1. What is a Software?

Software is a collection of programs to help us to perform a Task

1. **What is Testing?**

1-> Software testing is a part of the Software Development process

2-> It is an activity to detect and identify the defects in the software

3-> Reason to test the product is to provide Quality Software

1. Types of Software:

Device Software, Programming Software, Application Software.

System Software:

Drivers, Operation System, Utilities.

Programming Software

Compilers, Debuggers, Interpreters.

Application Software

Web Application, Mobile Application, Desktop Application

1. Software Quality

* 1-> Bug-free
* 2-> Delivered on time
* 3-> Within Budget
* 4-> Meets Requirements and Expectations
* 5-> Maintainable

1. Why Software has Bugs?

-> Miscommunication or No Communication

-> Software Complexity

-> Lack of Skilled Testers

-> Requirement Changes

-> Programming Errors

1. SDLC (Software Development Life Cycle)

Planning-> Analysis->Design->Implementation->Testing->Maintenance

STLC Phases;

* Phase 1 — Requirement Analysis. ...
* Phase 2 — Test Planning. ...
* Phase 3 — Test Design. ...
* Phase 4 — Test Environment. ...
* Phase 5 — Test Execution. ...
* Phase 6 — Defect Tracking. ...
* Phase 7 — Test Reporting.

1. Water Fall Model

* Requirement -> Design ->Development ->Testing -> Deployment ->Maintenance

1. Waterfall Model

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * 1) Quality of the Software will be Good * 2) Requirement Changes are not allowed so the chances of finding Bugs will be less * 3) Initial investment will be fewer because testers will be hired once Developing is done * 4) Preferred for small projects | * 1) Requirement changes are not allowed * 2) If there is a defect in Requirement that will be continued at Later Phases only * 3) Testing will start after coding |

1. **Spiral Model**

-> It is also called an Iterative Model (Version Control Model)

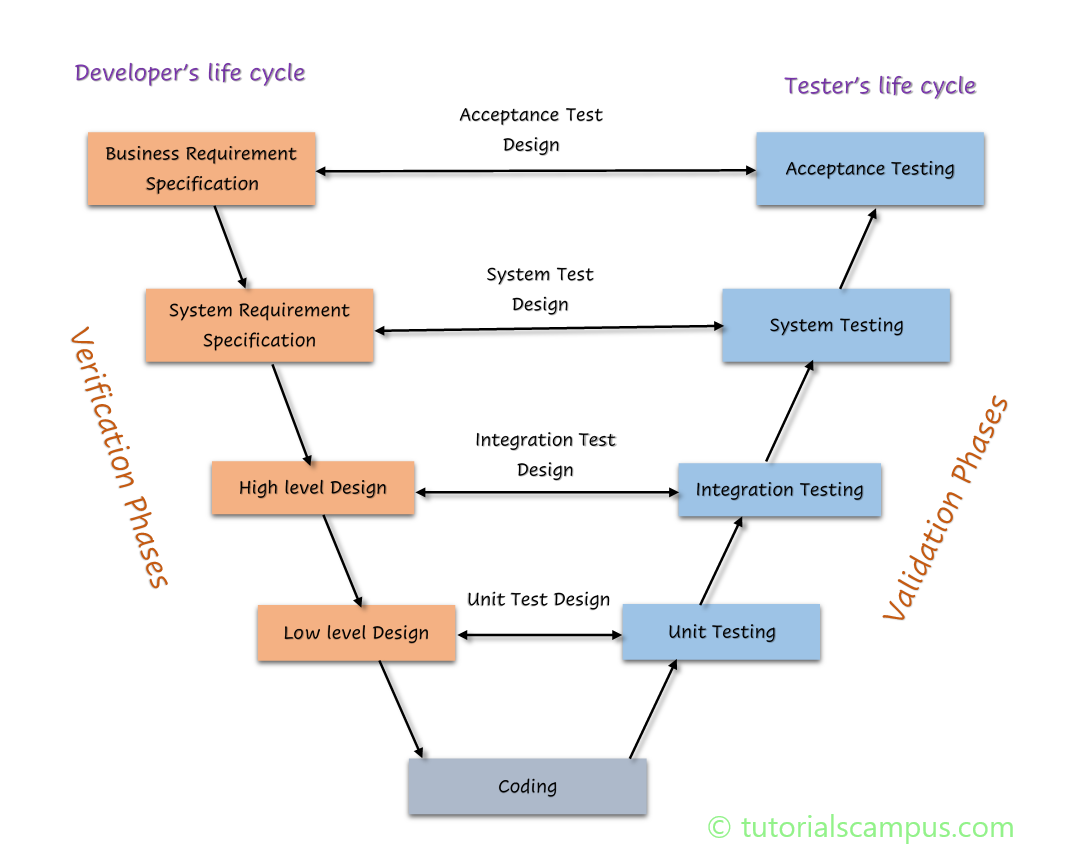
-> Spiral Model Overcome the drawbacks of the waterfall

-> In Every cycle new software will be released to the customer

-> Software will be released in multiple versions

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * 1) Testing is done in Every Cycle before going to the next cycle * 2) Customers will get to use the software of every module. * 3) Requirement changes are allowed after every cycle. | * 1) Every cycle of the spiral model looks like a waterfall model. |

1. V- Model



|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Testing is involved in each stage | * Documentation is more * Initial investment is more |

1. Verification and Validation

|  |  |
| --- | --- |
| Verification | Validation |
| * 1) Verification checks whether we are developing the correct product or not * 2) Focus on Documentation. * 3) Reviews * 4) Walkthroughs * 5) Inspections | * 1) Focus on Software only * 2) Actual Software is tested * 3) Unit Testing, Integration, System Testing, UAT Testing. |

1. **Static Testing (Verification)**

* **Review:** Conducts on the documents they should be correct & complete. (One or Many)
* 1) Requirement Reviews
* 2) Test Plan Reviews
* 3) Test Cases Reviews
* 4) Design Reviews

1. Walkthrough

* 1) It is an informal Review ( two or three people will be involved)
* 2) Author reads the documents or code and discuss with the Team Members
* 3) It is not Pre-planned and can be done anytime.
* 4) Particular duration is not defined.

1. Inspection:

1) It is a formal review (contains several people)

2) Inspection will be scheduled at the proper time.

3) Here 3 people (Reader, Writer, Moderator) are the main persons.

1. **Dynamic Testing (Validation)**

A Single Component or Module of a Software

1) Unit Testing will be conducted on a Single Program or Single Module

2) It is White Box Testing Technique.

3) It will be conducted by the Developers.

Integration Testing

* Integration Testing is performed on Several modules
* It is a White box Testing Technique
* It will check the data communication between multiple Modules.

System Testing:

GUI: Testing the User Interface of the Application

It includes elements such as Buttons, Colors, fonts, icons, sizes

Usability:

It will check how easily the user is able to access the Application.

**Functionality Testing:**

* It checks the Actual Behaviour of the Application
* Functional testing focuses on how your feature will work



* Object and Properties Testing
* Database Testing
* Error Handling Testing
* Calculation and Manipulation Testing
* Links Existence & Links Execution
* Cookies & Session
* Object and properties Testing: Check the properties of objects present on the Application or not

Ex: Enable, Disable, Visible

Database Testing: Database is an organized collection of structured information (Storage Location) A database

* Error Handling: Tester verifies the error messages while performing incorrect action on the application.
* It should be readable in simple language
* Incorrect or Invalid data

Calculation/ Manipulation:

Tester should check the calculations

Links and Execution:

* Where exactly the links are placed links are navigating to proper page

**Cookies and Sessions: Temporary files created by the browser**

Non-Functional Testing:



* Once the Application is working according to the requirement will we will start Non-Functional Testing.
* It will focus on Load and Security
* Performance Testing
* Security Testing
* Recovery Testing
* Compatibility Testing
* Installation Testing
* Garbage/Sanitation Testing

**Performance Testing:**

* Load: Increasing the load on the application gradually and checking the speed.
* Stress: Increasing/Decreasing the load on the Application Suddenly.
* Volume: It will check how much data is able to handle

**Security Testing:**

* **To check how secure our application.**
* **Authentication: Users are valid or not**
* **Authorization /Access Control: Permissions of the Valid User**

**Recovery Testing:**

* **Check the System change from abnormal to Normal**

**Compatibility Testing:**

* **Forward Compatibility**
* **Backward Compatibility**
* **Hardware Compatibility (Configuration Testing)**

**Installation Testing:**

* **Check Screens are clear to understand**
* **Screens navigation**
* **Simple or not**
* **Un-installation**

**Sanitation / Garbage Testing:**

* **If any application provides extra features/functionality then we consider it as a Bug.**

**Functional Vs Non-Functional**

**Functional Non Functional**

* **1) Validates the functionality of Software 1) Verify the performance, Security,**
* **Reliability of the software**
* **2) Functionality describes what software does. 2) Non-Functionality describers how**
* **Software works.**
* **3) Concentrates on user requirement 3) Concentrate on user expectation**
* **4) Functional testing takes place before Non-functional testing. 4) Non-Functional testing performed after**

**Finishing Functional testing**

**Regression Testing:**

* Testing conducts on modified modules to make sure that there will not be an impact on existing functionality because of changes like adding /deleting/modifying features.
* **Unit Regression Testing**: Testing only the changes/modifications done by the developer
* **Regional Regression Testing**: Testing the modified along with the impacted modules

**Full Regression:** Testing the main feature & remaining part of the application

**Re-Testing:**

* If the developer fixes the bug, Tester will test the impacted area which is called re-testing

Tester closes the bug if it is fixed. If not they reopen the bug and sent it back to the developer.

**End to End Testing:**

End-to-end testing is a type of software testing that tests flows from beginning to end, as opposed to only testing individual steps

Smoke Testing:

* In this testing method, the development team deploys the build in QA. A subset of test cases are taken, and then testers run test cases on the build. The QA team test the application.

Sanity Testing

**Sanity testing** is a type of software testing that aims to quickly evaluate whether the basic functionality of a new software build is working correctly or not. [It is usually performed on builds that are in the initial stages of development before the full regression testing is performed](https://www.geeksforgeeks.org/sanity-testing/)

**Smoke Vs Sanity**

|  |  |
| --- | --- |
| Smoke test is done to make sue the build we received from the development team is stable or not.  Smoke Testing is performed by both Developers and Testers  Build may be either stable or not It is done in beginning  It will be done everytime if there is a new Build release | 1) A sanity Test is done during the release phase to check for the main functionalities of the application.  2) Sanity testing is performed by testers alone  3)Sanity testing is done if the build is stable  4) It is done once the application is stable  5) It is planned when there is not enough time to do it. |

**Exploratory Testing:**

* Testers have to understand completely and test it.
* It will be done when the application is ready but there is no requirement
* Drawbacks: The tester might misunderstand any feature as a bug as a feature since you do not have requirements
* 2) Time-consuming
* 3) If there is any bug in the application, the tester is unable to identify it.

**Adhoc Testing:**

* Testing application randomly without any test cases (or) any business requirement document
* 2) Adhoc testing is an informal testing type
* 3) Tester should have knowledge of the application even though he doesn’t have the requirements
* 4) Tester is usually on unplanned activity

**Monkey testing:**

* Testing applications randomly without any test cases or any business requirement document.
* It is informal

Tester does not have knowledge of the application.

|  |  |  |
| --- | --- | --- |
| **Exploratory** | **Adhoc** | **Monkey** |
| No Documentation No plan Formal Testing Testers doesn’t have knowledge on the functionality Random Testing Intention is to learn or explore the application. Any Applications which is new to the Tester | No Documentation No plan Informal Testing Tester should know the Application functionality Random Testing The intention is to break the application Any Applications | No Documentation No plan Informal Testing Tester should know the Application functionality Random Testing The intention is to break the application Any Applications |

**END to End Testing:**

* Testing over all functionality of the System including the data integration

Among all modules is called End to End testing.

**Positive Testing:**

* Testing the application with valid inputs is called positive Testing

**Negative testing:**

Testing the application with invalid inputs is called Negative Testing.

Globalization & Localization

* Globalization Performed to ensure the system or Software application can run in ay culture and environment.

Localization is Performed to check system application is designed for specific region.

Requirement Analysis:

* Requirement analysis is the process of determining user expectations for a new or modified product and is vital for effective QA software testing as it lays down the basic foundation for various stages of the SDLC

Test Plan Contents:

* A Test plan is a document that describes the test scope. Test Strategy, objectives, schedule, deliverables and resources required to perform testing for a software product
* Test plan Template Contents:-
* Test Strategy -> Procedure to test step by step process (Smoke, Sanity and Functional)
* Defect reporting -> How we will report the Bug
* Roles/Responsibilities -> (Responsibility of Test Engineer, Test Lead, Author)
* Test Schedule -> (Duration and Activity)
* Test Deliverables - > Test cases, Test Execution ( Documents)
* Entry and Exit Criteria -> Where tester have to start and exit
* Suspension and Resumption criteria -> (If an application is broken need to report it)
* Tools – (Bug tracking, Automation tools, Test Management)

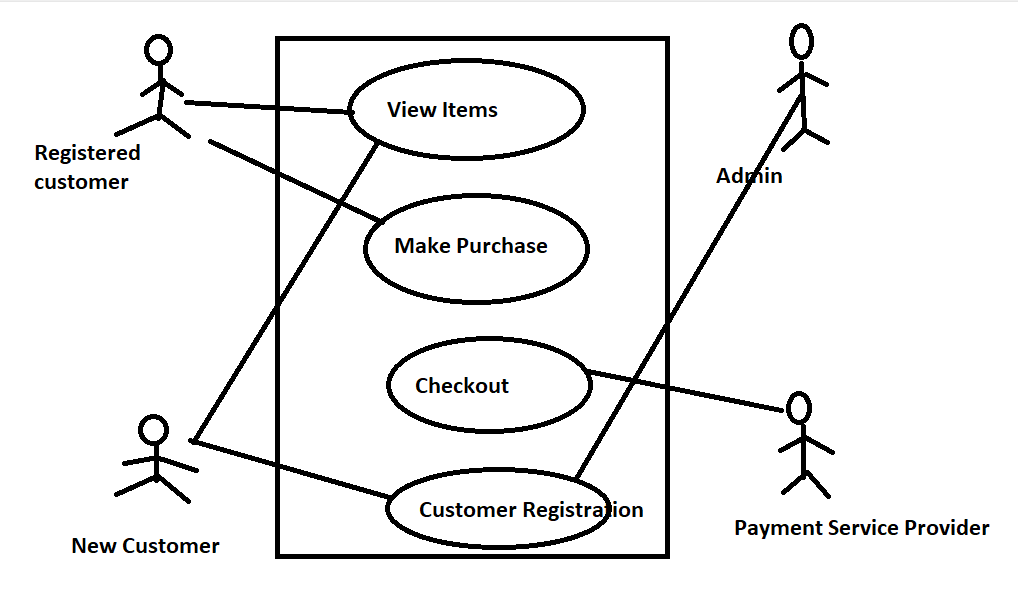
Use Case:

Use case describes the requirement containing three items

Actor- Which user is , interacting with process

Action- Which is the final outcome

Goal- To achieve the correct outcome.



Test Scenario: A Possible area to be tested (What to test)   
  
Test case: Step-by-step actions to perform and validate the functionality.   
  
Test steps, expected result, Actual result  
  
Use case Vs Test Case: use case describes the functional requirement, prepared by BA  
  
test case describes test steps prepared by Test Engineer.   
  
Test Scenario: What to be tested- How to Test  
Checking the functionality of the login button  
  
Test suite: Group of test cases which belong to the same category.

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  
  
Test Execution: The testing team will carry out the testing plans and test cases  
  
Entry criteria: Test cases, Test Data & Test plan  
  
Activities:   
  
1) Test cases are executed based on the test planning  
2) Status of the test cases are passed, failed and Blocked.  
3) Defects are tracked till closure.

Defect/Bug  
If the functionality of the application is not related to the actual exception it is called as Bug.   
  
Defect Reporting tools:  
Clear Quest  
Dev Track  
Jira (Bug reporting & Test Management)  
Quality Centre (Bug reporting & Test Management)   
Bug Jilla

Severity: If any defect can affect/impact the particular software it is related to the Severity

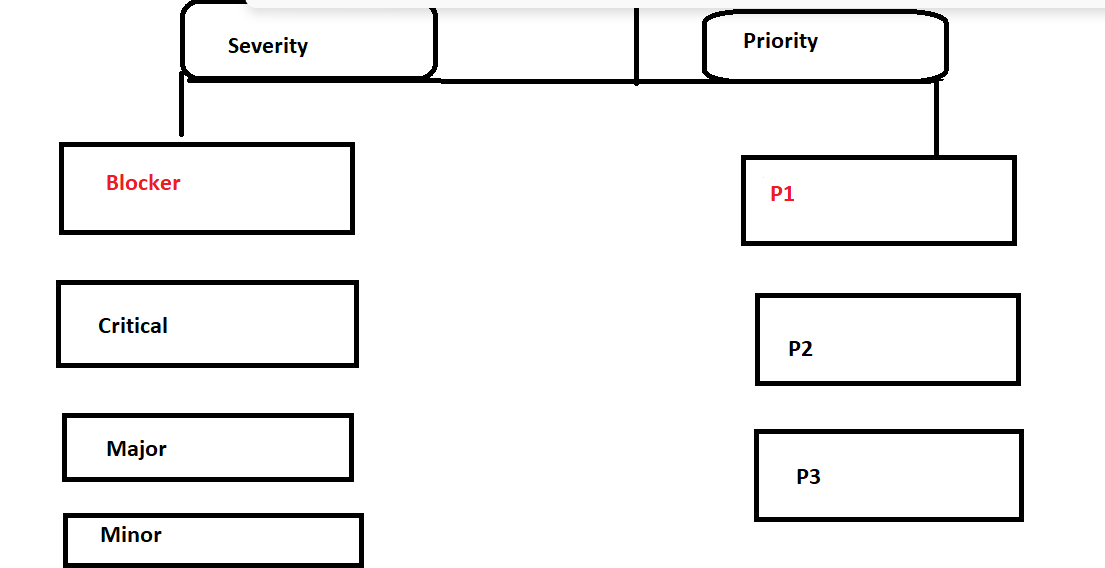
Types in severity

Blocker

Critical

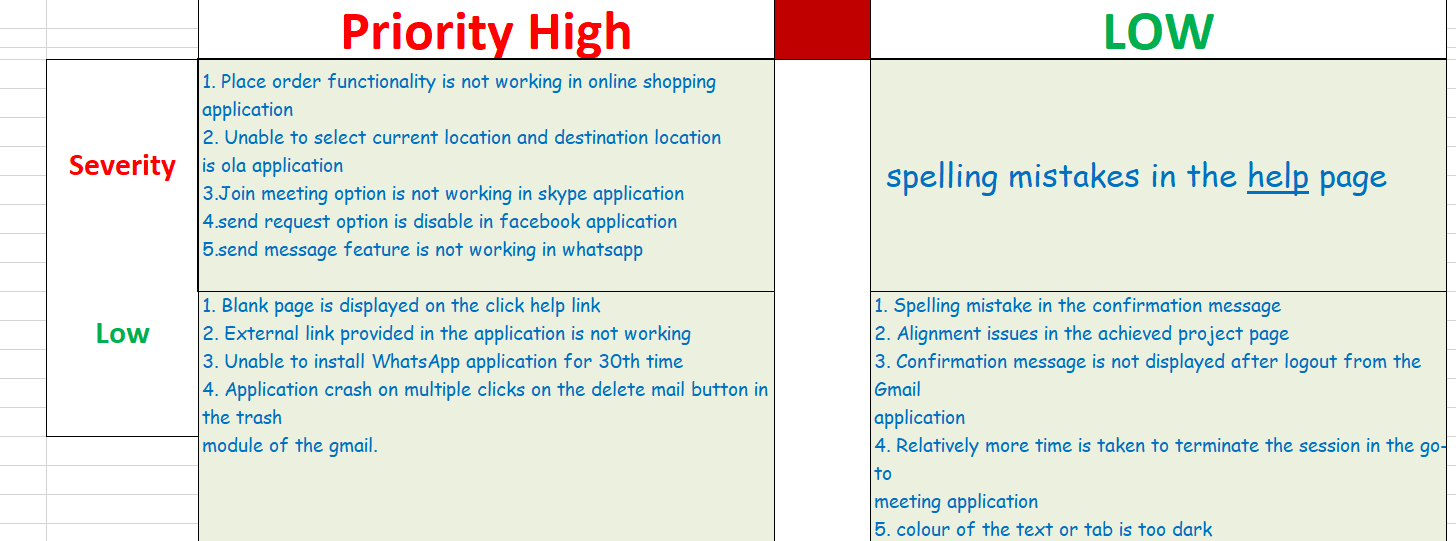
Major

Minor

**Blocker: This defect indicates nothing can proceed   
  
Ex: Application crashed, Login not worked  
  
  
Critical: The Basic functionality is not working  
  
Ex: Unable to add an item  
Ex: Unable to make the payment  
  
Major: It causes some Behaviour/feature of the application is still working  
Ex: No confirmation message  
  
Minor: It won’t cause a breakdown  
  
Ex: Spelling, Alignments**

Priority: It describes the importance of the defect

P0(High) : The defect must be released immediately as it affects the system and cannot be   
used until it is fixed  
  
  
p1(Medium) : It can wait until a new version/ Build is created  
  
  
p2 (Low) : Developer can fix it in Later releases.

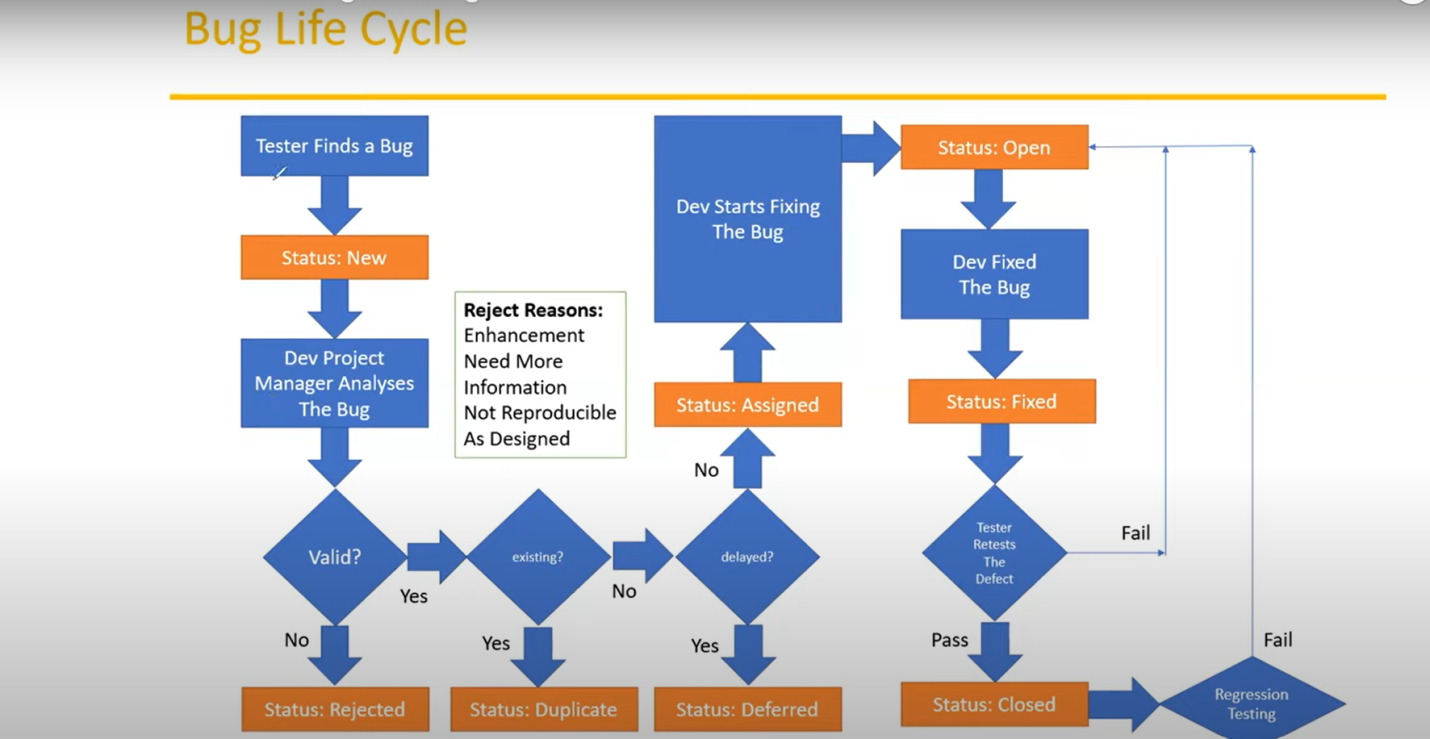


White box Testing: it is an examining the source code to validate its accuracy.

Ex: testing a program to calculate the area of a rectangle. Testers inspect the code logic, write test cases based on formulas, and execute various scenarios, including negative values, to ensure correct results.

Traceaaability Matrix: The relationship between test cases and requirements is shown with the help of a document. This document is known as a traceability matrix.

A **Requirements Traceability Matrix (RTM)** is a document that maps and traces user requirements with test cases. [It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software development life cycle 1](https://www.guru99.com/traceability-matrix.html)

**Bug Life Cycle**

**https://www.simplilearn.com/manual-testing-interview-questions-and-answers-article**