

**MANUAL
TESTING**

TESTING

- Checking the quality of Product is called Testing
- Checking if a product is defective or not is called Testing

SOFTWARE

- Some lines of codes, programs written in Programming language is called software.
- Software is collection of programs to help us to perform a task.

Let's say

Calculator Application in our laptop or desktop or Mobile.

WhatsApp is a Software Application

Microsoft word is a software

These are some example of Software.

TYPES OF SOFTWARE

Application Software

System Software

Programming Software

HOW SOFTWARE IS DEVELOPED OR CREATED

Let's we take example of calculator

Addition of two numbers

Subtraction of two numbers or anything

When we write program using programming language
and then combine all the program together in one single
application, then that is called software.

SOFTWARE TESTING

- Software Testing is a process of checking the quality of a Software.
- **Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.

Let's understand how to do Software Testing.

Take an example, Calculator

Calculator will have Addition, Subtraction and Multiplication etc....these are functionality of calculator

What is the Test scenario of calculator?

1. Testing the Addition Functionality
2. Testing the Subtraction Functionality
3. Testing the Multiplication Functionality

Actually if we prepare Test cases then Test cases is based on Test Scenarios

Test Scenario	Test case 1	Test case 2	Test case 3
Addition	Add 2 Numbers	Add 3 Numbers	Add a positive and a negative number
Subtraction	Subtract a small number from a big number	Subtract a big number from a small number	Subtract a positive and negative number
Multiplication	Multiply a positive number and a negative number	Multiply two positive number	Multiply two negative number

NEED OF SOFTWARE TESTING

- Software testing is imperative, as it identifies any issues and defects.
- Software testing helps in building the trust and satisfaction of the customer by assuring a defect-free application.
- Software Testing is a testing process to check whether the developed software product is 1) free of bugs for smooth functionality and 2) meets the actual requirements of the clients without any properties or features missing.

BUG, DEFECT AND ERROR

- Bug- Bug is an informal name specified to the defect.
- Defect- It is the difference between the actual outcome and expected outcome. Mainly Testers identified the defect while initial phase of Testing.
- Error- A problem or mistake in the application identified at the time of development. Mainly developers use this term. Due to this developer unable to compile or run a program successfully.

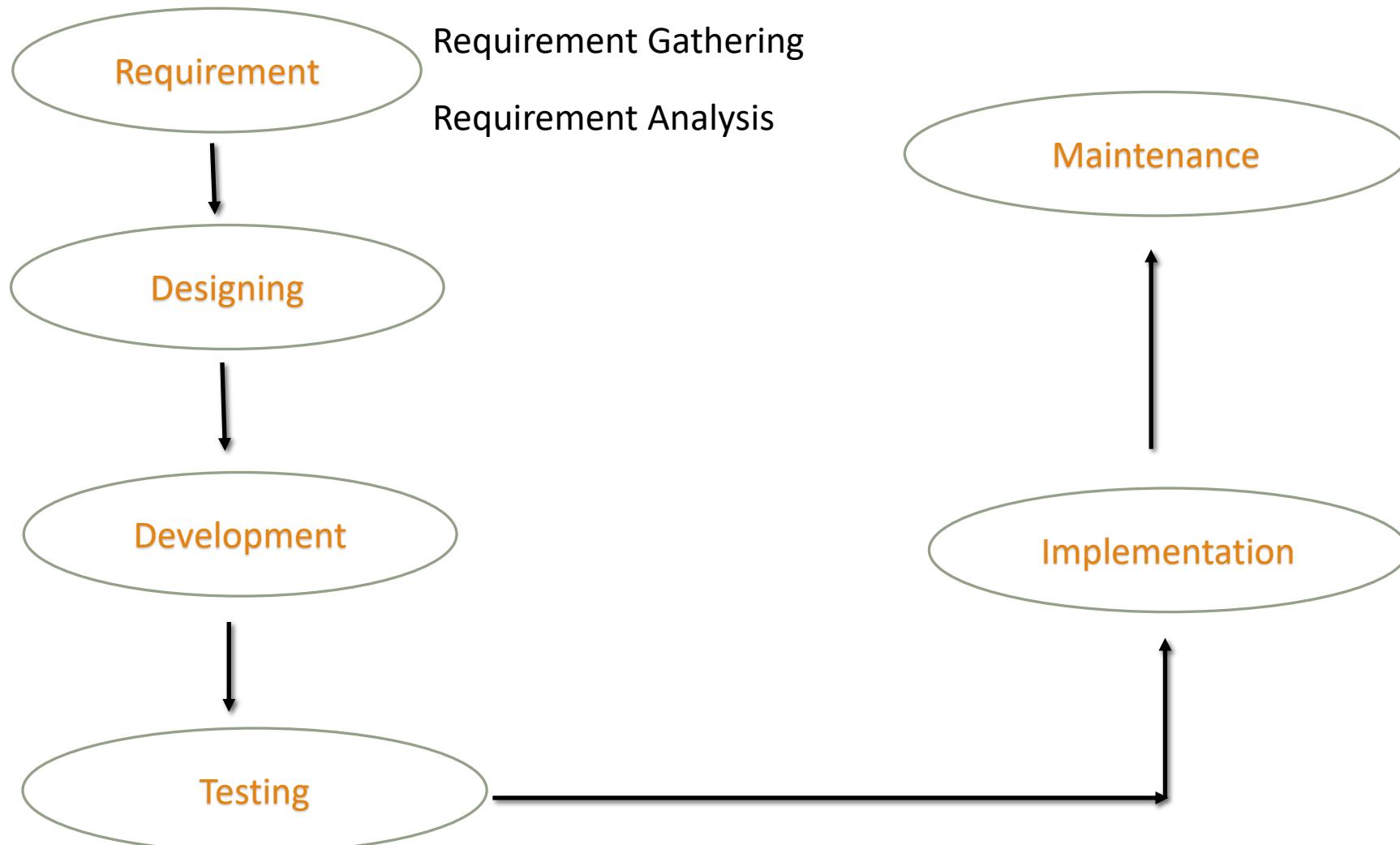
TESTING Vs DEBUGGING

- Testing is the process to find bugs and errors.
- Debugging is the process of correcting the bugs found during testing.

SCOPE AND FUTURE GROWTH OF SOFTWARE TESTING

- In recent times, testing is seen as a good professional career for many. From being a test engineer, one can grow to become a senior test engineer, from a test lead to a test manager; or alternatively, one can become a QA lead or QA Manager.
- Professionally trained software testers are in great demand nowadays and are sought after in the industry. One can either become part of an independent software testing company or join an internal software testing department in a leading firm.
- Testers get paid good salaries, depending upon their level of experience. Testers are paid as well as developers in most companies.

SDLC



It is a life cycle of software development

Software Development Life Cycle SDLC

SDLC is a **process** used by the software industry to **design, develop and test software**

SDLC consists of a **detailed process** that explains **how to plan, build, and maintain specific software.**

SDLC process **aims to produce high-quality software** that meets customer expectations.

Requirement Analysis



- We have to **collect and understand the requirement of the customer**. Normally BA's, Project Managers, and Product Managers are involved in this phase.
- They will talk to the customer, get the requirements, and **prepare a certain number of documents (BRS/SRS)**.

BRS - Business Requirement Specification

SRS - Software Requirement Specification

URS- User Requirement Specification

CRS - Customer Requirement Specification

BRS/CRS/URS-----Same (Non tech team)
team)

SRS (Technical team)

Project manager and Developer both are involve

Design



HLD
↓
LLD

software design documents are **prepared** as per the requirement specification document.

Project manager

A project manager is responsible for planning, executing and closing project. He focusing on tasks, timelines and resources.

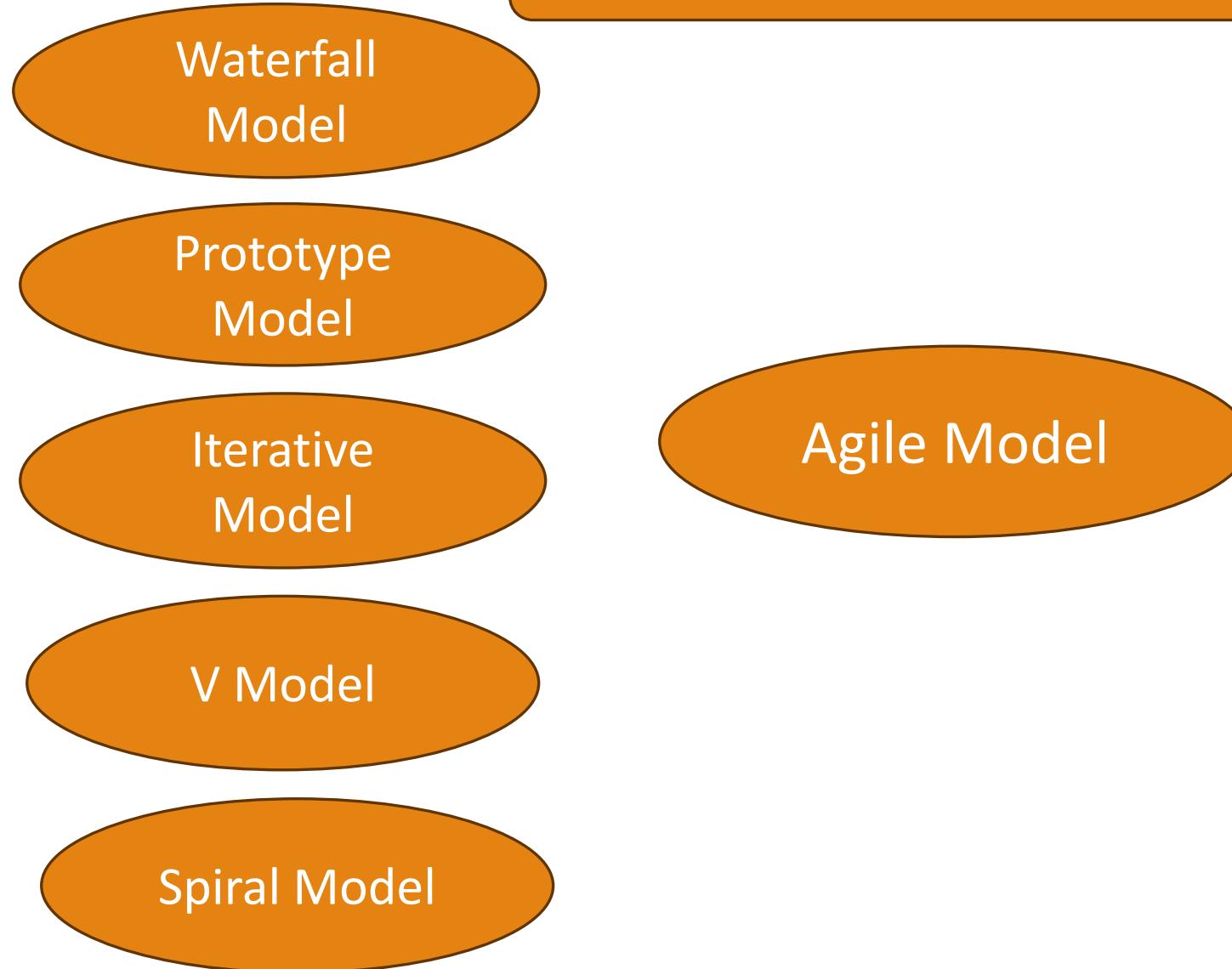
A project managers ensure timely project completion.

Product manager

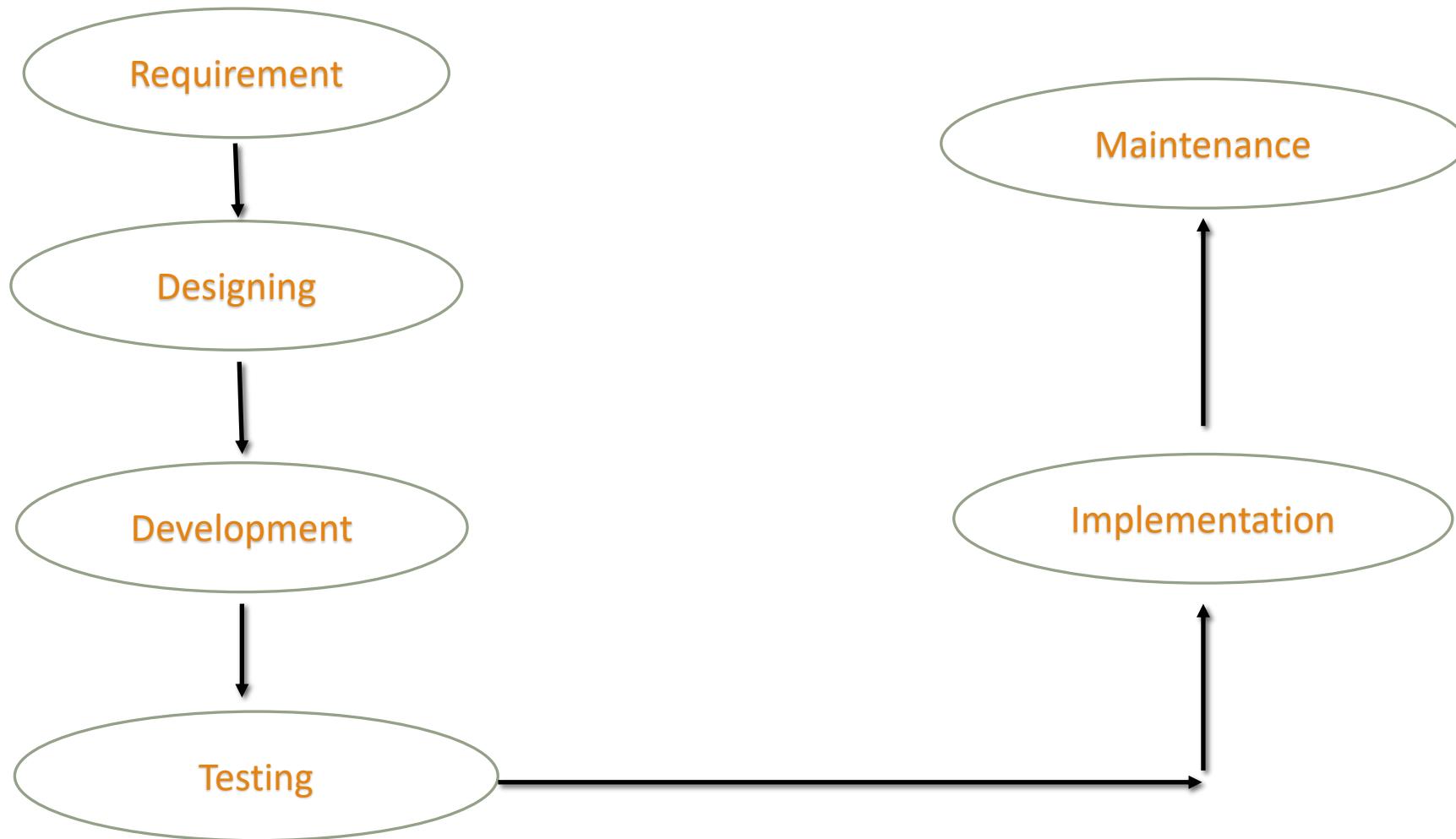
Product manager oversees the development and success of product throughout its lifecycle, involving market, strategy and user experience.

Product managers align the product with market needs and business goals.

SDLC Model



Waterfall Model



Waterfall Model



Requirement are not changing.

Application is not complicated and big.

Project is short.

Requirement is clear.

Quick development

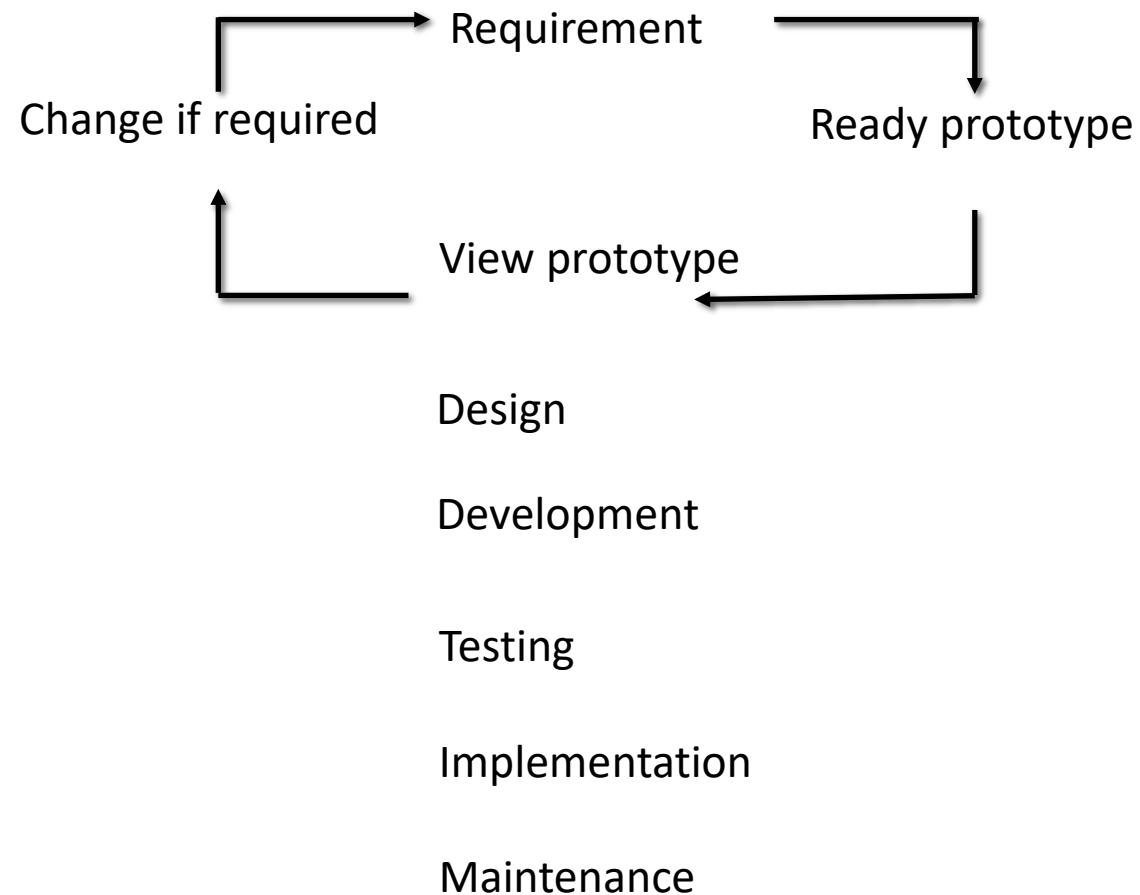
Before the next phase of development each phase must be completed.

Error can be fixed only during the phase and also fixed during maintenance.

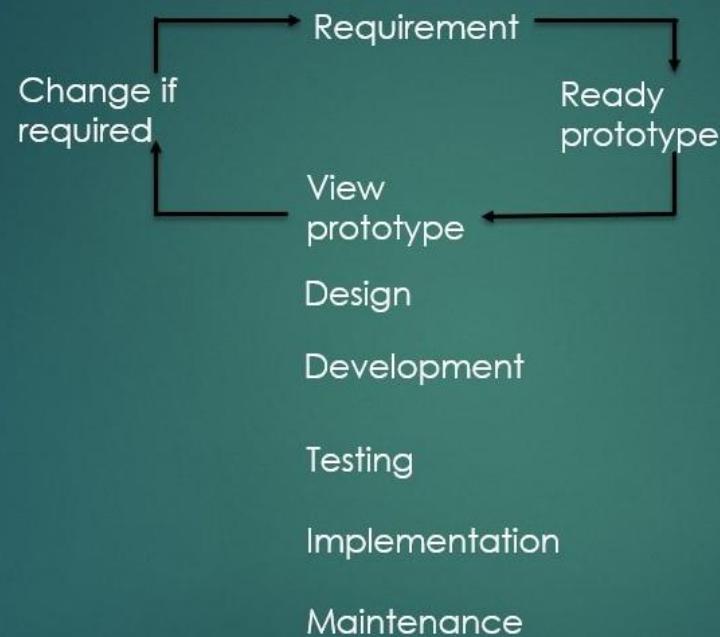
Back step is not allowed

Not satisfaction (low customer satisfaction)

Prototype Model



Prototype Model



This model is used **when user requirements are not very clear.**

Based on user feedback, the SRS document is finalized.

Here dummy software has to be shown to the client before making the actual software.

Iterative Model

Requirement

Design

Development

Testing

View

Design

Development

Testing

View

Implementation

Maintenance

Iteration 1

Iteration 2

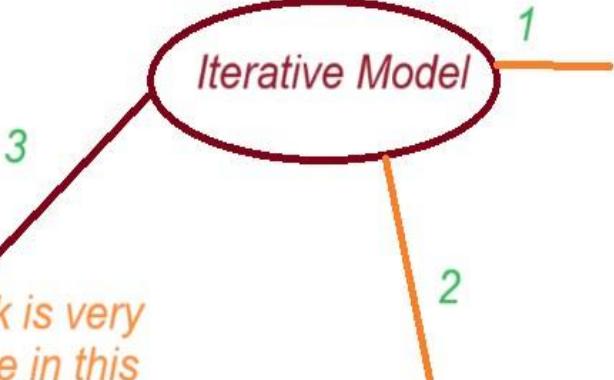
Iteration 3

Iteration 4

Disadvantage

1. Time taking
2. Costly

Iterative Model



We can say also, here develop the software in partially that means In iterative model the software is developed partwise according to clients requirement and also shown to the client.

Here the risk is very less because in this iterative model the client has seen the project again and again .

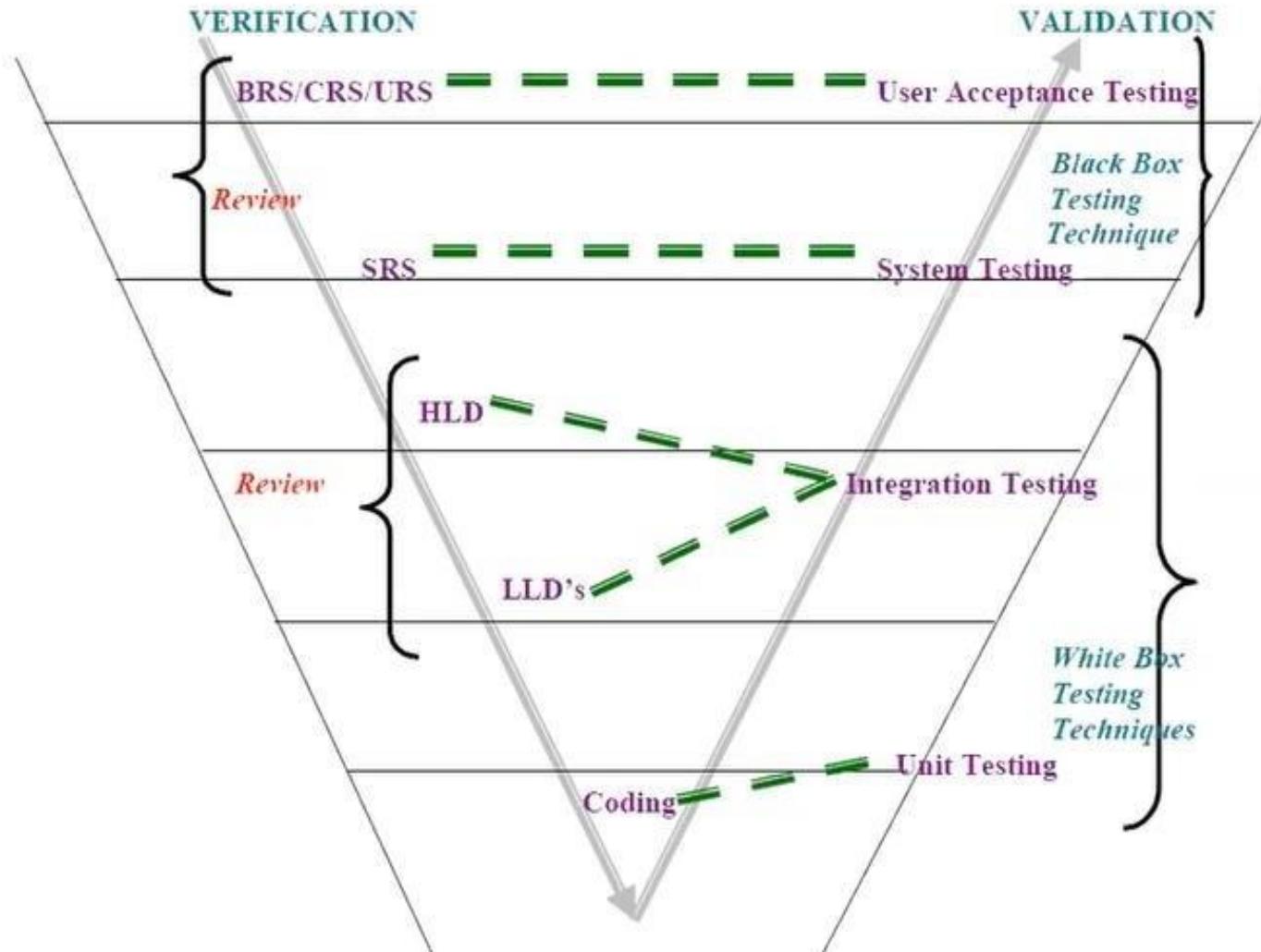
Here iteration is not fixed. In this iteration can also increase or decrease, its depend on size of the software/ project. Basically it is prepared for big project.

In this model, we can't start developing the complete software will full specification of requirement.

School management...
login
admission
enquiry
teacher attendance
student attendance
feedback.....

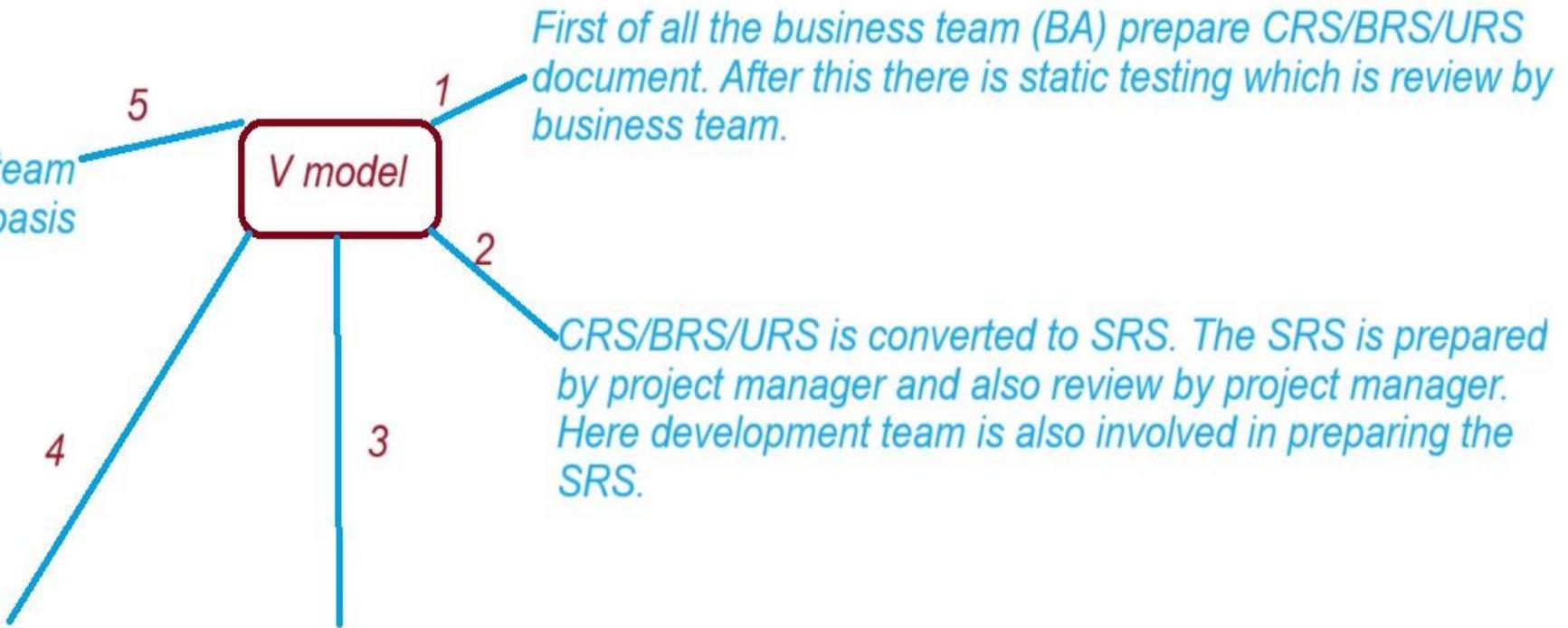
V Model

V-Model



Static Testing

After that the development team prepares the coding on the basis of LLD.



Dynamic Testing

V model

- 6 After coding, unit testing is done which is called white box testing. This testing is done by the developer.
- 7 After white box testing, integration testing will be done and after integration testing system testing will done. Actually these testing is done by QA team.
- 8 When the QA team completes the system testing then the product is deployed to the customer site. here client will do UAT testing.

1

Verification uses like reviews, walkthrough and inspection whereas validation uses Black box testing, White box testing, Integration testing, System testing and also UAT testing

Difference between Verification and Validation

2

Verification does not involve code execution while validation involves code execution

5

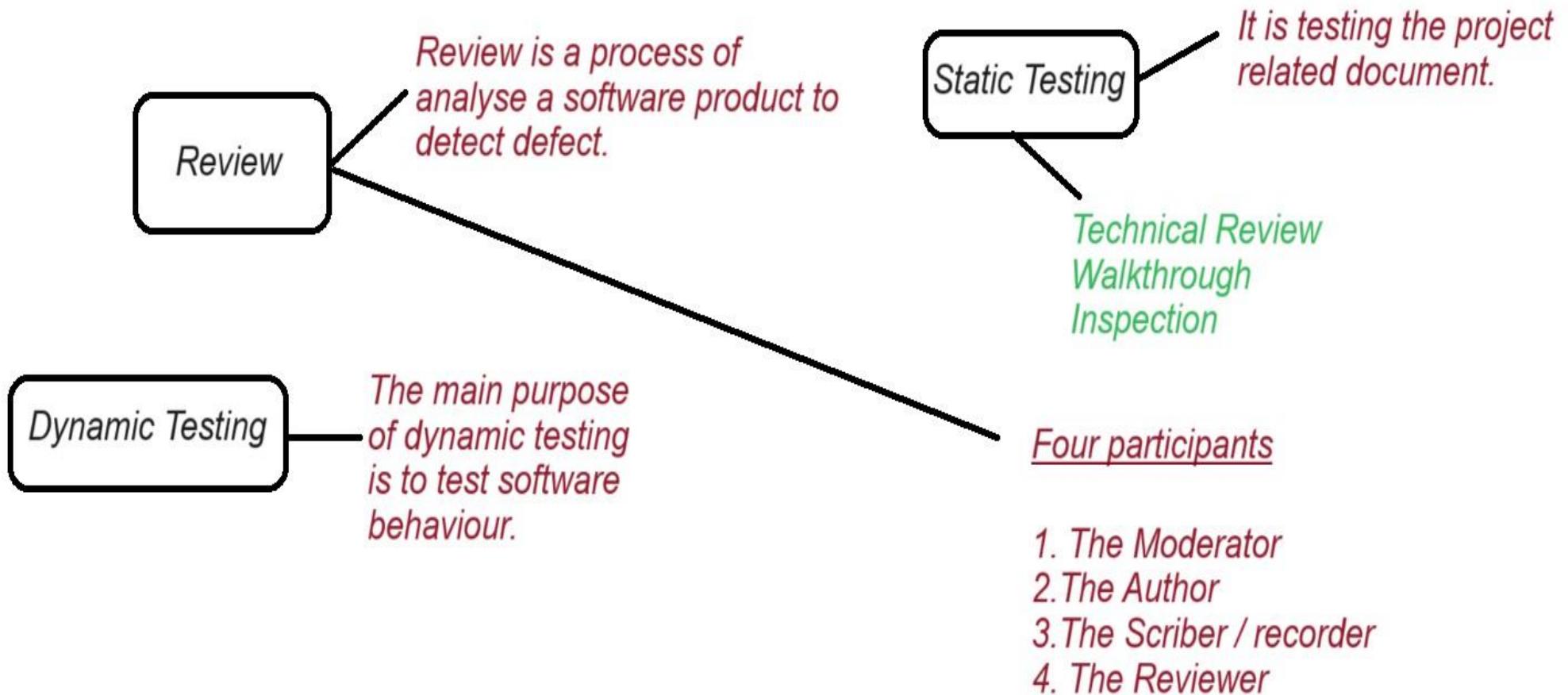
Verification checks whether the software confirms a specification whereas validation checks whether the software meets the requirements and expectations.

4

Verification process includes checking documents, design and code whereas Validation process includes testing validation of the actual product.

3

Verification finds the bugs early in development cycle, whereas Validation finds the bugs which verification could not catch



The Moderator

Moderator leads the review process. his role is to determine the type of review, scheduling meeting, distribute the document to other participants before the meeting.

The Author

Author is writer of the document under review.

The Scriber / Recorder

He is responsible to record each defect and comment or suggestion.

The Reviewer

To check defect to further improvement.

It just a formal review process

Technical Review

Technical review to ensure correctness and completeness of document.

It is lead by trained moderator. Here technical reviewer will read the document before the meeting for better understanding and to suggest of document

Which document will be reviewed in technical review ?

1. Requirement review
2. Design review
3. Code review
4. Test plan review
5. Test case review

Walkthrough

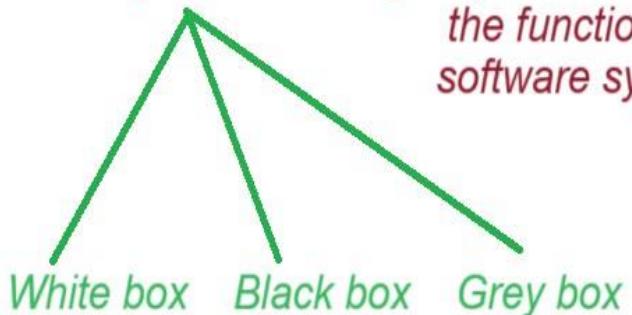
It is a informal review. In this, author reads the document and discuss with peers so that they note out defect and suggestion. It is not a preplanned.

Inspection

It is most formal review type. In which an author request the services of a Moderator, Scriber, Reviewer in a formal meeting. The Moderator distribute the document in between participant during meeting, scribe takes the notes of issues the reviewer during the meeting.

Static Testing - Static testing will analyze the code, requirement document and design document.

Dynamic Testing - Dynamic testing will look at the functional behaviour of software system



Q. How many types of testing?

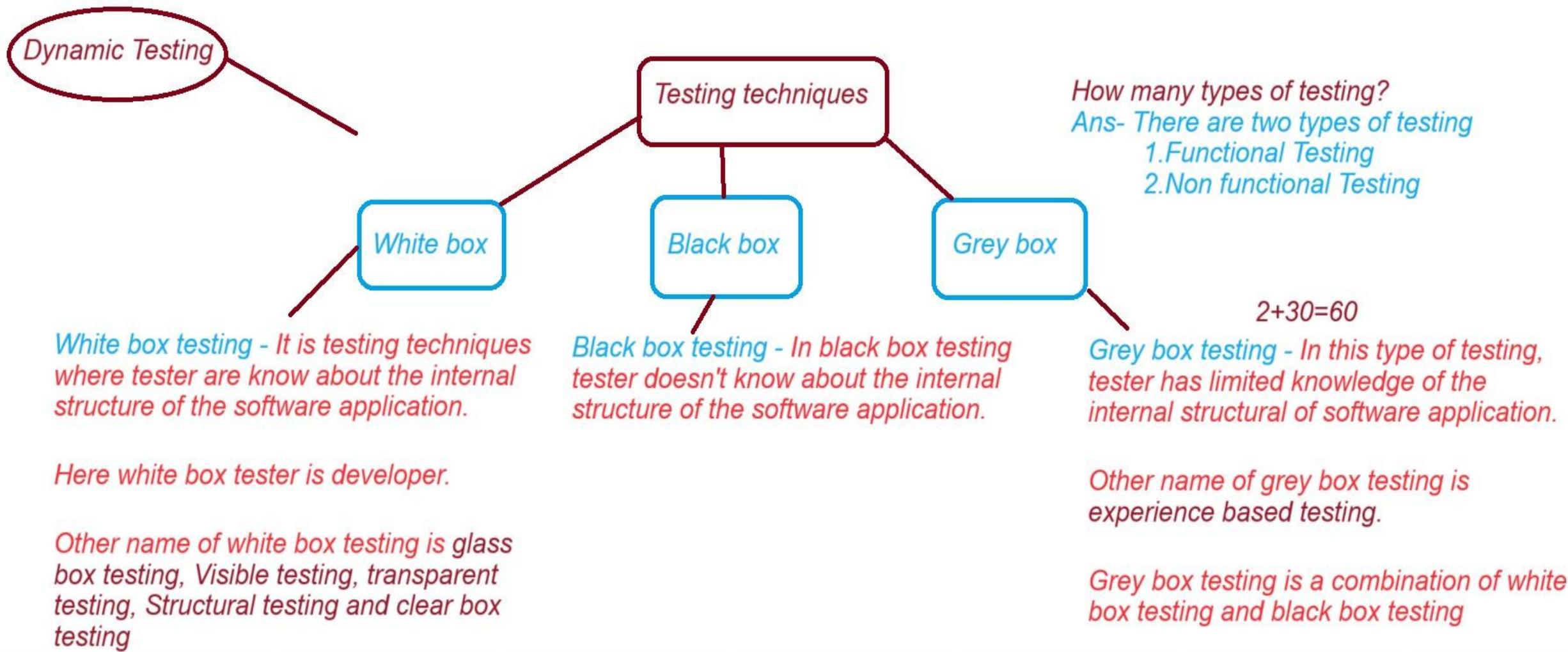
Ans- Functional and Nonfunctional

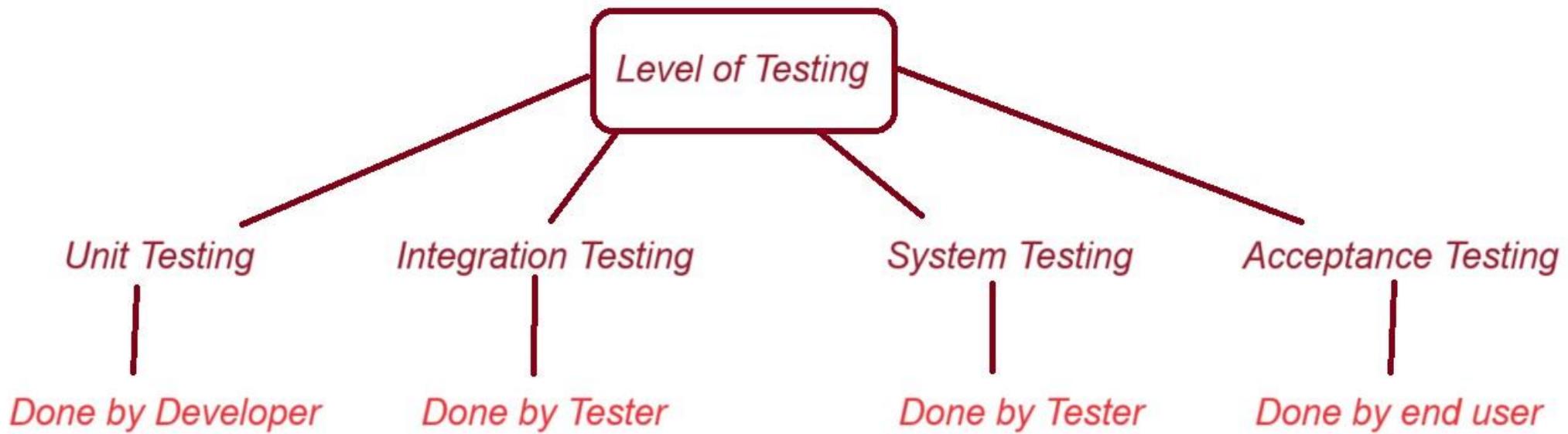
In functional testing we test behaviour of the software application. In functional site we will test the application UI, database working or not, elements are displaying or not that means we focus on more things on functional requirement

In Non-functional testing we test the performance of the software application such as speed, stability, scalability.

Q. How many types of testing techniques?

Ans - White box, Black box and Grey box.







Unit testing

Unit testing is a type of software testing where individual units of a software are tested. its done by developer. unit testing is a white box testing techniques.

Unit testing is a single component.

Integration Testing



In integration testing we test combination of two or more modules that means we have to check the data flow from one module to another module. Integration means combining.

For example - login page, mail box, draft and bin these are integrated logically. Such as when we enter login credentials and click on login button then to be directed to the mail box and when mail box select the email and click on cancel button then it should be appear in draft and when we delete from draft then it should be appear in the trash folder i.e bin.

Approaches of Integration Testing

- 1. Top down Integration
- 2. Bottom up Integration

Stubs

drivers

Different modules

Module-1
Module-2
Module-3
Module-4

Module functionality

Login page of the web application
Home page of the web application
Print set up
log out page

Top down Integration testing - In Top-Down Integration testing, firstly the high level modules are tested that means testing from main module to the sub module.

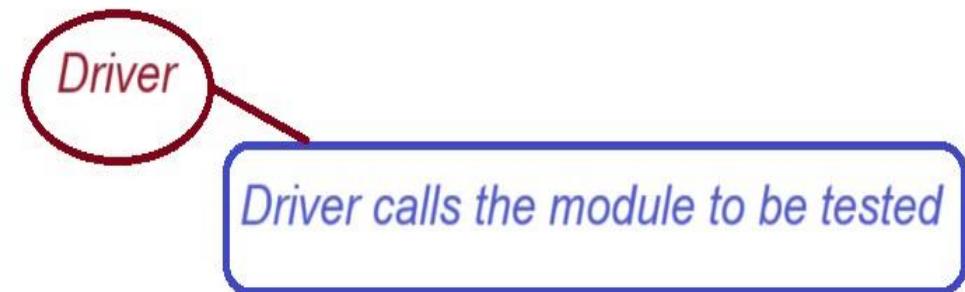


Stubs - In this approach stubs are used as a dummy module for the missing module. Missing module means these modules are under development or not available.

Bottom up Integration - In bottom up integration testing firstly the low level modules are integrated and tested that means here testing from sub module to the main module.



Drivers - In this approach drivers are used as a temporary module for missing module which are under development. It calls the module.



Big bang testing - In big bang testing, when all the component in the units are not completed then integration testing will not execute.

Incremental testing - In incremental testing two or more modules that are logically related to each other and then tested proper function of the application.

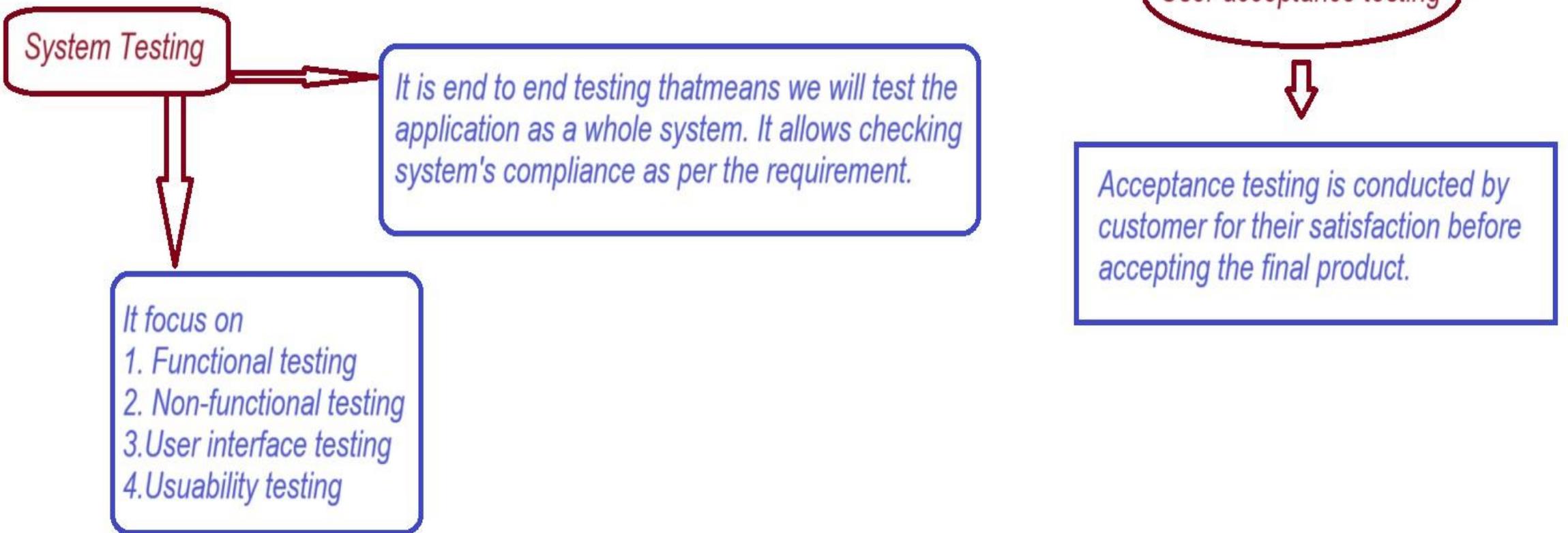
In Incremental approach there are two methods

1. Top down - Here integration testing takes place top to bottom.
2. Bottom up - In this testing lower level modules are tested that means here integration testing takes place bottom to top.

Sandwich testing



It is a combination of Top down and Bottom up approach. It is also called Hybrid integration testing. it makes use of both stubs as well as drivers.



Functional testing

In functional testing, we have to test the behaviour of the software application. In functional testing we will test the application UI, database working or not.

Functional testing is a type of software testing whereby the system is tested regarding functional requirement.

Functional testing ensures that the requirement are properly satisfied by client.

Example of functional testing is unit testing, smoke testing, sanity testing, Integration testing, white box testing, black box testing, user acceptance testing, Regression testing.

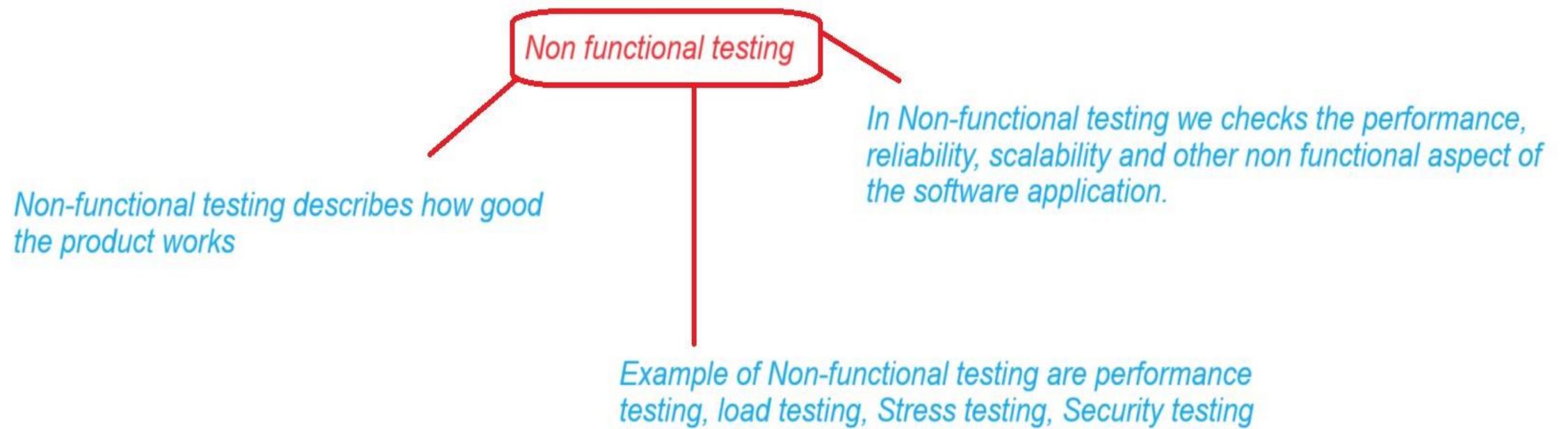
In functional testing , Testers are the follow the following steps

1. First of all testers are analyse the requirement regarding software application.
2. Create the test plan.
3. Create Test case.
4. Make traceability matrix
5. Execute the test case
6. we will create bug report.
7. At the end defect management should do to manage the defect resolving.

*Requirement
Traceability Matrix*

*RTM or Traceability matrix both
are same*

*Traceability matrix is a document in which we trace the
requirement regarding test scenario and test cases.*



Performance Testing

Its evaluate the overall performance/speed of the software application.

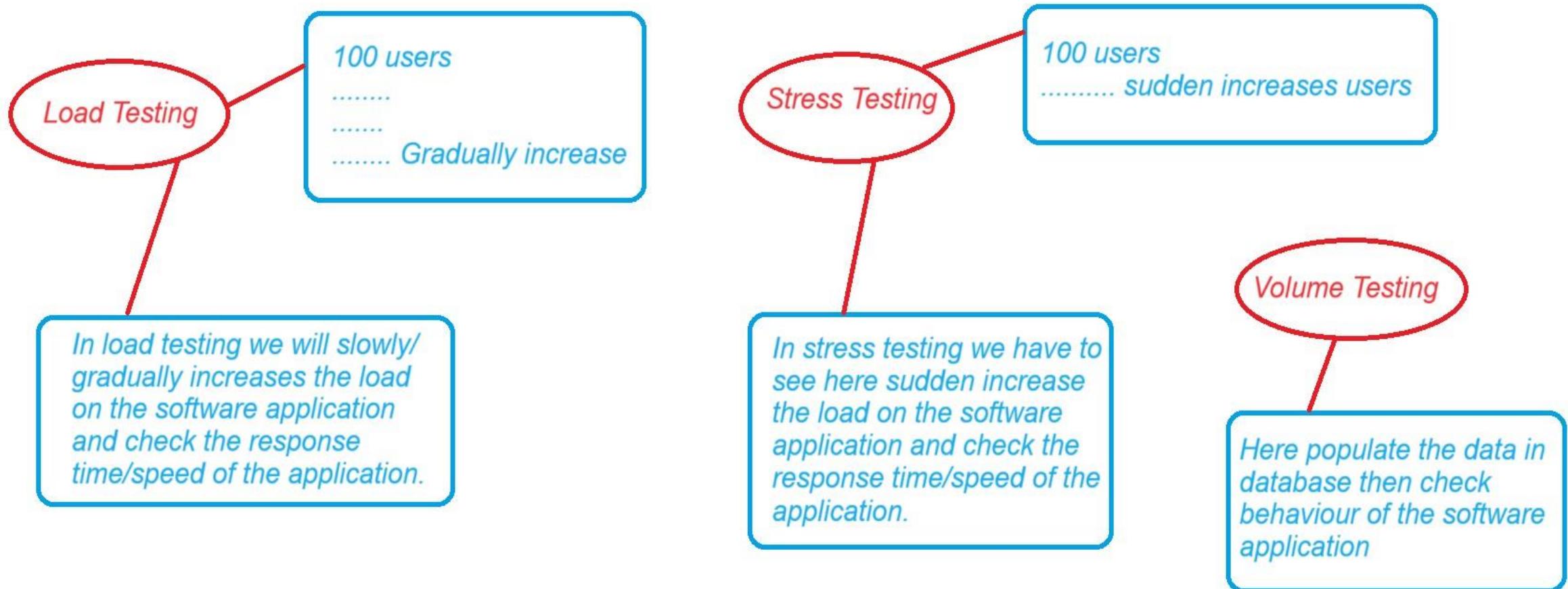
In performance testing we test speed, stability and scalability

Stability - We test behaviour of the software application with increases in users or on load.

Scalability - We check break point of the software application.

Performance testing

- 1. Load testing*
- 2. Stress testing*
- 3. Volume testing*



Non-functional Testing

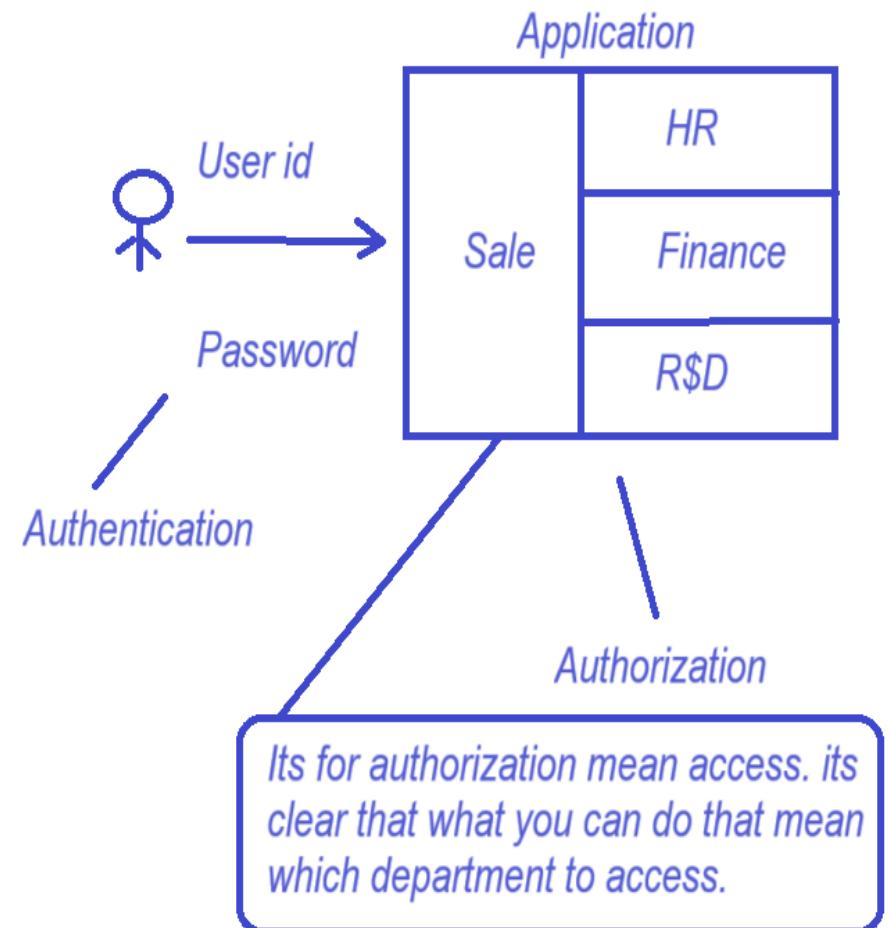


It evaluates how secure is our application that mean security testing includes testing of authentication and authorization.

↓ ↓

who are you what you can do

In security testing ,To ensure there are no loophole the application and leads to no data loss.



In this we ensure that application works without any issues in different environment.

Compatibility testing

Forward Compatibility



To check the behaviour and compatibility of the hardware or software with newer version.

Backward Compatibility



This testing is performed to check the behaviour and compatibility of the hardware or software with their older version.

We check whether the application is compatible with different environment like web browser, hardware platform, databases, operating system networks, different version, configuration etc.

MS word document

MS word document

MS office - 2010

MS office -2016

Backward Compatibility

Forward Compatibility

Recovery Testing

Here checks that application terminates gracefully in case of failure and data is recovered.

Instability testing

Here to be checks, the software installs and uninstalls correctly or not.

Sanity Testing / Garbage Testing



During this testing tester are finding extra functionality in build with respect to customer requirement.



Build refers to a version of a software application that is ready for testing.

User Interface Testing

In this we interact through text. ➡ Command line interface



(CLI)

CLI stands for Command Line Interface. It's a text-based interface used to interact with computer programs by typing commands.

For Example - C:\Users\91703>dir

➡ Inside this we check which directory is there?

Graphical user interface



(GUI)



In graphical user interface, images are used instead of text to interact with application.

GUI stands for "Graphical User Interface". It refers to the visual elements- such as windows, icon, buttons and menus -- that enable users to interact with a computer or software through graphical representation than text based command

Graphical user interface (GUI)



GUI testing is a software testing type in which we check the graphical user interface of the software to ensure the functionality of the software works as per requirement specification.

Checklist for GUI Test



1. we have to check error message and warning message are display correctly.
2. To be check font which used in an application is readable.
3. To be check colour of the hyperlink.
4. To be check for spelling.
5. To be test the scrollbar according to the size of the page.
6. Test the heading whether it is properly aligned or not

Usability Testing or User Experience Testing (UX Testing)



In Usability testing, we check the application for user friendly, efficiency and accuracy.

1. Easy to understand.
2. Easy to access.
3. Faster to access.
4. Good Error Handling.

Usability testing is a method used in design and development to evaluate a product's user interface by observing real users as they interact with it. The goal is to identify usability issues, gather feedback and improve the overall user experience.

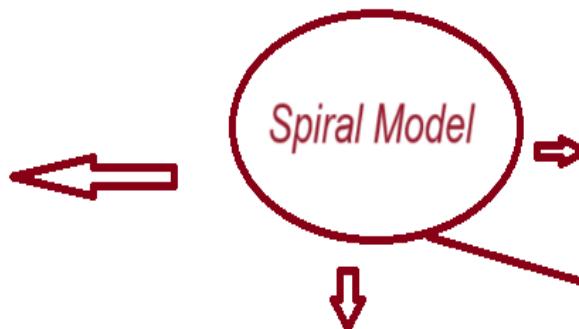
Spiral Model



Disadvantage

This model is not suitable for small project.

- 1. Time taking
- 2. Cost effective



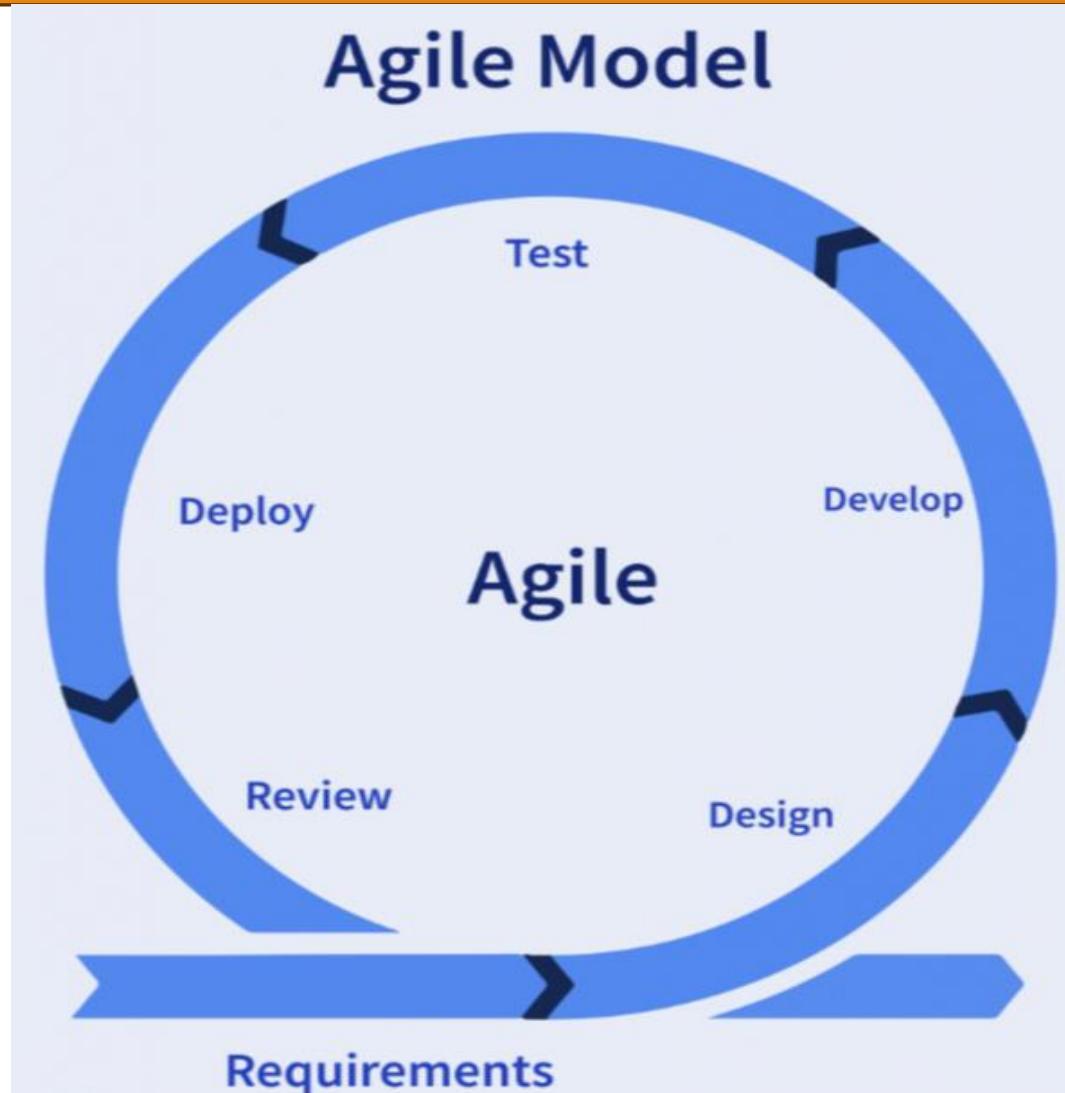
Spiral model is iterative in nature and it is also called version control model. It overcomes drawback of waterfall model.

It allows better risk analysis. The development can be distributed in to smaller parts.

WHY ?

Here client give part wise requirement after development these requirement again client give requirement for example let me client give 10 requirement after that again give 10 requirement that mean we will deliver software product in iterative nature

AGILE MODEL / AGILE METHODOLOGY / AGILE PROCESS



Agile Model / Agile Methodology / Agile Process



It is life cycle of software development. Agile model is combination of iterative and incremental process.

Iterative



Incremental Process

Incremental means add on new feature in to existing software.

Iteration means again and again that mean here deliver the software product in part wise according to client requirement on the basis of priority.

Note - There are some main frameworks of agile model such as Scrum, Kanban, Extreme programming (XP) etc. The framework provides a structured way of applying agile principles for which we use Jira tool.

Agile principles

- 1. Customer no need to wait for long time.
- 2. We will deliver piece of software which contains some functionality according to client requirement on the basis of priority.
- 3. Easily adopt to changes.

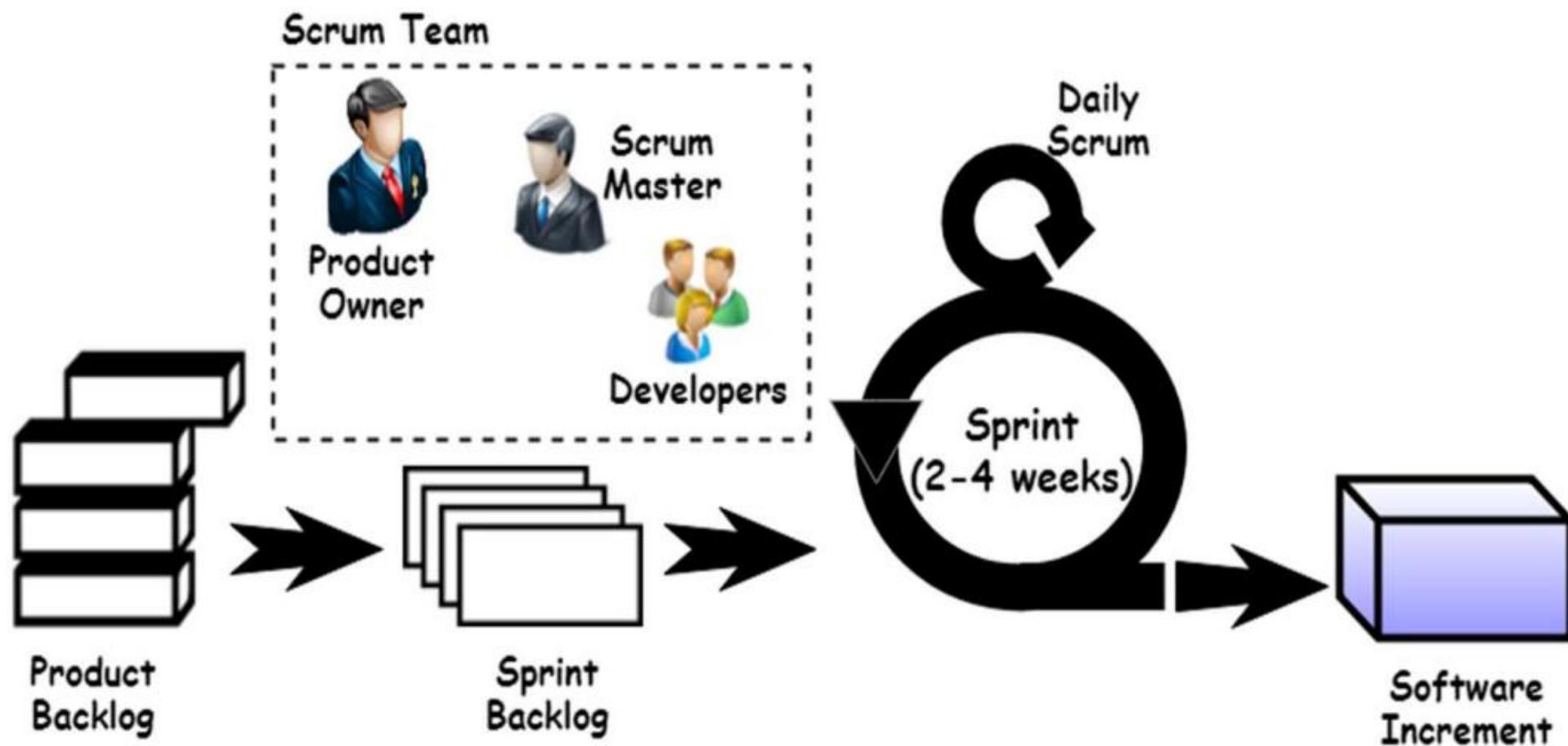
Disadvantage of Agile model

- 1. Lack of comprehensive documentation .
- 2. Dependency on customer availability

Advantage of Agile model

Here we can add on new feature in existing software.

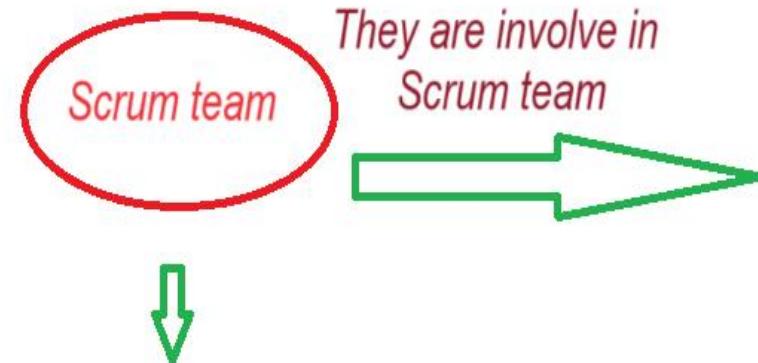
- 1. Release will be very faster.
- 2. Requirement changes are allowed in any stage of document.
- 3. There is good communication between customer, product owner, Developer and Tester.



Scrum master



Scrum master acting as leader. He will facilitating and driving agile process.



At the beginning of each sprint, the team selects items from the product backlog, defining them as the sprint backlog.

Product owner
Scrum master
Developer team



They are involve in Scrum team. Scrum team members decide how to do the work and overcome challenges.

Product owner



Product owner will take a requirement from customer after that he will define the feature of software in the form of Epic or User stories. He will accept or reject work result.

User stories



User stories are concise , informal descriptions of a feature or functionality from the end users that mean in user stories mainly focus on what they want to achieve with software product.

Epic



A collection of multiple user stories.

Product backlog



A list of user stories which is prepared by product owner.

Sprint backlog

It is a subset of product backlog.

Sprint planning meeting

In this meeting define what can be delivered to the team in the sprint duration.

Sprint

It is span of time to complete user stories. It is decided by product owner and team.

Sprint will be changing company to company but minimum duration is 2-4 week.

Daily Scrum meeting (Standup call)

It is conducted 15-20 minutes on daily basis.

In stand up call mainly focus on what did the task completed yesterday, what are the task today and what are the task plan for tomorrow and share if any face challenges or blocker.

This meeting conducted by our senior such as QA manager and Test lead.

Scrum



Scrum is a framework through which we build software product.

Sprint Review



A sprint review is an informal meeting held at the end of a sprint, during which the team shows what was accomplished, while the stakeholders provide feedback.

Scrum framework



The Scrum framework provides a structured and flexible approach to software development. It consists of roles and events also.

**Product owner
Scrum master
Development Team**

Sprint
Sprint planning
Daily Scrum
Sprint Review
Sprint Retrospective



The sprint retrospective is a recurring meeting dedicated to discussing what went well and what can be improved in a sprint. It also gives a chance to recover from a sprint and prepare for the next one.

Burndown



Burndown chart shows how much work is remaining to be completed.

Burnup



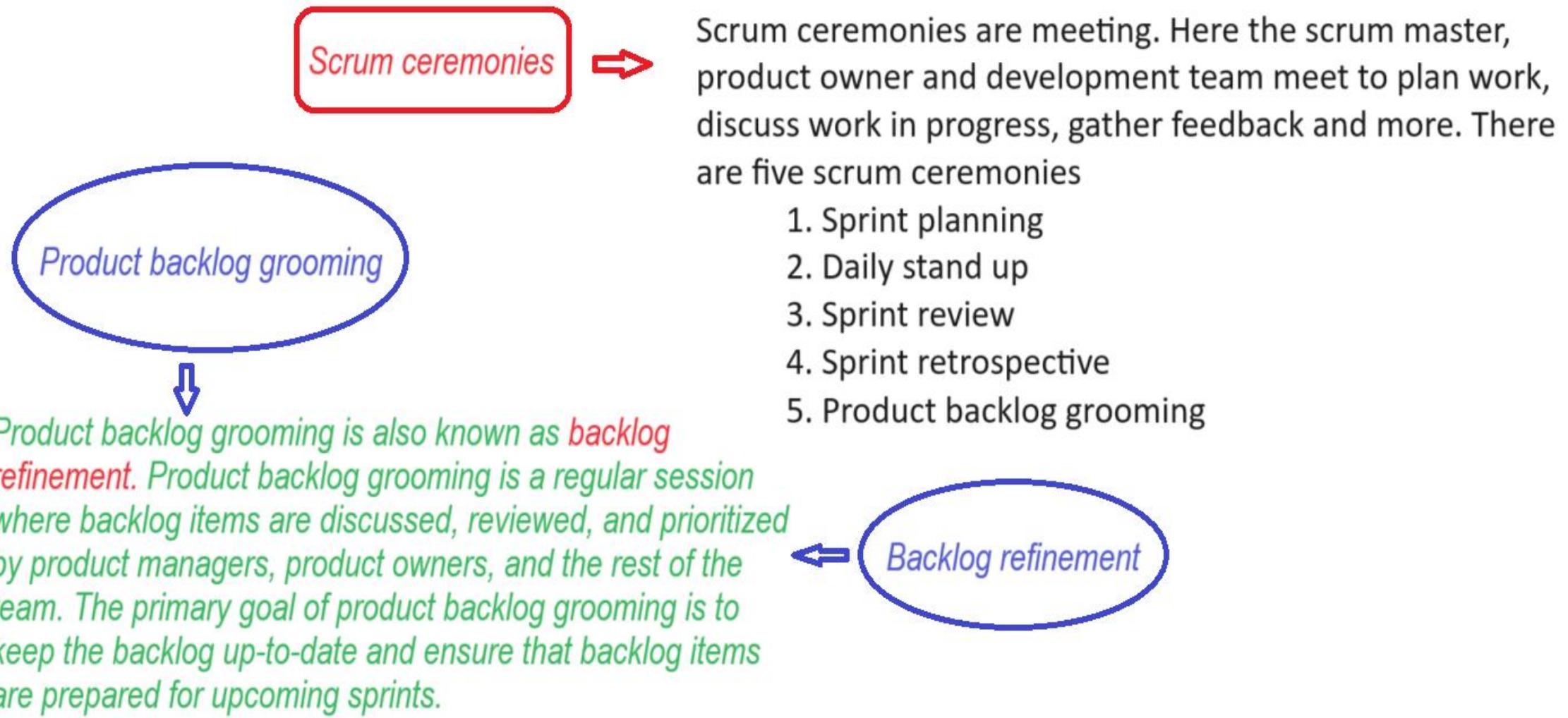
Burnup chart shows how much work has been completed.

Story point

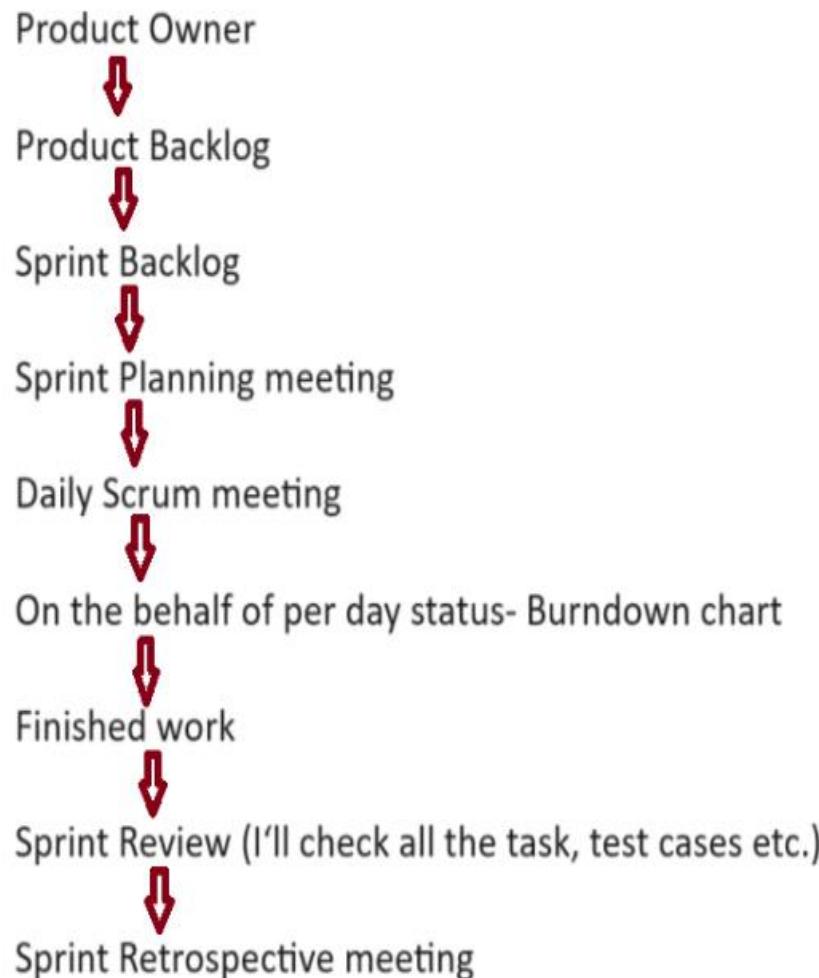


It is the rough estimation of time that mean how much time is required to complete a specific user story.

Story points are given by developer and QA



Work flow of Agile methodology



After product backlog and sprint backlog , there is a sprint planning meeting. In sprint planning meeting task is assign to team in the given sprint. After that there is standup call. In standup call takes 15-20 minutes on daily basis. In standup call focus on what did the task completed yesterday, what are the task today and what are the task plan for tomorrow and share if any face challenges or blocker. After scrum meeting (standup call) discuss to test lead regarding face challenges and blocker. After that There is sprint review and sprint retrospective meeting

Test Design Techniques

Test Design Techniques

Types of Test Design Techniques

1. Boundary value analysis (BVA)
2. Equivalence class partitioning (ECP)
3. Decision table
4. State transition
5. Error Guessing

Test design techniques is help for design the better test case and it helps to prepare the test data.

Test design techniques is comes under black box testing because in this techniques we will not use coding.

By using of test design techniques we cover the each and every area of the software application.

means

Reduce the test data
Reduce the no. of test case
increasing test coverage

Boundary value analysis (BVA)

It is test design techniques, by using these techniques we write test case in systematic way.

Boundary value analysis is a black box testing technique which is used to check the errors at the boundaries of the input domain.

Formula of Boundary value analysis.....

20 to 100

Min - 20 (Valid)

Min-1 - 19 (Invalid)

Min+1- 21 (Valid)

Max - 100 (Valid)

Max+1 - 101(Invalid)

Max-1- 99 (Valid)

Positive test case - When we verify the software with valid data is called positive test case.

Negative test case - When we verify the software with invalid data is called negative test case.

Ques. Let me , you have to test a field which accept age is 18-35. then what is the test data by using BVA.

Equivalence class partitioning (ECP)

It is also software testing techniques or black box testing that divides input domain in to equal classes of data.

By using equivalence class partitioning , we divide the test condition in classes or groups and from each group we check only one condition.

For example,

Let me given range is 20 to 100 then how to use equivalence class partitioning ?

10 to 20 = 18 (Invalid) ➡ Negative test case

21 to 30 = 26 (Valid)

31 to 40 = 38 (Valid)

41 to 50 = 45 (Valid)

51 to 60 = 56 (Valid)

61 to 70 = 64 (Valid)

71 to 80 = 77 (Valid)

81 to 90 = 90 (Valid)

91 to 100 = 96 (Valid)

101 to 110 = 107 (Invalid) ➡ Positive test case

101 to 110 = 107 (Invalid) ➡ Negative test case

Ques. Let me , you have to test a field which accept age is 18-35. then what is the test data by using BVA.

Ques. Let me, you have to test a field which accept age is 18-35. then what is the test data by using ECP.

1 to 17 = 16 (Invalid)
18 to 35 = 25 (Valid)
36 to 53 = 45 (Invalid)

Note - From these techniques reduces no. of test data and save time.

These two techniques is used for input domain testing.

Decision table (Cause-effect table)



This technique is appropriate for preparing test data to test functionalities which has logical relationship.

Using Decision table we can check and verify all possible combination of testing

Ques. If you want to login then you can enter the user ID and password. That mean here given two condition so what about how many test case, you can write?

Ans-

$$\begin{aligned} \text{No. of Test case} &= 2 \\ &= 2^2 \\ &= 4 \end{aligned}$$

No. of condition	Test case/Condition	T1	T2	T3	T4
	USER ID	TRUE	TRUE	FALSE	FALSE
	PASSWORD	TRUE	FALSE	TRUE	FALSE
	TEST CASE	POSITIVE TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE

Test case

1. To verify the login functionality with valid user id and valid password.
2. To verify the login functionality with valid user id and invalid password
3. To verify the login functionality with invalid user id and valid password
4. To verify the login functionality with invalid user id and invalid password

Ques. Suppose we have application like OLA and this application has kept three conditions such as first one is if someone is a new user then he gets 20% discount. second is if someone is a old user then he get 10% discount and if anyone has the coupon code then get extra 10% discount. then according to these condition how many write test case ?

$$\begin{aligned}
 & \text{Ans- No. of Test case} = 2^{\text{No. of condition}} \\
 & = 2^3 \\
 & = 8
 \end{aligned}$$

TEST CASE/CONDITION	T1	T2	T3	T4	T5	T6	T7	T8
N-20%	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE
O-10%	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
COUPON CODE-10% EXTRA	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
DISCOUNT	*	*	30%	20%	20%	10%	*	*
TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE

TEST CASE/CONDITION	T1	T2	T3	T4	T5	T6	T7	T8
N-20%	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE
O-10%	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
COUPON CODE-10% EXTRA	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
DISCOUNT	*	*	30%	20%	20%	10%	*	*
TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	POSITIVE TEST CASE	NEGATIVE TEST CASE	NEGATIVE TEST CASE

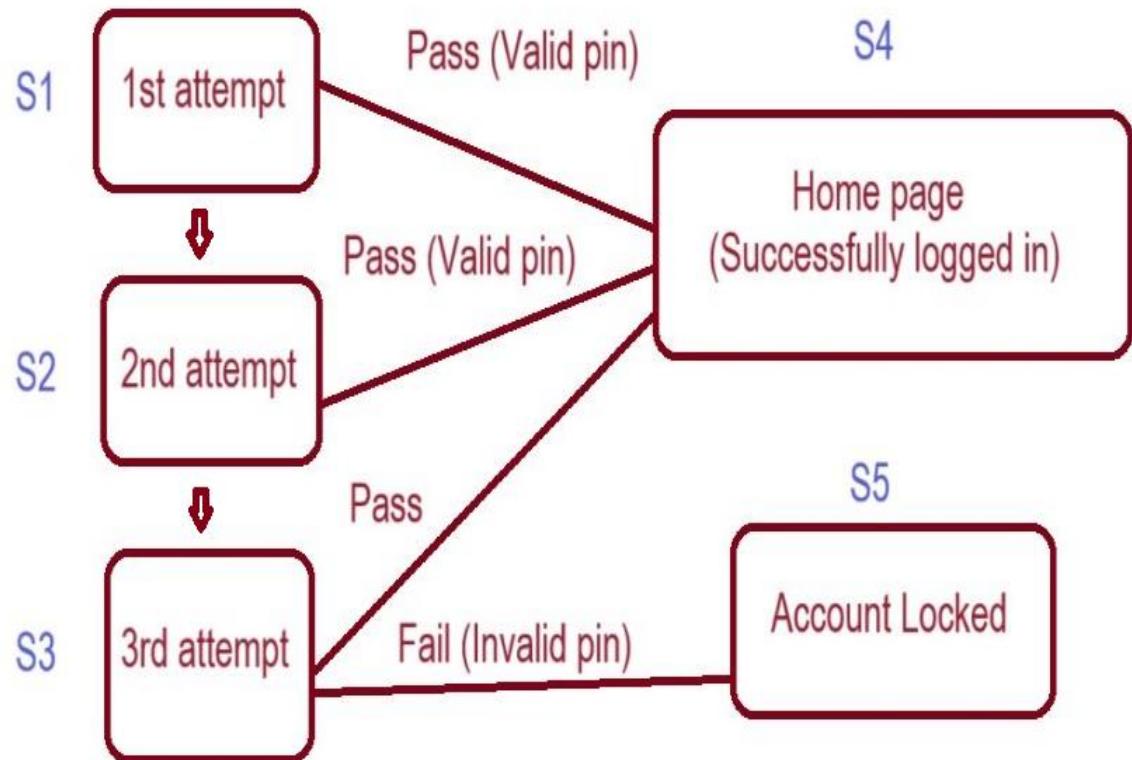
Test case

1. When we verify the functionality with select the new user, old user and select the coupon code
2. When we verify the functionality with select the new user, old user and not select the coupon code
3. When we verify the functionality with select the new user and coupon code
4. When we verify the functionality with select the new user
5. When we verify the functionality with select the old user and coupon code
6. When we verify the functionality with select the old user
7. When we verify the functionality with select the coupon code
8. When we verify the functionality with not select the new user, old user and the coupon code

State transition testing technique



State transition testing techniques is used when features of a system are represented as states which transform in to one another.



State transition diagram

Test case	TC1	TC2	TC3	TC4	TC5	TC6
State	S1	S1	S2	S2	S3	S3
Input	Correct Pin	Incorrect Pin	Correct Pin	Incorrect Pin	Correct Pin	Incorrect Pin
Output	Home Page	2nd Attempt	Home Page	3rd Attempt	Home Page	Account locked
Finished state	S4	S2	S4	S3	S4	S5

State transition table

Test case	TC1	TC2	TC3	TC4	TC5	TC6
State	S1	S1	S2	S2	S3	S3
Input	Correct Pin	Incorrect Pin	Correct Pin	Incorrect Pin	Correct Pin	Incorrect Pin
Output	Home Page	2nd Attempt	Home Page	3rd Attempt	Home Page	Account locked
Finished state	S4	S2	S4	S3	S4	S5

Test case

1. Validate that the system is able to do the transition from 1st attempt to Home page on correct pin input.
2. Validate that the system is able to do the transition from 1st attempt to 2nd attempt on incorrect pin input.
3. Validate that the system is able to do the transition from 2nd attempt to Home page on correct pin input.
4. Validate that the system is able to do the transition from 2nd attempt to 3rd attempt on incorrect pin input.
5. Validate that the system is able to do the transition from 3rd attempt to Home page on correct pin input.
6. Validate that the system is able to do the transition from 3rd attempt to account locked on incorrect pin input.

Error guessing technique

Error guessing is a software testing technique in which the tester has own experience and guess the types of error.



Error guessing technique is used to identify defect that may have been missed by other approach.

STLC

STLC

⇒ Software testing life cycle

It is a systematic process perform of testing. In STLC there are six phases. 1. Requirement analysis 2. Test plan 3. Test case creation 4. Test environment set up / Test bed 5. Test case execution 6. Test closer

1. Requirement analysis - We have to see proper requirement for testing in SRS document.

2. Test plan - Basically in test plan mainly focus on estimation that mean we have to see for build that whose responsibilities, how many team will be needed, whose write test case, whose execute test case, what will environment set up etc all are mention in test plan.

3. Test case creation - In test case creation basically there are set of action in which we test individual functionality.

4. Test environment set up (Test bed) - What is the requirement for testing regarding hardware and software.

5. Test case execution - In test case execution tester will do actual testing.

6. Test closer - During testing there are not found bug in software then we will do close or sign up.

Ques. What is the Entry criteria and Exit criteria in STLC ?

Ans - Entry criteria - Entry criteria gives the prerequisite items that must be completed before testing can begin.

Exit criteria - Exit criteria defines the items that must be completed before testing can be concluded.

ENTRY AND EXIT CRITERIA

Enter your sub headline here

Entry Criteria		Exit Criteria
Functional /non- functional requirement documents	Requirement Analysis	<ul style="list-style-type: none">RTM (Requirement traceability matrix)Automation feasibility report
<ul style="list-style-type: none">Functional /non- functional requirement documents RTM, automation	Test Planning	Test plan, test estimation, test strategy
<ul style="list-style-type: none">Requirements documentsRTM, test plan, test estimation, test strategy	Test Case Development	Test cases, test data
Environment setup plan	Environment Setup	Environment ready for testing
Test cases, test data, test environment	Test Execution	Test results, defect reports
Test completion, defect reports	Test Cycle Closure	Test closure document

Test Case

Test case

It is nothing but how to test.

Overall we can also, it gives details descriptions of our software application.

Test cases is a set of actions executed to verify a particular feature or functionality of your software application.

Facebook

1. To verify valid username and valid password.
2. To verify invalid username and valid password.
3. To verify valid username and invalid password.
4. To verify invalid username and invalid password.

1. To verify that user can create an account.
2. To verify that user can set password.
3. To verify that user length of password is sufficient.
4. To verify that password is having atleast 8 character ,atleast 1 some number and atleast one capital letter.
5. To verify that user can upload and change the profile pics.
6. To verify that user can delete and remove from profile pics.
7. To verify that user can post an image.
8. To verify that post is visible to selected user

Test scenario

Test scenario mean what to test.

Test scenario is defined as any functionality that can be tested.

Opencart

Features of these application are -

1. Register
2. Login and Logout
3. Forgot password
4. Search
5. Product compare
6. Product display page
7. Add to cart
8. Wish list
9. Shopping cart
10. Currencies
11. Home page
12. Home page
13. Checkout page
14. My account page
15. Order History
16. Refund and payment method
17. Download page
18. Contact us page
19. Menu option.....

Test case —

Test case ID
Testcase Scenario
Testcase Description
Test steps
Test Data
Expected Result
Actual Result
Status

Testcase scenario - what to Test
Test case - How to test (set of action)

Component of Defect - *Defect ID*
Defect description
Steps
Testcase Id
Reported Date
Reported by
severity
priority
Build version

Positive Test case- When we verify the functionality with valid data that's called positive test case.

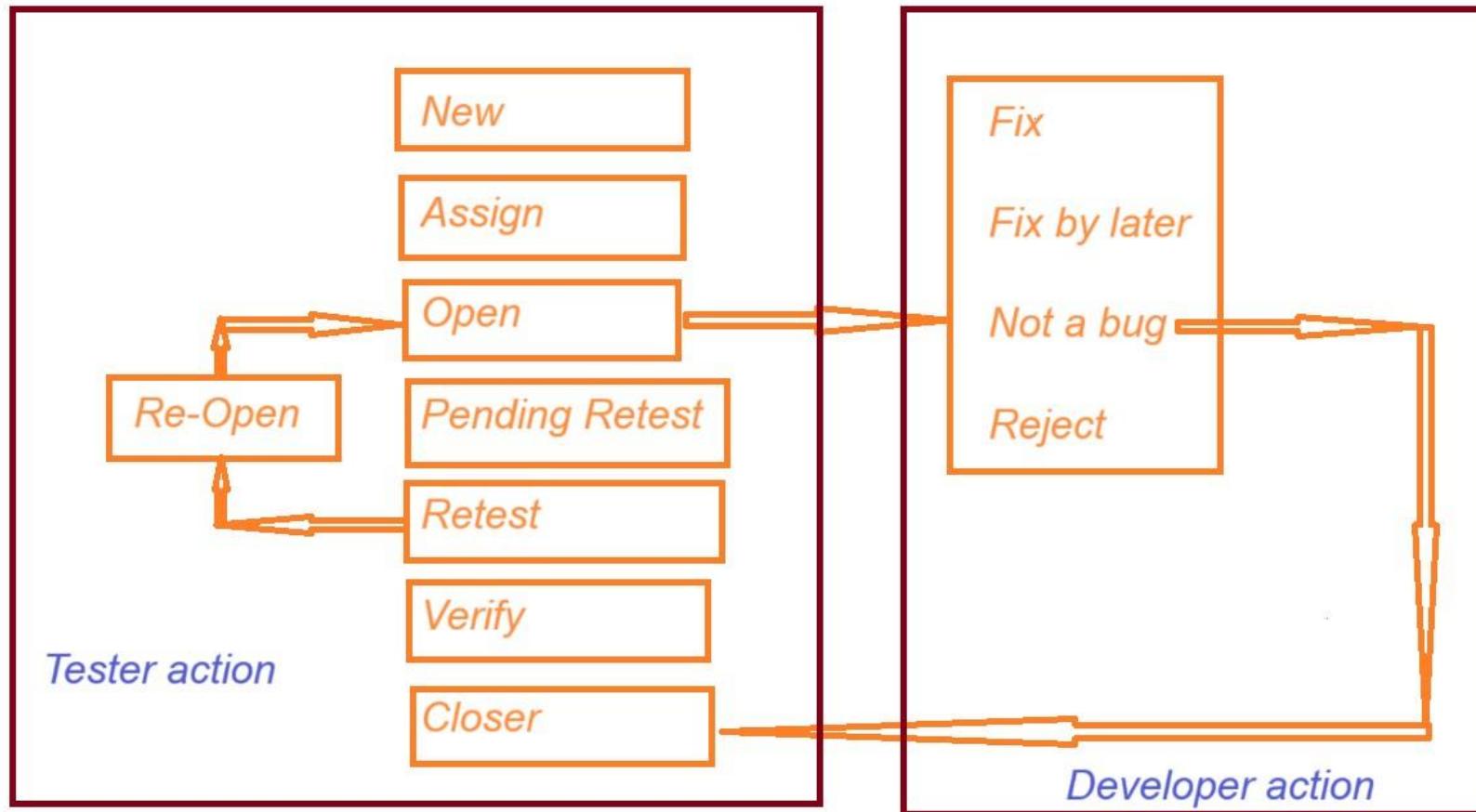
Negative Testcase- when we verify the functionality with invalid data that's called Negative test case.

Test Case ID	Test Scenario	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC_Login_001	Verify the login of gmail	Enter valid user name and valid password	1.Enter user name 2.Enter password 3.Click "login" button	<Valid user name> <Valid Password>	Successful login	Successful login	Pass
TC_Login_002	Verify the login of gmail	Enter valid user name and invalid password	1.Enter user name 2.Enter password 3.Click "login" button	<Valid user name> <Invalid Password>	A message "The email and password you entered don't match" is shown	A message "The email and password you entered don't match" is shown	
TC_Login_003	Verify the login of gmail	Enter invalid user name and valid password	1.Enter user name 2.Enter password 3.Click "login" button	<Invalid user name> <Valid Password>	A message "The email and password you entered don't match" is shown	A message "The email and password you entered don't match" is shown	
TC_Login_004	Verify the login of gmail	Enter invalid user name and invalid password	1.Enter user name 2.Enter password 3.Click "login" button	<Invalid user name> <Invalid Password>	A message "The email and password you entered don't match" is shown	A message "The email and password you entered don't match" is shown	

Manual Software Testing **LIVE PROJECT**



Defect life cycle



Defect life cycle

Defect life cycle

Bug life cycle is also known as defect life cycle. In defect life cycle there are two action.

- 1. Tester action*
- 2. Developer action*

Bug life cycle is a process in which defect goes through different stages in its entire life.

New - When tester find the defect, it is assigned a status as a new.

Assigned - Tester forward defect to the senior software tester or QA manager. Here Our senior analyze the defect or not. If defect is found then assign the bug to the developer team.

Open - After verify of bug its forward to open stage. Then developer start analyzing and works on defect fix.

Fixed - Bug report is given to development team. Firstly developer will check the bug accordingly to priority. when priority is high then developer will fix a bug immediately. and when priority is low then developer will fix the bug later.

Pending Retest - After fix a bug it goes to pending Retest further its goes to Retest.

Pending Retest - After fix a bug it goes to pending Retest. Further it goes to Retest.

Re-opened - In Retest phase when bug is found by tester then again it goes to Re-open phase.

Verify - When bug is not found then it will be verify.

Closer - It will goes to closed phase that means testing will closed.

In new phase tester raise the bug after that assign the bug to senior software tester or QA manager after that our senior analyze the bug then after they forward to open stage. In open stage bug report is given to development team. Firstly developer will check the bug then according to priority developer will fix the bug. when priority is high then developer will fix a bug immediately but when priority is low then developer will fix a bug by later. After fix a bug it goes to pending retest then it goes to retest. If in Retest phase bug is found by tester then it again it goes to Re-opened phase But when bug is not found then it will verify and will goes to in closed phase that means testing will closed.

Duplicate - If the defect is repeated twice or more then status goes to duplicate.

Rejected - If the developer feels the defect is not genuine then it goes to rejected.

Deferred - If the present bug is not priority it means status goes to deferred that means developer will fix by later.

Not a bug - If the defect does not have an impact on the functionality of the application then the status goes to 'Not a bug'.

Severity and Priority

Severity and Priority

It is decided by tester

Severity

In severity describes the impact of the defect on the application.

Types of severity

- Blocker (S1)
- Critical (S2)
- Major (S3)
- Minor (S4)

Degree of impact of bug in the software application during testing.

Blocker (S1)



In blocker, when problem in major flow then testing can not be continued.

Critical (S2)



In critical, when found problem in major feature/functionality then testing can not be continued.

Major (S3)

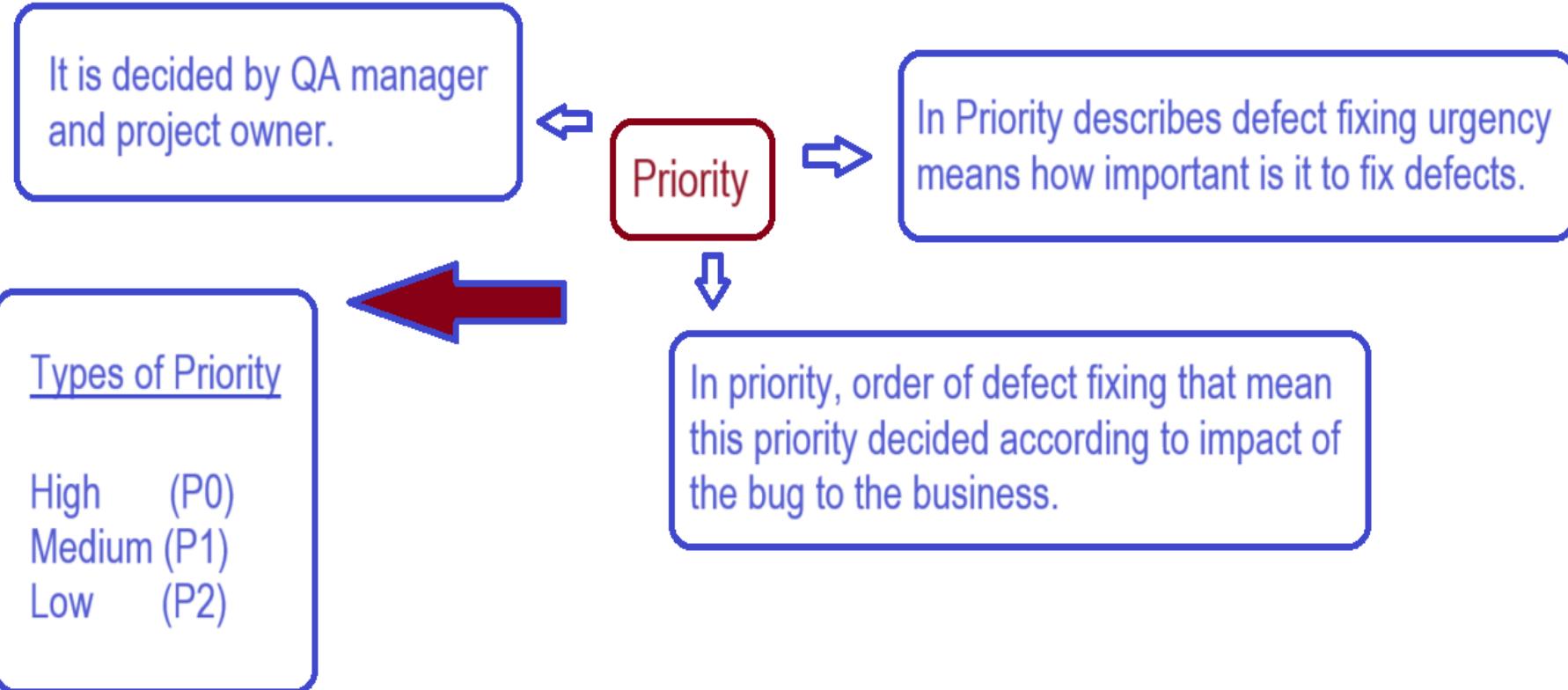


In major, tester found undesirable behaviour.

Minor (S4)



In minor severity, bug doesn't impact on the functionality of the system. Here defect related to look and feel, spelling, alignment.



High priority (P0)



That type of issue which affect the system severely and system can not be used until defect is fixed. High priority defects must be fixed immediately.

Medium priority (P1)



That type of issues which can be released in the next build comes under medium priority such issues can be resolved along with other development activities.

Low priority (P2)



An issues which can be fixed in later.

The Priority-Severity Matrix



1. High priority, low severity
2. High priority, high severity
3. Low priority, low severity
4. Low priority, high severity

1. High priority, low severity



Doesn't effect other module/functionality

High impact on business



For example - Wrong logo of any company, About us link of any company, spelling of any company

2. High priority, High severity



If major functionality of any application is not working then it goes to high priority and high severity.

High impact on business.

For example - Url is not working, login button is not working

3. Low priority, low severity



Business unaffected

Functionality unaffected

For example - GUI issues, colour, look and feel, alignment mistake

4. Low priority, High severity



A rarely occurring scenario affect functionality

For example - Annual report

Regression Testing and Retesting

Regression testing

We test impact of changes on existing software is called Regression testing.

Regression testing is ensure that the product works fine with new functionality, bug fixes or any changes in the existing features.

Types of Regression testing

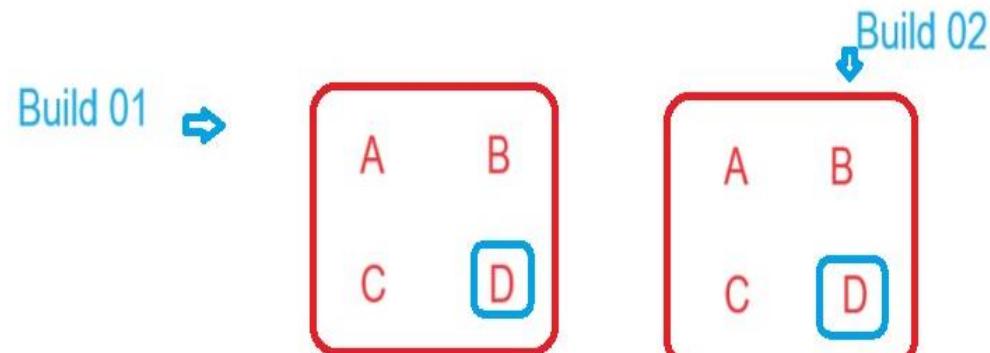
1. Unit Regression
2. Partial Regression
3. Complete Regression

Way of Regression Testing

1. Retest all
2. Selective Testcase
3. Priority Testcase

Unit Regression

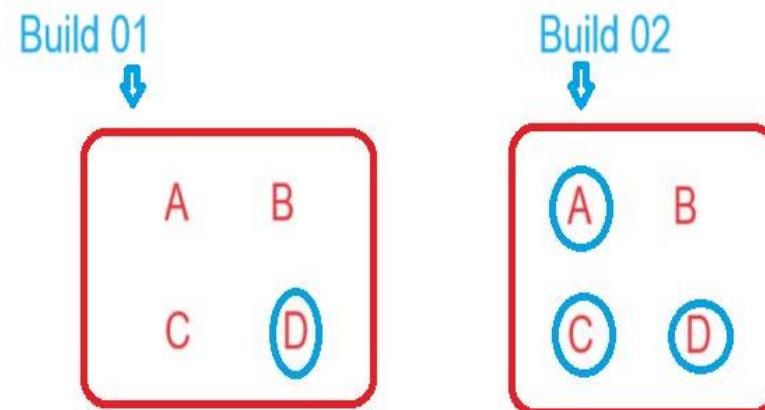
Testing only the changed feature is called the unit regression testing.



Partial Regression

Regional Regression

Here we test the modification along with the impact area.

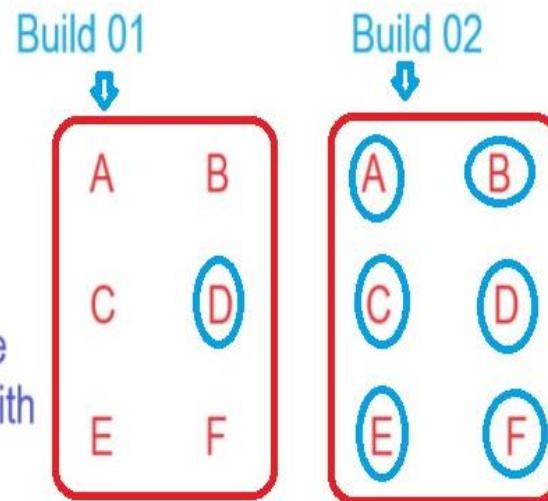


Impact analysis meeting

Here to find the impacted area during impact analysis meeting.

Complete Regression

In complete regression testing, we test the modification area along with the remaining areas.



Build

When the code of software application is written and converted in to executable (runnable) form that's called build.

Ques. On which build, will regression testing start ?

Ans - From Build 02

Ques. If you do regression testing, what is the worst scenario for you ?

Ans - The toughest work in regression testing is to decide which test case we execute and which test case we leave because we have to give output after executing minimum test case.

Retesting

➡ Retesting is testing a specific bug after it was fixed.



Retesting means testing the functionality or bug and again to ensure that bug is fixed or not

Build 01 ➡ Test cycle 1

TC1	Pass
TC2	Pass
TC3	Fail
TC4	Pass

Build 02 ➡ Test cycle 2

TC1	✗
TC2	✗
TC3	✓
TC4	✗

In this module found bug.

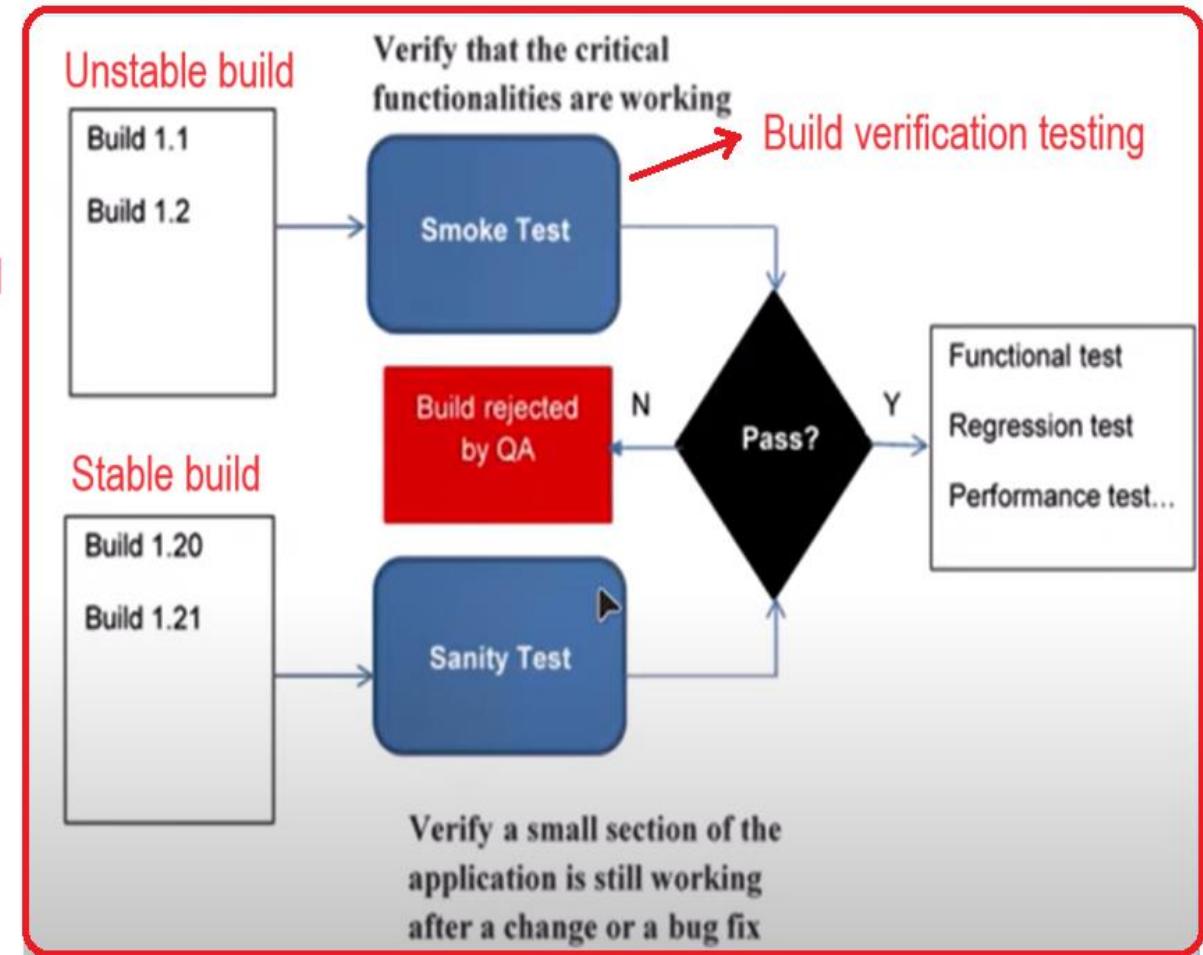
Ques. Which one will performed firstly Regression or Retesting ? Why

Ans - Retesting is performed before regression testing because retesting makes sure the detected bug is fixed, while regression testing checks the application for defects found after a code change.

Smoke and Sanity Testing

Smoke testing - 1. Smoke testing is performed after software build to find out the critical / basis functionalities of the software. This is to check whether software is testable or not.
2. It is performed by QA team but can be performed by developer also.
3. Smoke testing is performed on initial unstable builds and verifies whether the software is testable or not.

Sanity testing - 1. Sanity testing is performed on stable build to verify the basic features of the software is working or not.
2. It is always performed by QA team.



Quality Assurance and Quality Control

QA and QC

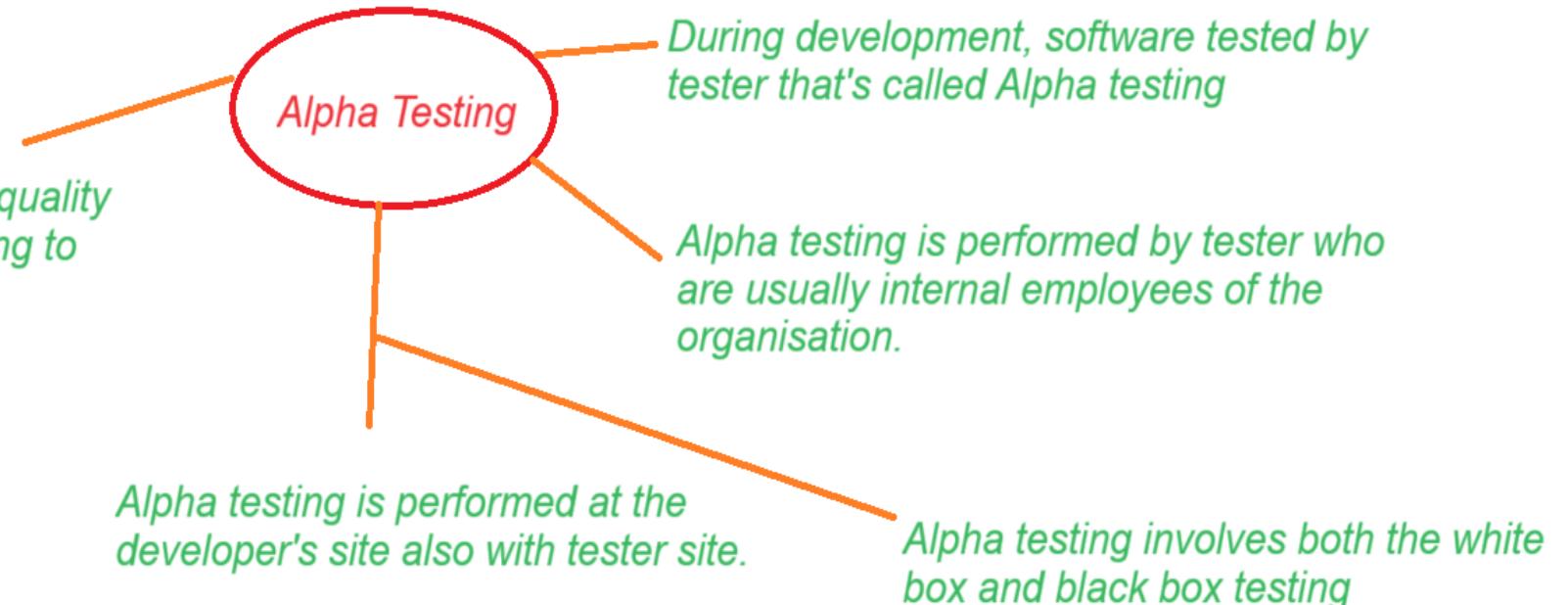
QA means quality assurance. Basically in quality assurance is talked about complete process. Complete process means in development phase Requirement, Designing, Development, Testing, Deployment and maintenance all process are comes in QA.

QC means quality control. Quality control only focus on testing activity. In qc all the tester part have been done.

Alpha Testing and Beta Testing

Alpha testing and Beta testing

Alpha testing ensures the quality of product before forwarding to beta testing.



Beta testing commonly uses black box testing

Beta testing

Beta testing is performed at the end user of the product

After completed testing, Software is tested by selected user by the client side, that's called Beta testing.

Beta Testing is performed by clients, who are not part of the organisation.

Alpha Testing and Beta Testing

Alpha testing ensures the quality of product before forwarding to beta testing

Alpha Testing

The tester who test the software during development is called alpha testing

Alpha testing is performed by tester who are usually internal employs of the organisation.

Alpha testing is performed at the developers site also with tester site. that why we can say in this testing involves both white box and black box testing

After the testing is completed, when selected users test the software by the client side is called beta testing

Beta Testing

Beta testing is performed by clients, who are not part of the organisation

Beta testing commonly uses black box testing

Beta testing is performed at the end user of the product

Adhoc Testing, Exploratory Testing \$ Monkey Testing (Gorilla Testing)

Adhoc Testing, Exploratory Testing \$ Monkey Testing (Gorilla testing)

In adhoc testing mostly negative test scenarios are tested with aim to break the system.

Adhoc Testing

Performing random testing without any plan is known as Adhoc testing.

In adhoc testing, there is no documentation, no test design and not test cases. These adhoc testing totally depends on capability of tester that mean tester have depth knowledge about the system.

Adhoc testing is informal testing process. It is performed after formal testing.

Exploratory Testing

Exploratory testing is performed when requirements are missing. In this testing firstly we explored the application under test, we understood the flow of application then test case is designed and test cases are executed.

Monkey Testing / Gorilla Testing

Monkey testing can be performed by an individual who does not have a good knowledge of the application. These testing is randomly test in any application that mean here tester test randomly by clicking on random objects and entering the random and invalid data to check is the application give an error or not.

Agile Test Management and Bug Tracking/Reporting With JIRA



What is jira tool



Jira is a product management tool in which we track project work status, Team work and also track bugs and issues.

Why we need jira tool

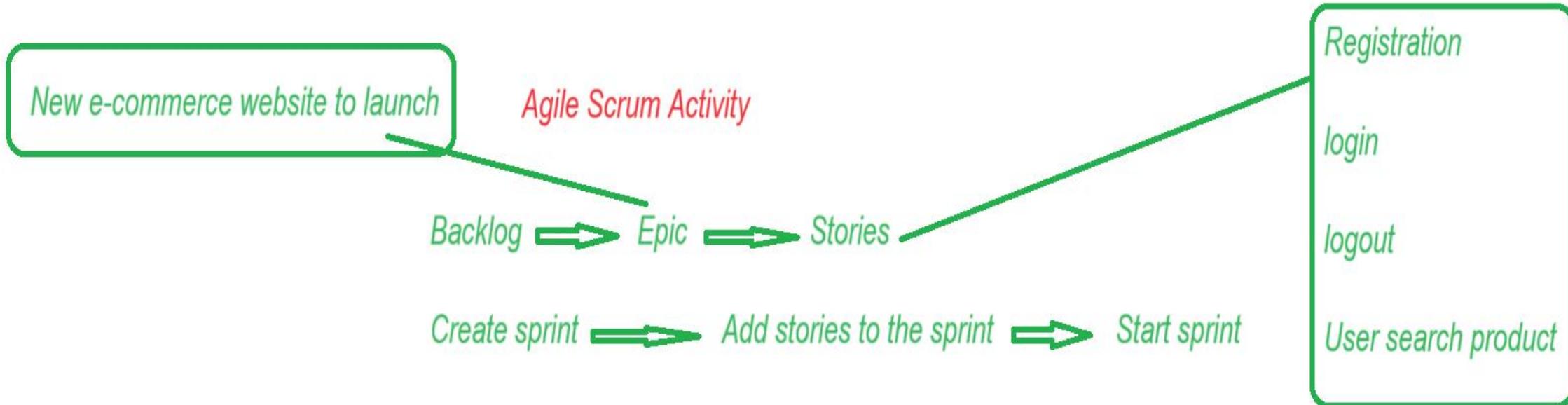


- 1.No need to make excel or any other document.*
- 2.Project work and bugs can manage easily.*
- 3.Here multiple project can handle.*
- 4.Team can communicate over the project*

Who use Jira tool



Jira tool is used by software developer, QA which are called testing team, product owner and also used by Team manager and management.



Manual Software Testing **LIVE PROJECT**

