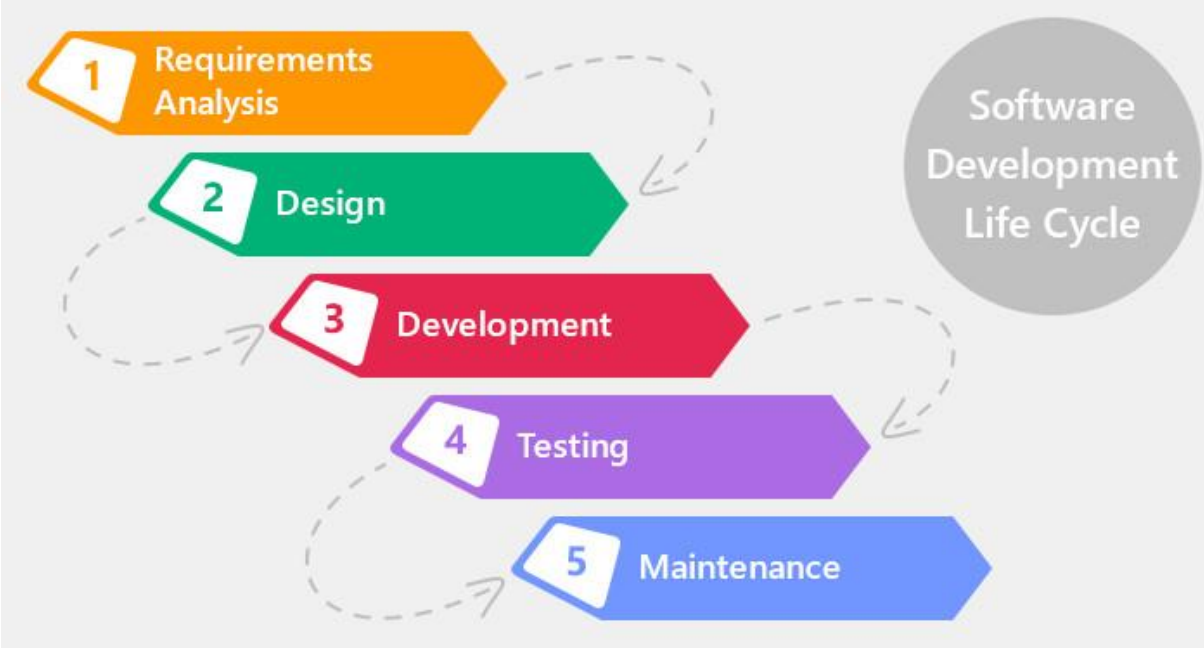
Manual Testing

- **Testing or Software Testing:** Try to find the differences between expected and Actual results, that is called Testing.

OR

comparing the actual behaviour of an application with expected behaviour.

- Project & Product: "Project is developed for a single customer on his own requirements by the software companies and the project will be used by the customer only."
  - "Product is developed for multiple customers on their consolidated requirements by the software companies and the product will be used by all customers."
- **Error, Defect, Bug, Failure:** A **mistake** in coding is called **Error**, **error** found by tester is called **Defect**, **defect** accepted by development team then it is called **Bug**, build does not meet the requirements then it Is **Failure**." ... It is the deviation (Difference) of the customer requirement.
- **SDLC (Software Development Life Cycle):** These phases may vary form one organization to another, but purpose is almost all same, that is "Develop and Maintain Quality Software". Below are the standard phases of SDLC:



SDLC Models: We are going to discuss about three types of models as below:

1. **Waterfall Model Oldest Model)**
2. **V Model**
3. **Agile Methodologies**

- **Waterfall Model Oldest Model):** Waterfall approach was first SDLC Model to be used widely in market. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model the outcome of one phase acts as the input for the next phase release sequentially.

  It is very simple to understand and use. This type of model is basically used for the project which is small and there are no uncertain requirements.

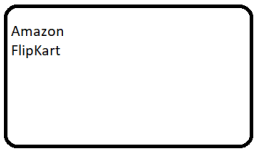
Note: Once requirements are finalized then we cannot change the requirements in Waterfall model. Means requirements are static.

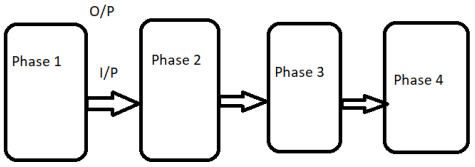
Software Testing/
TESTING Try to find the difference between expected and actual result that is called Testing.

Error: Any incorrect Human Mistake.
Defect: When that mistake is identify by the testing team/ tester that is called defect.
Bug: When the defect is accepted by the dev team then it is called Bug.
Failure: When the defect is missed by teasting team to detect. And issue identified by the Customer then it is called. Failure.

SDLC Models:
1. WaterFall Model
2. V Model
3. Agile Methodologies (We will study later)

1. WaterFall Model





1. How many days needs to develop?
2. Costing
3. Resource Management

6 months
3 to 4 months (300 Bugs)

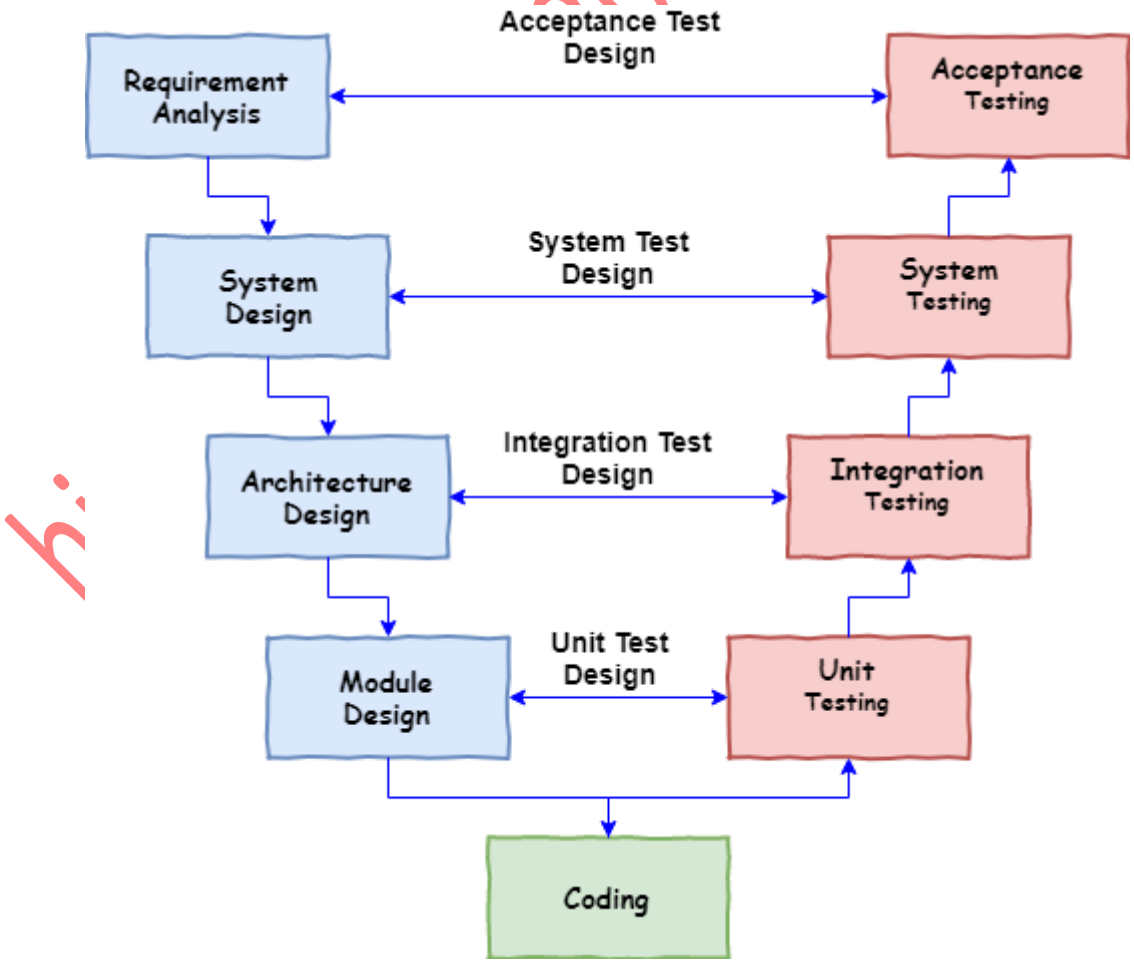
Disadvantage: Re-Work
2. Time increase
Requirements are Static

MAintainence Projects

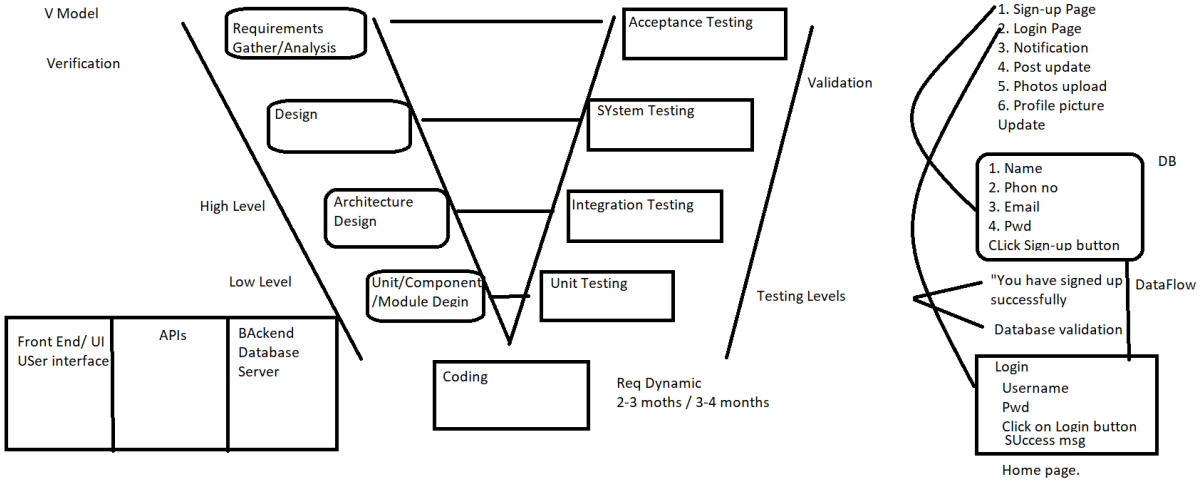
Dev team

V-Model: It is Verification & Validation model, known as V Model, in this model all development phases can be integrated with Testing phases.

The V-model tells how testing activities can be integrated into each phase of the software development life cycle.



The V- model should be used for small to medium sized projects where requirements are clearly defined and fixed.



❖ **Testing Methodologies:**

Black Box Testing: In black-box testing the tester is concentrating on what the software does, not how it does it. Testers have no knowledge of how the system or component is structured inside the box. Specification-based testing technique is also known as „black-box "or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

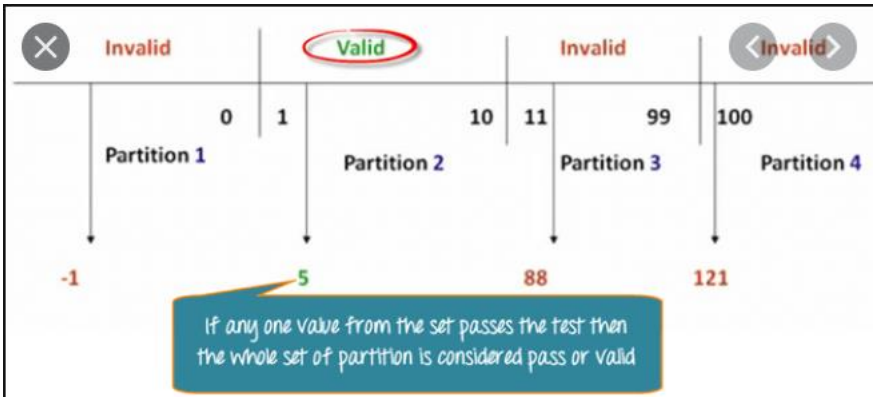
Black box testing covers both functional and non-functional testing. Functional testing is concerned with what the system does its features or functions. Non-functional testing is concerned with examining how well the system does. Non-functional testing like performance, usability, portability, maintainability, etc.

Below are the black box testing techniques: -

Equivalence partitioning

Boundary value analysis

1. **Equivalence partitioning or equivalence class partitioning (ECP):** It is a software testing technique that divides the input data of a application unit into **partitions** of **equivalent** data from which test cases can be derived. So, in the example below we will test for any value from the valid and invalid classes. So, we will consider same result for all the values from the Valid/Invalid partition.



2. **Boundary Value Analysis (BVA):** Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values. So in the above example boundary values will be 0,1,10,11,99,100

- **White box testing:**

"Structure-based testing technique is also known as „white-box" or „glass-box" testing technique because here testers require knowledge of how the software is implemented, how it works "

Developers use -structure based technique in component testing and component integration testing, especially where there is good tool support for code coverage."

- **Grey Box testing:**

Grey Box testing is a technique to test an application with to limited test knowledge of the internal working of an application."

Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has access to design documents and the database. Having this knowledge, the tester can better prepare test data and test scenarios.

- **Software Testing Levels:** There are mainly 4 testing levels as below:
- I) Unit Testing (Dev team) ii) Integration Testing (Dev team) iii) System Testing iv) Acceptance Testing
- **1. Unit Testing:** When we test single module or a piece of code independently that is UNIT testing. The purpose of unit testing is to validate that each unit of the application works as designed. Unit testing is done manually as well as automated. Generally, dev team perform unit testing. There are many tools to perform unit testing like Junit, Nunit, TestNG etc. and dev team used these tools to perform unit testing.
- 2. **Integration Testing:** When multiple modules or units are developed and tested as a group. That is called integration testing. The purpose of integration testing is to find out the defects while interaction between these integrated modules or Units. There are several tools to perform integration testing like protractor, Jasmine etc. Dev team performs integration testing.
  Ex: Application has 3 modules say 'Login Page', 'Mailbox' and 'Delete emails' and each of them is integrated logically. So here these modules already tested in Unit testing lets focus on when Login, Mailbox and Delete emails modules has been integrated. Below are some high-level integration test cases:

Test Case ID	Test Case Objective	Test Case Description	Expected Result
1	Check the interface link between the Login and Mailbox module	Enter login credentials and click on the Login button	To be directed to the Mail Box
2	Check the interface link between the Mailbox and Delete Mails Module	From Mailbox select the email and click a delete button	Selected email should appear in the Deleted/Trash folder

Types of Integration testing: There are 4 types of integration testing are below:

shubhangi3090@gmail.com

1. **Big Bang**
2. **Incremental Approach:**
   a. **Top down**
   b. **Bottom up**
   c. **Sandwich/Hybrid**

Big Bang Approach: This is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a complete unit. This combined set of components is considered as an entity while testing. If all of the components in the unit are not completed, the integration process will not execute.

Incremental Approach: In the **Incremental Testing** approach, testing is done by integrating two or more modules that are logically related to each other and then tested for proper functioning of the application. Then the other related modules are integrated incrementally and the process will continue until all the logically related modules are integrated and tested successfully.

In this approach we have two different methods:

Bottom up

Top down

Stubs and Drivers

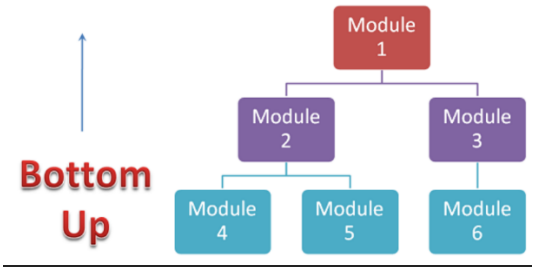
Stubs and Drivers refers to replica/dummy of the modules, which acts as a substitute to the undeveloped or missing modules. Stubs & drivers are specifically developed to meet the requirements of undeveloped modules and are immensely useful in getting expected results.

Stub: A stub is called from the software component to be tested.

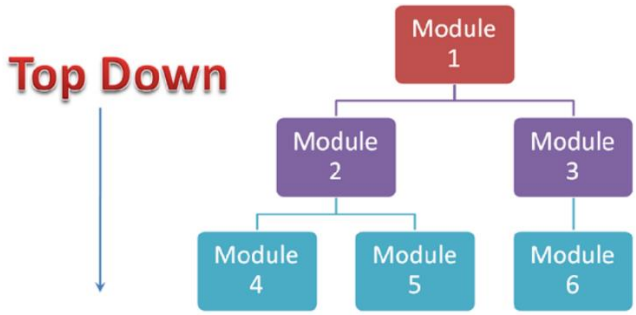
Driver: A driver calls the component to be tested.

Bottom-up Integration Testing: Bottom-up Integration Testing is a strategy in which the lower-level modules are tested first. These tested modules are further used to perform the testing of higher-level modules. The process will continue until all the modules at top level are tested.

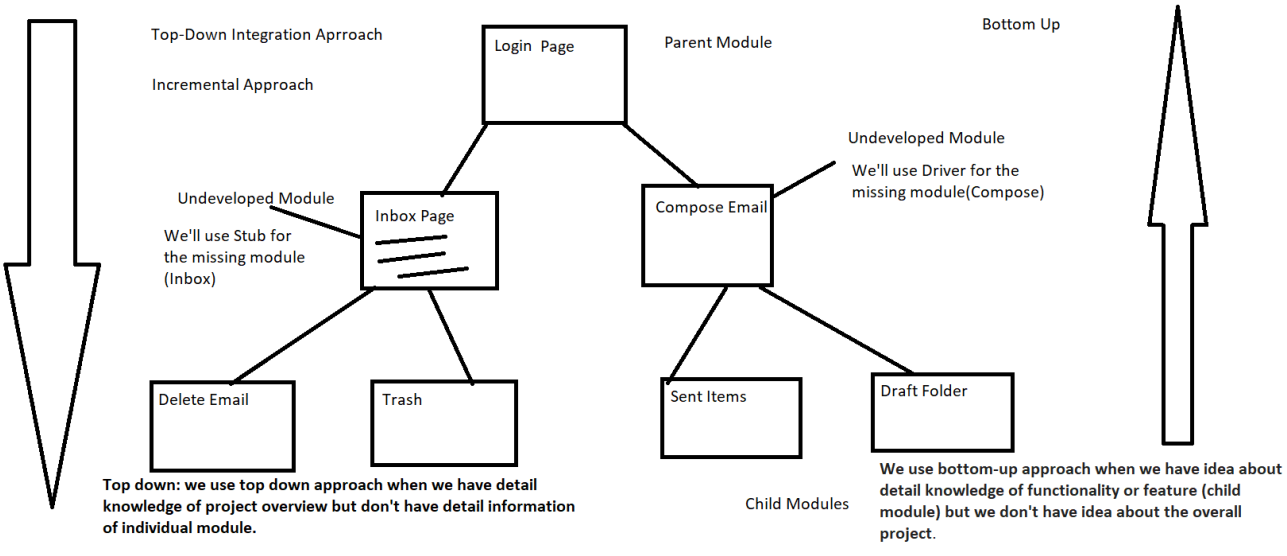
"**We use bottom-up approach when we have idea about detail knowledge of functionality or feature but we don't have idea about the overall project**."



Top-Down Integration Testing: It is a method in which integration testing starts from top to bottom. The higher-level modules are tested first and then lower-level modules are tested and integrated in order to check the application functionality. Basically, Stubs are used for testing if some modules are not ready.



Advantage and disadvantage: Critical modules are test first on priority and disadvantage is we may need many stubs to test in this approach.



Sandwich Testing: It is a strategy in which top level modules are tested with lower-level modules at the same time lower modules are integrated with top modules and tested as a system. It is a combination of Top-down and Bottom-up approaches therefore it is called **Hybrid Integration Testing**. It makes use of both stubs as well as drivers.

System Testing: System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

Example: If an application has three modules A, B, and C, then testing done by combining the modules A & B or module B & C or module A& C is known as Integration testing. Integrating all the three modules and testing it as a complete system is termed as System testing. It is just an example let suppose there are around 20 modules and then after performing integration testing we are doing system testing for all 20 modules.

Example 2: *A car manufacturer does not produce the car as a whole car. Each component of the car is manufactured separately, like seats, steering, mirror, break, cable, engine, car frame, wheels etc. After manufacturing each item, it is tested independently whether it is working the way it is supposed to work and that is called Unit testing.*
Now, when each part is assembled with another part, that assembled combination is checked if assembling has not produced any side effect to the functionality of each component and whether both components are working together as expected and that is called integration testing.
Once all the parts are assembled and the car is ready, it is not ready actually.
The whole car needs to be checked for different aspects as per the requirements defined like if car can be driven smoothly, breaks, gears, and other functionality working properly, car does not show any sign of tiredness after being driven for 2500 miles continuously, color of car is generally accepted and liked, car can be driven on any kind of roads like smooth and rough, sloppy and straight, etc and this whole effort of testing is called System Testing and it has nothing to do with integration testing.
So in our application we also needs to check different aspects like, Installation of application, Performance of application, Usability of application, responsiveness etc.

These are some focus areas for System testing as below:

1. External interfaces
2. Multiprogram and complex functionalities
3. Security
4. Recovery
5. Performance
6. Operator and user's smooth interaction with the system
7. Installation
8. Documentation
9. Usability
10. Load/Stress

Acceptance Testing: Acceptance testing is when the application has met the customer requirements or not. The main purpose of acceptance testing is checking the system compliances with the business requirements. And verify that application met the required criteria for delivery to end user.

There are various forms of acceptance testing:

- User acceptance Testing **(UAT)**
- Alpha Testing
- Beta Testing

User Acceptance Testing (UAT): It is a type of testing performed by the end user or the client to verify/accept the application/Release before moving the code to the production environment. UAT is done in the final phase of testing after functional, integration and system testing are done.

Alpha Testing: Alpha Testing is a type of acceptance testing; performed to identify all possible issues and bugs before releasing the final product to the end users. Alpha testing is carried out by the testers who are internal employees of the organization.

Beta Testing: Beta Testing is performed by "real users" of the software application in "real environment" and it can be considered as a form of external User acceptance testing It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer's environment.

Types of Software Testing: There are 2 types of software testing are below:

- ❖ Functional testing
- ❖ Non-functional testing

1. **Functional Testing**: This is normal testing which we perform by analysing the FR (Functional requirements) or BR (Business Requirements). FR and BR may be described in the form of User Stories.

Functional Testing is a testing technique that is used to test the features/functionality of the Software".

In functional testing we include Smoke, Sanity, Regression, Re-testing, End to End testing.

2. **Non-Functional Testing:** Non-functional testing is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of an application. These are the types of non-Functional testing.

User Interface Testing: "Graphical User Interface (GUI) testing is checking the application design of an application". Ex: Required/Optional, Fields Align, Lengths, Progress Bars, Scroll Bars, Alignments, etc.

Usability Testing: "In usability testing basically the testers test the ease with which the user interfaces can be used. It tests that whether the application is user-friendly or not.

"Usability Testing tests the following features of the software.

– How easy it is to use the software.

– How easy it is to learn the software.

– How convenient is the software to end user.

Stress Testing: "It is a form of testing that is used to determine the stability of a given system, Stress testing involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. "Stress testing is a generic term used to describe the process of putting a system through stress.

Load Testing: "Load testing is performed to determine a system's behaviour under both normal and at peak conditions. "A load test is usually conducted to understand the behaviour of the application

himanshusingh309@gmail.com

under a specific expected load. E.g. If the number of users are increased then how much CPU, memory will be consumed, what is the network and bandwidth response time.

Performance Testing: "Performance testing is testing that is performed, to determine how fast some aspect of a system performs under a particular workload. "It can serve different purposes like it can demonstrate that the system meets performance criteria.

Localization Testing: "Localization translates the product UI and occasionally changes some initial settings to make it suitable for another region." Localization testing checks the quality of a product's localization for a particular target culture/locale. The test effort during localization testing focuses on: - Areas affected by localization, such as UI and content - Culture/locale-specific, language-specific, and region-specific areas

Globalization Testing: "Globalization Testing is testing process to check whether software can perform properly in any locale or culture & functioning properly with all types of international inputs and steps to effectively make your product truly global."

This type of testing validates whether the application is capable for using all over the world and to check whether the input accepts all the language texts. Ex: Let's see another example of a Zip code field in Sign up form: - For globalized, it should allow to enter alphanumeric inputs - For localized (country like INDIA), it should allow only numbers in input field.

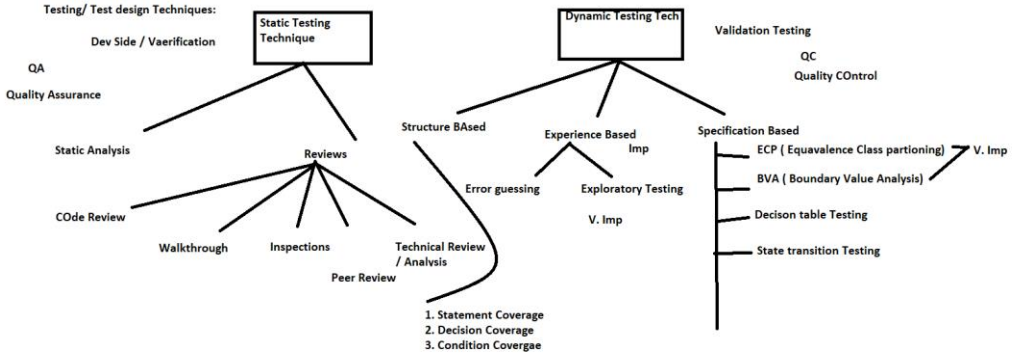
Security Testing: "Security testing is basically to check that whether the application or the product is secured or not. "

Compatibility Testing:" Compatibility Testing ensure compatibility of the application built with various other objects such as other web browsers, hardware platforms, operating systems etc." This type of testing helps find out how well a system performs in a particular environment that includes hardware, network, operating system, and other software etc.

Ex: **Browser Compatibility Testing**, **OS Compatibility Testing**

- **Change-related Testing (CR):** When changes are made to the existing functionality, either to correct a defect or because of new or changing functionality, testing should be done to confirm that the changes have corrected the defect or implemented the functionality correctly and are not impacting to other functionalities.
- **Software Testing Techniques: There are 2 types of test design techniques:**

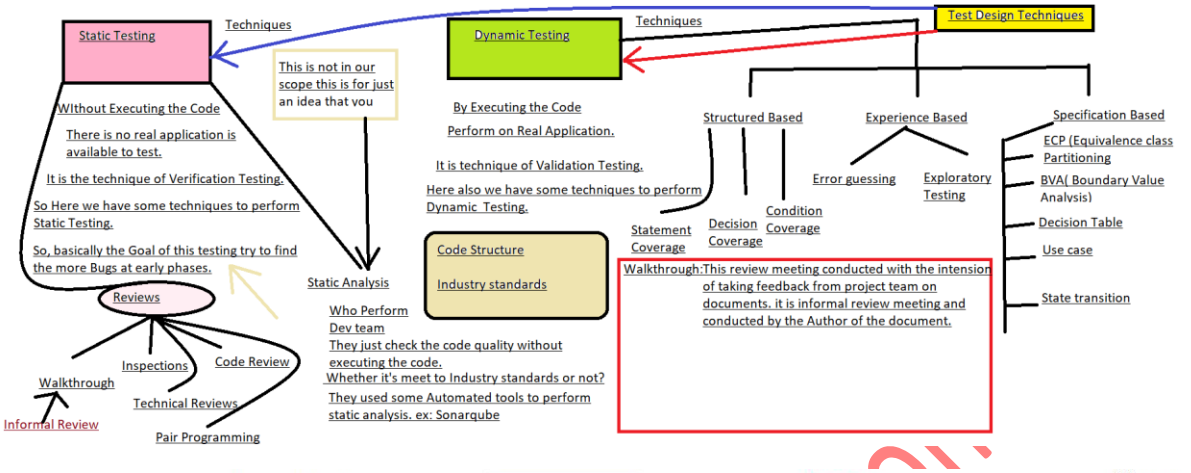
1. **Static Testing Technique** **2. Dynamic Testing Technique**

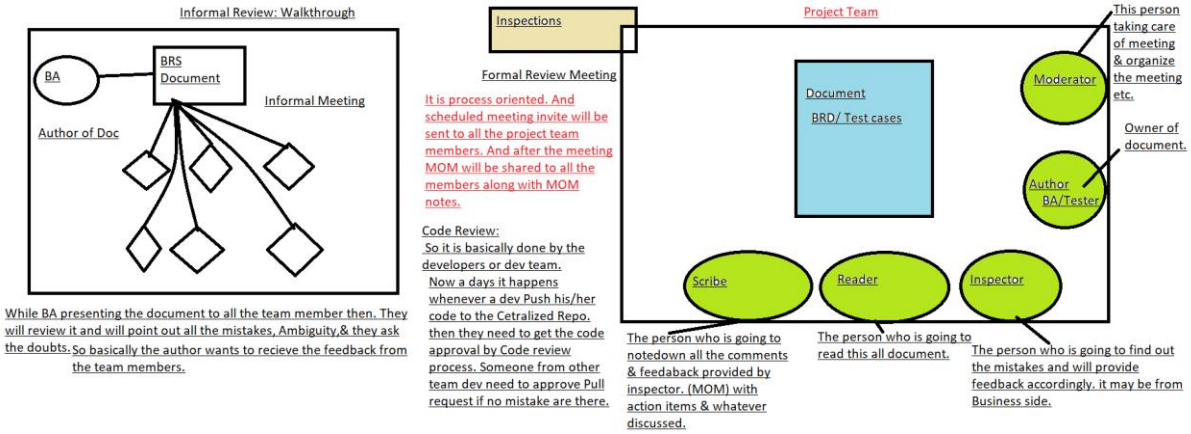


2.

hirenshusingh309@gmail.com

3.





4.

Peer Reviews: When two dev sitting at single workstation. And one dev is writing code and other dev is reviewing the code parallay. this process is known as Peer review.

TEchnical Review: An expert person identify the problems in the code & Document. this process is known as Technical review. Or sometimes Solution Architect provide Solution itself to dev team while working on a User Story req. So that Dev implement the solution in the same way SA explained or provided tech analysis in user story to implemet the code.

Experience Based Testing:

1. Error guessing: It is a experienced based testing technique in which a experienced person try to guess the problem. And where the Test Analyst uses his/her experience to guess the problematic areas of the application. This technique necessarily requires skilled and experienced testers.

2. Exploratory Testing: EXPLORATORY TESTING is a type of software testing where Test cases are not created in advance, but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is to explore the as many things they can explore in the application.

S.N.	Static Testing (Verification)	Dynamic Testing (Validation)
1.	Before doing the Whitebox/Unit testing done by the dev team before what an all activities we will do that comes under static testing . Whenever any req comes they will do req analysis, design, coding once after coding is done dev will perform Unit testing .	Whenever Dev is giving a build to the testing team and after getting builder what testing team will do that is Dynamic Testing. They will execute their test cases, they will do functional, Integration, System, UAT testing etc.
2.	Static testing is done as part of your verification activity.	Dynamic Testing is done as part of your activity.
3.	We review the code, review the requirement, review client req specification , SRS, High level design, Low level design, walkthrough of the code.	We will do component testing, Integration testing, system testing, acceptance testing , all your functional testing, all non-functional testing will be part of dynamic testing.
4.	We do static testing before the code is deployed.	We do dynamic testing once code is deployed.
5.	We do Static testing to prevent the defects.	We do dynamic testing to detect/catch the defects. And fix them.
6.	It is perfomed at the early stage of the development.	It is perform at the later stage of the development.
7.	The purpose of static testing is to prevent the defects as much as at early stages.	The purpose of dynamic testing is to find out the defects. More bugs should not come in dynamic testing.

Software Testing Life Cycle: There are different phases in testing life cycle which we follow:

a. Requirement Analysis

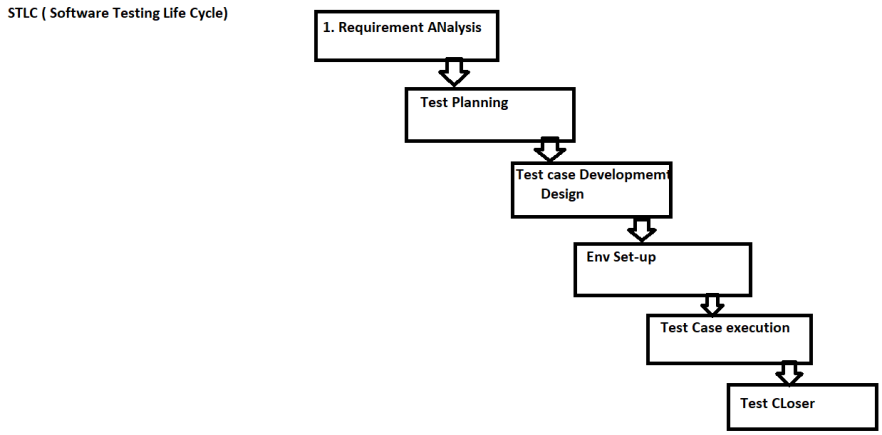
b. Test Planning

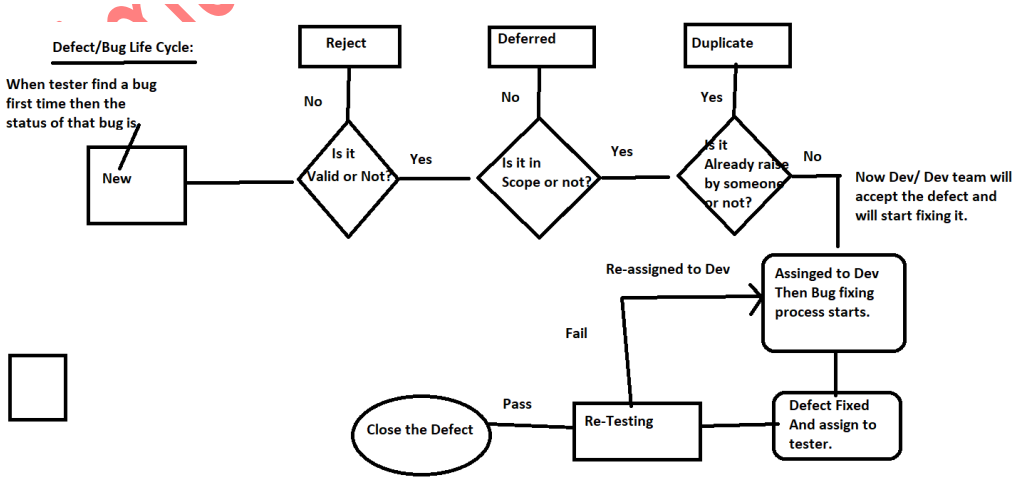
c. Test case design and development

d. Test Environment setup

e. Test case execution

f. Test closer





When we get a defect first time then it is in new state. After that we basically check 3 things like is it valid or not? If it is not valid then Dev team will reject the bug. If it is valid then again, we will check it is in scope or not? If is not in scope then we will mark it as deferred. If it is in scope then again, we will check last thing that is it already raised by someone or not? If it already raised by someone then we will mark it as duplicate if not raised by someone then finally we will assign the defect to the developer or someone from dev team who is working on the story. Now bug fixing process will start. Once defect is fixed then we do Re-testing on the defect if re-testing fails then again, we will re-assign that defect to the developer if Re-testing pass then we will close the defect.

Priority: It indicates to the scheduling. means How soon a bug should be fixed.

Severity: It indicates to the functionality, means seriousness of the defect on the product functionality.

High severity/ High priority: Let suppose there is Bank ATM machine. And a person is doing ATM transaction for the same bank for which he is holding the debit card. But He is getting charged 20 rupees/transaction. Which is against the bank policy. SO High severity and High priority will be here because these are lot of transactions are happening within an hour.

High severity/Low priority: Let suppose there is a banking application. and bank is giving 2 rupees/1000 rupees as interest to their customers. but bank found that there is a bug and due to that bug bank is giving 4 rupees/1000 rupees to their customers at the end day of the last month of the year. So due to the bug interest is going double and it cost very high to the bank. SO severity will be high and priority will be low because it happens last day of the year and next day is 1st day of the next year so we have lot of time to fix the issue.

Low severity/High priority: Let suppose there is a mistake in company Logo. so Severity will be low because there is no functionalities impact on the application and priority will be high because it create the question mark on the company image.

Low severity/Low priority: Let suppose there is just spelling mistake inside the application or you can say any little cosmetic UI Issue.

Difference between Test case and Test scenario: **Test case** consist of a set of input values, execution precondition, expected results and executed post condition, developed to cover certain **test** condition. While **Test scenario** is nothing but a **test** procedure. ... The **scenarios** are derived from used **cases**

A **Test Scenario** is any functionality that can be **tested**. It is also called **Test** Condition or **Test** Possibility. **Test** scenario can be a single or a group of **test** cases.

Test Case sample:

	TC_ID	TC_Name	Pre-Condition	Test Data	Test Steps	Expected Results
2	TC_01	Verify login feature without entering any data	User is on login page	URL	1.Click on the login button without entering any data in user name & Pwd field	1. User should get the validation error message " Username or password are incorrect." 2. User should not redirect to the Home/Inbox page.
3	TC_02	Verify login feature with entering valid data	User is on login page	URL User ID: abc@gmail.com PWd: xyz@123	1. Enter validation username in "Userid" input/text field	1. User should able to enter user name.
4					2. Enter valid password in "password" field	2. User should able to enter password.
5					3. click on the login button	3. User should redirect to Home/Inbox Page

	TC_ID	TC_Name	Pre-Condition	Test Data	Test Steps	Expected Results
6	TC_03	Verify the user upload the profile picture of his/her facebook account	1. User has logged in	Photo/pictures with proper size	1. Navigate to the User account/home page	User should redirect to the account page without errors.
7					2. Click on the profile photo	User should get the list of action items.
8					3. Choose the "Picture upload" option	User should get two options below: 1. From Galary 2. From Camera
9					4. Click on the "From Galary"	User should redirect to the Galary of his/her phone
10					5. Select the picture which you want to upload	User should get the option to crop the photo
11					6.Crop the image as per your need	Croped picture should get display.
12					7. Now click on the " Upload Photo" button	Progress report should get display.
13					8. validate the uploaded photo	1. Uploaded image should get uploaded. 2. Picture should get updated on user's account page. 3.Image should get updated in News feed sction. 4. Image should get updated in Photos section.

We can prepare the test scenarios for following list of items:

ATM
PEN
LIFT
SIGN-UP PAGE
WHATSAPP
FACEBOOK AFTER LOGIN
CAPTCHA CODE
CHAIR
BLUETOOTH (V.IMP)
FAN
COFEE MACHINE
DATE FIELD
FLIGHT RESERVATION, HOTEL RESERVATION

n309@gmail.com

	A	B
1	Requirements	Test Scenario
2	As a customer I should able to login into the application.	1. Verify the Login functionality with valid credentials
3		2. Verify the login functionality with Invalid credentials.
4	As a customer I should able to Re-set password by clicking on forgot Password button.	Please write Test scenarios
5		
6	As a customer I should able to select the Men's wear items from the Product section. And I can see the products category wise. I can choose the products and should able to add them into "Add to Cart" section. Then I should able to proceed with the Payment. Once the Payment is successfully done, Then I should get the message on my registered mobile number & on my registered email address with all the details.	
7	Test Scenarios, Then convert them into Test cases (Positive & Negative)	

Entry Criteria: Entry Criteria gives the Pre-condition items that must be completed.

Exit Criteria: Exit Criteria is defining the items that must be completed before giving the testing closer.

Test Planning: In this phase the Test Manager or Test Lead prepares the Test Plan and Test strategy documents.

Activities in this phase:

Test plan ID

Test environment
Features to be tested/Not tested
Entry/Exit criteria
Status
Types of testing

Brief Intro

Strategy defines what approach should be there for testing and Test plan has all the details how those approaches will be executed in a proper planned way. They both go hand in hand.

Test plan Vs Test Strategy: Generally, it doesn't matter which comes first. Test planning document is a combination of strategy plugged with overall project plan. According to IEEE Standard 829-2008, strategy plan is a sub item of test plan.

Every organization has their own standards and processes to maintain these documents. Some organizations include strategy details in test plan itself. Some organizations list strategy as a subsection in testing plan but details is separated out in different test strategy document.

Ex: Test plan gives the information of who is going to test at what time. For example: Module 1 is going to be tested by "X tester". If tester Y replaces X for some reason, the test plan must be updated.

On the contrary, test strategy is going to have details like – "Individual modules are to be tested by test team members. "In this case, it does not matter who is testing it- so it's generic and the change in the team member does not have to be updated, keeping it static. Deliverables: Test Plan with estimation

Test strategy contains:

a. **Scope and objective**
b. **Business Issues:**
c. **Testing approach:** What type of testing is needed?
d. **Defect tracking approach**:
e. **Risks**:

How many environments do we have?
"A Typical project can have following environments"
- Dev
- QA
- Pre-Production
- Production

Note: On high level we can have above 4 environments in any company. But it may depend upon company to company.

Explain what is Test Metric is software testing and what information does it contains?

Ans: In software testing, Test Metric is referred to standard of test measurement. They are the statistics narrating the structure or content of a program. It contains information like

- Total test
- Test run

sh309@gmail.com

- Test passed
- Test failed
- Tests deferred
- Test passed the first time
- How many defects are existed within the module?
- How many test cases are executed per person?
- What is the Test coverage %?

❖ **Test Data:** "In order to test a software application you need to enter some data for testing most of the features. Any such specifically identified data which is used while test execution is known as test data."

❖ **Defect Slippage Ratio**: Number of defects slipped (reported from production) v/s number of defects reported during execution.

Example: Customer filed defects are 15, total defect found while testing are 150, total number of invalid defects are 10. So, Slippage Ratio is [15/ (150-10)] X 100 = 10.71%

❖ **DRE (Defect Removal Efficiency)**: The defect removal efficiency (DRE) gives a measure of the development team ability to remove defects prior to release. It is calculated as a ratio of defects resolved to total number of defects found. It is typically measured prior and at the moment of release.

For example, suppose that 100 defects were found during QA/testing stage and 84 defects were resolved by the development team at the moment of measurement. The DRE would be calculated as 84 divided by 100 = .84*100 →84%

❖ **Mock ups**: In Design phase or for requirement phase, some sample screens replica of actual application will be provided for team.

❖ **Release notes:** Release notes is a document which will be prepared by the dev team during the release time. It will be delivered to the customers which contains the technical information about changes addressed in the current release.

❖ **Test Summary Report: Sample format is below:**

Test cases planned	Test cases executed	TCs Pass	Tcs Failed
80	75	70	5

	Registration	Booking	Payment	Reports	Total
Critical	6	7	5	7	25
Major	4	5	2	4	15
Medium	6	8	2	4	20
Cosmetic	1	2	1	1	5
Total-->	17	22	10	16	65

❖ **Status Call**: Status call will happen daily / weekly / monthly based on client/company rules where we will discuss about project status.

chusingh309@gmail.com

- **MOM(Minutes of Meeting)**: Minutes of Meeting are the written or recorded documentation that is used to inform attendees and non-attendees of the happenings during the meeting.

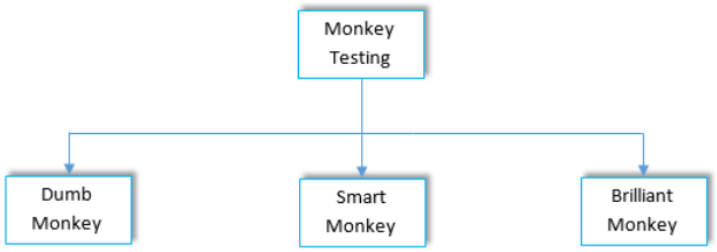
MOM Contains: The names of the participants, the agenda items covered, decisions made by the participants, the follow-up actions committed to by participants, due dates for the completion of commitments, and any other events or discussions worth documenting for future review or history.

- **Test coverage**: It is defined as a technique which determines whether our test cases are covering the application code and how much code is exercised when we run those test cases.
  If there are 10 requirements and 100 tests created and if 90 tests are executed, What is Use Case Testing? Ans. Validating a Software to confirm whether it is developed as per the use cases or not is called use case testing.
- **Defect Age**: The time gap between date of detection & date of closure of a defect.
- **Showstopper Defect**: A defect which is not permitting to continue further testing is called Showstopper Defect
- **Test Closure**: It is the last phase of the STLC, Where the management prepares various test summary reports that explains the complete statistics of the project based on the testing carried out then test coverage is 90%. Now, based on this metric, testers can create additional test cases for remaining tests.
- **Test Harness**:Test Harness is configuring a set of tools and test data to test an application in various conditions, which involves monitoring the output with expected output for correctness.
- **Fuzz Testing:** Fuzz testing is a black box testing technique which uses a random bad data to attack a program to check if anything breaks in the application.
- **Gorilla Testing:** Gorilla Testing, **a technique in which repetitive Manual Testing process, which a tester would have done several times before, is done again to test the robustness of the system**.

  A module can be tested over a hundred times, and in the same manner. So, Gorilla Testing is also known as “Frustrating Testing”.
- **Monkey Testing**: Monkey testing is a technique in software testing where the user tests the application by providing random inputs and checking the behaviour (or trying to crash the application). Mostly this technique is done automatically where the user enters any random invalid inputs and checks the behaviour.

Types of Monkey Testing:

Monkey Testing is further divided into several categories according to its way of implementation, See the following diagram for a quick idea of it;



1. **Dumb Monkey:** Testers have no idea about the system and its functionality, also no assurance about the validity of test case.
2. **Smart Monkey:** Tester has a precise idea about system its purpose and functionality. Tester navigates through the system and gives valid inputs to perform testing.
3. **Brilliant Monkey**: Testers perform testing as per user's behavior and can specify some probabilities of bugs to have occurred.

❖ Main Roles in Agile:

Product Owner	Scrum Master	The Team
◦ He defines features of the product.	◦ He manages the team and look after the team's productivity	◦ The team is usually about 5-9 members
◦ Product Owner decides release date and corresponding features	◦ He maintains the block list and removes barriers in the development	◦ It includes developers, designer and sometimes testers, etc.
◦ They prioritize the features according to the market value and profitability of the product	◦ He/She co-ordinates with all roles and functions	◦ The team organizes and schedule their work on their own
◦ He is responsible for the profitability of the product	◦ He/She shields team from external interferences	◦ Has right to do everything within the boundaries of the project to meet the sprint goal
◦ He can accept or reject work item result	◦ Invites to the daily scrum, sprint review and planning meetings	◦ Actively participate in daily ceremonies

What is Ad Hoc testing?

Ans: It is a testing phase where the tester tries to break the system by randomly trying the system's functionality. It include negative testing as well.

What is bug leakage and bug release?

Ans: Bug release is when software or an application is handed over to the testing team knowing that the defect is present in a release. During this the priority and severity of bug is low, as bug can be removed before the final handover.

Bug leakage is something, when the bug is discovered by the end users or customer, and missed by the testing team to detect, while testing the software.

RTM: It is a document in which we Map and trace our requirements with the Test cases.

TestCase #	Test Case	Test Steps	Test Data	Expected Result
1	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

When correct password and id entered, it should login successfully

T94 If userid and password are valid. Login

T94 is our technical requirement that verifies successful login

TestCase #	TR #	Test Case	Test Steps	Test Data	Expected
1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Note the Technical Requirement in the test case

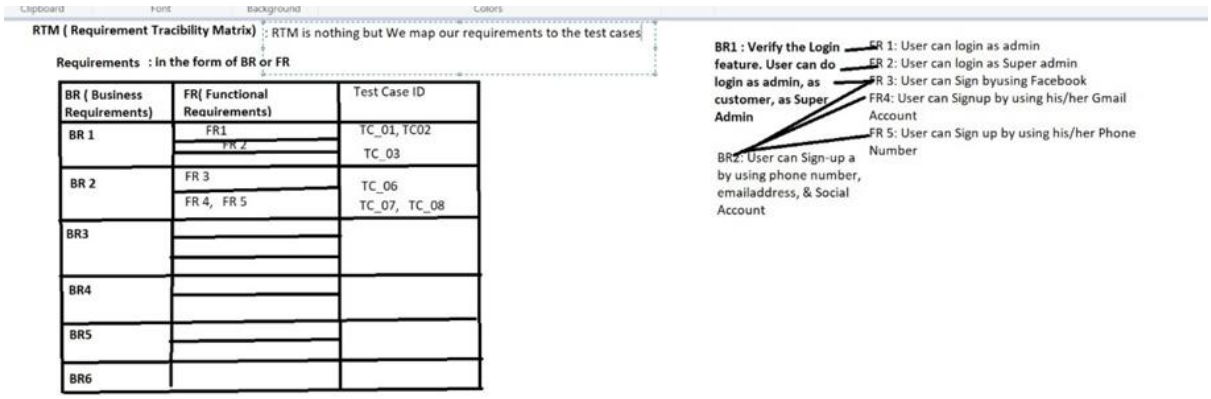
BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	**Customer:** A customer can login using the login page **Manager:** A manager can login using the login page of customer. Post Login homepage will show different links based on role

Identify the Business Requirement for which T94 is defined

TestCase #	BR #	TR #	Test Case	Test Steps	Test Data	Expe
1	B1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Business Requirement #	Technical Requirement #	Test Case ID
B1	T94	1
B2	T95	3
B3	T96	3
B4	T97	4

Requirement Traceability Matrix



Smoke Testing:

When we get a new software build first time from dev team then we perform Smoke testing. Actually, we do smoke testing on un-stable build. in this we have fixed set of test cases, if all test cases are passed then we accept the build and do further normal testing. If any of test case is getting fail then we reject the build and send it back to the development team. That is called Smoke testing.

Sanity Testing: We do sanity on stable build. when a build has clear multiple rounds of regression testing then we perform Sanity testing. Let suppose We raised a bug and there is new minor functionality is being added and there is some integration is also happening. Now first we will do re-testing for the raised defects and then we will test that new added functionality. Now we will perform a Sanity check on that build to test the impacts due to the bug fixing or integration things. means due to the changes or integration other major functionalities are working fine or not as earlier. That is called Sanity check. In this also we have fixed set of test cases.

Regression Testing: "We do test to confirm that a recent code change has not affected the existing feature functionality. That is regression testing. Let say you got an under story to perform testing and you did analysis on that story and you found that this feature is making impact on other integrated module or any existing functionality as well so in that case we'll write test cases for our new functionality which is going to be implemented based on the user story. And apart from that we'll include some more already written test cases for regression testing to test the existing functionality which may be impacted due to new changes. And we'll execute them once code is deployed in QA env. So, regression testing is nothing, but it is full or partial selection of already executed test cases which are re-executed to ensure that existing functionality is working fine or not.

And, we perform regression testing before each release.

- **Defect clustering:** When a small number of modules contains most of the bugs detected or show the most operational failures. Pesticide Paradox: If the same **tests** are repeated over and over again, eventually the same **test** cases will no longer find new bugs.

Pesticide paradox: Let's say you are testing an application. You have written a set of test cases.

Now you run one cycle of testing. You find few bugs and report them to the development team. Development team fixes the bugs and reverts to you with the updated code. You again execute the same set of test cases. This time you find that few of the bug were still not fixed and you report that back to the development team. They work on it and send an update to you. Once again, you execute the same set of test cases and don't find any bugs.

Now in a new release some changes were made in the application. You run the same set of test cases and they all pass. But what you miss here is the new bugs that may have introduced when the fix and new changes were applied. The old set of test cases are incapable of identifying these new bugs.

This is called **Pesticide Paradox**. To avoid this, you need to update your test cases with each cycle and add new cases to the old set.

Defect Clustering:

When a small number of modules contains most of the bugs detected or show the most operational failures.

Pesticide Paradox:

If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs. This is what pesticide paradox is referring to.

Defect Triage: Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc. Triage term is used in the Software testing / QA to define the severity and priority of new defects. The goal of Bug Triage is to evaluate, prioritize and assign the resolution of defects. The team needs to validate severities of the defect, make changes as per need, finalize resolution of the defects. Here, are some important factors that decide the frequency of Defect Triage Meetings:

These Important factors are:

- As per the project schedule
- Number of defects in the system
- Impact on schedules of team members' availability
- Overall project health

Who are the mandatory and other participants of 'Defect Triage'?

Mandatory Participants

Below project members always take part in Defect Triage Meetings.

- Project Manager
- Test Team Leader
- Technical Lead
- Development Team Leader

Optional Participants

- Developers
- Testers
- Business Analyst

- ❖ **What is Hot Fix:** A small piece of code developed to correct a major software bug or fault and released as quickly as possible. **hotfixes address very specific issues like: Adding a new feature, bug, or security fix**. **Changing database schema**

- **Patch -** Publicly released update to fix a known bug/issue.
- **Hotfix -** update to fix a very specific issue, not always publicly released.
  - ❖ **Bug Masking**: A **masked bug** is an existing error that has not yet caused a failure just because another error has prevented that piece of the code from being performed. The main feature of these bugs is that they hide other system defects.
  - ❖ **Latent Defect:** A **latent bug** is an existing error that has not yet caused a failure because the accurate set of conditions was never met. Such bugs can stay in a system for a long time and be detected in different software releases.
  - ❖ **Golden Defect:** The bug that is occurred in every instance of the application with severity level high and with high priority. These bugs can affect the critical functionality of the software app

❖ **QA Sign-off & Conditional Sign-off**

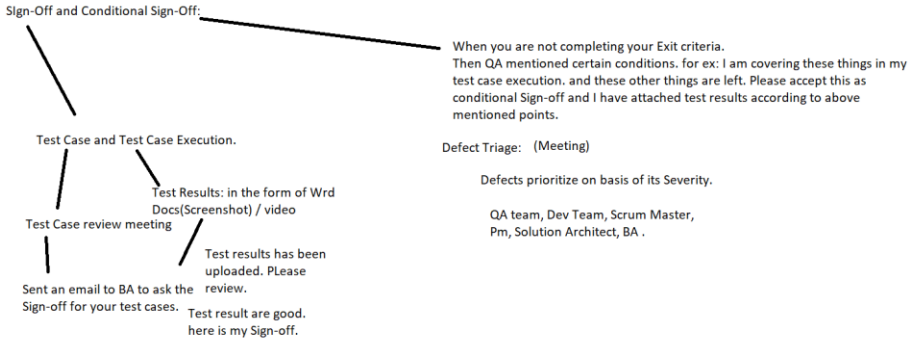
- **QA Sign-Off:** Means your test cases & test execution results are reviewed. And you are saying that testing is completed and ready to deliver. So Sign-off is a way to say that you have completed any task that could be test execution, Test case writing any other document.

- **Conditional QA Sign-off:**
- The **conditional sign-off** is ultimately a way for you to accept exiting the **testing**, but with a list of agreed conditions to move forward.

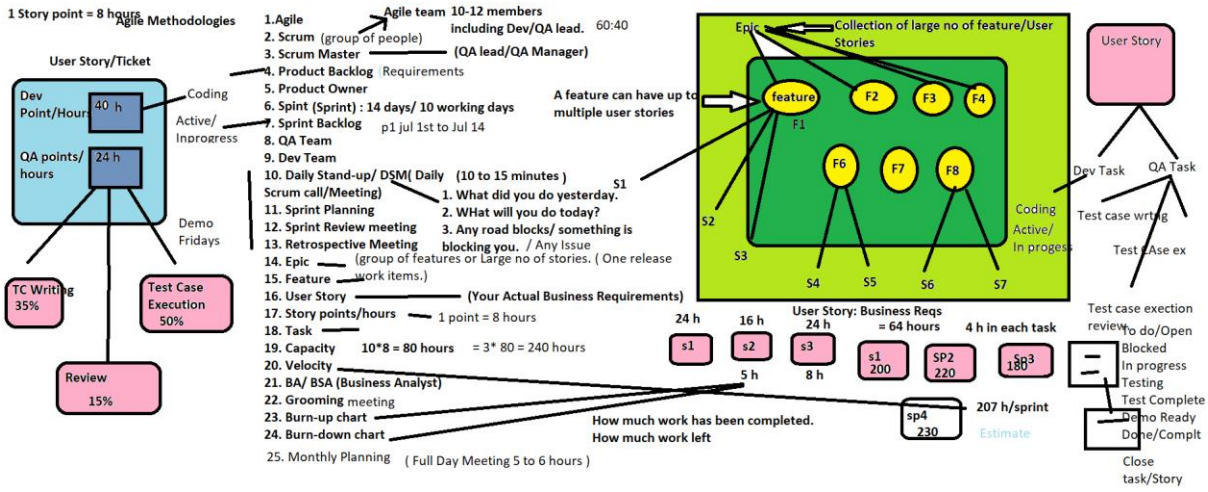
- When the application doesn't meet the exit criteria, the QA can do the Conditional Sign Off. For instance, when the application development exercises cross the cut-off time/deadline, and the tester has not finished the testing, they can do the Conditional Sign Off after speaking with their Test Manager.

  OR

- Apart from this QA, do conditional sign off when the application does not meet the exit criteria. The conditional sign-off is ultimately a way for you to accept exiting the testing, but with a list of agreed conditions to move forward.



Agile: Important Key terminologies:



himanshu

Email: himanshusingh309@gmail.com