Software Testing

By Mithilesh Singh

Testing

Process of verification and validation is known as Testing.

Or

Checking the functionality of an application as per customer rquirement

Or

Process of making an application bug free or stable.

Types Of Testing

Manual Testing

Automation Testing

Manual Testing

It is step by step process to test the component of a module of an application.

e.g.
Application--> Gmail
Module/Feature-->Compose
Componenet-->cc/send.

Types Of Manual Testing

Functional Testing

- Testing the functionality of an application.

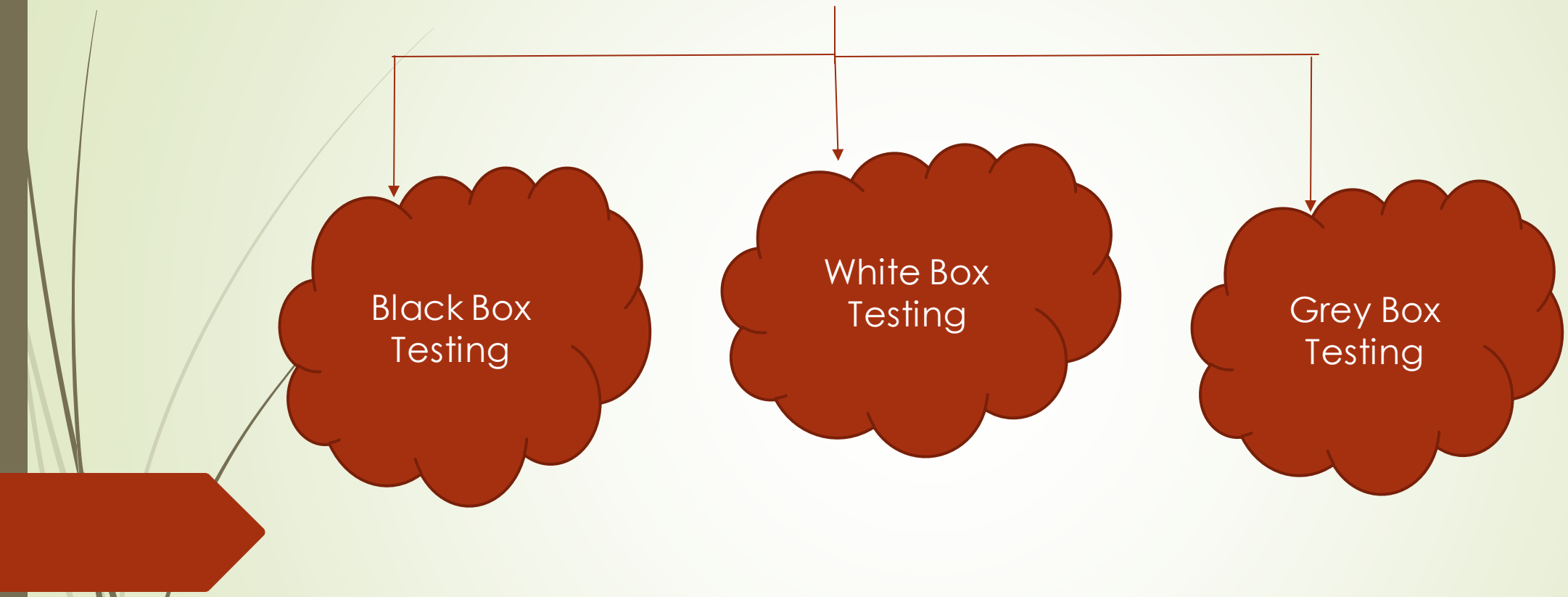
Non Functional Testing

- **Non-functional testing** is the testing of a software application or system for its **non-functional** requirements

Or

- the way a system operates, rather than specific behaviours of that system.

Types of Functional Testing



Types of BlackBox Testing

- Functionality testing
- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Regression Testing
- User Acceptance Testing

Functionality Testing

- Checking the Functionality of a feature with different kind of inputs(Positive and Negative) is know as Functionality Testing.

Integration Testing

- Process of testing the two or more than two features/Sub features simultaneously which is dependent to each other is know as Integartion testing.

- **Rules of Integartion Testing:**
  - Integration Testing always Perform once Functionality Testing has done.
  - For Integration Testing good Product Knowledge is require.
  - In Integration Testing Source, Destination and Data is require.
  - Data should be saved at Destination.
  - IT can be done in Module also

Types Of Integration Testing

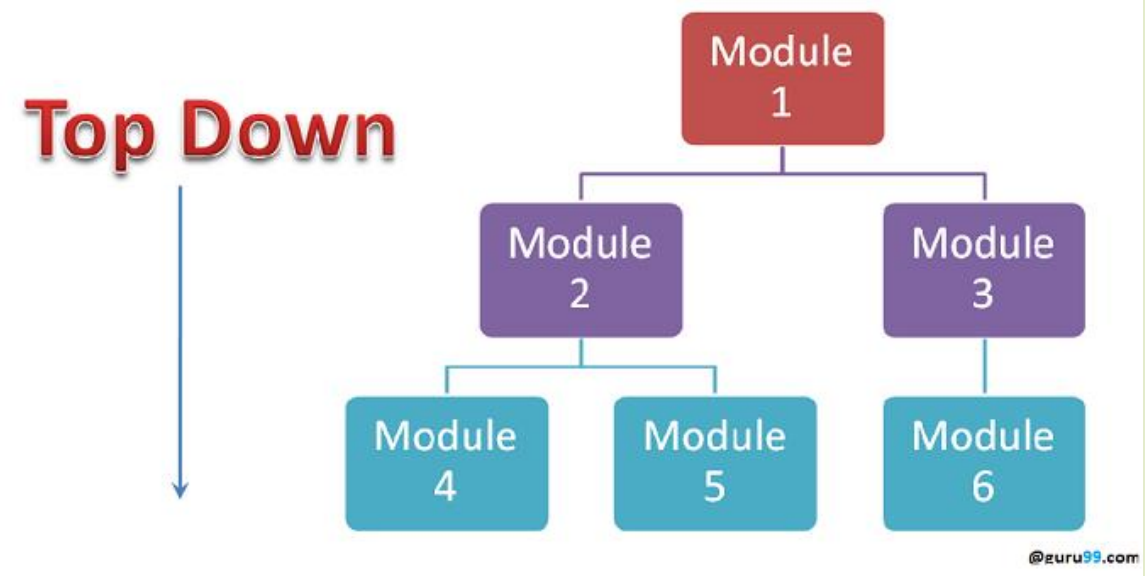
1. Incremental Integration Testing
   testing is done by joining two or more modules that are **logically related**

2. Big Bang Approach: all component are integrated together at **once** and then tested

Top Down Approach

- In Top to down approach, testing takes place from top to down following the control flow of the software system.

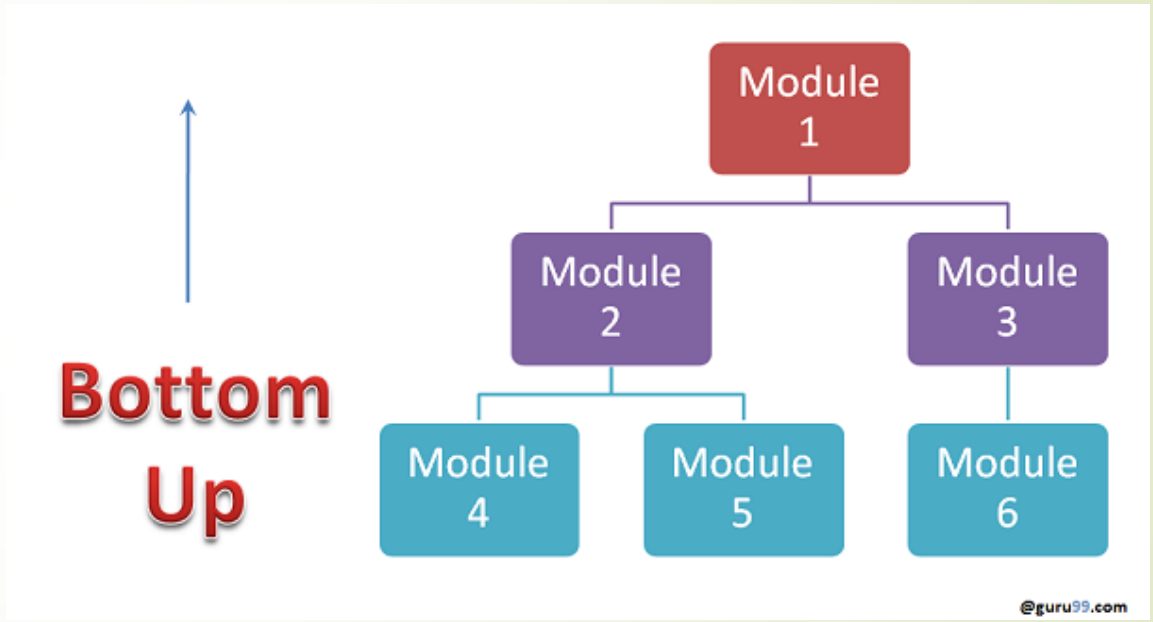
  Takes help of stubs for testing.



Bottom Up Approach

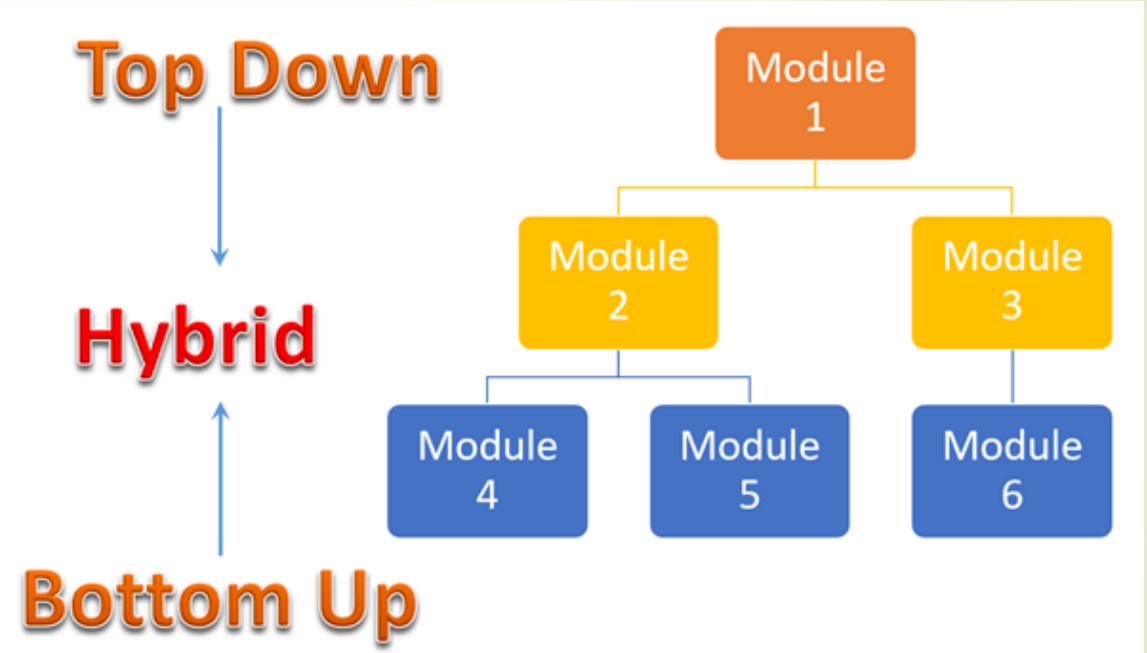
- each module at lower levels is tested with higher modules until all modules are tested.

  It takes help of Drivers for testing



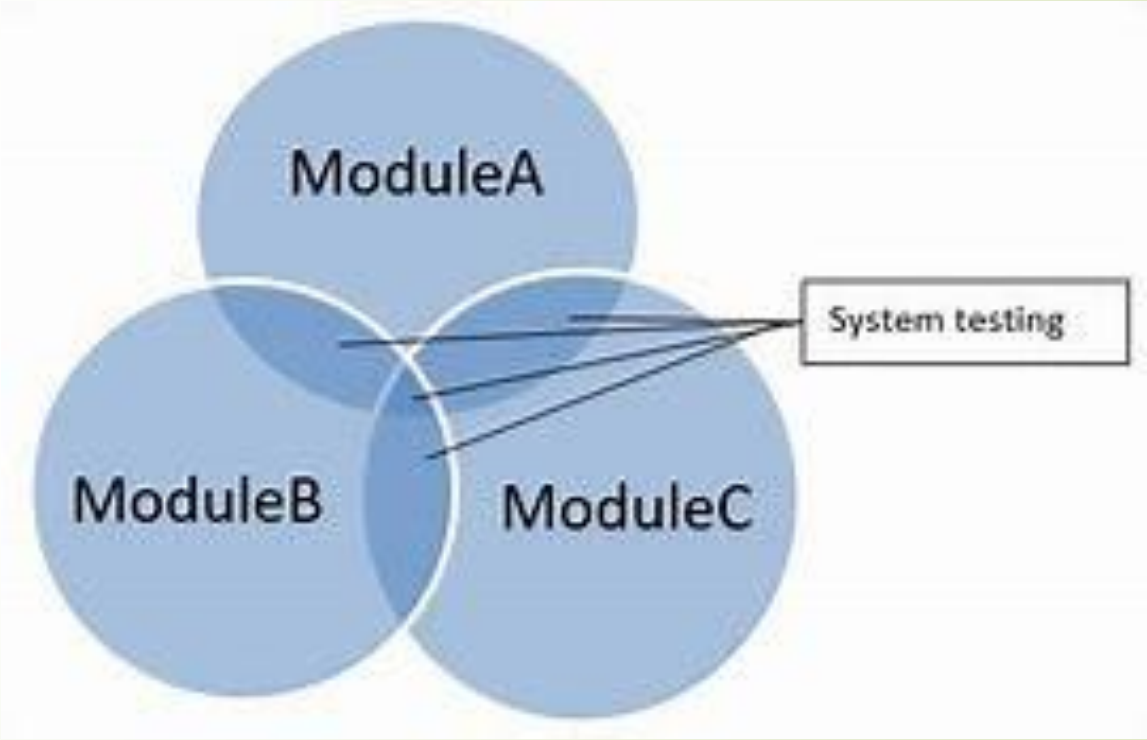
Sandwich Integration

- IT is a combination of Top Down and Bottom up approaches. Here, top modules are tested with lower modules at the same time lower modules are integrated with top modules and tested.
- This strategy makes use of stubs as well as drivers.



System Testing

- Checking the end to end flow of an application by navigating through all the necessary module of a software is known as System Testing



Sanity Testing

Sanity testing is to ensure that all the defects have been fixed and no further issues come in existence due to these changes.

Smoke Testing

Testing the basic and critical feature of an application in early stage as soon as get the recent build

Build: small part of an application

- NOTE:
- Identifying the business critical functionalities that a product must satisfy.
- Ensuring that the smoke test passes each and every build in order to proceed with the testing
- Smoke Tests can be manual or automated

Regression Testing

- Whenever developer does some code changes because of bug fixes or enhancement across the builds or across the release, these changes may have some impact area.
Checking these impact area or affected area is called as Regression Testing.

Challenges Of Regression Testing

- To Identify Impact Area.
- Time consuming process due to
  a. Test Cases increase in every release
  b. Less Resources
- No Accuracy because of Repetative Task.

User Acceptance Testing

It is a kind of testing done by the clien't before accepting the product to just ensure that whether product is working as per requirement or not.
It is end phase of testing.

White Box and Grey Box Testing

White Box Testing

- 1. It is type of test performed by developer.
- 2. In this type of testing once the coding is done the developer themself checks there own written code.
- 3. Since Code is visible it is known as WBT.

Grey Box Testing

- 1. It is the combination of Black Box and White Box Testing.

Types of Non Functional Testing

- 1. Performance Testing.
  a.Load Testing.
  b.Stress Testing.
  c.Stability Testing.
  d.Scalability Teesting.
  e.Volume Testing.
- 2. Compatibility Testing
- 3. Usability Testing.
- 4. Localization and Globalization Testing.

Performance Testing

- " Testing the behaviour of an application at different different load."

Types:

- **Load testing -->** with difference no. Of users.
- **Stree/Stability testing -->** with different user for particular period of time.
- **Scalability testing -->** with different user, user will be decreasing and increasing at particula scale range.

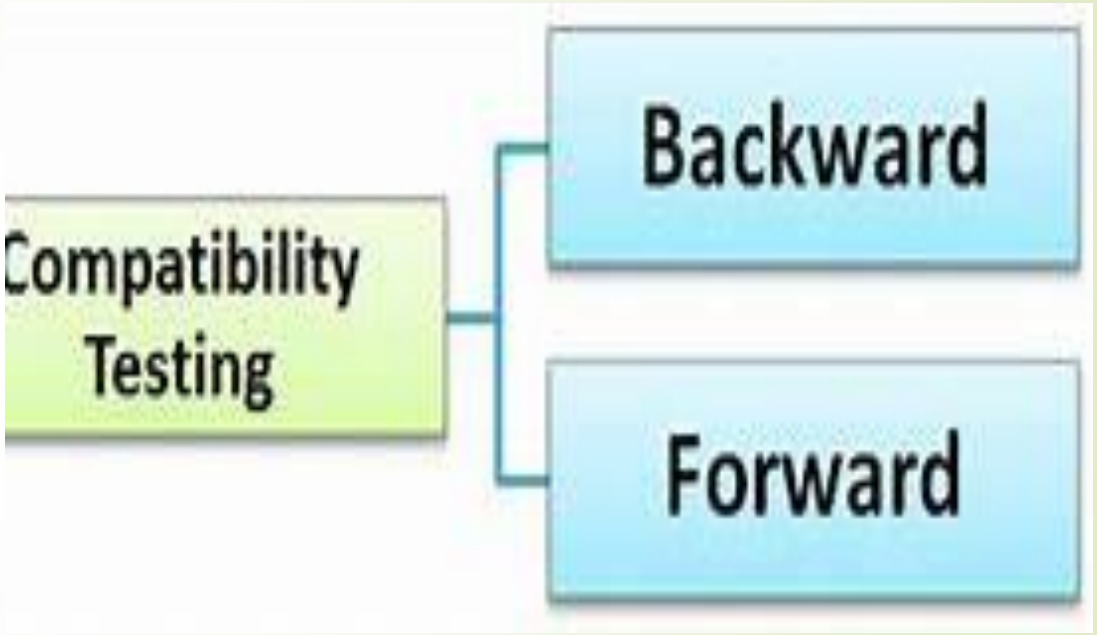
Compatibility Testing

" Checking the behaviour of an application at different different environment(combination of hardware and software)"

Types:

1. **Backward**: Checking the behaviour on older version.

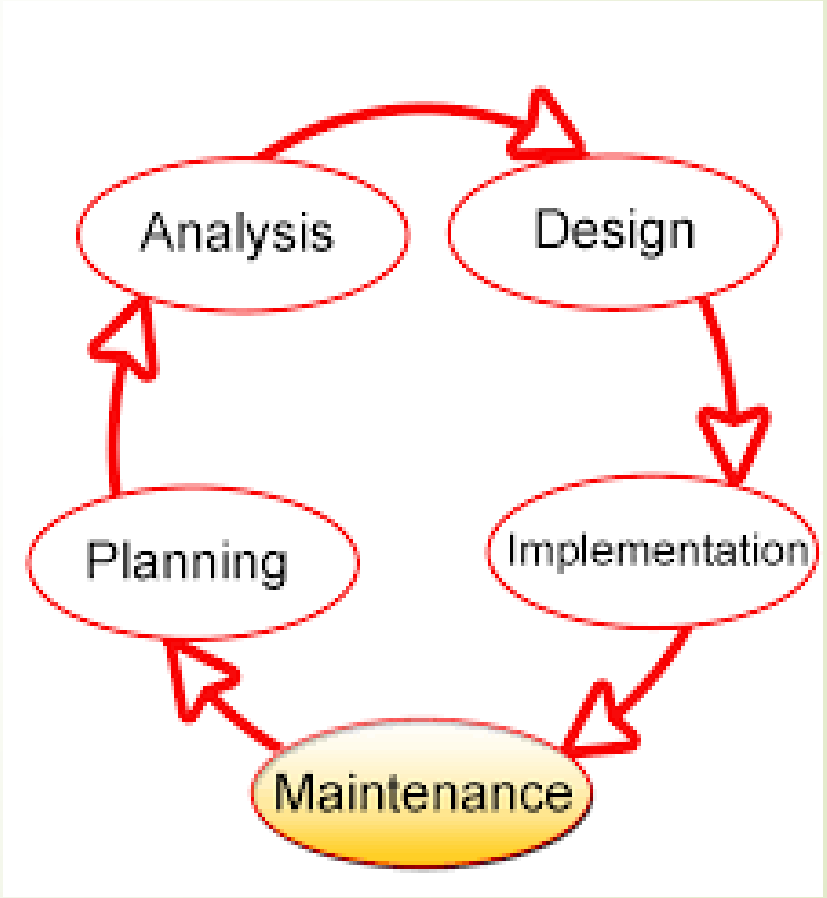
2. **Forward** : checking the behaviour on a new version



SDLC (Software Development Life Cycle)

It is a Step by Step Procedure to develop an application. It consist of various phases:

1. Requirement Collection
2. Feasibility Study
3. Design
4. Coding
5. Testing
6. Installation
7. Maintanance



1. **Requirement Collection**: In this phase the business analyst goes to the customer place to collect the requirement or business need of the customer.

2. **Feasibility Study** : Once the requirement is collected, team of high level people will sit togeter and decide the project is durable or not based on some factor.

3. **Design** : It is a blue print of an application.
Module level design is known as High level and component level design is known as low level design.

4.**Coding** : Once the Blue print of the application is ready, it is handed over to the developer, dev start writing the code based on the requirement as well as the design of the application, by choosing the particular programming language.

5. **Testing** : Testing the application untill it become bug free.

6.**Installation** : Product is carriedfrom company environment to customer environment.

7.**Maintanance** : overcome customer issue and do some necessary changes untill project is not out dated or become sunset

SDLC Models

- Waterfall Model
- Spiral Model
- Prototype Model
- Hybrid Model
- V model
- Agile Methodology

Waterfall model: It is an older and sequential model where downward flow of data happens like each stage outcome acts as input of other stage. Since progress of an application happens downward like waterfall it is known as waterfall model.

Stages:

1. **Requirement Collection**--> business need of client.
2. **Feasibility Study**--> analysis of business need
3. **Design**--> create architecture of flow of application
4. **Coding**--> develop the application.
5. **Testing**--> Test the application.
6. **Installation**--> make available at production server.
7. **Maintanance**--> enhancement or some newly added features.

Prototype model:

This model is very useful for the client to get interact with product. In this model we create dummy application for client in first stage and after getting confirmation we procees for actual development.

Stages:

1. **Requirement Collection**--> business need of client.
2. **Feasibility Study**--> analysis of business need
3. **Prototype design**-->dummy one
4. **Prototype testing**--> testing dummy app
5. **Customer review**-->customer feedback on the application.
6. **Design**--> create architecture of flow of application
7. **Coding**--> develop the application.
8. **Testing**--> Test the application.
9. **Installation**--> make available at production server.
10. **Maintanance**--> enhancement or some newly added features.

Spiral Model

- This model we use when customer want to use product at very early stage with some critical features. In this model in-between changes from the client is acceptable.
- Each module develop in a particular cycle, each cycle consist of(Requirement collection, designing, coding and testing pahses).
- Major changes sometimes increase the development time, we can add one or two more cycle due to some changes.

V model

- In this model development and testing both happens together, So each stage have to cross review process and downward flow of bug is very rare.
- This is very costlier approach that's why very few companies follow this approach.
- If some changes are require then have to do lots of rework, so it is time consuming as well.
- If we have low budget project then can not go with this model, because all resources need to add in early stage and have to pay also.

AGILE methodology

- **AGILE methodology** is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project.
- Or
- A project management methodology characterized by building products that customers really want, using short cycles of work that allow for rapid production and constant revision if necessary.

Or

- Agile methodology is an approach to project management that uses 4 key pillars and 12 principles to organize projects

The 12 Agile Principles

The Manifesto for Agile Software Development outlines 12 Agile principles that all projects should follow. These are:

- **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
  The first principle of Agile methodology states that customers should receive project deliverables or iterations across regular intervals throughout the life of your project, rather than just one product delivery at the end.
- **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
  One of the issues the Manifesto authors found with traditional project management is that it was difficult to accommodate last-minute change requests from the customer. This principle ensures that Agile projects have the ability to adapt to any changes, no matter how late in the game, with minimal delay.

- Deliver working software frequently, from a couple of weeks to a couple of months, with preference for the shorter timescale.
  Agile projects plan for frequent, short project timelines that allow for a fast turnaround of workable products. Often, Agile projects will be broken into 1 to 4 week-long sprints or project intervals, each one ending in a product delivery.

- Business people and developers must work together daily throughout the project.
  This Agile principle states that regular communication with all stakeholders is critical to the project's success. Commonly, this involves a short daily meeting with both the project team and any other key stakeholders.

- **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**
  A key concept of the Agile project management methodology is that the right people need to be placed in the right positions and given the autonomy required to do their jobs well. It's important to design a project team based on capabilities rather than job positions or titles within the company. The focus of the project manager should be on motivating the project team and supporting them, rather than micromanaging them.
- **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
  The creators of the Agile Manifesto strongly believe in the importance of co-locating teams and stakeholders whenever possible, as face-to-face communication is more effective than other options, such as email or phone. If your team cannot be co-located, video conferencing is an option that can still capture some of the same value, such as non-verbal cues.
- **Working software is the primary measure of progress.**
  The emphasis within the Agile methodology is on providing completed, working deliverables. This should always take priority over any supplementary requirements, such as project documentation. Also, other metrics, such as hours spent or time elapsed, are not considered as important as delivering working products.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

According to this principle, Agile projects should have a consistent pace for each iterative cycle or sprint within the project. This breakdown should eliminate the need for overtime or crashing schedules while promoting frequent output of workable products. It should also create a repeatable cycle that the team can continuously follow for as long as necessary.

Continuous attention to technical excellence and good design enhances agility.

A primary focus of an Agile project should be on improving the end product and achieving advancements consistently over time. In other words, each iteration should always be an improvement over the previous one, and the team should always be looking for new innovations Simplicity – the art of maximizing the amount of work not done – is essential.

The goal of an Agile project is to get just enough done to successfully complete the requested project and meet the requested specifications. Any additional documentation, steps, processes, or work that does not add value to the customer, or enhance the project outputs, should be avoided or eliminated.

The best architectures, requirements, and designs emerge from self-organizing teams.

Agile is based on the belief that motivated, autonomous, and skilled teams are required in order to deliver the best results and products. Teams should be empowered to organize and structure themselves as needed. They should have the freedom to collaborate and innovate as they see fit, without being hampered by too much oversight.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

A successful, self-motivated team requires a strong focus on how they can advance their skills and processes to continually grow and improve. The team should have regular reviews on their performance and outcomes, including discussion on how they can improve as they move forward.

benefits of adopting an Agile method:

- Continuous customer contact
- The ability to adapt
- Faster delivery
- Lower project risk

Bug

"Behaviour of Application/Software/Product/Build against the Client's requirement" is known as Bug.

Bug can be said:

- Defect
- Error
- Issue
- Failure

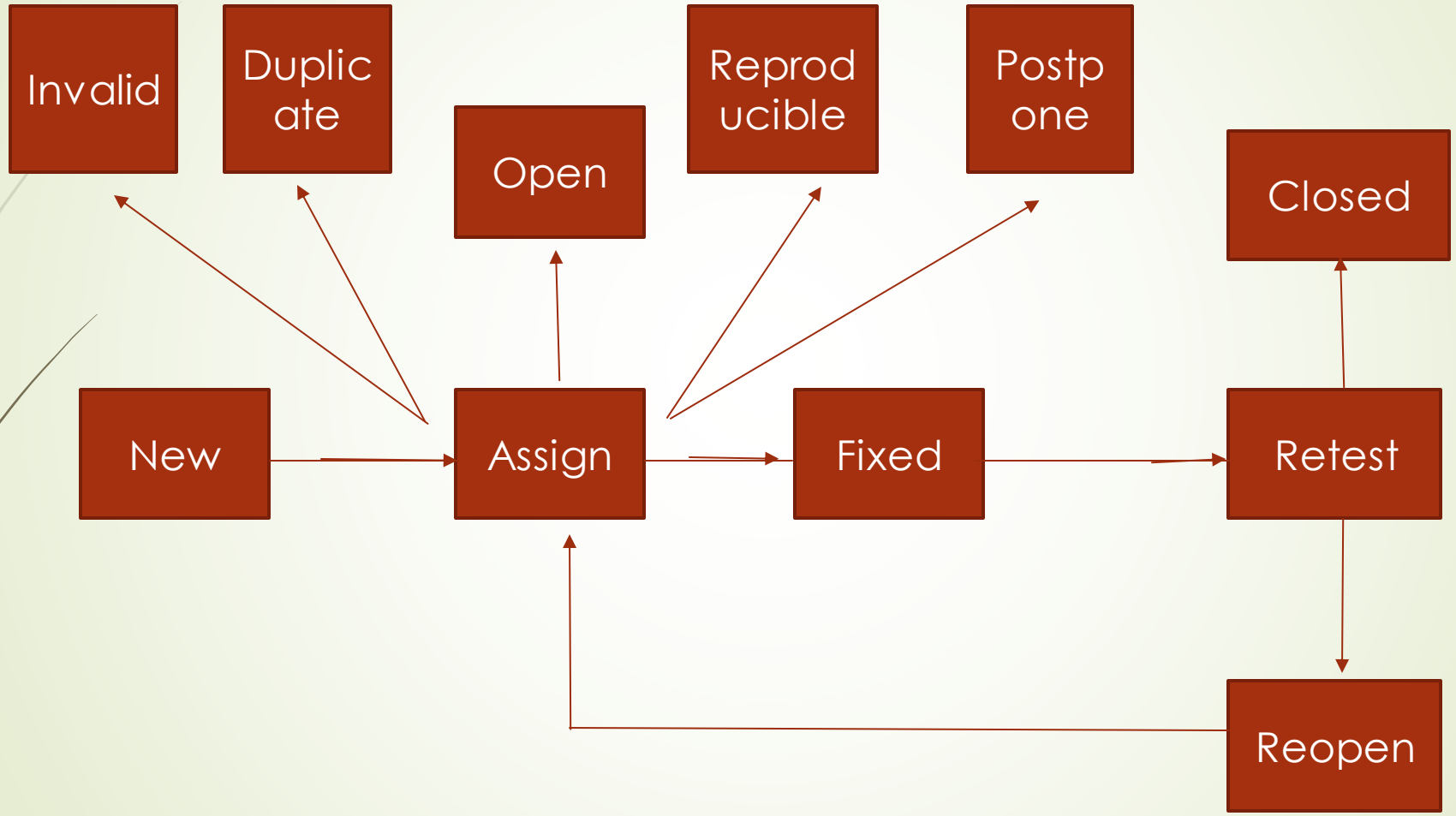
Error :Used by Developer.

Defect:Anyone in Team.

Issue : conv. with client.

Failure : Mostly used by Testers.

Bug Life Cycle Architecture



Status Of Bug

- **New** : When bug is newly logged in bug sheet.
  Assign : when bug is reported to Developer.
  Open : once Developer open the filed bug.
  Fixed : After fixing the code error by developer.
  Retest : At the time of retesting the fixed bug.
  Reopen : Assigned same bug to developer again.
  Closed : After bug fixed.
  Invalid :When developer reject the filed bug.
  Duplicate: when same bug is filed by two different test engineer.
  Postpone: when developer don't have time to fix the bug.
  Not Reproducible: developer not able to find out.

Test Document

- **Test Case:**
- 1. It is the low level or indepth documentation which we prefer to test the functionality of the application.
- 2. It contains input to test the functionality of a module of an application.
- 3. It helps other test engineer to know about the product.

- **Test Scenario**:
- 1. It is high level documentation which follow the end to end flow of the application.
- 2. It does not contain any kind of input.
- 3. it gives the overall product idea.

Difference between Alpha and Beta Testing

- **Alpha Testing**

  " This is a terminology which we use to describe all kind of Functional Testing in Service Base Company.

- **Beta Testing**

  'This Type of testing always comes under the Product base company,
  company develop the product and release a beta version for that product.
  After getting response from the customer company plan to improve the product accordingly.

Verifacation VS Validation

- **Verifacation**

1. It includes understanding requirement.

2.It includes gothrough review and walkthrough of requirements.

3.It includes writing reviewing and approval of the test documents

4.It also includes writing the code reviewing and approval of code.

5.Overall verification indicates the complete review process

- **Validation**
-

1.It includes WBT, BBT I.e functinal, Integration,System and UAT.

- 2. Overall the Validation indicates testing process.

Difference between QA and QC

- **QA**
- 1. We check if we are developing the product right
- 2. It includes gothrough/walkthrough and review.
- 3. It is verification process.
- 4. Done before app developed.
- 5. It is process oriented.
- 6. It is prevention method.

- **QC**

1. we check if we have developed the right product.

- 2. It includes WBT/FT/IT/ST/UAT
- 3. It is validation process
- 4. Done after app developed.
5. It is product oriented
- 6. It is detection method.

Stub and Drive

Stub and Drive use during the Integration Testing when there is some kind of dependency between the modules

- **Stub**
- **Stubs** are used in top down approach
- **Stubs** are the called programs.
- **Stubs** are used when sub programs are under construction.

- **Drive**
- **Drivers** are used in bottom up integration.
- **Drivers** are only used when main programs are under construction.
- **Drivers** are the calling programs.

L10N, Globalization and Sun Set mean in Testing

L10N: it stands for localization, it is a **process for checking the localized version of a product for that particular culture or locale settings**.

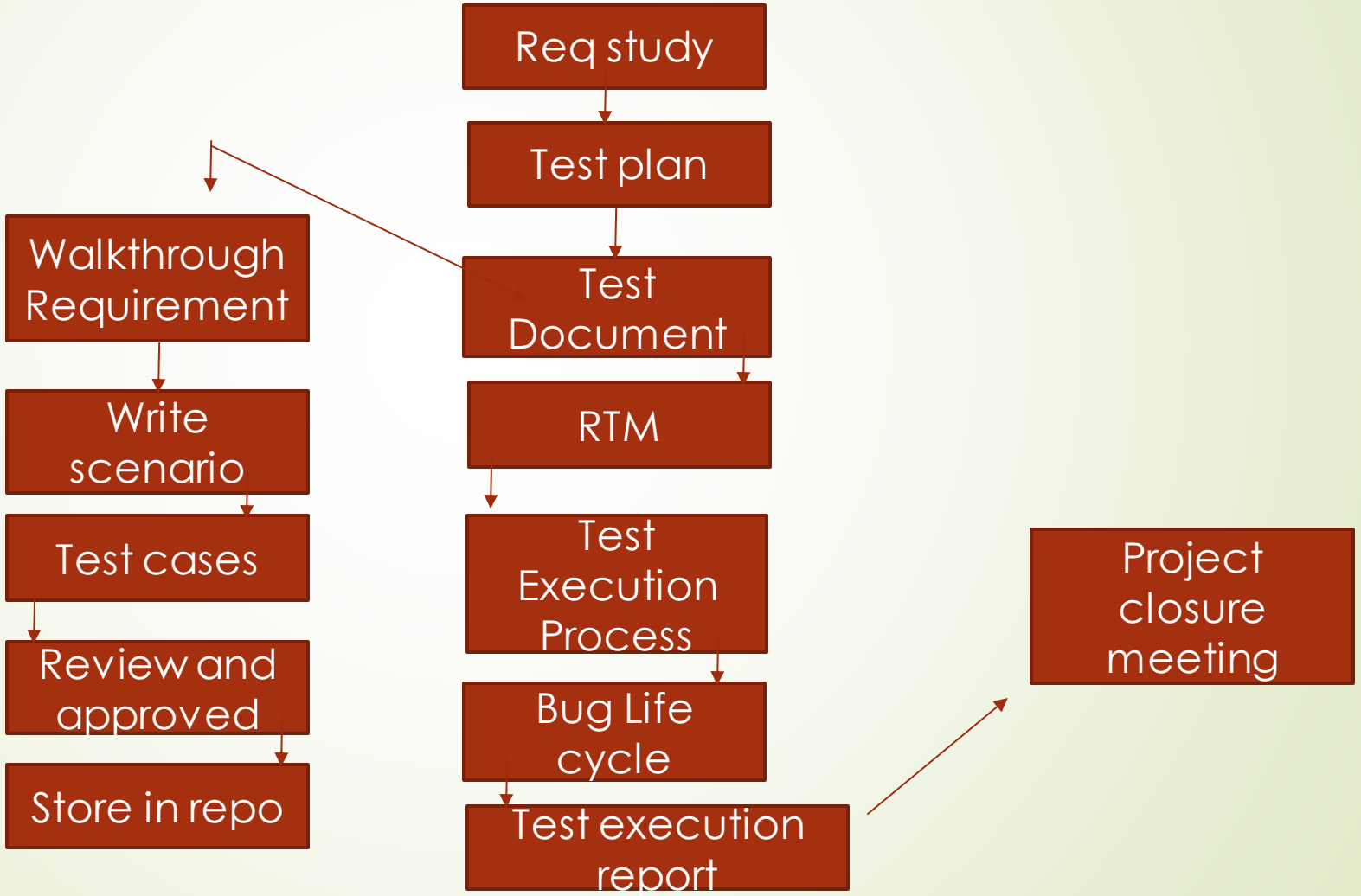
Globalization: Globalization testing is to ensure that application can function in any culture or locale (language, territory and code page) It is also called as Internationalization Testing.

Sun Set: when software is out dated in the market and company stop maintaining it, that is where sun set term comes for that particular application/software.

RTM(Requirement tracibilitty matrix)

- "Mapped the requirement based on written test cases" .
- 1. it is prepare to ensure that the TE has written test cases for every requirement.
- 2. It is easy to track cases written by the particular test engineer.
- Note: RTM is prepare as soon as the test cases are written(Before test execution process).

STLC


Req study
Test plan
Test Document
RTM
Test Execution Process
Bug Life cycle
Test execution report
Project closure meeting
Walkthrough Requirement
Write scenario
Test cases
Review and approved
Store in repo