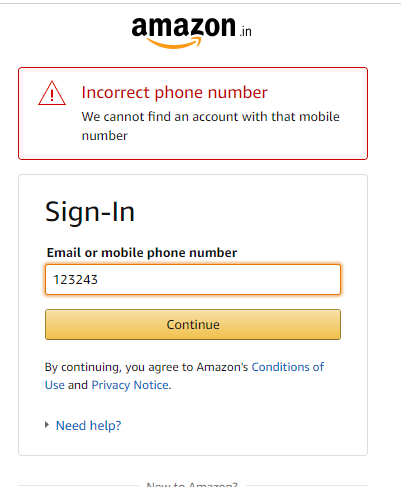
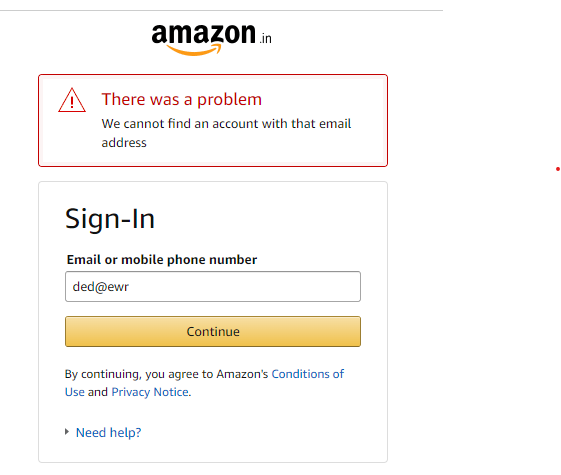
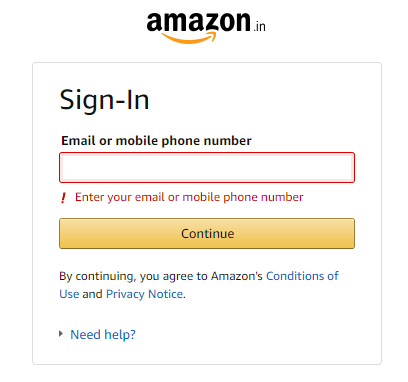
Validate invalid mobile no. with 3 digit

Validate invalid mobile no. with 4 digit

1. **E**rror Handling Coverage Testing-

Validating different error present in application while passing invalid input

Ex. Amazon 🡪Sign in 🡪 mobile/email 🡪 passing invalid input

* Null input – please enter valid email / mobile
* nnfofnrws@jvkfsj – There was problem
* 1234 – Invalide no.
* 
*  

1. **Backend coverage testing**

We are validating Backend / API / Request & Response

Validating users Data ( History) of Application

Ex. Amazon 🡪 ICICI bank, Amazon 🡪 UPI, Paytm 🡪 Airtel

1. **Service Level coverage**

Validating 🡪 Sequence of Operation of build / application

Ex. Amazon purchase 🡪 Select Product 🡪 Add to cart 🡪 Payment 🡪 address 🡪 thanks massage

1. **Calculation base coverage**

Validating **arithmetic operation/calculation**

Ex. Discount coupon (10% ), Promo Code (Offers) , No. of quantity (Multiply with quantity)

**Non Functional Testing:**

* In Non Function , We are validating External Function/Features of Application
* Non Function testing – (RCCIISPG)

1. Recovery coverage testing

* Validating , either application handle abnormal condition.
* Ex. Payment Process of Amazon 🡪Select product🡪Add to cart 🡪 payment selection 🡪address (Abnormal Condition) 🡪 in next login it should start from Address page - Pass

1. Compatibility coverage testing

* Validating , application accept client platform (Environment)
* Type: 1. Forward Compatibility 2. Backword Compatibility
* Forward Compatibility

Application 🡪 Operating System

**OS**

Application **Forward Compatibility** n

* **Backword Compatibility**

**OS 🡪 Application**

**OS**

Application **Forward Compatibility** n

**In backword we are covering Browser Two type of Testing**

* **Cross Browser 🡪Validating Module / Application working in Different Browser**

**Pass 🡪 Working should be same in all browser**

**Ex. Chrome, Mozilla Firefox, Opera , safari, MS, IE etc**

* **Version Control browser 🡪 Validating Application / Module in different version of Same Browser**

**Older versions are specify in Story Ex. Older 5 version**

**Pass 🡪 Application working same in all older version**

1. **Configuration coverage testing**

Validating configuration / Hardware support to application / module

EX. Paytm 🡪 Ticket Book 🡪 Payment 🡪 Download / Print

1. **Installation coverage testing**

Application / Build install in Client Env. ( Prod. / Live)

We are not performing Installation testing.

1. **Intersystem coverage testing**

**Validating , Interaction of our application with other application**

Network Operator

Paytm

Electricity Board

Bank Gateway

1. **Sanitisation coverage testing**

Validating Extra Feature / Function develop by Developer

+91

Ex.

1. **Parallelization coverage testing**

Validating / Comparing Our Application with other same Application

Ex. Amazon & Flip cart , Facebook & Instagram ,

1. **Globalization coverage testing**

Validating application / module support all languages

Standard language – English

Regional Language - Hindi , Marathi , Tamil , Telgu , Bangali

**Re- Testing**

* Re-testing Perform in SIT Env.
* Retesting include in all System & Functional Testing
* It is Define as perform similar testing again and again in application with different test data
* If we found Bug 🡪 to ensure that bug we perform retesting
* Test Data 🡪 will get from Data base or we will prepare our own test data or we will request to PO for Test Data
* If Test data required which is from other Application then we will request to PO.

**Regression Testing**

* It is perform in SIT env.
* If we found Defect 🡪 we will create that defect in Jira & raise it to Dev.--> Developer fix that defect 🡪Modify existing code
* Modification of bug 🡪 will affect on other Function 🡪 then will perform Regress Testing
* In Regression Testing 🡪We are re-executing all TC which is depend on Bug 🡪check bug is resolve or not also check modify build affecting on other function or not
* Ex. Paytm (Assume Browser Plan found Bug ) 🡪 Fix by Dev. 🡪 regression Testing (Adding all dependant test cases for Re-execution)
* Regression Testing Perform

1. If we found Defect 🡪 SIT Env
2. When Build / Module Moving from one Env. To other Env. ( SIT to UAT 🡪Regression / UAT to PROD. 🡪 Final Regression)

* Regression Testing We are prepare regression Suit ( List of TC)

1. Fail Test Case ( browser plan)
2. High Priority TC
3. Dependant TC
4. Extra Function
5. If time permit then will cover other TC as well

* Regression testing Perform 🡪 1 to 2 Hrs manual testing
* Ex (US 🡪 30 TC) 🡪 Regression Suit (5 to 8 TC)

**Interview Question:**

1. **What testing you have perform in your testing**
2. **Which testing you have perform after finding defect**
3. **What is re-testing , How we got data ?**
4. **What is regression testing , Which suit you will use for Regression testing?**
5. **What is System & Function testing?**
6. **What is Techniques to Design Test cases**
7. **What is Decision Table?**
8. **What is Globalization ? How to test it?**
9. **What are no Function testing?**
10. **What is Difference between Sanity & Smoke**

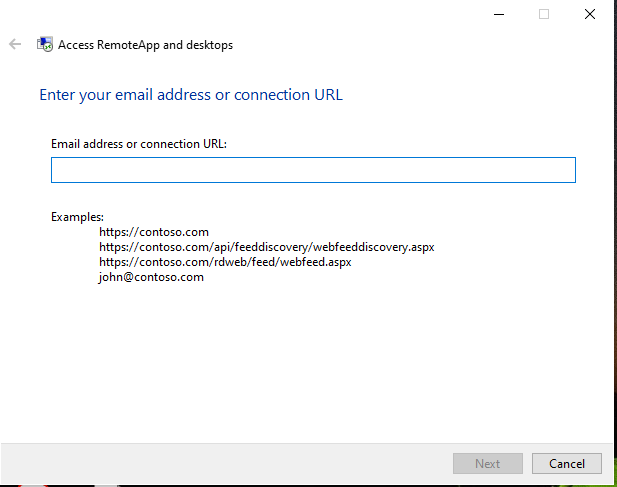
**UAT Env Testing: ( UAT replica of PROD)**

* UAT Team Size 🡪 2 Dev. (Pune ) + 1 Tester ( Client )
* UAT 🡪 System & Function ( Functional & Non Functional)
* After Completion of Above Testing in UAT :

1. Alpha Testing
2. Beta Testing

|  |  |
| --- | --- |
| **Alpha Testing** | **Beta** |
| Alpha Testing perform in Web Base Application / Service base Application  Ex. Paytm, Amazon, IRCTC | Beta Testing Perform in Product base Application  Ex . Adobe Reader , Splitwise , HT etc. |
| In Alpha Developer & Tester both are present | Only Developers are present |
| If found Defect / Issue 🡪 Its resolve immediately | If found defect / Issue 🡪 not resolve immediately 🡪 issue resolve in next version |
| Client interaction is more | Client Interaction less |
| Duration Required for 1 US 🡪 1 to 2 days |  |

* In my project, we have done Alpha Testing
* Tester will not work in both Env. ( SIT & UAT)
* **In my Previous project I have done Alpha Testing.**
* **I have work for 2 to 3 month in UAT ( Because UAT Tester is on leave )**
* **I have used Access Remote Desktop to access UAT Env.**
* **If we found Bug 🡪 Raise it to Dev. 🡪 Fix by Dev. 🡪 Again Validate ( Jira/ Scrum Board)**

****

**Production Issue**

* After Completion of UAT Testing, Then build will Deploy in PROD / Live Env.
* If Bug/ Issue found in PROD. It is called Production Issue
* Reasons for Prod. Issue

1. If any function / Feature Wrongly develop from Our company / missing Bug to test 🡪 Called Hot Fix 🡪 Development Wrong 🡪 Testing Miss in SIT + Miss from UAT 🡪 From Client Raised Hot Fix 🡪Hot Fix are resolve at high Priority 🡪 PO/SM asking reasons for Hot Fix + they send Escalation mail to your Service Base company.
2. From Client , Client forgot to give correct info for particular feature 🡪 Client raised CR (Change Request ) 🡪 CR client should Pay

* CR impacting more on Current Development 🡪 Wiil inform to Client ( Budget/ new Requirement)
* CR impacting less on Current Development 🡪 Will accept CR and Start Modification

Company 🡪 Hot Fix

Client 🡪 CR ( Change Request)

**Scenario Base Testing/ Approaches / Testing Terminology:**

1. **Monkey Testing / Random Testing:**

* When We have more Test Case Execute (30 to 40 TC )& have limited Time ( Half Day ) 🡪 Build migrating in UAT
* In Monkey Testing we are Execiting only High Priority Test Case & Core Function

1. **Ad-hoc Testing**

* When We have less test data but we have good knowledge of application / Build
* In such situation user Story assign to experience tester ( Good knowledge )
* Ex. Payment Tab 🡪 Damy card ( ICICI ) 🡪 Test all function

1. **Exploratory Testing**

* If we have less knowledge about Build 🡪 but we have all test data
* When your team member taking emergency leave
* In such Situation 🡪 We are Exploring Story 🡪 Implement using test data

**Test terms:**

**Error- When developer performing wrong coding / Exceptions in program**

**Bug- In TCE, If tester will found any wrong function/Development called bug**

**Defect- When we raise bug to developer & it accept by developer it called defect**

**Issue – If these defect are more critical / because defect testing stop**

Priority & Severity :

* Priority & Severity is define for Bug / Defect
* Priority 🡪 Define against Client Business Impact
* Priority Level 🡪Very High, High, medium , low
* Severity 🡪 Define impact against functionality of Application
* Severity Level 🡪 Critical, High, Medium , Low

1. High Priority & High Severity-

Ex. – SignIn Button, Payment Tab option ,

1. High priority & Low Severity

Ex. Application Logo, Flag ,

1. Low priority & High Severity

Ex. Menu Button, Filter , Promo Code, offer

1. Low Priority & Low Severity

Ex. Spelling Mistake in Description, Colour of buttons,

Questions:

1. What is UAT Testing
2. Have you got chance to work with UAT
3. What is approach if defect found in UAT
4. What is approach if defect found in Prod.
5. What is Error , Bug ,Defect , & Issue
6. Ex. Priority & Severity
7. Who will deploy Build in SIT Env.
8. What is your approach when User Story not completed in current Sprint
9. What is your approach when defect not completed in current Sprint
10. Who will deploy Build in UAT & PROD Env. 🡪 SRE + Dev. + Tester