# Test Documentation –

**Q. What is your organization Test documentation?**

* Test Document hierarchy

Quality control – QC/

Testing Head -TH

Company Level document

**Test Scenario/ Case**

**Test Plane**

**Test Methodology**

**Test Strategy**

**Test Policy**

Test Strategist – TS & PM

Project manager – PM

TRM

Team Lead – TL

Team Lead – TL

Tester – Tester

Project Level document

**Test Procedure/Design**

**Test Script/ Execution (Test Proof)**

**Defect Report**

**Test Summary Report**

**Final Report/ Test closer Report**

What is Test Policy? What does it contain?

FacebookLinkedInTwitterEmail

Test Policy is one of the key documents that exists in an organization. These are some of the broad categories of documents that exist in every testing project.

* **Test Policy** – It explains the **goals that the organization** wishes to achieve through testing activities.
* [**Test Strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/) – This document details the **general testing methods** used by the organization. These test methods are independent of any project.
* [**Master Test Plan**](http://tryqa.com/what-are-master-test-plans-level-test-plan-examples-when-to-use/#what_is_master_test_plan) – Also called the **project test plan**, it explains project specific testing strategy and test implementation. The **master test plan** is a document that describes in detail how the testing is being planned and how it will be managed across different [**test levels**](http://tryqa.com/what-are-software-testing-levels/).
* [**Level Test Plan**](http://tryqa.com/what-are-master-test-plans-level-test-plan-examples-when-to-use/#what_is_level_test_plan) – Also referred as the **phase test plan**, this document gives details about the **testing activities** that must be performed for every test level.

Actual existence of each of these documents is different for each project and/or organizations. For larger projects / organizations, they could possibly be split across multiple documents, however for smaller projects/organizations, they could possibly be part of a single document. Every document is explained in detail here.

Larger organizations which are more formal and larger projects have more detailed documents compared to smaller, less formal organizations & smaller projects.

## Test Policy

The overarching **objective of an organization** in **performing test activities** is described in **Test Policy** document. It is created by **seniors in the test management team** in association with senior managers of the stakeholders’ groups.

Sometimes, **test policy is part of** a wider **quality policy** adopted by the organization. In such cases the quality policy will explain the overall aim of the management with respect to quality.

### Test policy contents

Test policy is a short document, summarized at a high level that contains the following:

* Outline the advantages of testing, business value delivered to the organization which justifies the [**cost of quality**](http://tryqa.com/what-is-cost-of-quality-in-software-testing/)
* **Define**[**test objectives**](http://tryqa.com/what-is-the-software-testing-objectives-and-purpose/) like confidence building, defect detection and reduce quality risks
* Describe the **methods to measure test efficiency**and effectiveness in fulfilling test objectives
* Summarize the processes used in testing using ISTQB primary [**test process**](http://tryqa.com/what-is-fundamental-test-process-in-software-testing/)
* Describe ways for the organization to [**enhance its testing processes**](http://tryqa.com/software-testing-process-improvements-for-test-qa-managers/)

Test policy must also include testing activities necessary for maintenance of current project & also development of new projects.

Test Strategy

## Test Manager should be able to decide on a suitable testing strategy for the project based on the project requirements as well as the organizations needs.

(AMMSRRC)

### Table of contents

1. [**Types of testing strategies**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#types_of_testing_strategies)
   1. [**Analytical strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#analytical_strategy)--the [**testing conditions**](http://tryqa.com/what-is-test-condition-test-analysis-advantages-disadvantages-level-of-detail/) to be covered after analyzing the [**test basis**](http://tryqa.com/what-is-test-analysis-or-how-to-identify-the-test-conditions/), be it **risks** or **requirements**
   2. [**Model based strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#model_based_strategy)--testing team chooses an **existing or expected situation** and **creates a model for it**, taking into account inputs, outputs, processes and possible behavior. scenario of mobile application testing. To carry out its [**performance testing**](http://tryqa.com/what-is-performance-testing-in-software/), models may be developed to emulate outgoing and incoming traffic on mobile network, number of active/inactive users, projected growth, etc.
   3. [**Methodical strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#methodical_strategy)--follow a **predefined quality standard** (like ISO25000), **checklists** or simply a **set of test conditions**. Standard checklists can exists for specific types of testing (like security), application domains.
   4. [**Standards or process compliant strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#standards_or_process_compliant_strategy)--testers follow the **processes or** **guidelines established by committee for standards** or panel of industry experts to identify test conditions, define test cases and put testing team in place.
   5. [**Reactive strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#reactive_strategy)--**testing is based on defects found** in actual system. scenario where [**exploratory testing**](http://tryqa.com/what-is-exploratory-testing-in-software-testing/) is being used.
   6. [**Consultative strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#consultative_strategy)--**uses consultations from**[**key stakeholders**](http://tryqa.com/how-to-identify-stakeholders-for-testing/)**as input** to **decide the scope** of test conditions as in the case of user directed testing. the [**compatibility**](http://tryqa.com/what-is-compatibility-testing-in-software/) of any web based application with possible browsers is to be tested.
   7. [**Regression averse strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#regression_averse_strategy)--**focus on** [**reducing regression risks**](http://tryqa.com/how-to-manage-regression-risk-and-evolve-manual-automated-test-cases-in-agile-methodology/) for functional or non-functional product parts.
2. [**Details included in test strategy**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#details_included_in_test_strategy)
3. [**Test strategy selection**](http://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/#test_strategy_selection)

## Details included in test strategy

The final **test strategy should include details** about these factors:

* Test levels
* Entry as well exit conditions for each test level
* Relationships between the test levels
* Procedure to integrate different test levels
* Techniques for testing
* Degree of independence of each test
* Compulsory as well as non-compulsory standards that must be adhered
* Testing environment
* Level of automation for testing
* Tools to be used in testing
* Confirmation and regression testing
* Re-usability of both software and testing work products
* Controlling testing
* Reporting on test results
* Metrics and measurements to be evaluated during testing
* Managing defects detected
* Managing test tools and infrastructure configuration
* Test team members roles and responsibilities
* \*\*\*\*Test Strategy defines which Strategy/approaches we can apply for full fill the objective of the project
* Ex. Project 🡪 Automation testing 🡪 Java/ C#/ Python / etc + Selenium tool
* Test Strategy documents will be prepared by Test Strategist – TS & PM
* Test Strategy documents company level documents\*\*\*\*

[@arun motori]

Step1:create different scenario and test cases for different functionality

We apply test design techniques:equivalent class portioning ,boundary value analysis ,desigen table, state transitiontesting

Expertise 1}error guessing

Exploratory testing

Priortise testing

Step 2:we get application for testing :we perform smoke testing ---

**Test Methodology------trm**

Mapping between test factor and quality factors to the development stages.

Factors are—authorization ,access control cease to use,easy to operate ,community, correctness, coupling, data integrity ,performance ,recovery,portable,servie level maintainable,methodologies and audit detail

Project manager will prepare test methodology

**Test Methodology-**

* Test methodology defines 🡪 Environment follow/use for Strategy/approaches
* Test methodology documents will be **prepared by PM**
* Test methodology documents Project level documents
* Test methodology documents 🡪 PM will prepared the **TRM (Test responsibility matrix)**
* TRM defines **development stage mapping with testing factor**
* While preparing the TRM consider

1. Project requirement
2. Project scope
3. Risk in project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Development Stage**  **Testing factor** | **Information gathering/ Analysis** | **Design** | **Coding** | **Testing** | **Maintains** |
| System & functional Testing | No | No | Yes | Yes | Yes |
| Security | No | No | No | Yes | Yes |
| Performance Testing | No | No | No | Yes | Yes |
| GUI/ UI | No | Yes | Yes | Yes | Yes |

**Test Plane-**

* Test plane will contains Resource allocation, Job allocation, estimation, etc
* Test plane documents prepared by Test lead
* Test plane documents Project level documents

**Identify test scenario & Test case design –**

* Tester will identity test scenario & Test cases design
* Test scenario & Test cases design prepared by Tester
* Test scenario & Test cases documents Project level documents

**Test cased execution & defect report-**

* In TCE, if we found defects then we will raised to developer
* Modified build we will perform regression & Retesting.

Test plan

Plan our coming day test activities for smooth execution we neet to create test plan .

Sample test plan document we have to download .project name:tutorial ninja, client tutorial ninja ,prepared by whom :company name date

1overview:

2 Scope :provide url . functionality testing whoch are in the scope it is called as inclusion and decided n kick off meeting. Exclusion those functionality is nhot check test environment

* Planning, monitoring and control
* Analysis
* Design
* Implementation
* Execution
* Evaluating exit criteria and reporting
* Test closure activities

# Software Testing Life Cycle (STLC) –

**Q. What is your organization Test process?**

**Q. What is STLC?**

Test Initiation

Test Plane

Test Scenario

Defect sent to developer

Test Case Execution- Test Proof

Test Case Design

Requirement Analysis

Test Closer/ Test Summery Report

**Test Initiation-**

* In Test initiation stage PM is working
* PM will prepared the **TRM** (Test responsibility matrix)

**Test Plane –**

* **Test plane will** prepared by test lead
* Test plane will contains **Job allocation, Resource allocation, Estimation**

**Test scenario-**

* **Test scenario will** prepared by **tester against US**
* Test scenariowill defines “**how to test”**

**Test cases design-**

* **Test cases will** prepared by **tester against Test scenario**
* Test **cases** will defines “**what to test”**

**Test cases execution & Defect report-**

* When we got build from developer then we will execute test cases
* If in TCE, when we **got the defect** then we will **create these defect in JIRA/ Azure Develops tool**
* Tester will inform to developer and developer has **fixed these defect**
* Tester will do **Re-testing & Regression**

**Test Closer/ Test Summery Report-**

* Test Closer/ Test Summery Report will prepped by test lead
* Test Summery Report will prepped against sprint
* Test Closer Report will prepped against Module

# Testing Process-

**BRS**

Testing

Development

**SRS/FRS/CRS**

**Design Test Initiation Stage- TRM**

**Coding Test Plane –Sprint**

**STLC**

**Unit Testing Test scenario & Test Case Design**

**SDLC**

**Integration Testing Test Case Execution & Closer**

**(Install Build)**

**Level 0** Sanity**/ Smoke Testing (**Check theStability of build)

**Level 1 BBT/ System & function testing (**Intern & External**)**

**(**If found **defect** sent to developer) (Inform throw JIRA/Azure)

**Level 2 Regression Testing on Modified built**

**Level 3 Final Regression Testing**

# Test Plane-

* Test plane will be **prepared by Test Lead**
* Test plane will be prepared **against the Sprint**
* While preparing these test plane, test lead focus on

1. Job allocation
2. Resource allocation
3. Estimation
4. Test plane main purpose to defines **start and end day of testing**

* In Test plane documents will contains

1. Test plane ID
2. Testing task
3. Feature to be tested
4. Feature not to be tested
5. Pass & Fail Criteria
6. Test delivery (entry and exit Criteria)
7. Test environments
8. Defect life cycle
9. Test risk
10. Responsibility
11. Signature & Approval

* I have **worked with Test Lead for preparation of Test Plane**

# Test Scenario-

SRS/ FRS/ CRS

Use Case / User Story

Test Scenario

Test cases design

**BA BA Tester Testers**

**SRS/ FRS/ CRS-**

* SRS defines system requirement specification
* SRS will be **prepared by BA**
* SRS will be **derived from BRA**
* SRS will **contains multiple Use cases**
* SRS will contain

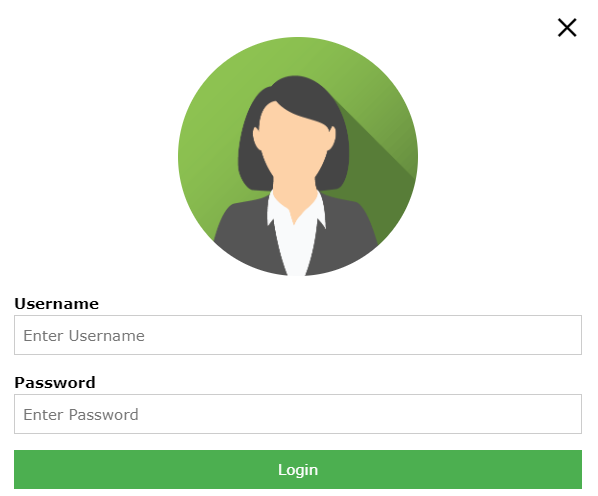
1. Function requirements
2. Function flow diagram
3. Use Cases
4. Screenshot / Prototype

**Use Cases-**

* Use Cases defines system single specification requirement
* Use Cases will be **prepared by BA**
* Use Cases will be **derived from SRS**
* Use Cases will **contains multiple Use stories**
* Use cases will contains

1. Description
2. Acceptance Criteria

**Test Scenario –**

* Test scenario will be **prepped by Tester**
* Test scenario will be **derived/ identified from US**
* Test scenario defines **“Ways to test functionality”**
* Test scenario will be written in **+VE ways**
* Test scenario will **contains multiple test cases**
* **Ex. US =** Login page – Username = email id/ Mobile number accept, Password = 4 to 8 Charter (1 Upper letter, 1 small letter, 1 number, 1 Special no)****

**Test Scenario** **-**

1. Verify the login page by passing email id data into Username test box
2. Verify the login page by passing mobile number data into Username test box
3. Verify the login page by passing data into password test box
4. Verify the login page by pressing Login with data into Username & Password test box

**Test Cases-**

* Test cases will be **prepped by Tester**
* Test cases will be **derived/ identified from Test scenario**
* Test cases defines **“How to test functionality”** OR “**Input, Process, Output**”
* Test cases will be written in **+VE ways & -VE ways**
* Test cases will **contains multiple steps for execution**

**Test Scenario** **-** Verify the login page by passing email id data into Username test box

**Test cases-**

1. Verify the login page by passing gmail id (abc@gmail.com) into Username test box
2. Verify the login page by passing yahoo id (abc@yahoo.com) into Username test box
3. Verify the login page by passing Outlook id (abc@outlook.com) into Username test box
4. Verify the login page by passing Hotmail id (abc@hotmail.com) into Username test box
5. Verify the login page by passing Redfmail id (abc@ Redfmail.com) into Username test box
6. Verify the login page by passing company id (priti.patil@ Infosys.com) into Username test box
7. Verify the login page by passing invalid gmail id ($#$&^@ gmail.com) into Username test box
8. Verify the login page by passing null/blank into Username test box

**Test Scenario** -- Verify the login page by passing mobile number data into Username test box

**Test Scenario** - Verify the login page by passing data into password test box

**Test Scenario** - Verify the login page by pressing Login with data into Username & Password test box

# Test Case review-

* **Review-** Review defines to **check** **correctness and completeness of your documents**
* Test cases review 4 types

1. Self Review
2. Peer Review
3. Internal Review
4. External Review

**Self Review-**

1. In self review, Tester will do review their **own test case**

**Peer Review-**

* In Peer review, Test cases reviewed by **Senior tester/ Colleague / Test lead**
* When Tester has **completed test cases design**, Tester will **inform** to Senior tester/ Colleague / Test lead throw **Mail/ Meeting**

**Internal Review-**

* In internal review, Test cases review will **reviewed by BA**
* When Tester has **completed test cases design**, Tester will **inform** to BA throw **Mail**
* BA will set up **one meeting (Team Tool)** with Tester

**External Review-**

* In external review, Test cases review will **reviewed by Client/ UAT**
* When Tester has **completed test cases design**, Tester will **inform** to Test lead
* **Test lead** will set up **one meeting (Team Tool)** with **Client/ UAT**
* In these meeting, Every Tester will reviewed/ explain the test cases
* In my Project, we are following internal review OR **external review**
* In review, if we got the any suggestion/ feedback related test cases. Tester will **accepted these suggestion/ feedback** and Tester will modify/ change/ add test cases 🡪 **Directly modify/ change/ add test cases in the tool** OR In excel will modify/ change/ add into column “**Comments/suggestion“** present in excel sheet
* **While doing Test cases review check-**

1. Test cases should cover all functionality mentation in the US
2. Test cases should cover client business/ business logic related test cases
3. Test cases should not be duplicated
4. Test cases should cover standard format
5. Test cases should be simple / Understandable
6. Test cases should not grammatically mistake
7. Test cases should not spelling mistake

# Test Case execution-

* In TCE, Tester will **prepared the Test proof**
* In Test proof documents will contains Test cases, Screenshot of functionality related Test cases & Tables data stored
* In TCE, if we found defect then we will create/ raised these defect in JIRA/ Azure DevOps tool and inform to developer

# Defect life cycle-

* **Defect –** If tester found error in functionality then these error are called as defect
* if we **found defect** then we will **create/ raised these defect in JIRA/ Azure DevOps tool** and inform to developer
* **Defect life cycle –** The **journey of defect from there start to end stage**
* **Defect stage different**
* **Defect stage 🡪 New, Open, Fixed, Closed, re-open, rejected, differed**

**Closed**

**Fixed**

**Re-open**

**Reject**

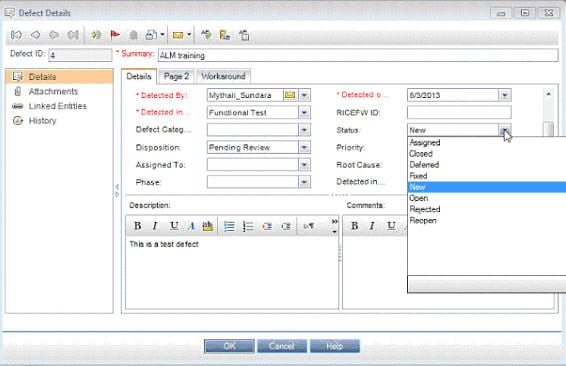
**Open**

**New**

**Differed**

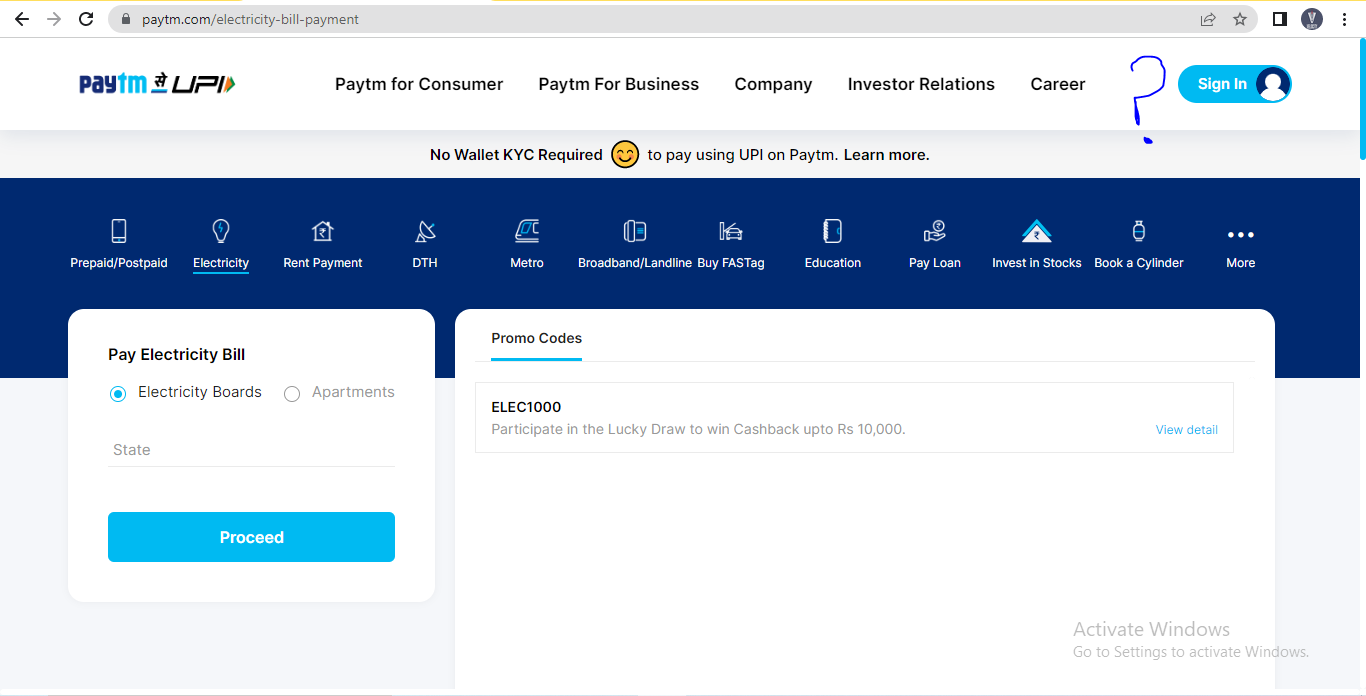
**Tester Developer Tester**

* If we found defect then we will **create/ raised these defect in JIRA/ Azure DevOps tool.** Tester will mark different stage
* **NEW-** When **tester** will created/raised new defect then we will mark **defect status as New.** Tester will inform to developer throw mail
* **OPEN -** When **Developer** is **looking/analyze into defect** then he will mark **defect status as Open**
* **FIXED –** When developer found that it is **valid defect** then developer will fix these defect then he will mark **defect status as Fixed.** Developer will inform to tester throw mail
* **1. CLOSED-** When tester found that **defect has been fixed** then we will mark **defect status as Closed.** Tester will mark different stage
* **2. RE-OPEN-** When tester found that **defect has not been fixed** then we will mark **defect status as Re-open.** Tester will mark different stage
* **REJECT -** When developer found that it is **In**-**valid defect** then developer will mark **defect status as Rejected.** Developer will inform to tester throw mail
* **DIFFERED –** When Client has changed the priority of that US **OR** those defects has not been fixed in current sprint **OR** defect will take more time for fixing, then these defect has been fixed into next sprint then the **status of these defect will be mark as Differed** and these **decided by PM, Designer & BA.**



# Types of defect-

* **Duplicate defect-** Those defect which is in similar nature OR those defect which is created by changing test data these defect are called duplicate defect
* **EX.** Paytm – Recharge module ––Test data – 9988001122, Circle= MP, Operator - Airtel – Airtel mobile number is not working - Defect ID= 2738
* Test data – 7788001100, Circle= UP, Operator - Airtel – Airtel mobile number is not working - Defect ID= 2739 (duplicate defect)
* **Re-producible defect-** Those defect which is producing again and again in SIT environments after fixing also those defect are called Re- producing defect
* In JIRA/ Azure Devops toll we **mark Re- producing defect = Y / N**
* **Ex.** Paytm – Ticket booking travels – 10 peoples email pass – 10 different types emails – SIT environments company emails id – **paytminfo@wipro.com**

****

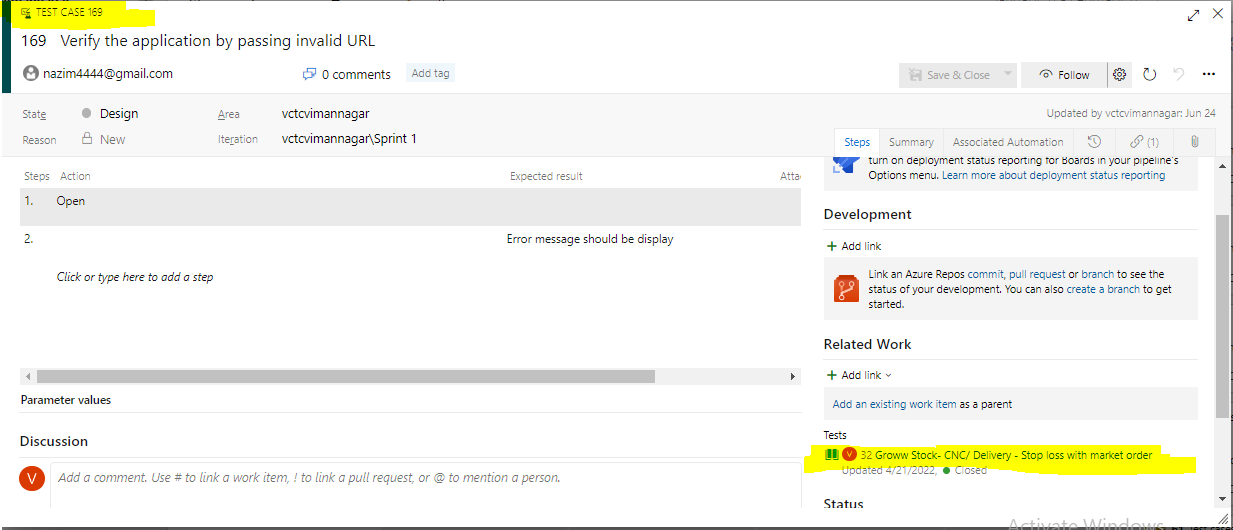
* **Show stopper defect/ Blocker defect –** Those defect which is stop the testing activity OR Those defect which is related core functionality/ Main functionality which is stop the testing activity these defect are called show stopper defect/ blocker defect
* **Ex.** Paytm – Recharge module – Operator is not selecting / working
* **Critical defect -** Those defect which related core functionality/ Main functionality but which is not stop the testing activity these defect are called show critical defect
* **Latent defect-** Those defect which is hidden from another defect these defect are called latent defect
* **Ex.** Paytm – Recharge module – Mobile number , Operator, Circle, Amount- If operator is not working or operator dropdown not shows ant values – Defect raised but to these circle defect has been miss
* **Bug leakage –** Bug leakage definesDefect which is missed from SIT and found in UAT
* **Bug released-** Bug released term defines miner defect has seen present in functionality OR know defect in functionality but still be we are deployment
* **Condition sign-off / Partial sign-off -** know defect in functionality but still be we are deployment to client
* **Ex.** Paytm – Ticket booking travels – functionality = 10 peoples name, mobile, lname, email pass – 10 different types emails sent – But in SIT environments company emails id –[**paytminfo@wipro.com**](mailto:paytminfo@wipro.com) **(In** SIT environments these functionality not support)
* **Defect density –** NumberofDefect of found in total line of code
* **Ex. Defect density =** 10 defect / 200 total line of code = 5%
* **Defect cluster-** Defect cluster defines If we have found more defect in small functionality

# Traceability matrix/ RTM (requirement Traceability matrix)-

* In Traceability matrix, **Test cases & defect are mapped/ link with US/ requirement**
* 2 types Traceability matrix

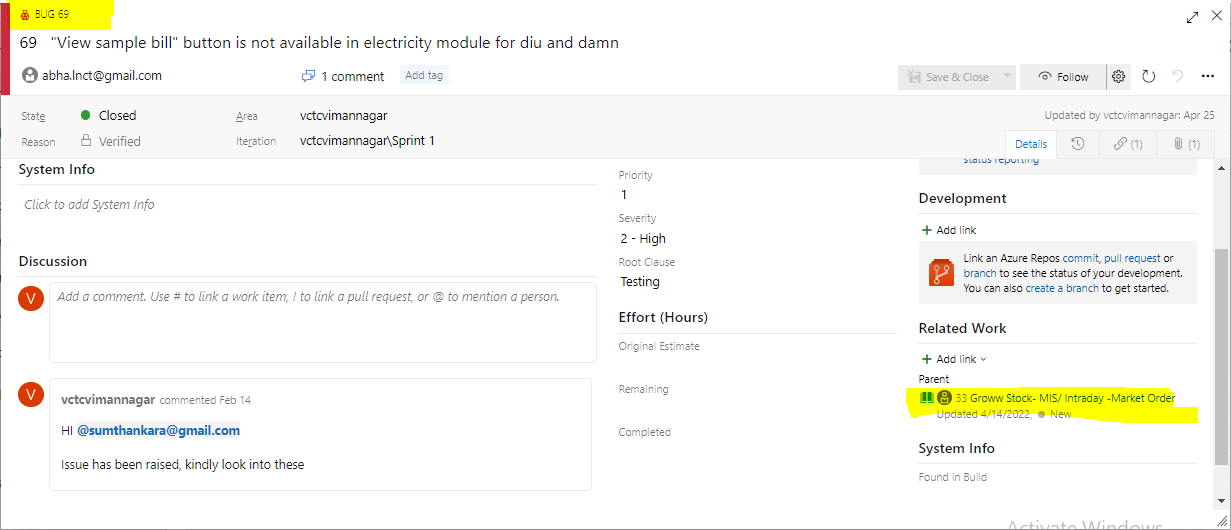
1. **Froward Traceability matrix –**

* In Froward Traceability matrix, Test cases are mapped/ link with US/ requirement

****

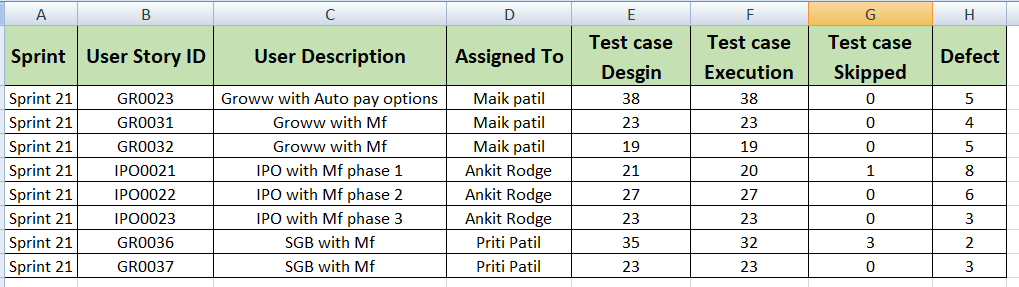
1. **Reverse Traceability matrix –**

* In Froward Traceability matrix, defect are mapped/ link with US/ requirement

****

# Report –

* Test plane 🡪 Test lead 🡪 Against Sprint
* Test cases design 🡪 Tester 🡪 Against US
* Test summery report 🡪 Test lead 🡪 Against Sprint
* Test closer report 🡪 Test lead 🡪 Against Module
* Daily wise report 🡪 Tester 🡪 Against day work
* Defect report 🡪 Tester 🡪 Defect present in every US
* **Test summery report –** How many TCD, TCE, TCS, Defect has been raised / written against sprint

****

* **Defect report –** In a sprint how many defect has been raised

