**What is Software?**

It’s a collection of computer program that’s helps us to perform a task

**Types of Software?**

* System Software (All types of drivers like keyboard, Audio, Mouse, Operating system, servers, Utilities (Antivirus))
* Application Software (Web application, mobile application, Desktop application)
* Programming software (Complier, Debugger, Interpreter)

**What is software Testing?**

* Basically, software testing is a part of software development process.
* It’s an activity to detect and identify the defects in the software’s.
* The objective of testing is to release the quality product to the client.

**Software Quality:-**

* Meets the Requirement
* Bug free
* Within the budget.
* Delivery on time
* Maintainable

**Project and Product:-**

**Project :-** Which is developed for a single client.

**Product:-** Which is developed for a many clients.

**Defect, Bug, Failure, Error:-**

**Defect:-** it’s an error. That doesn’t allow the intended action to be completed.

**Bug:-** Bug is an error or fault in an application. This causes the application to produce unexpected results. (eg:- crashing, Functional error, Content bug etc…)

**Failure:-** An failure is an inability of a software system or component to perform its required function within the specified performance requirement. (eg:- when the intended tasks of application doesn’t work in the client end it is called Failure)

**Error:-** Incorrect Human Act. A Mistake in coding is error. Error found by tester is called defect. Defect accepted by development team then it is called bug. Build doesn’t meet the requirement is Failure.

**Why the software has a Bug?**

* Miscommunication
* Software complexity
* Programming Error
* Changing Requirement
* Lack of skill for the testers and developers

### SDLC:-

SDLC is a step by step process used by the software industry to develop and test the software.

Diagram

Description automatically generated

**Testing Methods :-**

**Black Box Testing:-**

**White Box Testing:-**

**Gray Box Testing:-**

**Types of Testing:-**

**Unit testing:-** which is done in code level by developers

**Integration Testing:-** which is done for two products(eg:- medchart and iseries)

**System Testing:-** which is also called as functional testing.(End-to-End Testing)

**User Acceptance Testing:-** Which is done in Client Side.(UAT)

**System Integration Testing:-** it is also called as SIT(functional testing, Integrated product testing)

**Sanity Testing:-**  high level check. Whether functionality is working or not. It wont have test cases.

**Smoke** **Testing:-** high level check. Whether functionality is working or not. It will have test cases.

**Regression Testing:-** Basically it conducts on modified build. To make sure there will not be any impact on existing functionality. (eg:- add feature, modifying the feature, deleting feature)

**( Unit Regression, Regional Regression, Full Regression)**

1. **Unit Regression:-**  only the changes or modifications will be tested.
2. **Regional Regression:-**  Testing the modified along with impacted module.(Note:- This testing is done only after discussing with the team(dev, testers))
3. **Full Regression:-**  Testing the main feature of the remaining part of the application. (Note:- One round of full regression)

**Security Testing:-**  Mainly whether the application is secure or not.(Note:- Payment, unwanted ads, porn related stuff, data breach, data collection)

**Exploratory** **Testing:-**

**Adhoc** **Testing:-**

**Monkey** **Testing:-**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Phases of Testing:-**

**Alpha:-**

**Beta:-**

**Unit Testing:-** White box testing

**Performance Testing:-** Load , Volume, Stress, Long Duration, Spike, Endurance.

**UAT(User Acceptance Testing):-**

**Phase of testing (DXC)**

**Module** **Testing:-**

**SIT** **Testing:-**

**UAT** **Testing:-**

**Testing Techniques:-**

**Equivalence partitioning:-**

### Boundary value analysis:-

### Decision table testing

### Error Guessing

### State Transition Technique

### Bug Life Cycle:-

### New>Assigned to developer>fixed>Awaiting alpha test>Pass(close)/Fail(alpha test fail)

### STLC:-

### FHIR(Fast Healthcare Interoperable Resource):-

### Test Plan:-

### RTM(Requirement Tracability Matrix):-

**Traceability matrix:-**

Requirement is mapped with the test case.

### RCA(Root Cause Analysis):-

### Re-Test:-

**Medchart-**

Patient details

Encounters,

MOA,

Discharge medications,

Infusions

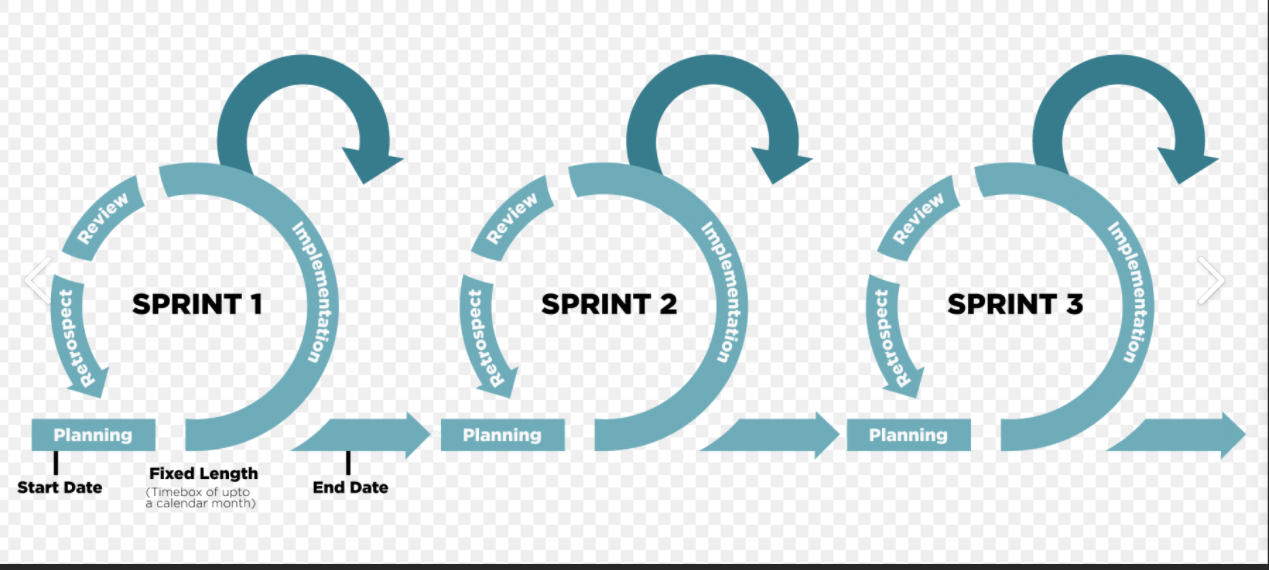
Co-sign

Messaging component

**PBI(Product Backlog item):-**  List of deliverables(items either bug or PBI).

New>inprogress>resolved>done

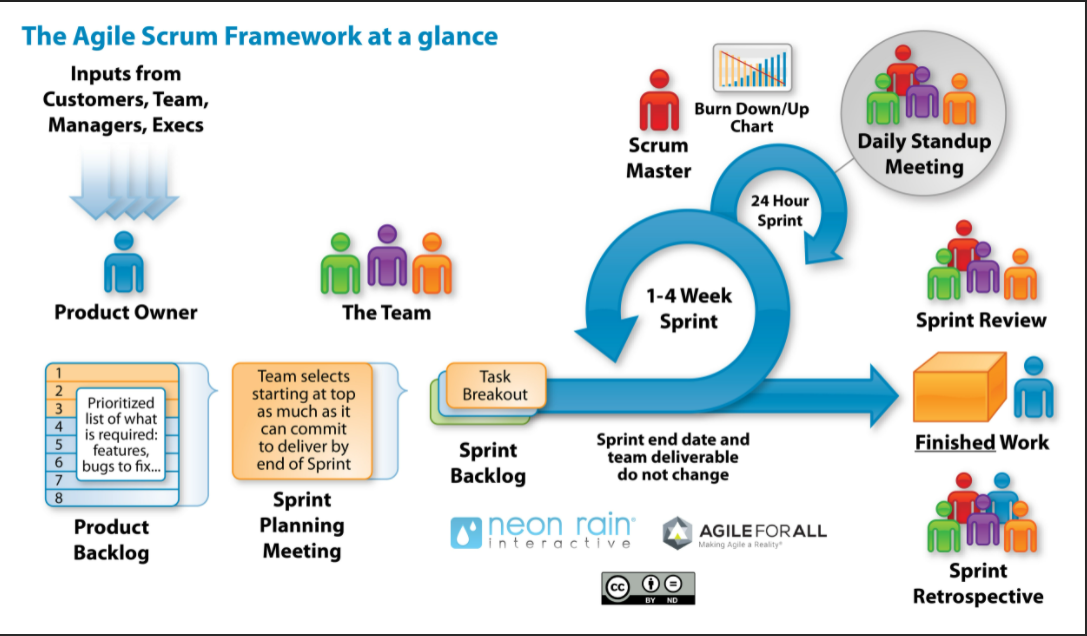
**Sprint:-** each sprint 2-3 weeks.



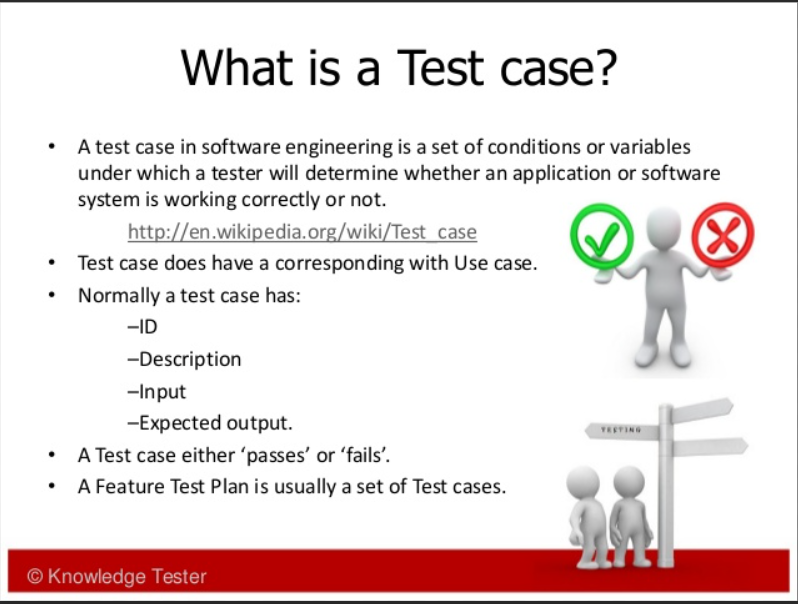
**API Testing:-**

**Get,Put,Post,Delete(200- OK, 200 -OK, 201- Created)**

**Agile:-**



**SCRUM**:-  Scrum is most often used to manage complex software and product development, using iterative and incremental practices. Scrum significantly increases productivity and reduces time to benefits relative to classic “[waterfall](https://www.cprime.com/2011/01/integrating-waterfall-and-agile-development-hybrid-model/)” processes

**Test Case:-   
**

**EPIC:-**    
ex:- under 10 epics. each epic is having 5 userstories.. **User Story:-** New Feature.

**Sprint Planning:-** ( Scrum master, Product owner, Team member(Dev, Test, Design))

**Sprint Review:-** demo.

**Retrospective:-** what went well, what went wrong, what need to improve.(or) What we have achieved in the previous sprint. What we have to achieve in future sprint or upcoming sprint(Next Sprint)

**Priority :-** How important to fix the bug(Priority 1,2,3,4()

**Severity:-** How the bug is impact the application.(Severity 1,2,3,4 (Very High, High, Medium, Low or Critical, Major, Minor))

## **Types of Severity**

In Software Testing, ***Types of Severity*** of bug/defect can be categorized into four parts :

* **Critical(Sev-1)**: This defect indicates complete shut-down of the process, nothing can proceed further
* **Major(Sev-2)**:: It is a highly severe defect and collapses the system. However, certain parts of the system remain functional
* **Medium(Sev-3)**:: It causes some undesirable behavior, but the system is still functional
* **Low(Sev-4)**:: It won’t cause any major break-down of the system

## **Priority Types**

***Types of Priority*** of bug/defect can be categorized into three parts :

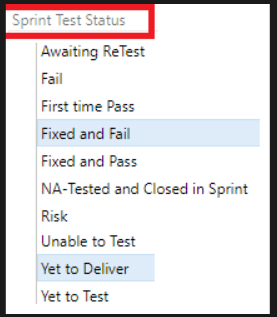
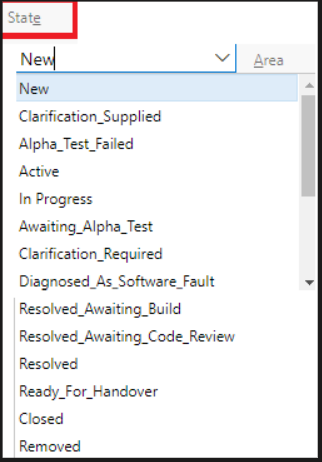
* **Low:**The Defect is an irritant but repair can be done once the more serious Defect has been fixed
* **Medium:**During the normal course of the development activities defect should be resolved. It can wait until a new version is created
* **High:**The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed

**User story status:-**

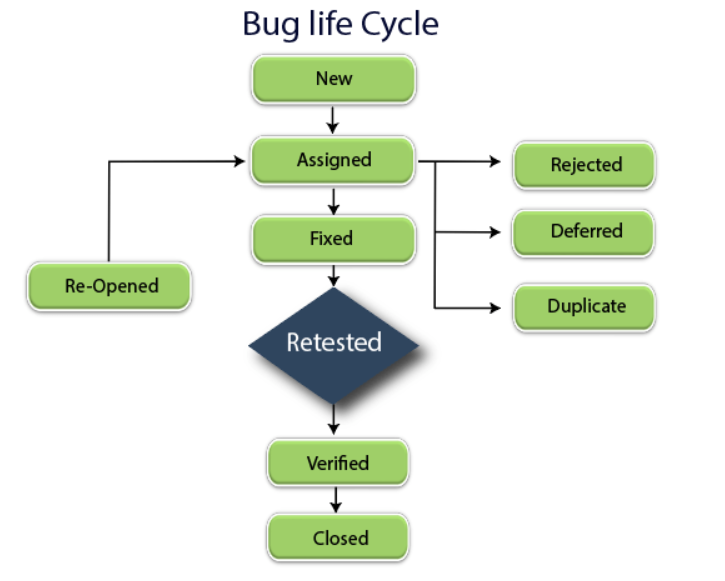
State:- New->Inprogress->Resolved->Done

Sprint Test Status:- Yet to deliver->Awaiting Build->Yet to test->First time pass(or ) Fail-> Fixed and pass(or) fixed and fail

**Bug Life Cycle:-**

**Bug life cycle in JIRA:-**



**Verification and Validation:-**

Graphical user interface

Description automatically generated

**Tester’s Accomplishment:-**

**As a tester  
1. I will try to find** at least one bug per day

2. **supposed to work under tight schedules**

**3. I am** passionate about testing job

4. Highlight If you have developed/implemented any automation frameworks or if you have provided contribution in bringing up the test coverage to maximum level.

5. **Perform Software Testing properly on time with good quality**

6. identify gaps in requirements

**Test Coverage:-**

Test Coverage is [testing every requirement](https://www.softwaretestinghelp.com/how-to-test-software-requirements-specification-srs/) at least once.

Eg:- If there are 10 requirements and 100 tests created and if 90 tests are executed then test coverage is 90%.

**Key Points:-**

Every tester should be knowledgeable of the requirements as well as the testing procedures.

**Resume:-**

# **Test Scenario**

The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

These are the high-level documents that talks about all the possible combination or multiple ways or combinations of using the application and the primary purpose of the test scenarios are to understand the overall flow of the application.

Diagram

Description automatically generated

# **User stories**

Diagram

Description automatically generated

**Challenges:-**

1. Date is getting saved one day before not able to replicate in the other region.