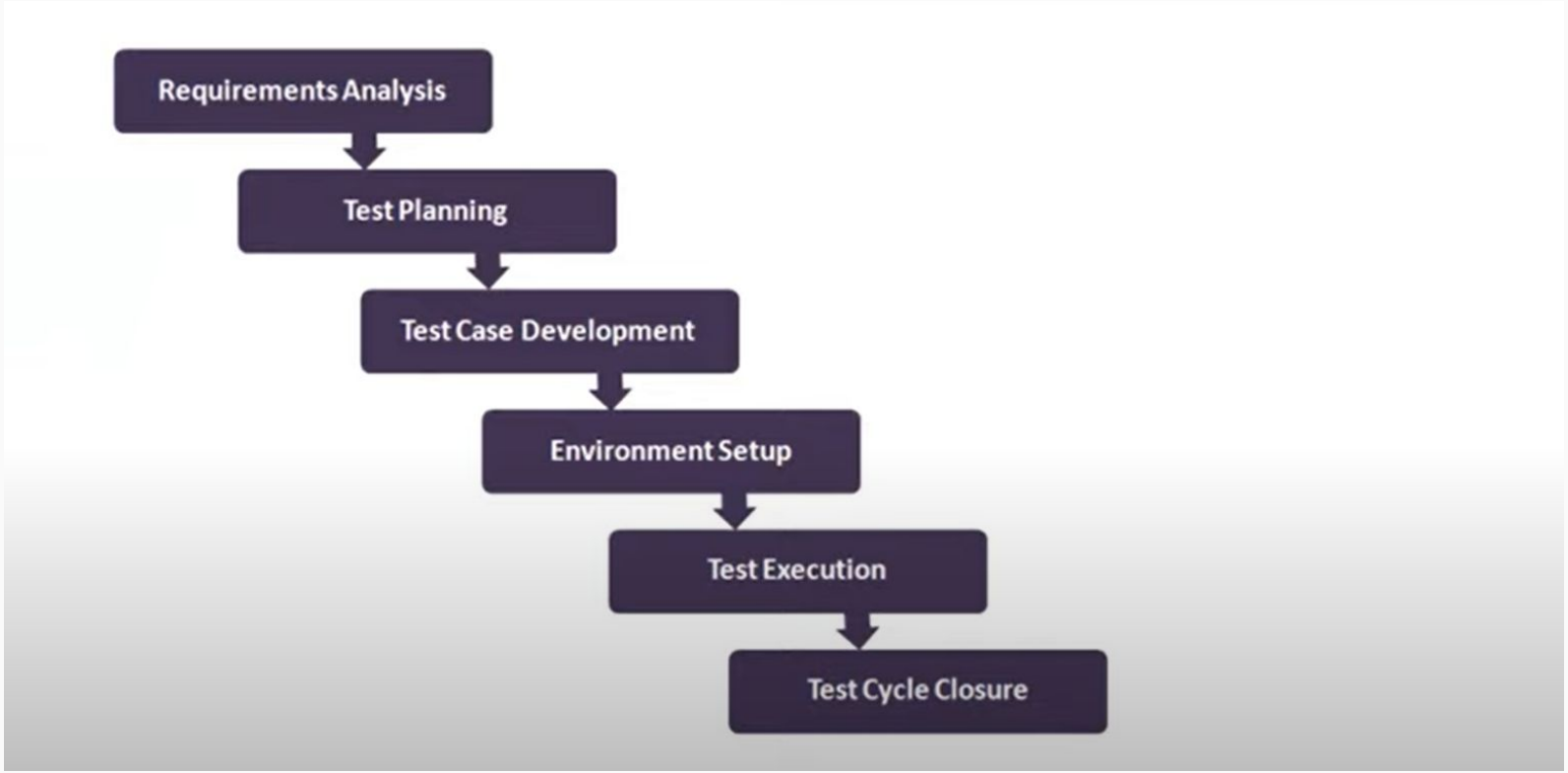
# STLC

## Software Testing Life Cycle

# Steps in STLC

1. Requirement Analysis
2. Test Planning
3. Test Design
4. Test Execution
5. Defect Reporting / Bug Reporting
6. Test Closure

# Software Testing Life Cycle (STLC)



# STLC

| SNO | PHASE | Input | Activities | Responsibility | Out Come |
|---|---|---|---|---|---|
| 1 | **Test Planning** | Project Plan | ➢Identify the Resources | Test Lead/Team Lead (70%) | Test Plan Document |
| | **What to test** | Functional Requirements | ➢Team Formation | Test Manager (30%) | |
| | **How to test** | | ➢Test Estimation | | |
| | **when to test** | | ➢Preparation of Test Plan | | |
| | | | ➢Reviews on Test Plan | | |
| | | | ➢Test Plan Sign-off | | |
| | | | | | |
| 2 | **Test Designing** | Project Plan | ➢Preparation of Test Scenarios | Test Lead/Team Lead(30%) | Test Cases Document |
| | | Functional Requirements | ➢Preparation of Test Cases | Test Engineers( 70%) | Traceability Matrix |
| | | Test Plan | ➢Reviews on Test Cases | | |
| | | Design Docs | ➢Traceability Matrix | | |
| | | Use cases | ➢Test Cases Sign-off | | |
| | | | | | |
| 3 | **Test Execution** | Functional Requirements | ➢Executing Test cases | Test Lead/Team Lead(10%) | Status/Test Reports |
| | | Test Plan | ➢Preparation of Test Report/Test Log | Test Engineers (90%) | |
| | | Test Cases | ➢Identifying Defects | | |
| | | Build from Development Team | | | |
| | | | | | |
| 4 | **Defect Reporting & Tracking** | Test Cases | ➢Preparation of Defect Report | Test Lead/Team Lead(10%) | Defect Report |
| | | Test Reports/Test Log | ➢Reporting Defects to Developers | Test Engineers (90%) | |
| | | | | | |
| 5 | **Test Closure/Sign-Off** | Test Reports | ➢Analyzing Test Reports | Test Lead/Test Manger(70%) | **Test Summary Reports** |
| | | Defect Reports | ➢Analyzing Bug Reporting | Test Enginners(30%) | |
| | | | ➢Evaluating Exit Criteria | | |

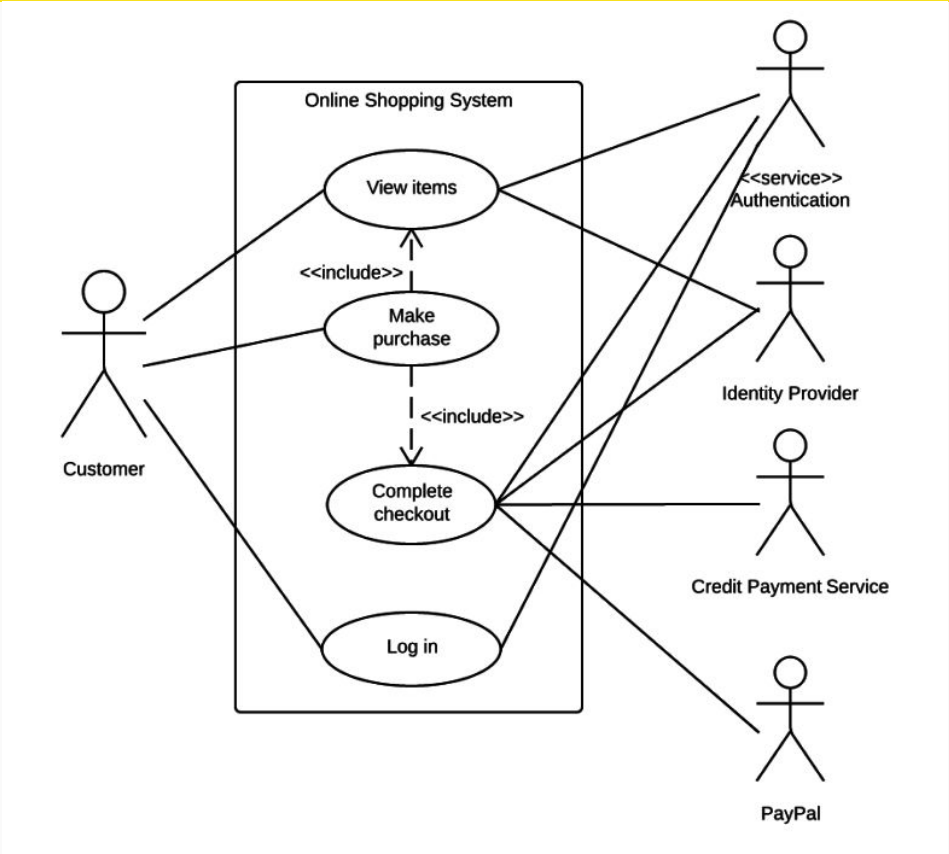
# Test Plan Contents

- A Test plan is a document that describes the test scope, test strategy, objectives, schedules, deliverables, and resources required to perform testing for a software product.
- **Test plan template contents:**
  - **Overview**
  - **Scope**
    - **Inclusions**
    - **Test Environment**
    - **Exclusions**
  - **Test Strategy**
  - **Defect Reporting Procedure**
  - **Roles/ Responsibility**
  - **Test Schedule**
  - **Test Deliverable**
  - **Pricing**
  - **Entry and Exit Criteria**
  - **Suspension and Resumption Criteria**
  - **Tools**
  - **Risks and Mitigation**
  - **Approvals**

# Use case, Test Scenario and Test Case

- **Use Case:**
  - Use case describes the requirement
  - Use case contains THREE items
    - **Actor**, which is the user, which can be a single person or a group of people, interacting with a process.
    - **Action**, which is to reach the final outcome.
    - **Goal/Outcome**, which is the successful user outcome.
- **Test Scenario:**
  - A possible area to be tested (What to test)
- **Test Case:**
  - Step by step actions to be performed to validate functionality (How to test)
  - Test case contains test steps, expected result and actual result.

# Sample Use Case



# Use Case vs Test Case

- Use Case
  - Describes functional requirement, prepared by Business Analyst

- Test Case
  - Describes Test Steps/ Procedure, prepared by Test Engineer

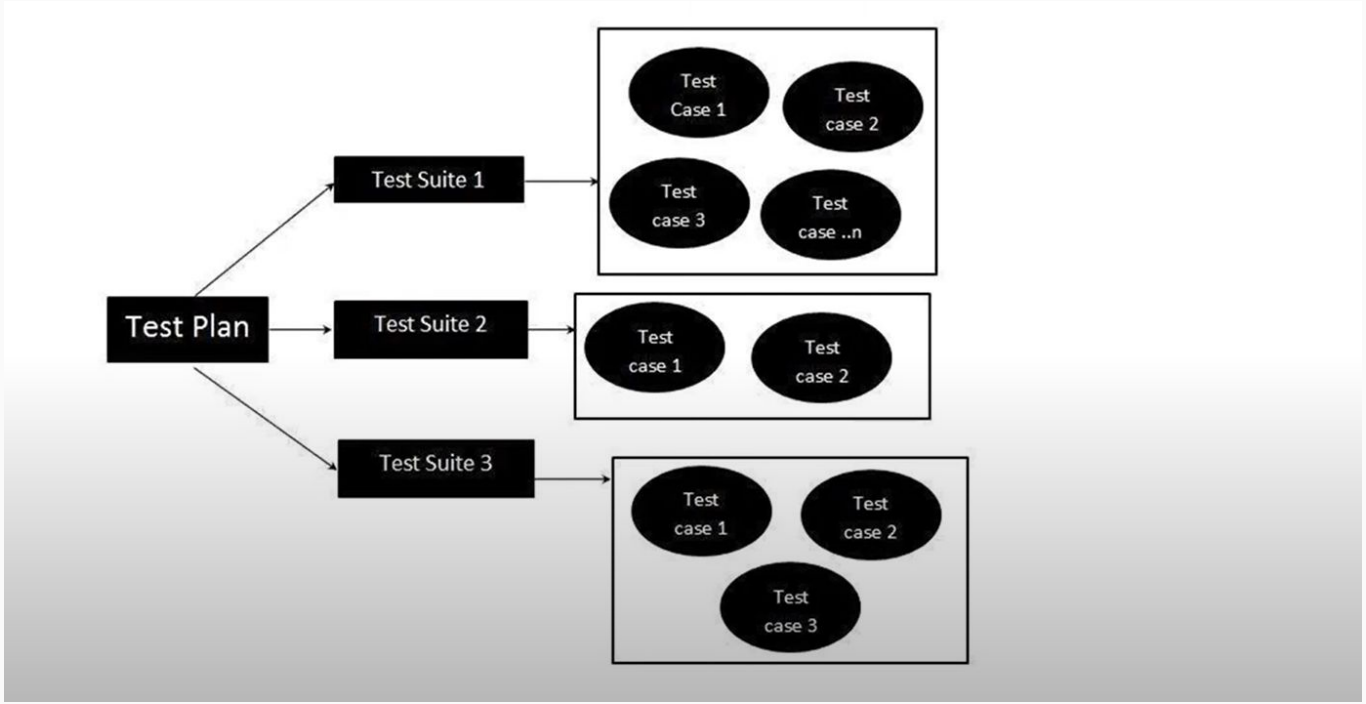
# Test Scenario vs Test Case

- Test Scenario is “**What to be tested”,** and Test Case is “**How to be tested”**

- **Example:**
- Test Scenario: Checking the functionality of Login button
  - TC1: Click the button without entering username and password
  - TC2: Click the button only entering Username
  - TC3: click the button while entering wrong user name and wrong password

# Test Suite

- Test Suite is ground of test cases which belongs to same category.



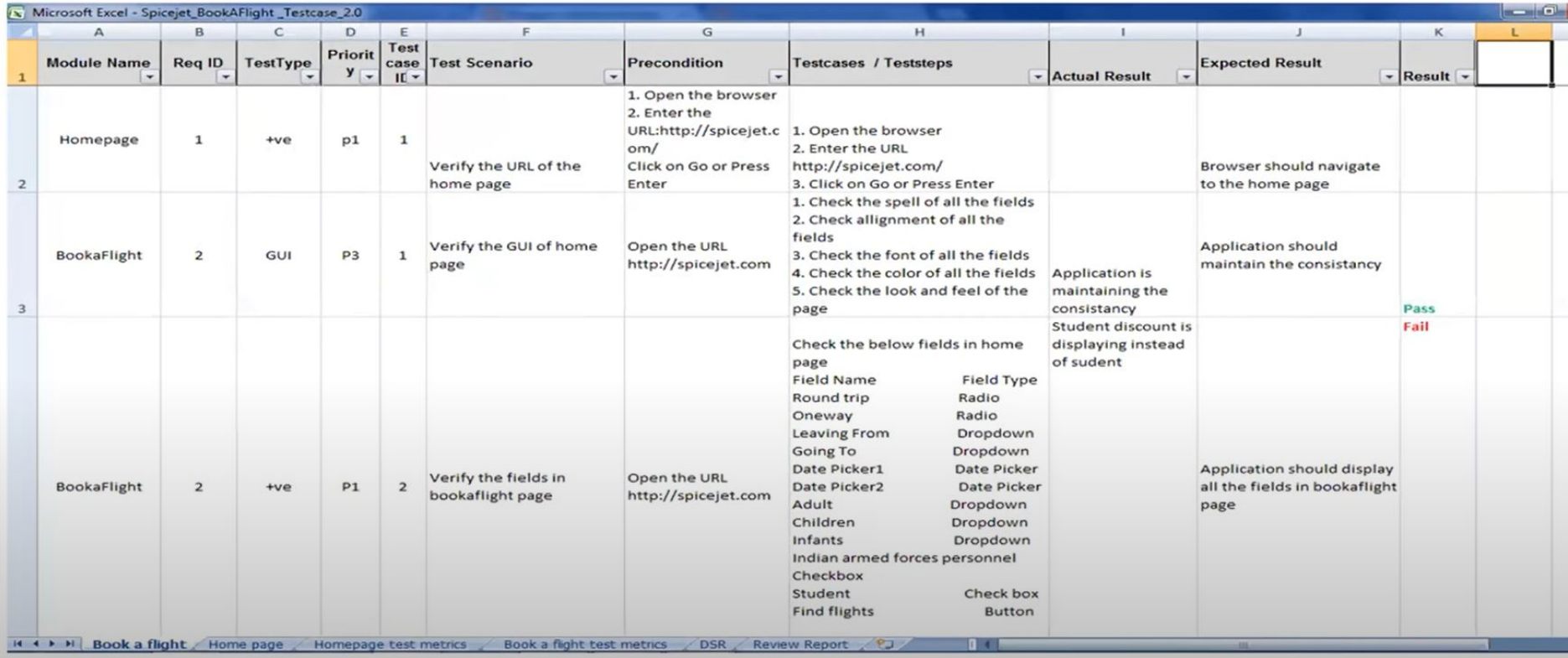
# What is Test Case?

- A Test Case is a set of actions executed to validate particular feature or functionality of your software application.

# Test Case Contents

- Test Case ID
- Test Case Title
- Description
- Pre-condition
- Priority (P0, P1, P2, P3) - order
- Requirement ID
- Steps/Actions
- Expected Result
- Actual Result
- Test data

# Test Case Template



Microsoft Excel - Spicejet_BookAFlight _Testcase_2.0

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Module Name | Req ID | TestType | Priority | Test case ID | Test Scenario | Precondition | Testcases / Teststeps | Actual Result | Expected Result | Result | |
| 2 | Homepage | 1 | +ve | p1 | 1 | Verify the URL of the home page | 1. Open the browser<br>2. Enter the URL:http://spicejet.com/<br>Click on Go or Press Enter | 1. Open the browser<br>2. Enter the URL http://spicejet.com/<br>3. Click on Go or Press Enter | | Browser should navigate to the home page | | |
| 3 | BookaFlight | 2 | GUI | P3 | 1 | Verify the GUI of home page | Open the URL http://spicejet.com | 1. Check the spell of all the fields<br>2. Check allignment of all the fields<br>3. Check the font of all the fields<br>4. Check the color of all the fields<br>5. Check the look and feel of the page | Application is maintaining the consistancy | Application should maintain the consistancy | Pass | |
| | BookaFlight | 2 | +ve | P1 | 2 | Verify the fields in bookaflight page | Open the URL http://spicejet.com | Check the below fields in home page<br>Field Name — Field Type<br>Round trip — Radio<br>Oneway — Radio<br>Leaving From — Dropdown<br>Going To — Dropdown<br>Date Picker1 — Date Picker<br>Date Picker2 — Date Picker<br>Adult — Dropdown<br>Children — Dropdown<br>Infants — Dropdown<br>Indian armed forces personnel Checkbox<br>Student — Check box<br>Find flights — Button | Student discount is displaying instead of sudent | Application should display all the fields in bookaflight page | Fail | |

Book a flight | Home page | Homepage test metrics | Book a flight test metrics | DSR | Review Report

# Requirement Traceability Matrix (RTM)

- What is RTM (Requirement Traceability Matrix)?
- RTM describes the mapping of Requirements with the Test cases.
- The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while doing Software testing.

- Requirement Traceability Matrix - Parameters include
  - Requirement Id
  - Requirement Description
  - Test case IDs

# Sample RTM

| Req No | Req Desc | Testcase ID | Status |
|---|---|---|---|
| 123 | Login to the application | TC01,TC02,TC03 | TC01-Pass<br>TC02-Pass |
| 345 | Ticket Creation | TC04,TC05,TC06,<br>TC07,TC08,TC09<br>TC010 | TC04-Pass<br>TC05-Pass<br>TC06-Pass<br>TC06-Fail<br>TC07-No Run |
| 456 | Search Ticket | TC011,TC012,<br>TC013,TC014 | TC011-Pass<br>TC012-Fail<br>TC013-Pass<br>TC014-No Run |

# Test Environment

- Test Environment is a platform specially build for test case execution on the software product.
- It is created integrating the required software and hardware along with proper network configurations.
- Test environment stimulates production/ real time environment.
- Another name of test environment is Test Bed.

# Test Execution

- During this phase test team will carry out the testing based on the test plans and the test cases prepared.
- **Entry Criteria:** Test cases, Test Data and Test Plan
- **Activities:**
  - Test Cases are executed based on the test planning.
  - Status of  test cases are marked, like Passed, Failed, Blocked, Run and others.
  - Documentation of test results and log defects for failed cases is done.
  - All the blocked and failed test cases are assigned bug ids.
  - Retesting once the defects are fixed.
  - Defects are tracked till closure.
- **Deliverables:** Provides defects and test case execution report with completed results.

# Guidelines for Test Execution

- The build being deployed to the QA environment is the most important part of the execution cycle.
- Test execution is done in Quality Assurance (QA) environment.
- Test execution happens in multiple cycle.
- Test execution phase consists Executing the test cases + test scripts (if automation)

# Defects/Bugs

- Any mismatched functionality found in a application is called as Defect/Bug/Issue.
- During Test Execution Test engineers are reporting mismatches as defects to developers through templates or using tools.
- Defect Reporting Tools:
  - Clear Quest
  - DevTrack
  - Jira
  - Quality Center
  - Bug Jilla etc.

# Defect Report Contents

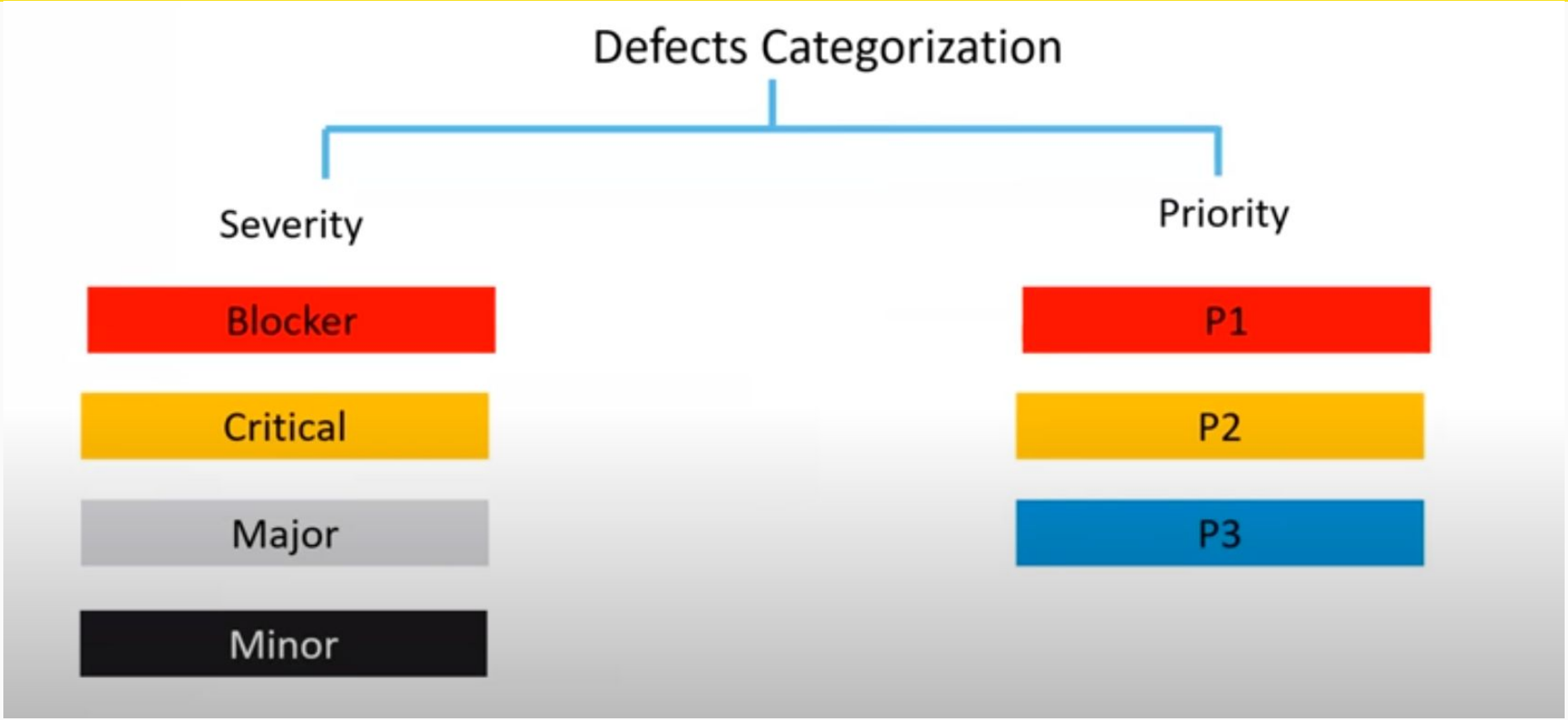
- **Defect_ID:** Unique identification for defects
- **Defect_Description :** Detailed description of the defect including information about the module in which defect was found
- **Version:** Version of the application in which defect was found
- **Steps:** Detailed steps along with screenshots with which the developer can reproduce the defects.
- **Date Raised:** Date when the defect is raised.
- **Reference:** where you provide reference to the documents like requirements, design, architecture or may be even screenshots of the error to help understand the defect

# Defect Report Contents

- **Detected By:** Name/ID of the tester who raised the defect
- **Status:** Status of the defect
- **Fixed By:** Name/ID of the developer who fixed it
- **Date Closed:** Date when the defect is closed
- **Severity:** which describes the impact of the defect on the application
- **Priority:** which is related to defect fixing urgency.

  Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively.

# Defect Classification

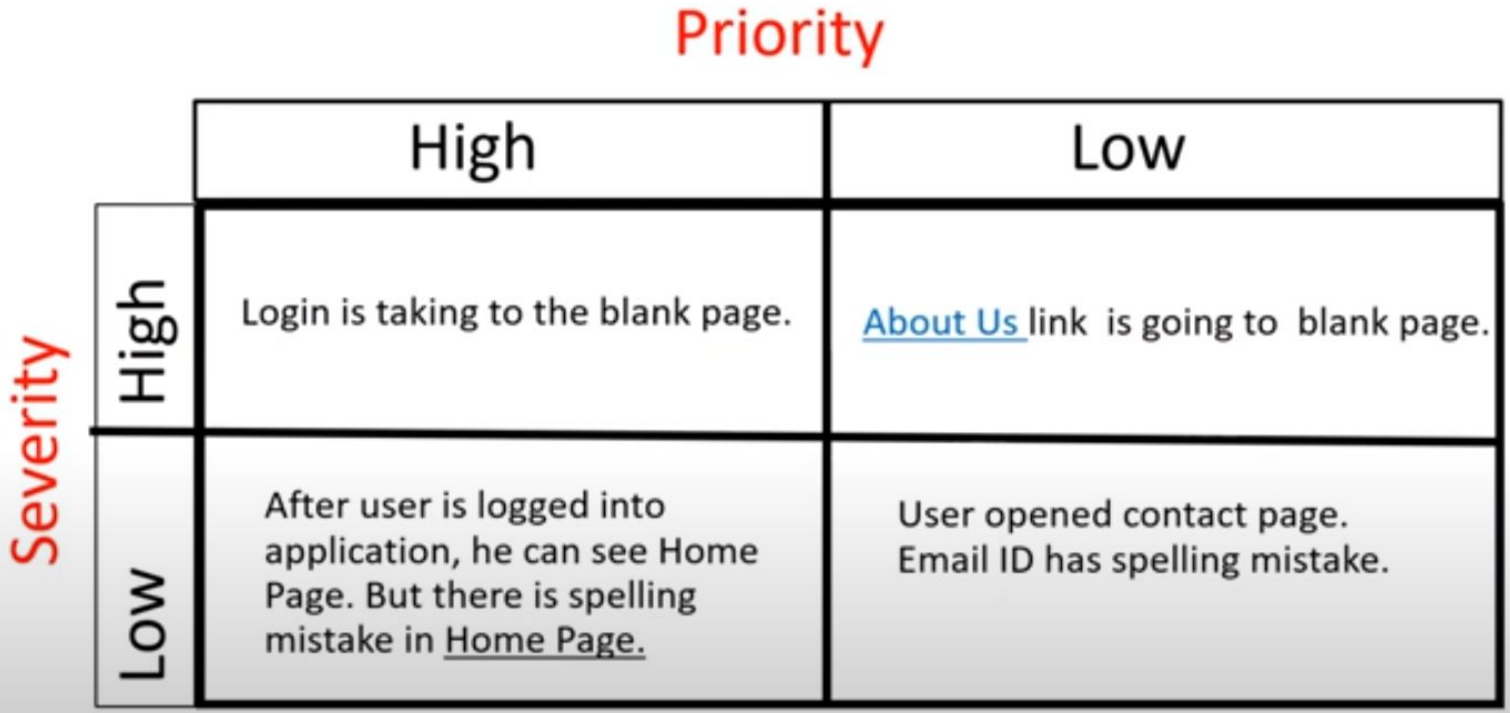


# Defect Severity

- Severity describes the seriousness of the defect and how much impact on Business workflow.
- **Defect severity can be categorized into four class**
  - **Blocker (Show stopper):** This defect indicates nothing can proceed further.
    - Ex. App crashes, login not working
  - **Critical**: The main/ basic functionality is not working. Customer business workflow is broken. They cannot proceed further.
    - Ex. Fund transfer is not working in banking.
    - Ex. Ordering product is not working in online shopping.
  - **Major:** It cause some undesirable behaviour, but the feature/application is still functional.
    - Ex. After sending email there is no confirm message.
  - **Minor:** It won't cause any major break-down of the system.
    - Ex. Look and feel issues, spellings, alignment

# Defect Priority

- Priority describes the importance of defect.
- Defect Priority states the order in which a defect should be fixed.
- **Defect priority can be categorized into three class**
  - **P0 (High): The defect must be resolved immediately as it affects the system severely and cannot be used until it is fixed.**
  - **P1 (Medium): It can wait until a new version/builds is created.**
  - **P2 (Low): Developer can fix it in later releases.**


Priority
High
Low
Severity
High
Low
Login is taking to the blank page.
About Us link is going to blank page.
After user is logged into application, he can see Home Page. But there is spelling mistake in Home Page.
User opened contact page. Email ID has spelling mistake.

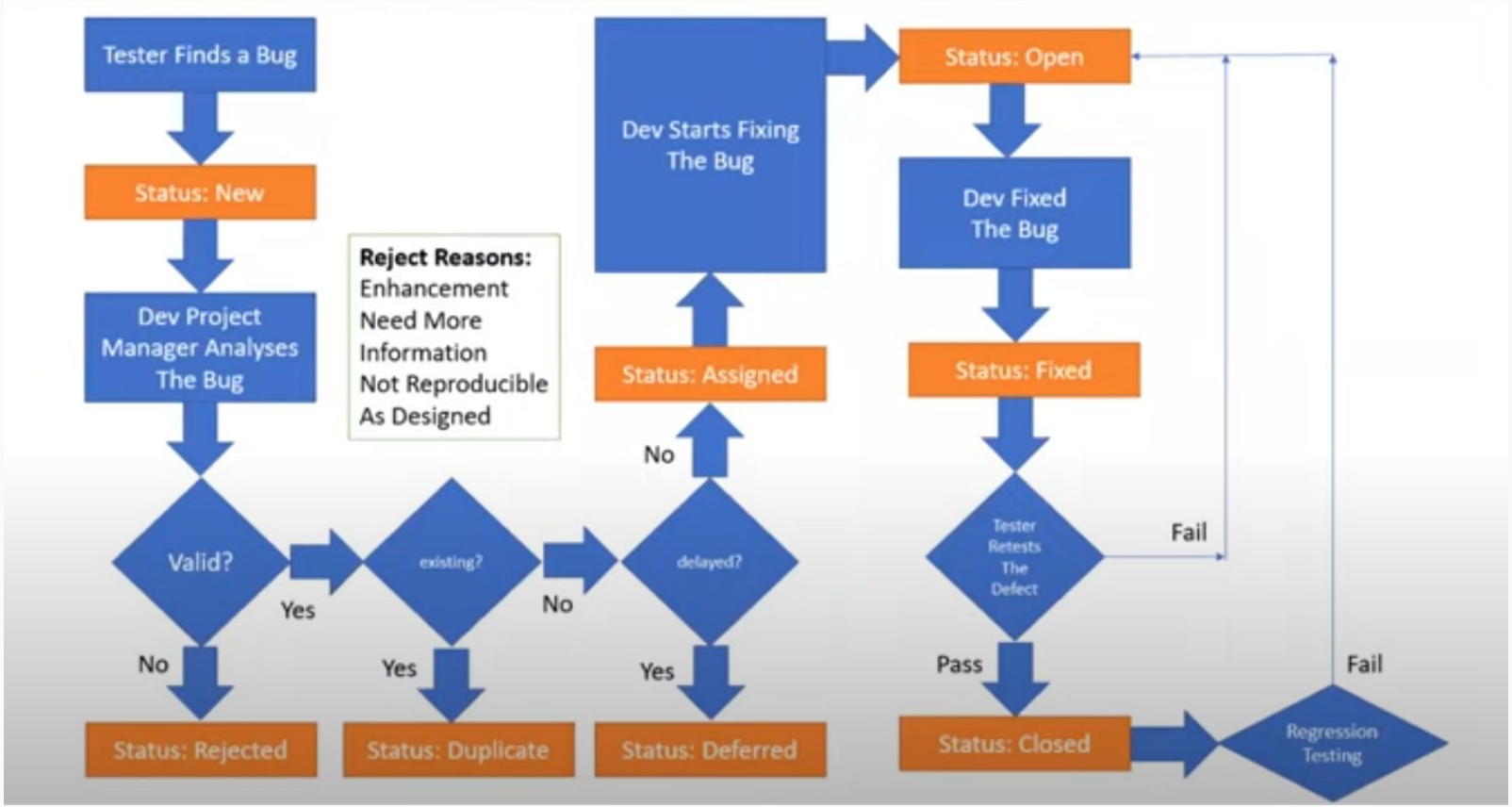
# More examples….

- **Lp-Ls -** A spelling mistake in a page not frequently navigated by users.
- **LP-Hs-** Application crashing in some very corner case.
- **Hp-Ls-** Slight change in logo color or spelling mistake in company name.
- **Hp-Hs-** Issue with login functionality.
- **HS-Ls-** Web page not found when user clicks on a link.
- **Lp-Ls-** Any cosmetic or spelling issues which is within a paragraph or in the page.

# Defect Resolution

- After receiving the defect report from the testing team, development team conduct a review meeting to fix defects. Then they send a Resolution Type to the testing team for further communication.
- **Resolution Types:**
  - **Accept**
  - **Reject**
  - **Duplicate**
  - **Enhancement**
  - **Need more information**
  - **Not Reproducible**
  - **Fixed**
  - **As Designed**

# Bug Life Cycle



# Test Cycle Closure

- **Activities**
  - Evaluate cycle completion based on Time, Test coverage, Cost, Software, Critical Business Objectives, Quality
  - Prepare test metrics based on the above parameters
  - Document the learning out of the project
  - Prepare Test Summary report
  - Qualitative and quantitative reporting of quality of the work product to the customer.
  - Test result analysis to find out the defect distribution by type and severity
- **Deliverables**
  - Test Closure report
  - Test Metrics

# Test Metrics

- No. of Requirements
- Avg. No. of Test Cases written Per Requirement
- Total No. of Test Cases written for all Requirement
- Total No. of test cases executed
- No. of Test cases passed
- No. of Test cases failed
- No of test case blocked
- No of test cases executed
- Total no of defects identified
- Critical, Higher, Medium, Low defects count
- Customer Defects
- No of defects found in UAT

# Test Metrics

- % of Test cases executed
- % of Test cases not executed
- % of Test cases passed
- % of Test cases failed
- % of Test cases blocked

# Test Metrics

- Defect Density: Number of defects identified per requirement/s
  - No of defects found / size (no of requirements)
- Defect Removal Efficiency (DRE)
  - (A / A + B) * 100
  - (Fixed Defects / (Fixed Defects + Missed defects)) * 100
  - A - Defects identified during testing (Fixed Defects
  - B - Defects identified by the customer (Missed defects)
- Defect Leakage
  - (No. of defects found in UAT / No of defects found in Testing) * 100

# Test Metrics

- Defect Rejection Ratio:
  - (No. of defect rejected / Total No. of defects raised) * 100
- Defect Age
  - Fixed date - Reported date
- Customer satisfaction
  - No of complaints per period of time

# QA/Testing Activities

- Understanding the requirements and functional specifications of the application.
- Identifying required Test Scenarios.
- Designing test cases to validate application
- Setting up test environment (Test Bed)
- Execute test cases to valid application
- Log test results (Hot many test cases pass/fail)
- Defect reporting and tracking
- Retest fixed defects of previous build
- Perform various types of testing in application

# QA/Testing Activities

- Reports to test lead about the status of assigned tasks
- Participate in regular team meetings
- Creating automation scripts
- Provides recommendation on whether or not the application / system is ready for production.

# 7 Principles of Software Testing

- Start software testing at early stages. Means from the beginning when you get the requirements.
- Test the software in order to find the defects.
- Highly impossible to give the bug free software to the customer.
- Should not do Exhaustive testing. Means we should not use same type of data for testing every time.
- Testing is context based. Means decide what type of testing should be conducted based on type of application.

# 7 Principles of Software Testing

- We should follow the concept of Pesticide Paradox. Means, if you are executing same cases for longer run, they won't be finding any defects. We have to keep updating test cases in every cycle/release in order to find more defects.
- We should follow defect clustering. Means some of the modules contains most of the defects. By experience, we can identity such risky modules. 80% of the problems are found in 20% of the modules.