

# SOFTWARE TESTING

## MANUAL TESTING

1. what is software testing? All definitions
2. what is difference between manual testing and automation testing?
3. what is SDLC explain SDLC?
4. models of SDLC?
5. why testing is important?
6. explain white box testing black box testing Grey box testing?
7. explain functional testing, integration testing, system testing?
8. explain retesting and regression testing with example?
9. what is scenario explain types of scenario?
10. give example for positive scenario and negative scenario for one object and for one application?
11. tell about different testing tools(test management tools, functional testing tools, Performance testing tools?)
12. explain STLC?
13. explain defect life cycle (or) bug life cycle?
14. explain severity and priority?
15. explain defect report?
16. difference between scenario and test case?
17. explain smoke testing and sanity testing?
18. explain compatibility testing ? its types?
19. explain performance testing ? its types?
20. explain globalization testing ? its types?
21. explain usability testing ?
22. explain accessibility testing ?

23. explain acceptance testing ? its types?
24. explain test case template?
25. explain test case design techniques?
26. explain static and dynamic testing?
27. why did you testing?
28. how will you prove you are a good test engineer?
29. explain exploratory testing?
30. explain ADHOC (or) monkey testing?
31. explain types of integration testing?
32. tell about RTM? Types of RTM?
33. which are white box testing techniques and black box testing techniques?
34. explain stubs and drives?
35. explain test plan and its attributes?
36. explain difference between QA & QC?
37. what is defect, bug, error, issue and failure?
38. what is build explain?
39. what is a release?
40. what is a test bed?
41. have you done any real time testing explain it?
42. explain security testing and its types?
43. difference between functional testing and non functional testing?
44. Tell about principles of software testing?
45. explain test methodologies?
46. what is the difference between developers and test engineers?
47. what is test coverage? How will you ensure that everything is covered in testing?
48. what is API testing?

49. what is fuzz, recovery, aesthetic testing?

50. what is Hot fix?

## **SDLC:**

It is software development life cycle

It has difference stages

- Requirement collection
- Feasibility study
- Design
- Coding
- Testing
- Installation
- Maintenance.

## **TYPES OF SDLC MODELS:**

- **Waterfall model** - when customer freezes the requirement we go for this model.
- **Spiral model** - when customer gives the requirement part by part we go for this model.
- **V & V model** - verification and validation when customer needs high quality product we go for v&v model.
- **Prototype model** - when customer is confused with his requirement we go for this model.
- **Agile model** - when customer want application very fast with less time we go for this model.

## **STLC:**

It is software testing life cycle

It is a procedure to test a software application

It has different stages like:

- System study
- Prepare test plan
- Write test case
- Prepare RTM ( requirement traceability matrix)
- Execute test case
- Defect tracking
- Test execution report
- Retrospective meeting or post mortem meeting.

## **DEFECT LIFE CYCLE/BUG LIFE CYCLE/STATUS OF DEFECTS:**

**New or open --> assign --> fixed --> retesting --> closed or reopen.**

- Once test engineer finds a defect the status of defect will be new or open.
- Once it is assigned to developer the status is assign.
- Once developer fixes the defects the status is changed to fixed.
- Once it is fixed test engineer should do retesting and if defect is fixed properly the status is closed if defect is not fixed properly the status is reopen.

## **SOFTWARE TESTING:**

- Testing an application along with its functionality based on customer requirement is software testing.
- It is testing an application with intent of finding defects.

## **MANUAL TESTING:**

Testing an application without using tool.

## **AUTOMATION TESTING:**

Testing an application using tool (selenium).

## **WHY TESTING IS IMPORTANT?**

Every application is developed for business purpose if testing is not done it can lead to defect which can be founded by end user (public).this will appeared negative in the market, due to this many people will stop using an application, to avoid all this things. testing has to be done so testing is very important.

## **WHAT IS WBT,BBT,GBT?**

**WBT (White box testing)** - it is testing source code of an application it is done by developer.

**BBT (Black box testing)** - it is testing user interference of an application it is done by test engineer.

**GBT (GREY box testing)** - it is testing both source code and user interference of an application it is done by a person who is good in testing source code and user interference both parallely.

## **TYPES OF BLACK BOX TESTING:**

**Functional testing** - testing each and every component indepentally and throughly with respect to requirement.

**Integration testing** - testing the data flow between two or more dependent module.

**System testing** - testing an application from end to end just like a real user. It is done in a testing severs limitation to production server.

## **SMOKE TESTING/CONFIDENCE TESTING/BUILD VERIFICATION TESTING:**

Testing the basic or critical features of an application before doing through testing like functional testing , integration testing , system testing.

Here we check positive scenarios.

## **SANITY TESTING:**

Testing the new features or bugs fixes of an application.

## **ADHOC TESTING / MONKEY TESTING:**

Testing an application randomly without following requirement.

Here we check negative scenarios.

## **EXPLORATORY TESTING:**

Testing an application by exploring it thoroughly when there is no requirement we can do this testing.

## **PERFORMANCE TESTING/SCALABILITY TESTING :**

Testing the response time of an application by applying load.

### **TYPES OF PERFORMANCE TESTING :**

- Load testing
- Stress testing
- Volume testing
- Soak testing.

## **COMPATIBILITY TESTING :**

Testing an application with different hardware and software.

### **TYPES OF COMPATIBILITY TESTING :**

- Hardware compatibility
- software compatibility
- mobile compatibility
- browser compatibility.

## **GLOBALIZATION TESTING :**

Testing an application which is developed for different languages.

### **TYPES OF GLOBALIZATION TESTING :**

- I 18 N-INTERNATIONALIZATION
- L 10 N-LOCALIZATION.

## **USERABILITY TESTING :**

Testing the user friendliness of an application.

## **RELIABILITY TESTING :**

Testing the functionality of an application for a long duration of time.

## **ACCESIBILITY TESTING/ADA TESTING/508ACT TESTING:**

Testing the application which is developed for physically challenged people.

## **ACCEPTANCE TESTING :**

Testing the business scenarios of an application which is done by customer.

## **TYPES OF ACCEPTANCE TESTING :**

➤ Alpha

➤ Beta

## **RETESTING :**

Testing defect fix of an application. It is done after developer fixes the defect.

## **REGTESSION TESTING :**

Testing the IMPACT AREA of an application.

A) After defect is done

B) After changes done in the application

(changes can be adding , modifying and deleting a feature).

## **STATIC TESTING :**

Testing which is done on document is static testing.

(documents can be requirement , design , test scenarios , test cases etc).

## **DYNAMIC TESTING :**

Testing which is done on application is dynamic testing

(EX: WBT , BBT , GBT all comes under dynamic testing).

## **SECURITY TESTING :**

It is to check whether all the DATA and RESOURECES of an application accessible only for authorized users and it is highly secured.

Here we focus on hardware , software and network securities.

### **TYPES OF SECURITY TESTING :**

- Vulnerability scanning
- Security scanning
- Penetration testing
- Risk assessment
- Security auditing
- Ethical hacking
- Posture assessment.

## **SEVERITY :**

Severity will tell how much the defect is affecting/impacting the customer bussiness.

### **TYPES OF SEVERITY :**

- Blockers / shows topper
- Critical
- Major
- Minor
- Trivial

**OR**

- high
- Medium
- Low

Severity is used in defect report while raising the defect

.



## **PRIORITY :**

Priority says which defect has to be fixed first by the developer.

For every defect we have to set priority.

### **Different types of priority**

- high
- Medium
- Low

OR

- urgent
- high
- Medium
- Low

OR

- P1
- P2
- P3

## **SDLC :**

It is software development life cycle

It is a procedure to develop the software application

It has difference stages

- Requirement collection
- Feasibility study
- Design
- Coding
- Testing
- Installation

➤ Maintenance.

➤ **Requirement collection / gathering :**

- Usually customer gives the requirement in the form of CRS (or) BRS that is converted into SRS by BA.
- Collecting the requirement and giving it to software is requirement collection.

**CRS** - Customer requirement specification

**BRS** - Business requirement specification

**BA** - Business analyst

**SRS** - Software requirement specification

**FRS** - Functional requirement specification.

➤ **Feasibility study :**

All the team member gather together to like HR , architecture , Finance , Project manager and business analyst and discuss whether we can continue the project or not.

**HR** - he involved in hiring the resources

**Architecture** - architecture involves in design and technology

**Finanace** - they think about buget and profit.

**Project manager** - project manager will take the final decision based on the discussion with other team member.

**Business analyst** - business analyst will explain the requirement to all team member.

➤ **Design :**

In design we have two types

**HLD** - High level design

**LLD** - Low level design

**HLD** - HLD is like a blue print it tells about external architecture of an application.

**LLD** - LLD will tell about internal architecture of an application.

- Design will be done by architecture and senior developer
- It is a document which is prepared based on requirement.

### ➤ **Coding :**

- Once design is done developers will write the source code to develop the application based on the requirement and design.
- Coding is done in development server.

### ➤ **TESTING :**

- Once application is given by developer, testing will be done on the application based on the requirement and it will find defects (or) bugs.
- Testing is done in testing server.
- Once application matches the requirement and there are no bugs, then we go for stages.

### ➤ **INSTALLATION :**

Once the application is ready it has to move from testing server to production server for end user to use. This process is called installation or deployment.

### ➤ **MAINTAINANCE :**

Once application went to end users, if they find any bugs/defects. It will be informed to software company again company should provide support or service that is maintenance.

Here we have free service and paid service

**Free service** - free service will be limited time as per the agreement

**Paid service** - later it will be paid service.

During maintenance defect fix will be done and changes will be handled.

(changes can be adding, modifying, deleting).

### **What is server :**

- A server is a computer with high configuration.

(i5 processor, 100TB hard disk, 800GB ram)

- It is a super computer
- A server accepts the request
- Recognizes the user (based on IP address)
- Sends the response

Request goes and response come from the internet.

### **TYPES OF SERVER :**

- Development server (used by developer)
- Testing server (used by test engineer)
- Production server (used by end user or public)

We have three different server because it should not disturb public when any changes is happening to have privacy for end users to avoid disturbance.

### **MODELS OF SDLC :**

#### ➤ **WATERFALL MODEL :**

- It is one of the oldest model in SDLC.
- It is the mother of all model.

#### **When do we go for waterfall model**

- when customer freezes the requirement we go for this model.
- For any short term project.
- For developing simple application

EX: calculator , calender , notepad etc.

#### **Advantages of waterfall model :**

- We can expect stable application (if requirement does not change)
- There will be no disturbance for the team members if the requirement does not changes.

#### **Drawback of waterfall model:**

- Testing happens only after coding requirement and design are not tested.
- Developers used to do testing before (currently it is done by test engineer)
- If requirement changes it leads to lot of rework.

Currently waterfall model is used very less in IT industry.

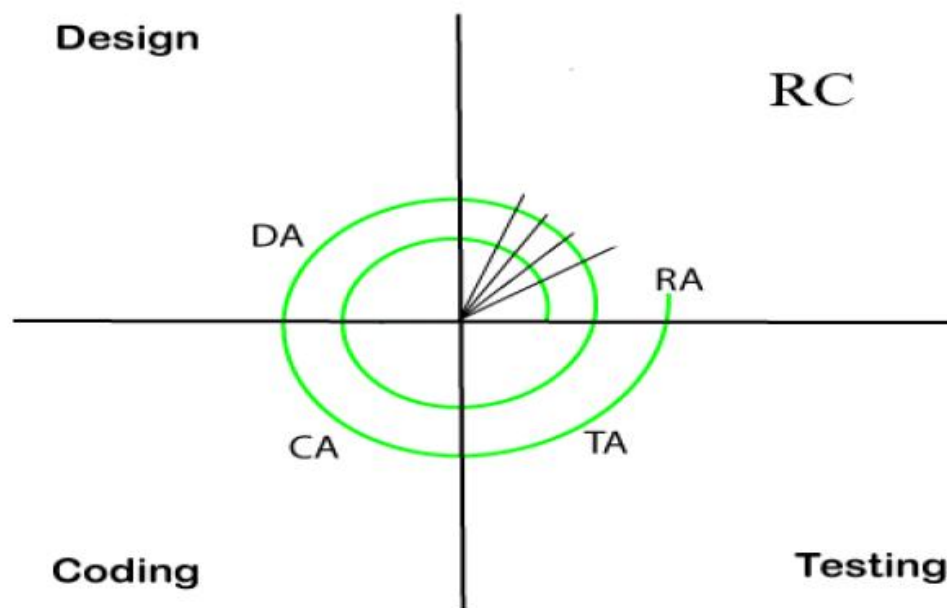
### Why developers should not involve in testing now days.

- Developer always focus how to build the software not to break the software.
- They always be over confidence what they do.
- They does not like to find their own mistake since it is not easy to find their own mistake even though mistake is there they will hide it.

### Why requirement changes :

- To stay in the business for customer.
- To face the competition.
- To upgrade the technology.

### ➤ SPRIAL MODEL / INTERATIVE :



A- Login module

B- Compose module

C- Inbox module

D- Sent module.

- Once login module is ready customer will check login module.
- If he is happy he will give requirement for compose module.

## **When we go for spiral module**

- when customer gives the requirement part by part we go for this model.
- When there is a lot of dependency between the module.

### **Advantages of spiral model:**

- Customer can see the application partially and can get confidence.
- Requirement changes can be done.
- Compare to waterfall model rework is less.

### **Drawbacks of spiral model:**

- Testing happens only after coding requirement and design are not tested.
- Developers used to do testing before (currently it is done by test engineer)
- If there is any requirement changes it may delay the project.
- Customer can cancel the project if he is unhappy.

### **Requirement changes is two types :**

- Major
- Minor

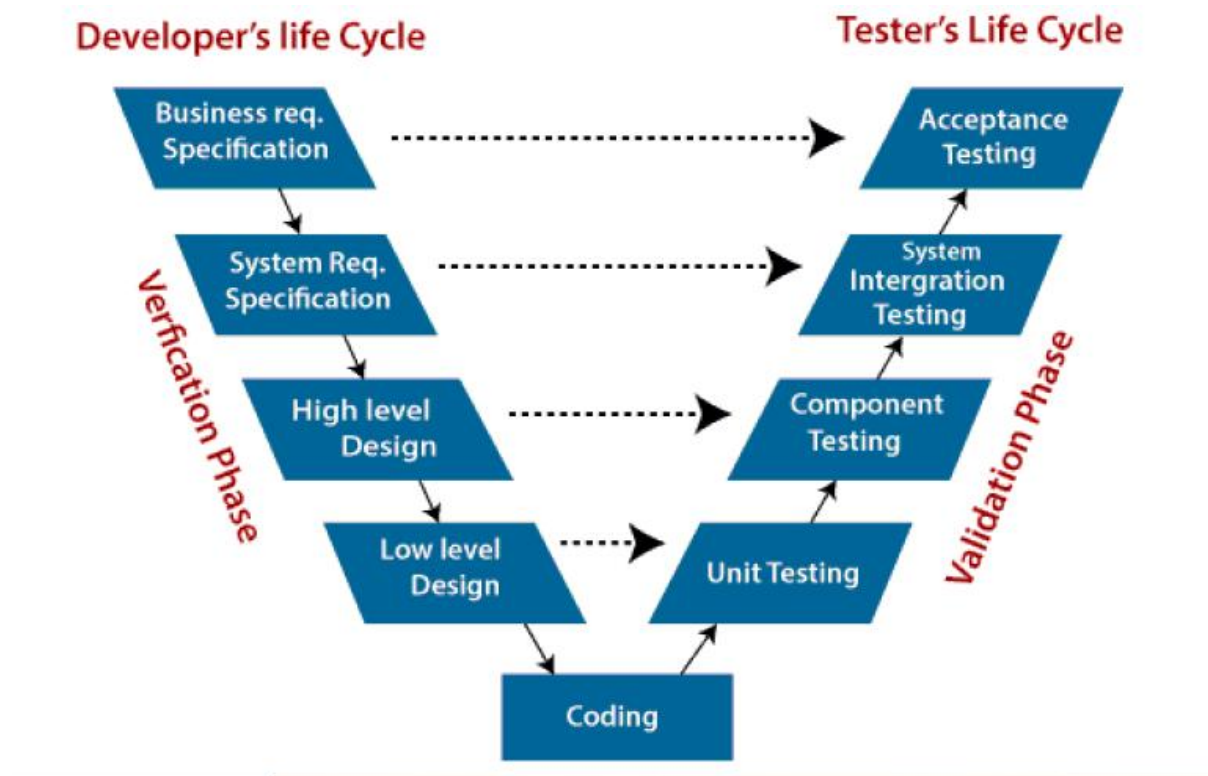
#### **Major :**

Whenever major changes we can work only on the changes we cannot work on the new module.

#### **Minor :**

Whenever there is a minor changes we can work on changes and also on new module.

## V & V MODEL (Verification & Validation model) :



- V & V means verification and validation model.
- Verification is a process of checking “are we building product right”.
- Validation is a process of checking “are we building right product”.
- V & V is done only by test engineer.
- Verification is QA process.
- Validation is QC process.

### When do we go for V & V model :

- When the customer needs high quality products.
- For complex application (EX : banking , health care , space application , airlines , navy application).
- For long term projects (usually more than one year).

### Advantages of V & V model:

- Testing is started at initial stage (document check)

- Requirement and design are tested.
- The downward flow of defect is less.
- Requirement changes can be done.
- Quality will be high compare to other models.
- Rework will be less.

#### **Drawbacks of V & V model :**

- Documentation work will be more.
- Too much of resource are needed.

#### **Explain V & V model :**

- Verification and validation model is one of the best model in SDLC.
- In this model done by parallely development and testing.
- The left side done by development and right and middle is done by testing team.
- When the customer gives the requirement in the form of CRS it is converted into SRS by BA at the same time review CRS is done by test engineer.if there any mistake in CRS will be send to customer.
- Parallely we prepared test plan and test case.
- Once development process is done with design and coding the software is ready for testing.
- First testing is WBT. This testing done by developer. Then functional testing , integration testing , system testing is done by test engineer. At the same time the execution of the test cases are also done.
- After functional testing , then it is done , later system testing is done. Then automation testing is done by customer then it is released to end users.
- The review of the document is verification and it is based on QA.
- The testing of application is validation and it is based on QC.



### **Difference between QA & QC :**

<b>QUALITY ASSURANCE (VERIFICATION)</b>	<b>QUALITY CONTROL (VALIDATION)</b>
It is a process oriented	It is a product oriented
It is a verification process	It is a validation process
It is a preventive method	It is a detective method
Here we check “ are we developing product right”	Here we check “ are we developing right product”
It involves only static testing	It involves only dynamic testing

### **PROTOTYPE MODEL :**

It is a dummy model and is non working application

#### **When do we go for this model**

- When the customer is not clear about requirement.
- When the software company is new to the domain, then they will go for prototype model.
- When the developers are new to the technologies.
- When the customer and software company are new to the business.

#### **Advantages of prototype model :**

- Initially customer can get to know what we get at last.
- Initially itself developers will also come to know what they should deliver at last day.
- Requirement changes can be done initially itself
- Initially investment is very less.

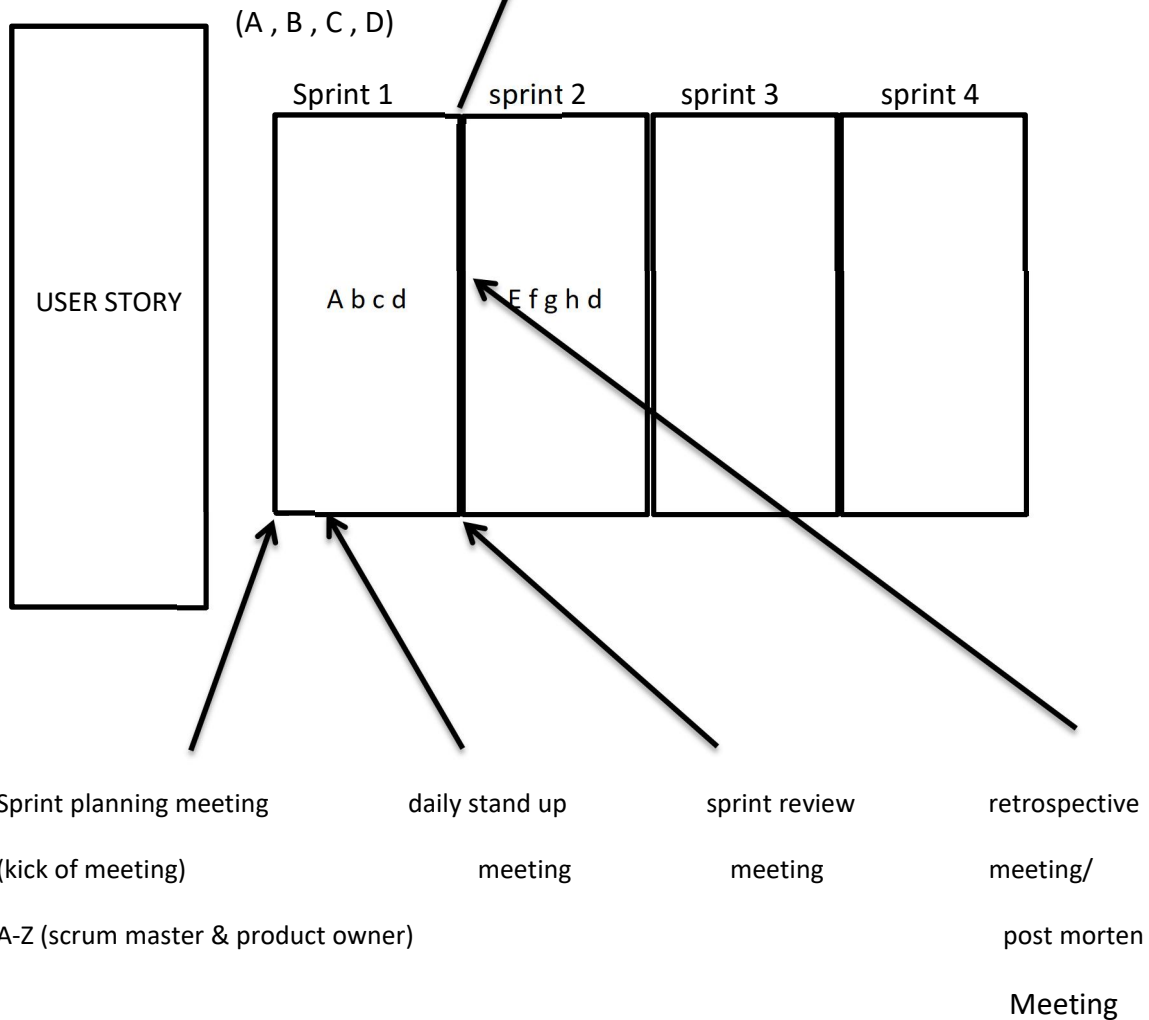
#### **Drawbacks of prototype model :**

- There will be delay in the actual start at the real project.
- Investment is done on non working product.
- Too many changes can disturb the rhythm of the company.

## AGILE MODEL :

Epic (requirement )

(Product backlog)



Agile model is one of the best model of SDLC currently.

### Types of version we are using in agile

- Scrum (80% - 90% industry uses this)
- Xp (Extreme programming)
- RAD (rational verified process)
- KANBAN process etc.

Currently we are using scrum

## **Scrum terminology**

Scrum is a version of agile.

- Epic
- Product backlog
- Sprint backlog
- User story
- Sprint planning meeting
- Sprint daily meeting
- Sprint review meeting
- Retrospective meeting / postmortem meeting.

### **Epic :**

It is complete set of requirement given by customer

EX: A-Z

Epic is also called product backlog

User story / story card

It is a part of the requirement

Example:

A (Login) - user story

B(compose)-user story

C(inbox)-user story

### **Sprint :**

- It is the duration time taken to work on 1 or more user stories
- Each sprint can be either 2/3/4 weeks it depends upon customer decision.

### **Sprint planning meeting :**

- It is a meeting which is conducted before sprint starts
- In this meeting product owner and scrum master will be involved

- They will decide which user story has to be worked on that particular sprint. Also they will decide the duration of the sprint.

#### **Who is product owner**

- He is a person who is a customer or representative of a customer
- He decides which are the important user stories.

#### **Who is scrum master**

- He is a person who is representative software company
- Scrum master can be project manager / developer lead / test lead depends on project decision

#### **Daily stand up meeting :**

- All team members will gather together like project manager developer lead , test lead , test engineers , developers , BA.
- Customer also involves in this meeting.
- Here each employee should give work update every day
- If the meeting is held in the morning the team members have to convey what they did yesterday and what are they going to do today.
- If the meeting is held in the evening the team members have to convey what they did today and what are they going to do tomorrow.
- This is very much helpful for a better communication between team members.
- Scrum master is the host of this meeting

#### **Sprint backlog :**

- It is user story which is not completed in that particular sprint that is carried into forward to the next sprint.
- EX: if 'D' user story is not completed in sprint 1, it is a backlog which taken from sprint1 to sprint2.

#### **Sprint review meeting :**

- In this meeting, scrum master will check whether all the user stories are completely developed and tested and is it ready to release customer or not

This happens at the end of the sprint.

## **Product backlog :**

- It is the user stories which is taken for a new sprint
- It is under epic

EX : e,f,g,h are from product backlog for sprint 2

In the same way till all the user stories are completed , multiple sprints will be going on and full application will be ready.

## **Advantages of agile model :**

- From this model work is getting completed very fast.
- It is the most successive model currently
- We can work more within less time
- Communication is very good between company and customer
- We can handle any pressure
- It is best model for existing projects.

## **Disadvantages of agile model :**

- Less focus on design and documentation since we delivery very fast.

## **STLC:**

It is software testing life cycle

It is a procedure to test a software application

It has different stages like:

- System study
- Prepare test plan
- Write test case
- Prepare RTM ( requirement traceability matrix)
- Execute test case
- Defect tracking
- Test execution report

➤ Retrospective meeting or postmoter meeting.

➤ **System study :**

It is going through the requirement given by the customer and understand how the system works.

➤ **Prepare test plan :**

- It is a document which is prepared for future testing activities.
- It is done by test lead or test manger because the plans will be done by experienced people.

➤ **Write test case :**

It is a step by step procedure to perform the testing on the application. It is done by test engineer. Once we go through the requirement , we identify the scenario and then converted into the test cases. To write the test case we need requirement and test case template or tools (QC , ALM , JIRA etc)

Test engineer will write test cases

QC - quality center

ALM - application life cycle management

➤ **Prepare RTM ( requirement traceability matrix) :**

- It is a document which is prepared to check whether every requirement has atleast one test case or not.
- To prepare RTM we need both requirement and test case test engineer should create this document .

➤ **Execute test case :**

- Once the requirement is given to the test engineer he write test cases for the applications. After the developer gives the developed application then the cases are executed and it is compared with expected result and actual result. If the expected results and actual results are same then the status is pass if the expected result and actual result are not same status will be fail this is called execute test case.
- So,to execute test cases we need test cases and software application. This is where exactly the software is been tested.this is the most important phase of software testing life cycle.

➤ **Defect tracking :**

- While executing the test cases we may come across defects these defects are raised/logged/reported to the developer.
- Once we raise , we should also check what is happening to that defect.

This is called defect tracking.

It is done by using tool called bugzilla , QC , ALM , JIRA etc.

➤ **Test execution report :**

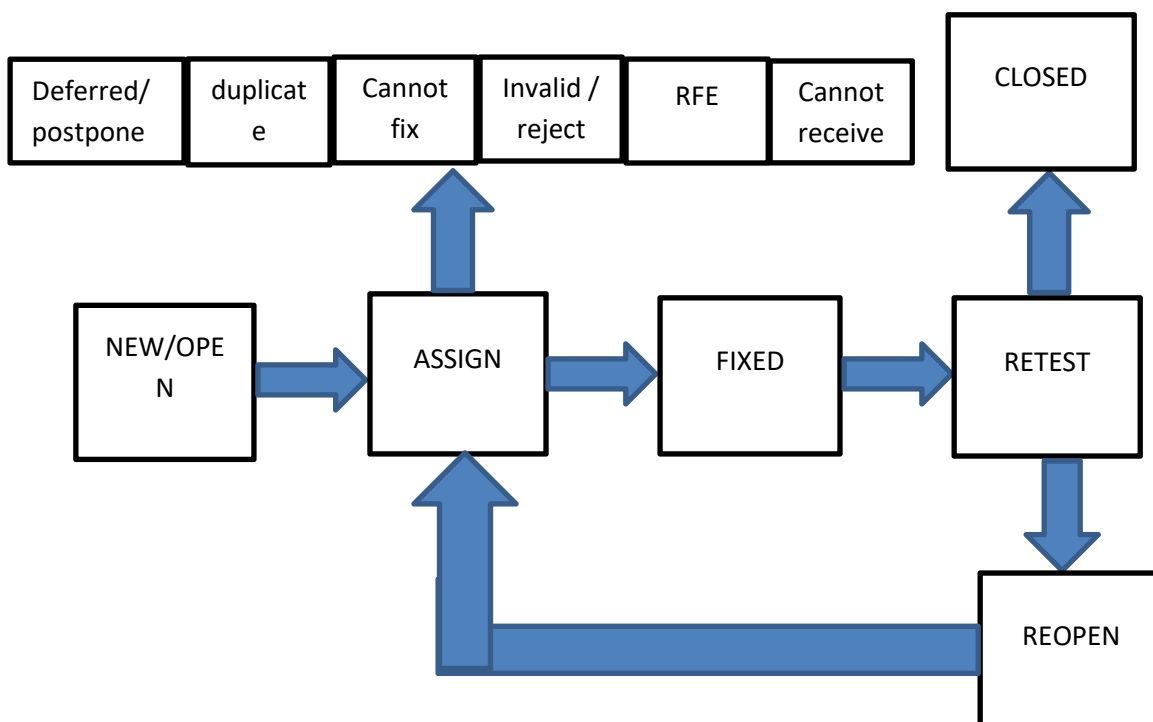
Once we execute the test case we should prepare one document called test execution report, this document tells about how many test cases are pass and how many test cases are fail. Also we can check how many are executed and not executed.

Till here customer involvement will be there according to customer this is the end of the project.

➤ **Retrospective meeting or postmoter meeting :**

In this meeting al team member will gather together like project manager , test leas , developer lead , developer , test engineer , BA and they will discuss about the good and improvement needed once the project is done. This will be very useful for next release or next project.

**DEFECT LIFE CYCLE :**



When we are executing the test case if the expected result and actual are not same come across the defect. Then the defect is raised to the developer, the status will be **New/open**. That defect will be assigned to the developer or development lead (we don't know the correct developer) then the status is **Assigned**.

The developer will reproduce the defect and if it is reproducible it will be accepted. He starts fixing the defect in the developer server and it has to be installed to the testing server. Then he should change the status to **Fixed**.

The test engineer starts retesting the defect in the testing server. If the defect is properly fixed, then the status will be **closed**. If the defect is not properly fixed, then again, the defect is reopen to the developer. This process continues till defect goes to closed status.

### **What is deferred / postpone status ?**

Whenever test engineer raises the defect, the developer accepts the defect but he will not fix the defect immediately. We will fix it later, then the status of those defects will be done as deferred / postpone by developer.

EX: when there is a major defect and minor defect. Developer has less time to fix the major defect. In this case the developer will fix the major defect and for minor defect he will make status as **deferred / postpone**.

**New/open-->Assign-->deferred-->fixed-->closed** (Do retesting before closing)

### **What is duplicate status ?**

Whenever the test engineer finds the defects and he raises it. If another test engineer or same test engineer finds the same defects and raises it unknowingly / mistakenly, then status of that second defect will be changed as **Duplicate** by developers.

### **To avoid Duplicate status :**

A) Check whether the same defect is already raised or not in the repository EX: (Bugzilla, QC, ALM, JIRA) before raising any defect

B) Communicate with team members before raising a defect, if you are raising for common module defect.

### **What is can't reproduce / unable to reproduce status :**

When test engineer finds the defects but developer cannot be able to reproduce that defect so he will change the status as cannot be reproduce.

EX: mobile hang etc.



### **Reasons for can't be reproduce :**

- Installation problem (developer server , test server)
- Improper defect report (chrome , fire fox)
- Inconsistent bug (a bug which occurs sometimes and does not occur sometimes )

### **What is can't be fixed / cannot fix status :**

Whenever the developer is unable to fix the defects that is raised by the test engineer , then he changes the status as **"can't be fixed"** and it is a valid defect , but he can't fixed the defect.

EX: whenever technology does not support to fix that defect , then developer will change the status **"can't be fixed"**.

**New/open-->Assign-->can't fix-->closed** (after speaking with manager)

Finally what do for this defect will be decided by managers/customer.

### **What is invalid / reject / not a defect status :**

Whenever the test engineer raises the defect , but developers will not accept the defect as valid and he changes the status to **invalid/reject**.

### **Reason for invalid :**

#### **Due to the misunderstanding of the requirement**

- If test engineer misunderstand requirement

**New/open-->assigned-->invalid-->closed.**

- If developer misunderstand the requirement

**New/open-->assign-->invalid-->assign-->closed.**

**NOTE: INVALID** is bad for test engineer.

### **What is request for enhancement (RFE) :**

Whenever the test engineer raise the defect which is not given the requirement so the developer will take it as suggestion. In this case he will change the status as RFE.

### **What is review :**

- It is process of finding mistakes in the document.

(requirement , test scenario , test case etc)

- Reviewing the document is also known as static testing.

### Types of review :

- Self review (finding our own mistake)
- Peer review (finding other test engineer mistake)
- Manager review (finding test engineer mistake).

FORMAL REVIEW	INFORMAL REVIEW
It is documented	It is not documented
It has proof	It has no proof
It has procedure	It has no procedure
It has a process	It has non process

### What is Domain :

Domain is different industries (or) categories of industries.

### BFSI :

- Banking , Finance services and insurance.
- It is three different domains.

**Banking Domain :** [Example : SBI , yes bank , G pay , Ph pay , Paytm]

**Finance :** [Example : shares and stocks]

**Insurance :** [Example : bike Insurance , car Insurance , LIC]

**E- commerce :** [Example : Flipkart , amazon , Meshoo]

**Health care :** [Example : apollo]

**Telecom :** [Example : airtel , jio].

### SOFTWARE TESTING:

- Testing an application along with its functionality based on customer requirement is software testing.
- It is testing an application with intent of finding defects.
- Testing is a process QA and QC.

## Types of black box testing :

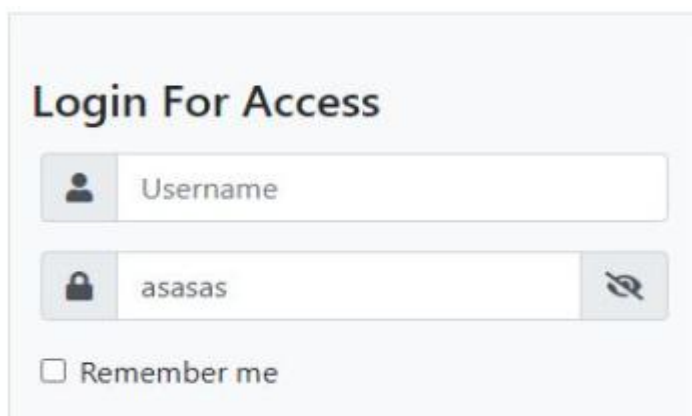
- Functional testing
- Integration testing
- System testing.

## Functional testing (or) field level testing (or) component testing :

- Testing each and every independently and thoroughly with respect to requirement.
- We perform functional testing for all components inside a module of an application.

### EXAMPLE :

#### ➤ Text field



Login For Access

☐ Remember me

#### ➤ Radio button :

☐ Male

☐ Female

☐ others

Whenever we want one option we go for this

➤ **Check box :**

**Foods Eaten**

<input checked="" type="checkbox"/> Fresh celery	<input type="checkbox"/> Grapes	<input type="checkbox"/> Blueberries	<input type="checkbox"/> Breast milk
<input type="checkbox"/> Skim milk	<input type="checkbox"/> Cheddar cheese	<input type="checkbox"/> Peaches	<input type="checkbox"/> American cheese
<input type="checkbox"/> Strawberries	<input type="checkbox"/> Orange juice	<input type="checkbox"/> Apple juice	<input type="checkbox"/> Fresh tomatoes
<input type="checkbox"/> Beef jerkey	<input checked="" type="checkbox"/> Sour cream	<input checked="" type="checkbox"/> Vienna sausages	<input checked="" type="checkbox"/> 2% milk
<input checked="" type="checkbox"/> Ice cream	<input checked="" type="checkbox"/> Turkey	<input checked="" type="checkbox"/> Whole milk	<input type="checkbox"/> Raw carrots
<input checked="" type="checkbox"/> Bean sprouts	<input type="checkbox"/> Butter	<input checked="" type="checkbox"/> Cooked bacon	

When ever we want one value or multiple value we go for this

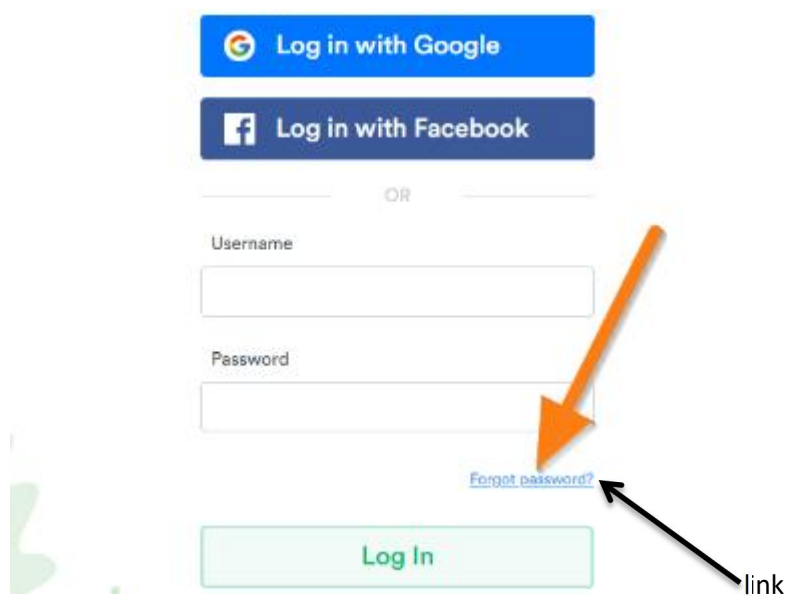
➤ **Drop down :**

<p>Color</p> <p>- Select -</p> <p>Red</p> <p>Orange</p> <p>Yellow</p> <p>Green</p> <p>Blue</p> <p>Purple</p> <p>Pink</p>	<p>Color</p> <p>Select color</p> <p>Red</p> <p>Orange</p> <p>Yellow</p> <p>Green</p> <p>Blue</p> <p>Purple</p> <p>Pink</p>
--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

➤ **Button :**

The image shows a login form for 'ocula'. It has a header with the 'ocula' logo and a 'Log In to Ocula' title. Below the title are two input fields: 'E-Mail' and 'Password'. At the bottom of the form is a dark blue button with the text 'SIGN IN' in white. A blue arrow points from the word 'Button' to this 'SIGN IN' button. There is also a link that says 'Forgot your password?' below the button.

➤ **Link :**



➤ **Text area :**

It is a field which allows us to enter more text so it is called as text area. It will allow 100,200,300 characters etc.



➤ **Labels :**

It is a name given for every component



**NOTE :**

- For every module , we have to do functional testing separately.
- We have to check with positive and negative scenario for every component.
- We have to apply test case design techniques like
  - ❖ Boundary value analysis
  - ❖ Equivalence partitioning
  - ❖ Error guessing

While doing functional testing (wherever applicable)

After functional testing completed for every module we go for integration testing.

**Integration testing :**

Testing the data flow between two or more depended module.

**Example :**

- Whenever send an Email to another person and that person received a same Email is know as Integration testing.
- Searching the product in flipkart (I phone) and click the search button the next page should shows only I phone is know as integration testing.
- Sending 5000 in paytm to friend and check whether the same amount is debited and credited is know as integration testing.

Once functional and integration is completed we go for system testing.

**System testing :**

Testing an application from end to end like real user in testing sever limitation to production server.

**Example :**

- Verify by opening flipkart , search the item , add to cart , make payment , logout.
- verify by opening Email , sending message , see sent box , logout.
- Verify by opening Email , login , click on setting , select privacy , click password , change new password , logout and login with new password.

## **Retesting and Regression testing :**

### **➤ Retesting :**

Testing the defects fix of an application. it is done after developers fix the defect.

### **➤ Regression testing :**

Testing the impact area of an application.

A) After defect fix is done

B) After changes done in the application.

(changes can be adding , modifying and deleting a features)

### **Example :**

Taking mobile phone as an example. Test the mobile front camera is not working and test engineer report mobile front camera is not working and test engineer report the defect to the developer. once developer accept the defect and fixes the defect and given to test engineer and test engineer recheck the front camera is retesting and test engineer do regression testing on impact area like back camera , whatsapp camera , face lock , video call , timer , slow motion etc. And it is known as regression testing.

### **Example :**

Sending E-mail to 5 people but E-mail got only for 3 members and 2 members did not get message it is defect. Test engineer call the developer to fix the defect. Developer fixes the defect and send the application to the test engineer. test engineer do retesting by sending same E-mail to same 5 person and check whether E-mail is sent or not it is retesting .

After retesting is done on impact area of an application by verifying sending mail to 4 people and verifying by sending to 6 people by applying boundary value analysis and verifying by sending mail to same 5 person with attachment and verify by sending mail to own mail ID is known as regression testing.

### **Why we do regression testing :**

After some changes are done for an application we check whether the whole feature is working fine or not. So we do regression testing.

## **How will you prove you are a good test engineer / How will ensure everything is covered in testing :**

As a test engineer , I will find scenario both positive and negative scenario for a requirement.

This scenario will be converted into test cases using a template in a understandable way. While writing test cases I will use techniques like boundary value analysis , equivalence partitioning , error guessing. I also prepare RTM document to ensure every requirement has test case or not . I will also review the test cases of by peer after testing , once I find the defect I will report it clearly along with screen short . by all this things I will say I am a good engineer.

## **What are the drawbacks of manual testing**

**(or)**

## **Why we go for automation testing**

- Manual testing consumes more time
- We need more man power
- Testing is repetitive activity so we go for automation testing .

## **Tools used in IT industry :**

### **Functional testing tools :**

- Selenium
  - Selenium can be used with java , python , c# etc.
  - Java selenium has highest demand.
  - Selenium was found by thought works
  - It is an open source tool (free of cost)
- QTP (or) UFT
  - Quick test professional
  - Unified functional test
  - QTP (or) UFT can be used with VBS (visual basic script) and JS (java script)
  - VBS with QTP is very good combination.
  - QTP is licensed tool.(paid tool)
  - QTP is licensed with HP before it was with mercury.
  - HP - Hewlett Packard.



- Win runner
- Silk test
- Test partner.

### **SMOKE TESTING/CONFIDENCE TESTING/BUILD VERIFICATION TESTING:**

Testing the basic or critical features of an application before doing through testing like functional testing , integration testing , system testing.

Here we check positive scenarios.

#### **Why we do smoke testing :**

- To check whether basic features are working fine or not
- To get a confidence that the basic features are working fine.
- If there is any defect in the important feature , that can be found earlier and report it to developer , so that they will get time to fix the defect fast.

Note :

In smoke testing we check only positive scenario.

Example :

For login page of face book below one the smoke testing scenario

- Verify by opening the face book
- Verify by entering valid username
- Verify by entering valid password and click on login button
- Verify by entering valid phone number and valid password click on login button
- Verify by clicking the forgot password and check whether the page is open
- Verify by clicking the create new account and check whether the page is open

#### **When do we do smoke testing :**

- As soon as the build is given , we do smoke testing.
- For every build first we have less time to test the application , we do smoke testing first.

### When time is very less as a test engineer how will you handle the situation

I do smoke testing first to ensure the module are working or not and after that can go type testing like functional testing , integration testing and system testing before give application to customer I do these all testing.

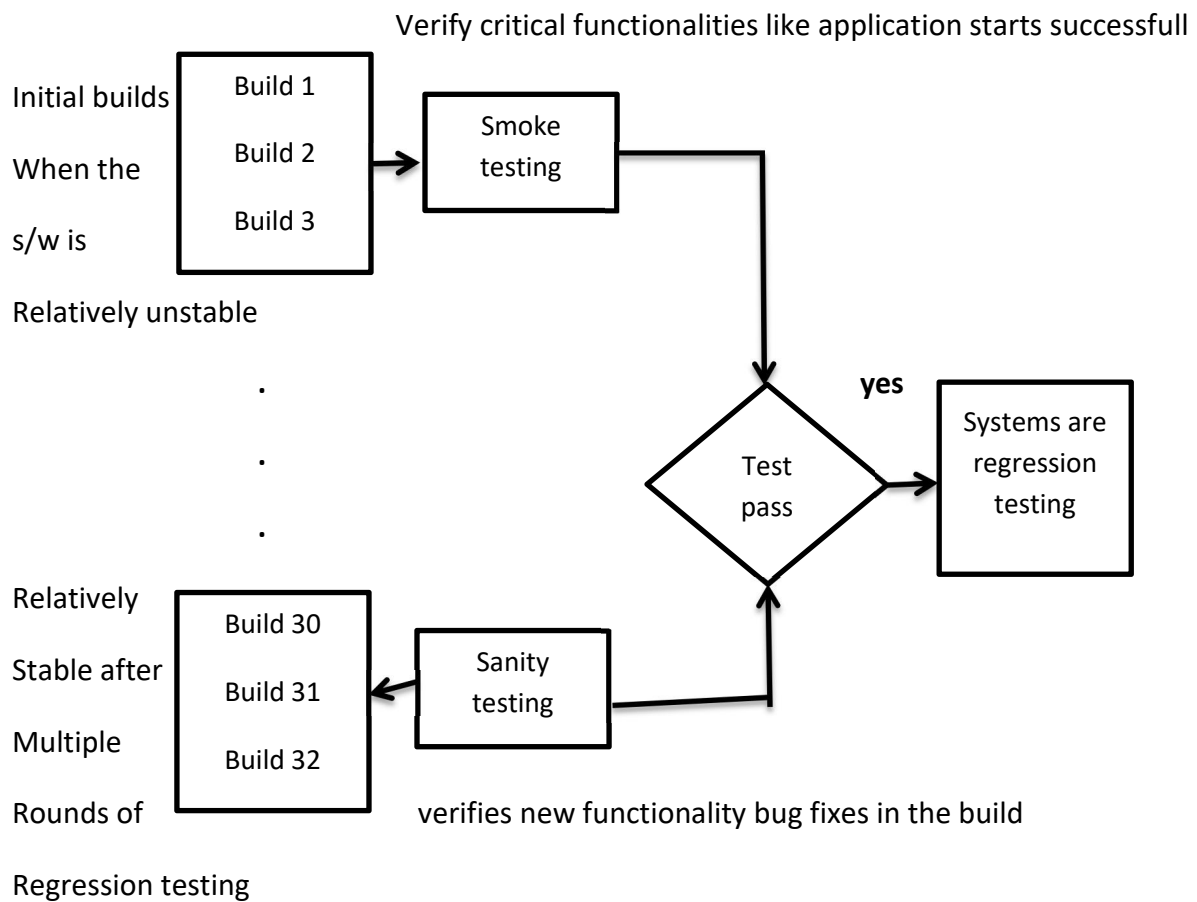
### What is formal smoke testing and informal smoke testing

Formal smoke testing	Informal smoke testing
It is documented	It is not documented
It has proof	It has no proof
It has procedure	It has no procedure
It has a process	It has non process

### Difference smoke vs sanity

SMOKE	SANITY
It is testing the basic / critical features only	It is testing the new features and bug fixes
It is shallow and wide testing	It is narrow and deep testing
It is done on unstable builds	It is done on stable builds
Only positive testing is done	Both positive and negative testing are done
It is done on initial builds	It is done on last builds
	It is the subject of regression testing

**Testing and sanity testing can be learned with help of following diagram**



### **PERFORMANCE TESTING TOOLS :**

- Load runner
- J meter
- Silk performer
- Neo load
- QA load

### **PERFORMANCE TESTING :**

Testing the response time of an application by applying load.

## **Types of performance testing :**

- Load testing
- Stress testing
- Soak testing
- Volume testing.

### **Response time :**

It is a time taken to load the page or screen based on the action performance

### **Load :**

Load is no.of.users

### **Load testing :**

Testing the response time of an application by application the load which is less than or equal to designed number of users (it will be given by customer).

### **Example :**

if customer want application for 1 lakh users then the designed number of users is 1 lakh

### **Stress testing :**

Testing the response time of an application by applying the load which is greater than the load.

### **Example :**

Requirement 100000 users -----response time should be within 2 sec

Stress testing -110000 users -----response time 2sec/3sec

### **Volume testing :**

Testing the response time of an application by transferring huge volume of data through the application.

### **Example :**

Huge data sharing through share it , bluetooth , google drive by uploading more photos and checking the time taken to transfer the data.

## **Soak testing (longevity testing or capacity testing or endurance testing ) :**

Testing the response time of an application by applying load continuously for long duration of time.

### **Example :**

72 hours more users are using application continuously and we will check the response time for 72 hrs.

Apply load for 3 days continuously and check response time .

### **Spike testing :**

Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load.

### **Example :**

Seeing exam result when the result is released.

### **Can we do performance testing manually :**

Yes, but there are some drawbacks

- High cost (man power , device control , infrastructure )
- Low accuracy
- More time
- Managing is difficult

So we go for automation (load runner , J meter etc)

### **Test management tools :**

- QC or ALM is a test management tool
- QC - quality center
- ALM - application life cycle management
- QC/ALM (it is a licensed tool . it is with HP. It is a paid tool)
- JIRA (best for agile model)
- OTM (oracle test manager)
- Test link.

Using this tools we can manage the testing activites like below

EX: QC / ALM below are the module of QC / ALM

Add requirement / edit requirement / delete requirement

**Test plan** : write test case

**Test lab** : execute test case

**Defect** : report / raise / log a defect track a defect .

Developers are all so see the tools

Every company will have atleast one test management tool . login credentials will be given by company people based on manager approval.

**Defect management tools : (defect tracking tool (or) bug tracking tool ) :**

- Bugzilla (open source tool)
- QC / ALM
- JIRA (partially free)
- OTM
- Test link etc.

In buzilla we cannot add the requirement , write test cases and execute test cases .

We can only report the defect and track the defect.

**Import / Export :**

- We can write test cases using excel sheet or tools like QC / ALM , JIRA etc
- Always use the tool if available in the company if not available , then use template given by manager or create own template.
- We can import and export test cases between excel and tools (QC / ALM).
- QC / ALM export test cases to excel import test cases to excel / import test cases from QC / ALM to excel.

QC / ALM - excel

Exporting (sending) importing (receiving)

## **SEVERITY and PRIORITY :**

Severity will tell how much defect is affecting the customer business .

- Blockers / shows topper
- Critical
- Major
- Minor
- Trivial

**OR**

- high
- Medium
- Low

Severity is used in defect report while raising the defect

Priority says which defect has to be fixed first by the developer.

For every defect we have to set priority.

### **Different types of priority**

- high
- Medium
- Low

**OR**

- urgent
- high
- Medium
- Low

**OR**

- P1

➤ P2

➤ P3

- Both severity and priority used in defect report this to be decide by test engineer.
- Severity will be converted many times priority keeps on changing based on the situation.

Example for blockers severity and high priority defect

Example: banking application

When I send the money to other person the money will not received because transfer button is not working.

Severity : Blockers

Priority : High

Example : when I select the seat in red bus but didn't select the seat selecting option is not working.

Example for critical severity and medium priority defect

Example : transaction history is not displayed in SBI app.

Example : bus spare is not shown in the red bus app.

Severity : critical

Priority : medium

Example for major severity and medium / low priority

Example : when we transfer the money conformation message is displayed with little delay.

Example : when I am selecting the seat option but its different seat number .

Severity : major

Priority : medium / low priority

Example for minor severity and low priority

Example : if there is any spelling mistake like sucessfully instead of successfully

Example : destination place have some spelling mistake like chennani.

Severity : minor

Priority : low priority



Example for trivial severity and low priority

Example : if there any font change colour changes in the app

Example : if there any font colour changes in the app.

### **If severity and priority not given what will happen**

If this not given developer will fix the defect randomly ass per his wish.but they have to fix important defect first for this reason severity and priority is very important

### **DEFECT REPORT :**

It is report which has to be prepared by test engineer once defect has to be found

To report a defect we can use tools like

- JIRA
- QC / ALM
- Bugzilla

To know how to report the defect using QC / ALM go through below:

In QC /ALM we use defect

In Bugzilla we use Bug

In JIRA we use issue

Note : it is the report which is used for raise the defect , there is no tool to raise the defect we should use excel or word document.

Defect report :01

Defect ID : 001

Build no : Bo1

Test case number : TC\_IT\_05

Status : assigned

Severity : major

Priority : high

Test environment : chrome browser

Module name : order online (super lean meals )

Reported by : Dinga ( test engineer)

Assign to : Dingi (developer)

Brief description : veg shake is displayed in red colour instead of green colour (beet root juice)

Test data : NA (not applicable )

### **WHITE BOX TESTING :**

Testing the source of an application line by line is white box testing. It is done by developer. Other name of white box testing is

- Open box testing
- Clear box testing
- Structure box testing
- Glass box testing
- Transparent box testing

### **BLACK BOX TESTING :**

Testing the user interference of an application is know as black box testing. It is done by test engineer Other name of black box testing is

- Functional testing
- Behavioral testing
- Closed box testing

### **GREY BOX TESTING :**

Testing both source code and user interference of an application parallel is grey box testing it is done by a person who is good with source code and user interference. (SDET)

**SDET** - software development engineer in test.

### **COMPATIBILITY TESTING :**

Testing an application with different hardware and software platforms is called as compatibility testing.

### **Why do we do compatibility testing :**

- To ensure that application is working for multiple platform because there might be different types of users.
- To check whether the application is consistently working in all platforms or not.

### **Types of compatibility testing :**

#### **● Software compatibility testing :**

Testing an application in different software platforms is software compatibility testing.

Example : In different OS like:

Windows ( w7 , w8 , w8.1 , w 10 etc)

Linus ( ubuntu , fedora etc)

MAC (MACINTOSH) ( Tiger , lion , leopard , snow leopard , mountain lion , sierra etc)

Example : telegram , zomato , first cry , paytm etc.

Platform will be decided by customer based on no.of.user that particular platform.

#### **● Hardware compatibility testing :**

Testing an application in different hardware platforms is hardware compatibility testing.

Example : in different processor like - Intel , AMD.

In different mother board like - intel , ASUS

In different RAM - (random access memory)

In different ROM - (read only memory)

In different hard disk - bits , byte , kb , mb , gb , tb , zb

#### **● browser compatibility testing :**

Testing an application in different browser.

Example : chrome , mozilla fire fox , internet explorer , opera , uc browser , netspace navigator , duckduck go , safari.

#### **● Mobile compatibility testing :**

Testing an application in different operating system and also for each different versions like kitkat , windows , ios and for each brand of mobile.

Example : samsung , redmi , nokia etc

For each brand

For different models like A series , M series , galaxy for samsung

X peria c4 , c5 , z5 for sony

V15 , V7 , V19 for VIVO.

### **What kind of bugs / defects we find in compatibility testing**

- Look and feel changes ( Ex : front and colour changes)
- Object overlapping (login and cancel button sitting on one another )
- Certain image will not displayed in certain browser.
- Scroll issues (horizontal and vertical scroll bar may work in one browser and may not work in other browser )
- Alignment issues / problems
- Certain button links and components may work in one browser and may not work in another browser.

### **GLOBILIZATION TESTING :**

- Testing the application which is developed for different languages is called globalization testing.
- To do this testing we have to excellent in different languages
- Here translating the content is very important translation can be done using machine or human.

Machine translator cannot understand

- Exact meaning
- Exact grammar
- Exact spelling
- Exact feeling

So we go for human translator

Example :

I love you

Nan kadhal unnai -- > wrong

Naan unnai kadhalikiran --> correct

### **Types of globalization testing :**

- I 18 N - internationalization
- L 10 N - localization

#### **I 18 N - internationalization :**

Testing whether the right content is displaced in right place in right language is I18N language.

Example : for languages like tamil , english , hindi , kanada etc.

At should be displayed from left to right

For languages like urudu , arabic etc it should be displayed from right to left.

#### **L 10 N - localization :**

Testing the application with respective local standard or culture is L 10 N testing

Example : in Amazon if we choose India region currency should be in rupees , if we choose America region currency should be in dollars.

Note : this testing will be done based on customer requirement this will help in increasing the customer business

### **USERABILITY TESTING :**

Testing user friendliness of an application is user ability testing (or) checking whether we can perform the operations easily with less number of actions is called userability testing.

Example : VPS ( virtual positing system ) , GPS ( global positing system )

- Touch screen phones are more user friendliness than keypad phones
- In zomato (or) swiggy if current location is taken automatically that is more user friendly than we entering the current location.

### **Who can do user ability testing**

- End user is the first option
- Customer is the second option
- Test engineer is the third option

### **RELIABILITY TESTING :**

Testing functionality of an application for long duration of time is know as reliability testing.

Example : testing the mobile phone by using continuously for three days and check it is working fine or not and it is know as reliability testing.

### **ACCESSIBILITY TESTING :**

ADA - American disability applicant

- Testing the application from physically challenged person point of view
- This testing is important for people who having colour blindness (or) wrist injures

Example : colour blindness person cannot see RGB colour

RGB - red green blue

- If RGB colour is present in the application it is defect according to accessibility testing
- All the components should be accessible using mouse and keyboard both. If it is not there it is defect

### **Tools used for accessibility testing**

- Wave
- In focus

### **EXPLORATORY TESTING :**

Testing an application without any requirement is exploratory testing.here we have to explore the application using common sense and identify scenarios.

### **When we do exploratory testing**

- When there is no requirement we do this exploratory testing
- Requirement is there but it is confusing or not clear.

### **Drawbacks while doing exploratory testing**

- If any feature is missing will never come to know that is missing.
- Feature will be misunderstood as defect , defect will be misunderstood as feature.

### **How to overcome the problem**

Communicate politely with peers , customers , BA

Leave ego to survive in the company

### **AESTHATIC TESTING :**

Testing the beauty of the application.

Example : checking colour combination , font style , font size and attractive of an application.

All this can be done using front end technologies like HTML , java script , angular java script.

### **SECURITY TESTING :**

It is to check whether all the data and resource of an application is accessible only for authorized users and it is highly secured.

#### **Types of security testing**

##### ➤ **Vulnerability scanning :**

It is identifying all the risk area of an application with respect to security.

##### ➤ **Security scanning :**

Once the risk area are identified we have to secure them the additional safety measures that is security scanning.

##### ➤ **Penetration testing :**

Once the security is implemented on the risk areas. Check again whether still any risk or data leakage is happening this is penetration testing.

##### ➤ **Risk assessment :**

Once the security risk are identified categorize them into high , medium , low and take precaution as needed.

➤ **Security auditing :**

This is an internal inspection of application and operating systems for security flows. An audit can also be done by line by line inspection of code.

➤ **Ethical hacking :**

It is a process of attacking a system by an internal employee to see whether security measures which is taken is really worth or not

➤ **Posture assessment :**

This combination security scanning , ethical hacking and risk assessment to show an overall security posture of an organization .

**Example :**

- https is more secured more than http
- Virtual keypad is more strong and secured compared to normal keyboard
- Password should be encrypted ( \*\*\*\*\*)
- Bank application have session timeout

https - hypertext transfer protocol secure (data is encrypted)

http - hypertext transfer protocol (data is not encrypted)

HTTP	HTTPS
It is not encrypted	It is encrypted
http is operate at application layer	https is operate at transport layer
http operate on port 80 http is faster than https	https operate on port 443 https is slower than http

Very important security testing

- Verify by login to the application.
- Copy the URL.
- Logout the application
- Paste the URL in the search and search the application the application page comes it is defect , login page comes it is not a defect.



**Note :**

- G mail application login in browser tab and copy the URL link and add the new page in browser and paste the URL to search the page and its comes with logged in its defect and its not come its not defect.
- Use the bank application in browser login and money transfer after from browsing history click the same history again its comes the defect.

**Note:**

- Functional testing , integration testing , system testing are mandatory for every testing application
- Smoke testing , sanity testing , exploratory testing , adhoc testing are situation based testing
- Performance testing , compatibility testing , globalization testing , reliability testing , usability testing , security testing , accessibility testing are non functional testing. These are done based on additional request from customer

These testing will help increasing business for customer.

**ADHOC TESTING :**

Testing the application randomly without following requirement

While do this testing we have to think like monkey , kids , uneducated people , illetrate , anger over , excited people , depression , drunk people

**Why do we do adhoc testing**

Whenever we don't get defects from regular testing like functional testing , integration testing , system testing than we think about adhoc testing to find more defects

When we don't get defects within the requirement we have think out of the requirement and find defects

**Note :**

In adhoc testing we check negative scenario.

Example : whenever we do right click , perform left click or both click

Gaming apps ( EA sport , scientific games , loft)

## **Negative scenario for gaming application**

In the bike racing game road rash

- Verify by clicking forward , backward , left , right at the same time
- Verify by clicking forward and backward button at the same time
- Verify by clicking forward and left and right button at the same time
- Verify by driving the bike on wrong direction
- Verify by driving the bike on sand and grass
- Verify by writing a board near the road
- Verify by hitting a tree
- Verify by trying to cross the bridge
- Verify by trying to enter into the water
- Verify by doing nothing for more hours
- Verify by climbing the mountain.

## **RECOVERY TESTING :**

Testing the application whether it is able recovery from crash state or not

### **Example :**

- Internationally use multiple application at the same time continuously and make the application crash ( not responding or not opening or 404 page not found )
- Later close the application & open it again & see whether all previous data is still available or not

This is recovery testing

- If all data are available & application is working normally after crash then it is not a defect.
- If any data is lost application is not working properly after crash , then it is a defect.

## **FUZZ TESTING :**

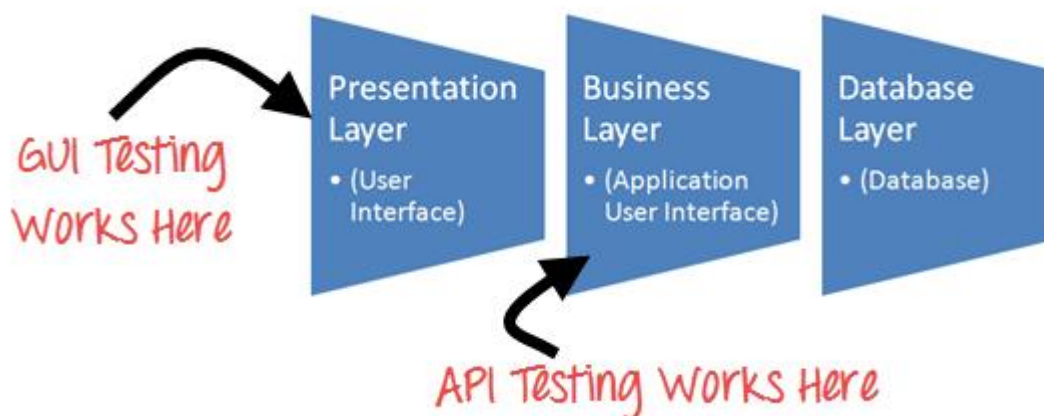
- It is testing the application with invalid data (or) unexpected data randomly check hoe the system works & handle it

- This is a sort of adhoc testing
- It is also to check the vulnerability of the system. ( this is sort of security testing )
- Overall fuzz testing is almost a combination of adhoc and security testing where we perform through automation.

## API testing ( application programming interface )

API testing is a software testing type that validates application programming interface (APIS)

- The purpose of API testing is to check the functionality , reliability , performance and security of the programming interface.
- In API testing instead of using standard user inputs (keyboard and output) you use software to send calls to the API get output and note down the system response
- API tests are very different from GUI tests and wont concentrats on the look and feel of an application
- It mainly concentrates on the business logic layer of the software architecture.



## ACCEPTANCE TESTING :

In this testing customer will check business scenario of an application

### Why acceptance testing is done :

- It gives confidence to customer
- If development team or testing team missed any important feature to verify that acceptance testing has to be done.
- Due to business pressure software company might develop wrong feature
- To ensure this does not happen acceptance will be done by customer.

### Types of acceptance testing :

- Alpha testing
- Beta testing

ALPHA testing	BETA testing
Alpha testing is performed by tester who usually an internal employee of an organization	Beta testing is performed by client or end user who are not employee of an organization
Alpha testing is performed at the developer sides (software company)	Beta testing is performed at the client location or end user of product
Alpha testing involves both black box and white box testing techniques	Beta testing typically uses black box testing techniques
Alpha testing requires lab environment or test environment	Beta testing needs real time environment
Virtual environment	Real time environment
Done at offshore (company)	Done at onsite (customer place)
Alpha testing done under controlled environment	Beta testing is done under uncontrolled environment
It is closed for public	It is opened for public

**Note :** customer will decide about alpha and beta testing and also he will decided who should do this testing.

**Example :** we are taken courier application

An test engineer will check below features for courier business - placing a courier , shipping , tracking , delivery date , weight , amount , agent , status , return option , OTP , address , kind of product , bill , customer details is tested by test engineer.

### We may miss to test cancel courier feature

If that is tested by customer and if cancel is not working this will be bad impression for testing team.

## Approaches of acceptance testing :

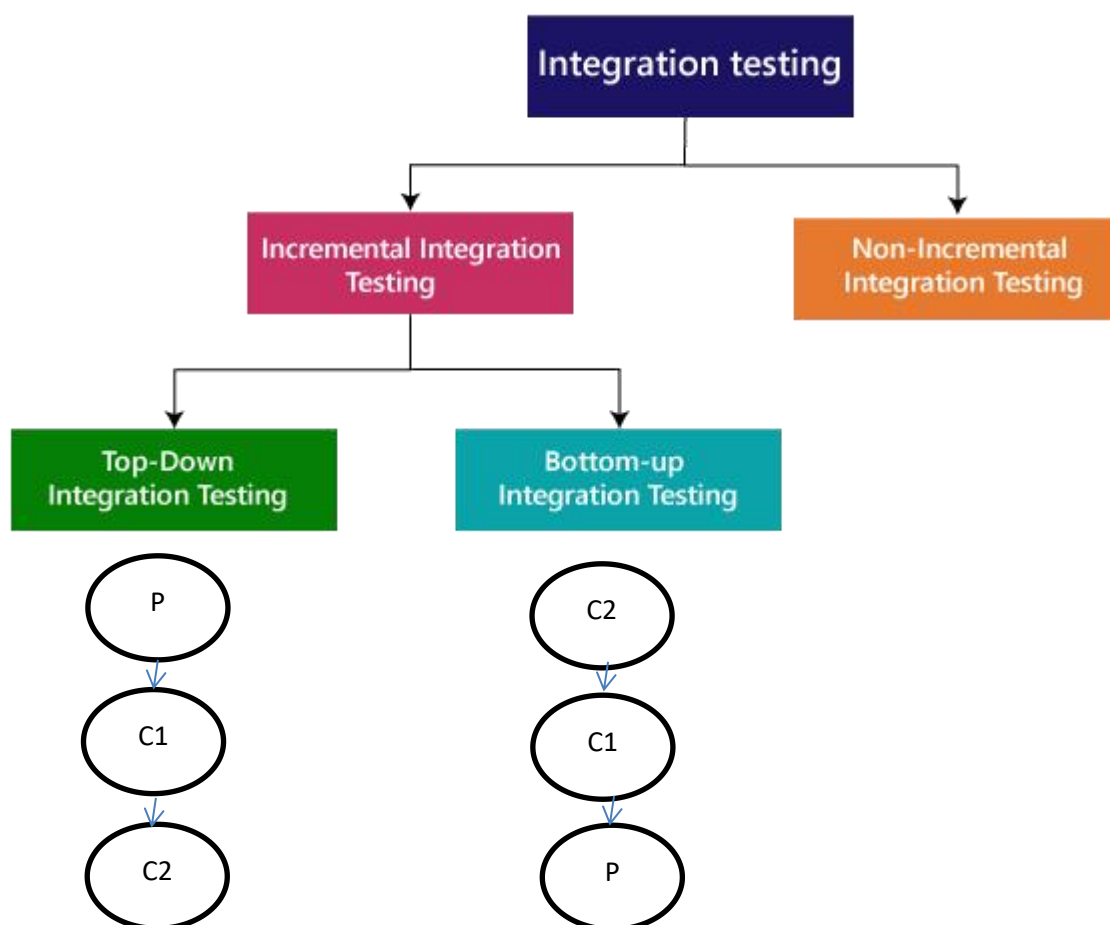
Approach no.1 : BA or IT engineer of customer will do acceptance testing at customer place

Approach no.2 : employee or test engineer of customer will do acceptance testing at customer place.

Approach no.3 : testing of software company will do acceptance testing at customer place

Approach no.4 : test engineer or BA of software company will do acceptance testing under the control of customer at developer place.

## Types of integration testing :



### Incremental example :

ATM : insert card --> language-->PIN-->account type-->withdraw-->enter amount.

Flipkart : search --> cart --> buy --> payment.

### Non-Incremental example :

G mail : compose

Inbox

settings  
Messages

### **Incremental integration testing :**

When parent module is there , child module is added and we do integration testing between the module (TOP DOWN)

(or)

Whenever child module is there , parent module is added and we do integration testing between module (bottom up)

### **Non Incremental integration testing :**

When we cannot find which is a child and which is a parent module.even then we are integration testing between them this is called non incremental integration testing (BIG BANK)

### **Sandwich integration testing :**

It is combination of incremental and non incremental testing

**Example :** facebook : sign up --> login (incremental)  
facebook : notification (non incremental)

### **Stubs and driver :**

- Whenever one module is ready and other module is not ready we use stubs and drivers.
- When parent module is ready , we use stubs
- When child module is ready and parent module is not ready we use drivers.
- Both stubs and drivers are dummy modules once real modules are ready we replace real modules by removing stubs and drives

.

#### **Example :**

Stubs and drivers are used for critical application like banking , healthcare , navy , space , airlines , submarine , army

For any application its takes long time to develop we use stubs and drivers

### Difference between functional and non functional :

FUNCTIONAL	NON-FUNCTIONAL
Functional testing Integration testing System testing	Performamnce testing Compatibility testing Globalization testing Security testing Accessibility testing Usability testing Reliability testing
checking for single users and single platform and single language focus on normal page	Checking for multiple users Multiple platforms Multiple languages
it is mandatory for all applications.it is important to start the business	It is done on additional requirement to increase the business

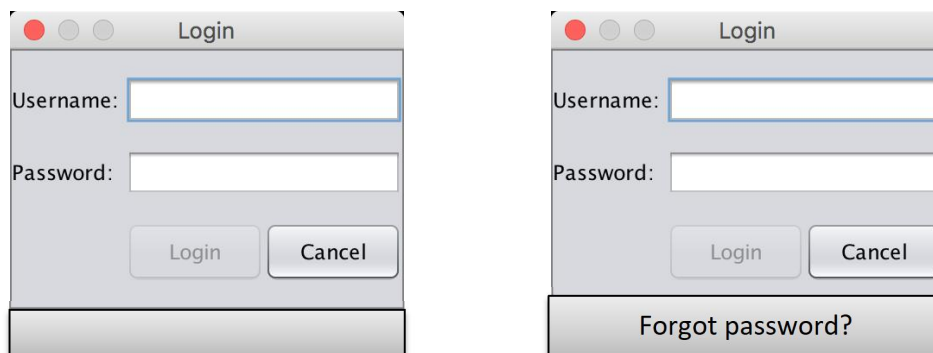
### Types of regression testing :

We have three types of regression testing

- Unit testing
- Regional testing
- Full regression testing

### Unit regression testing :

Checking the impact area of an application within a singlr module.



### Regional regression testing :

Checking the impact area of an application which affects some couple of modules is regional regression testing.

**Example :** In G-mail app if compose modified , inbox , sent , draft , spam has to be checked again.

## **Full regression testing :**

Checking the impact areas of an application thoroughly and completely because of major changes

**Example :** whats app. If whats app is changed into IOS from android for that we have to do check full app again it is full regression testing .

## **What is build**

- Build is also called as a version of an application
- Build is a compile and compress formats of a source code
- A build is given by development team

### **Note :**

In agile model for 4 weeks sprint we may get around (7 to 8) builds

For 3 weeks sprint we may get around (5to6) builds

Once we find a defect in one build it will be raised to developer he will fix it and gives another build so, we will do retesting in the next build

## **What does the build contains**

### **Test engineer answer :**

A build contains compile and compress format of the source code of new module old module and defect fixes.

We start with unstable build and we end with stables build. Test engineer should conform whether the build is stable or not

### **Developer answer :**

A build contains compile and compress format of the source code of new module old module and defect fixes.

We start with unstable build and we end with stable build. Test engineers should conform whether the build is stable or not

## **What are the formats of build**

### **Build can be of two formats**

- Compress
- Archive



Compress	Archive
For compress zip	For archive , jar , war , tar <div style="text-align: center;"> </div> Combination of path
Multiple file will be converted into single file	Multiple file will be converted into single file
Size of the file will be reduce	Size of the file will be almost same <ul style="list-style-type: none"> <li>• Jar → back end</li> <li>• War → front end</li> </ul>

### What is test cycle

It is a duration (or) time taken to test an application once a build is given

#### Example :

1<sup>st</sup> test cycle → build no.1 (ST , FT , IT , ST)

2<sup>nd</sup> test cycle → build no.2 (ST , FT , IT , ST , RT , reg T)

### What is release

Starting from gathering the requirement , developing an application testing an application and finally releasing the application to the end user is called release.

**Note 1 :** one application having release (or) multiple release usually for waterfall model. We have one release

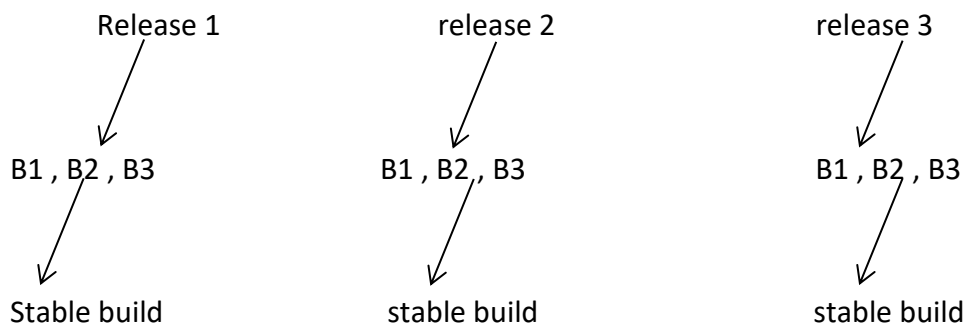
For agile model we have multiple release for each sprint.

#### Note 2 :

One release can contains multi builds

The final stable build will go to production server for customer (or) end user.

Example :                      whats app



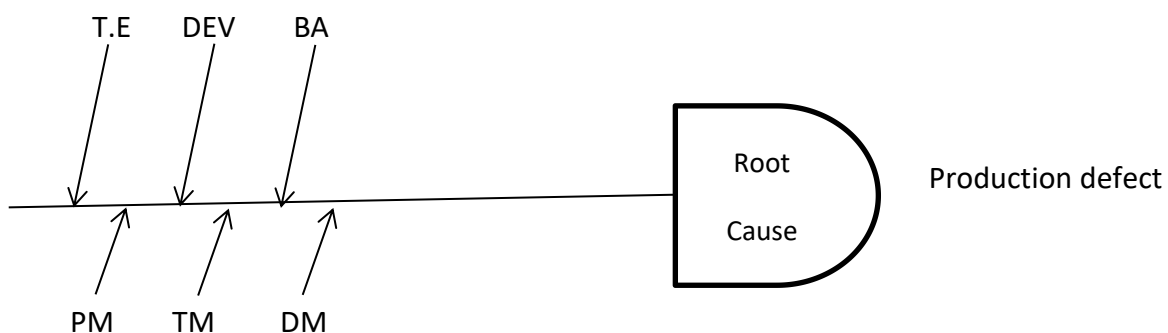
## What is Hot fix :

Once application is given to production server for end user to use . once end user / public finds the defect that will be informed / reported to the company and the software company will do check it and it will be fixed immediately and giving back to production server for end user / public this whole process is called hot fix.

**Note :** hot fix is a situation which is a negative impression for software company

Hot fix is happens in production server.

## Fish bone diagram



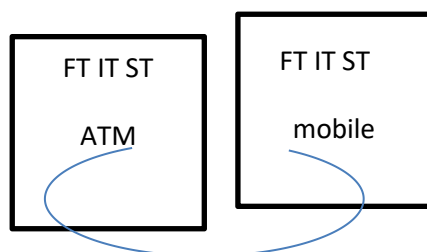
- It is a which is written to identify the root cause of production defect
- Each arrow represents the reason for the production defect
- Fish bone diagram diagram concept comes only if hot fix

## System integration testing :

Testing the data flow between two or more system is system integration testing.

### Example :

While using instagram in between some and feed is displaying like myntra , amazon , flipkart and some connections is between two more application and system.



## What is test bed

It is testing environment or a platform some times its also called as test data.

## What is test coverage :

It is metric in testing to ensure the everything is tested or not.

We can do this with techniques like

- Statement coverage
- Path coverage
- Branch coverage
- Cyclomatic complexity

Also we have two to do below activities to ensure everything is tested properly.

- Review
- Walk through
- Inspection
- Preparing RTM etc.

If test coverage is good we can prevent defect leakage.

**Example :** if a application having 500 lines of code ( LOC ) and if we execute 50 lines of code then test coverage is  $( 50 / 500 ) * 100 = 10\%$

## RTM and types of RTM

Requirement traceability matrix

It is the document prepared to check whether all the requirement having atleast one test case or not

- i. Forward RTM
- ii. Backward RTM
- iii. Bi-directional RTM

### Forward RTM :

Mapping from base document to derived document is forward RTM

**Example :** mapping from requirement to test case ID , test case ID to status

### Backward RTM :

Mapping from derived document to base document is backward RTM

**Example :** mapping from status to test case and test case to requirement

## Bi – directional RTM :

Mapping from derived to base and derived to another derived is bi-directional traceability matrix.

Requirement no	Requirement description	Testcase ID	Status
123	Login to the application	TC01 , TC02 ,TC03	TC01 – pass TC02 – pass
345	Ticket creation	TC04 , TC05 , TC06 TC07 , TC08 , TC09	TC04 – pass TC05 – pass TC06 – pass TC07 – fail TC08 – no run
456	Search ticket	TC011 , TC012 , TC013 , TC014	TC011 – pass TC012 – fail TC013 – pass TC014 – norun

## Bug leakage (or) defect leakage :

- Any defect or bug which is not found by test engineer but found by customer or end user is called bug leakage or defect leakage.
- The bug not founded by test engineer is called defects leaked
- The defect leakage should be 0% for a every good testing team.

## Defect masking :

One defect hiding another defect because of a defect we cannot find another defect

**Example :** if there is a defect in login module and homepage module . I cannot see the defect in home page module because the defect present in login module will hide the defect present in homepage module .

## Bug (or) defect seeding :

Intentionally injecting a defect in an app this is done to check the efficiency of test engineer and to see whether he is seriously working or not

This will be done secretly / confidentially by managers if they get doubt whether testing is properly happening or not

## Latent defect :

A defect which is there for long time in an application but that is found very late

It is bad situation for testing team.

### **Mutant testing :**

Mutation testing is the form of WBT , here the changes will be done to the software with the intention to cause errors in programming and to check the source code how it works

<b>Testing</b>	<b>Debugging</b>
It is process of finding defects or bugs	It is process of fixing bugs
It is done by test engineer	It is done by developer
It can be done both with manual & automation	It should be done by manual process

### **Defects / bugs / error / issue / failure**

#### **Defects :**

Any feature which is not working (or) working against the requirement

It is formal name

#### **Bugs :**

It is a informal name of a defect

Whenever test engineer finds the defect and report back to developer then developers will reproduce and accept that defect then the defect is name as bug

#### **Error :**

The mistake which is done by developer in the source code lead to error

#### **Issue :**

Problems faced by the customer is called issue

#### **Failure :**

Multiple issues in an app can lead to failure in customer business

#### **Bug release :**

A bug release is when a particular version of software is released with a set of known bugs / defects these bugs are usually low severity or low priority bugs.

## Software testing methodology :

Testing methodology is a strategy to check whether application is meeting the client exception this includes unit testing , integration testing , system testing , performance testing etc.

## What are the principles of software testing :

- Testing shows presence of defects
- Exhaustive testing not possible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence of error fallacy

### i. Testing shows presence of defects :

The main purpose of testing to find the defect and that is nothing but presence of defects.

### ii. Exhaustive testing not possible :

Exhaustive testing nothing but over test practically over testing is impossible. So exhaustive testing is also not possible

**Example :** requirement says text should contain 50 – 60 for this requirement we cannot test with all the numbers it is practically impossible

Because last number is infinity.

### iii. Early testing :

Early testing is nothing but we should start testing initial stage that is nothing but static testing

**Example :** review v and v model is best example for early testing

### iv. Defect clustering :

If there are any defect in one module we can expect more defects in same module so we have a principle call pareto principle which says approximately 80% of the defect in 20 % of the module

By experience we can identify such risky module

**v. Pesticide paradox :**

Sometimes if we do the same testing repeatedly we cannot find new defects

So we have to revise our test cases and should change it frequently to find new defects.

This is something like to avoid the insects we will have different methods to kill it.

**vi. Testing is context dependent :**

For every application every testing is not important but some testing is will be very important for every application

- 1) Functional testing
- 2) integration testing
- 3) system testing
- 4) monkey testing
- 5) performance testing
- 6) reliable testing
- 7) compatibility testing
- 8) globalization testing
- 9) security testing
- 10) User ability testing

For game app performance , reliability , adhoc , performance , compatibility.

Testing is always done based on domain.

**vii. Absence of error fallacy :**

Sometime development and testing will happen for the wrong requirement , if we don't check our own emails or any update from customer finally even through application is working fine for wrong requirement it is still unusable for customer

This can be avoided by doing static testing.

**Difference between UI and UX :**

UI	UX
UI is mainly used for interaction like buttons, icons , text field , display screen etc	Ux is mainly the feel what we get by interacting with entire application
UI is user interface UI is needed for UX	UX is user experience

## Configuration management process :

It is the process of maintaining systems such as computer hardware and software in a desired date.

## What is test closure :

It is the report which describes all the testing activities which is performed by QA team (or) test team

## Difference between test plan and test strategy

Test plan	Test strategy
It is defined for project level	Strategy is set and organization level it can be used at multiple projects
Test plan can change	Test strategy usually not be change
Test plan gives us exactly what to be performed	Strategy will says guide lines Strategy used as a general approaches

## What is test plan and explain it

Test plan is document it is doing future testing activity

It is done test lead (or) manager because test plan will be do for experience people .

Test plan contains different attributes or section like

- Objective
- Scope
- Schedule and milestone
- Entry and exit criteria
- Defect tracking
- Assumptions
- Risk
- ( contingency plan or mitigation plan or backup plan )
- Roles and responsibilities
- Environment / platforms
- Deliverables
- Graphs and metrics

## Note :

Test plan prepared by test lead (or) manager mainly

If needed I will also prepare test plan



### **1. Objective :**

This section tells about the purpose / aim of preparing the test plan

### **2. Scope:**

This will say the limitation

- a) Feature what to be tested (Ex : G mail)
- b) Features what not to be tested (Ex : FB , Insta )

### **3. Schedule and milestone :**

This section will tell which activity has to be done first and which actually has to be done next

It is just a time table of the project

Example : requirement (1 April) > write test case (20 April) > execute test case (10 April) > UAT (30 May ) > go live (5 June )

### **4. Entry and exit criteria :**

This section tells about when to enter and when to exit each type of testing

**Example :**

#### **entry criteria for functional testing :**

WBT should be done

Build has to be installed to testing server

#### **Exit criteria for functional testing :**

Pending defects should be lesser than or equal to

0 blockers , 1 critical , 5 major , 10 minor

#### **Entry criteria for integration testing :**

0 blockers , 1 critical , 5 major , 10 minor

#### **Exit criteria for integration testing :**

0 blockers , 0 critical , 3 major , 5 minor

#### **Entry criteria for system testing :**

0 blockers , 0 critical , 3 major , 5 minor

### **Exit criteria for system testing :**

0 blockers , 0 critical , 0 major , 2 minor

### **5. Defect tracking :**

This section will tell about whenever a defect is founded how to track the defect and which is the tool are using to track defect and what are the terminology we are using to raise the defects.

#### **Example :**

ALM tool

P1 or p2 or p3 → priority

Blockers , critical , major , minor , trivial → severity

Open , confirm , in progress , resolved verify status

### **6. Assumption :**

All the assumption for a project will be mentioned here

#### **Example :**

All employees will be available in the office everyday that is called assumption

### **7. Risk :**

All the risk which can occurred will be mentioned here

**Example :** some employees may take a sudden leave

### **8. Backup plan / contingency plan / mitigation plan :**

Before risk occurs we should arrange a regular among employees when they are available or not

### **9. Roles and responsibility :**

The section will tell about role and responsibility of the team members

#### **Example 1 :**

Test engineer ( manual testing → go through requirement → write scenario and test case  
→prepare RTM →execute test case →prepare test execution report → report defects  
→track defects

## Example 2 :

Software test engineer (automation)

All the responsibilities of manual test engineer and additionally below responsibility

Go through test cases and prepare test scripts

Execute test scripts maintain report

### 10. Environment / platforms :

In this section manager will tell in which platform we should do testing

**Example :** windows 10 , 11

Mac os

### 11. Deliverables :

Documents which is prepared for projects that is called as deliverables

**Example :**

Scenario , test case , defect track , RTM , test closure , test execution reports this all documents and deliverables.

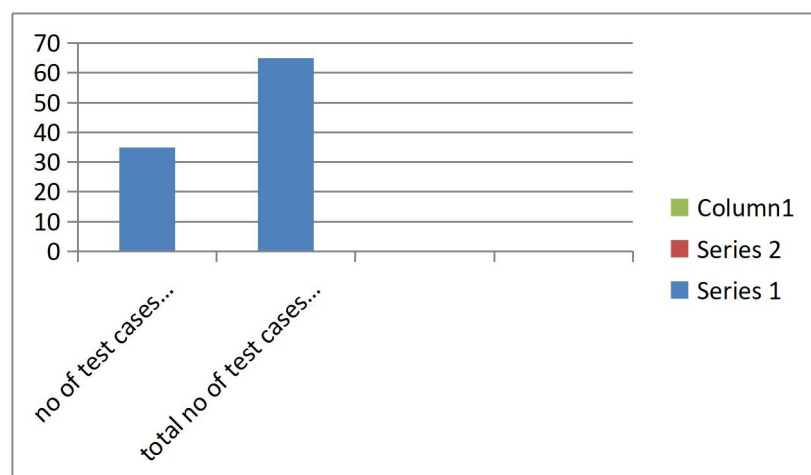
### 12. Graphs and metrics :

This section will tell about the graphite and the metrics which will be prepared for the projects.

**Example :** total number of test cases 100

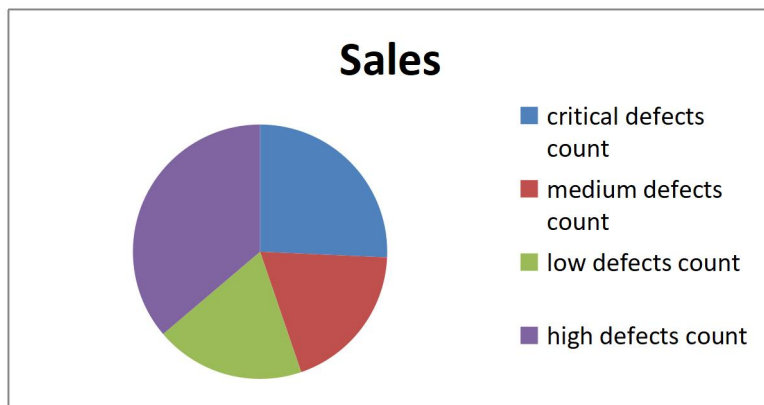
Executed 65

Not executed 35



Bar chart test metrics

## Pie chart



## Matrice and metrices

### Manual test result

Test cycle	Total no.of test cases	# of test cases executed	# of test cases passed	# of test cases fail
Cycle #1	88	88	88	0
Cycle #2	126	126	125	1
Cycle #3	174	174	173	1
Cycle # total test cycles	388	388	386	2

### Automation test result

Sprint	Total no.of test sprint	# of test scripts executed	# of test scripts passed	# of test scripts failed
Cycle #1	45	45	42	3
Cycle #2	110	110	109	1
Cycle #3	150	150	146	2
Cycle # total test cycles	305	305	299	6

## Scenario :

- Testing an application in all possible ways is an scenario
- Once requirement is given test engineer should think scenario to find the defects

### We have two types of scenario :

- Positive scenario
- Negative scenario

#### Positive scenario :

Testing with valid or expected data

#### Negative scenario :

Testing with invalid or unexpected data

#### When scenario is writing

- While developers are writing coding for the application test engineer will write scenario
- Scenario is converted into test cases.
- Scenario is different and defect is different.

### Difference between scenario and test case

Scenario	Test case
It will tell all the possible ways we can test the application	It is step by step procedure to test the application
Scenario says what to test	Test case says hoe to check
Scenario is high level document	Test case is low level document
Scenario doesn't have navigation steps	Test case have navigation steps
Scenario doesn't says were the exact defect is present , scenario takes less time to write	Test case says were the exact defect is present test case takes more time to write

## Test case template :

HEADER SECTION						
<b>Test case ID</b>		It is a unique name/ID given for every test case				
<b>Project name</b>		It is the name of the project which we are working				
<b>Requirement no</b>		It is the requirement number of the test case which we are writing here				
<b>Module name</b>		It is the name of module which we are testing				
<b>Test data</b>		It is the data which has to be ready before we execute test case				
<b>Preconditions</b>		It is the condition which has to be satisfied before we start execution				
<b>Testing type</b>		it is the type of testing which we are performing here				
<b>Priority</b>		It is the important of this test case to manage the execution				
<b>Testing environment</b>		it is the platform where we need to do testing				
<b>Brief description</b>		It tells what is the aim of this test case				
Body section						
Step number	Description / action	Input data	Expected result	Actual result	Status	Comments
Footer section						
<b>Author</b>						
<b>Reviewer</b>						
<b>Approver</b>						
<b>Approved date</b>						

## **Test case :**

- Test case is a document which has step by step procedure to test a application
- Once we ready with scenario we will converted into test cases
- To write test case we need requirement and test case template
- A test case template is a format to write the test cases
- This format will differ from company to company
- We can create test case template using excel.

## **Explain test case template :**

It is a format to write test cases it contains usually three section

- Header section
- Body section
- Footer section

### **HEADER SECTION :**

Header section contains different attributes like test case ID or name , project name , requirement number , module name , test data , precondition , testing type , priority , testing environment , brief description

### **BODY SECTION :**

Body section contains different attributes like step number , description (or) action , input data , expected result , actual result , status , commands

### **FOOTER SECTION :**

In footer section we have a attributes like author, reviewer , approver , approved date

- While writing test cases we fill header section completely , body section till expected result and footer section till author
- Once application given we start executing test cases while executing test cases we fill actual result , status and commands (if defect is there we fill command )

We have different types of test cases like functional test case , integration test case , system test case , smoke test case , regression test case.

## **Test scenario for ATM :**

- Verify ATM is ongoing
- Verify the touch screen of ATM
- Verify the ATM whether the card slot is there are not
- Verify the ATM has keypad
- Verify the ATM machine giving entered money
- Verify the ATM by inserting defect card
- Verify the ATM by inserting blocked cards
- Verify the ATM card chip is present
- Verify the ATM card without chip is working
- Verify the ATM card is connected with bank
- Verify the ATM machine display is clearly visible
- Checking the deposit money is correctly deposit in the account
- Check the ATM after removing the magnetic strip from the ATM
- Check the ATM is working with the wet ATM card
- Check the ATM IS working in which types of the temperature
- Check the ATM card is releasing after transaction is done
- Check the display size of ATM
- Check how many can be stored in ATM machine
- Check ATM has card less transaction
- Check ATM has display in how many language
- Check while touch the display it has sound feature
- Check ATM is working properly or it is hanging
- Check ATM has security alarm
- Verify by inserting other ATM bank card
- Verify by inserting the other cards ( aadhar card , pancard )
- Verify ATM is under security camera
- Check whether the ATM can open manual and take amount
- Check whether the ATM display has any crack
- Check whether the cancel button is there
- Check whether the clear button is there
- Check the transaction time taken in the ATM
- Check whether the entered money is correctly coming from ATM
- Check the weight of the ATM machine
- Check the weight of the ATM card
- Check the ATM by entering the card in wrong direction
- Verify the height of the ATM
- Check whether there is abnormal sound in ATM machine
- Check whether the message is coming after transaction
- Check whether it shows the balance money after transaction



## Scenario for face book module :

- Verify by entering valid Email and valid password and click on login button
- Verify by entering invalid email and valid password and click on login button
- Verify by entering valid email and invalid password and click on login button
- Verify by entering invalid email and invalid password and click on login button
- Verify by leaving email and password blank and click on login button
- Verify by entering email and leaving password blank and click on login button
- Verify by invalid email and leaving password and click on login button
- Verify by leaving email and entering invalid password and click on login button
- Verify by entering valid phone number and valid password and click on login button
- Verify by entering invalid phone number and valid password and click on login button
- Verify by entering valid phone number and invalid password and click on login button
- Verify by entering invalid phone number and invalid password and click on login button
- Verify by leaving phone number and password blank and click on login button
- Verify by entering phone number and leaving password blank and click on login button
- Verify by invalid phone number and leaving password and click on login
- Verify by leaving phone number and entering invalid password and click on login button
- Verify by entering email and password and click on forgot password
- Verify by leaving email and password blank and click on forgot password
- Check the face book logo
- Check the correct spelling of the logo
- Check the upper case of the logo
- Check the lower case of the logo
- Check the font size of the logo
- Check the colour of the logo
- Check the spelling of the login button
- Check the font size of the login button
- Check the colour of the login button
- Check the colour of the login
- Check by clicking login button and ensure it is working
- Check by entered wrong password forget password is highlighting
- Check the forgot password spelling
- Check the question mark is there at forgot password
- Check the create new account spelling
- Check the create new account font
- Check the colour of the font
- Check the colour of the create new account login
- Check the login button is working properly
- Check the logout button is working properly
- Check the background colour of the application
- Check the language of the face book application
- Check the like button of face book
- Check the privacy of the face book application

## Write scenario for pen

- Verify by looking appearance of pen
- Verify by print company name without spelling mistake
- Verify by outer colour of pen
- Verify by cap colour
- Verify by pen holding rubber cover colour
- Verify by at the top cap holes is there or not
- Verify by holding pen rubber grip
- Verify by quality of plastic
- Verify by pen length
- Verify by pen outer diameter
- Verify by pen inner diameter
- Verify by pen height
- Verify by pen cap height
- Verify by pen cap inner diameter
- Verify by pen cap outer diameter
- Verify by printing done on the refill
- Verify by pen nipple colour
- Verify by pen nipple diameter
- Verify by pen nipple cone length
- Verify by cap dimension and pen dimension fix properly
- Verify by barcode of the pen
- Verify by pen rubber grip length
- Verify by pen rubber grip outer diameter
- Verify by pen rubber grip inner diameter
- Printing sticker done on the pen
- Checking the elasticity of the pen
- Checking by writing in the paper
- Checking by writing in the rough surface ( wall and road )
- Checking sharpness of the pen nipple
- Checking comfort of pen while writing
- Check the refill is blank
- Checking whether the pen is waterproof
- Checking strength of the pen by applying tensile load
- Checking the pen by breaking in hands
- Checking the continues flow of the ink
- Verify the plastic is poison or not by beating
- Verify the pen and cap fixing properly
- Verifying the breakage by throwing fast of the ground surface
- Checking the refill strength by bending
- Checking whether ink is erase able or not erase able
- Checking the refill by cutting the refill with knife
- Checking the printing letters font size of the pen
- Writing on different papers like ( carbon paper , bus tickets , money)
- Checking ink capacity that how many pages can be write by the ink

## **Test scenario by applying ( boundary value analysis , equivalence partitioning , error guessing )**

From account number : it should accept only 10 digits

To account number : it should accept only 10 digits

Amount : it should accept between 100 - 5000

Permanent account number : it should accept 10 alpha numerical value no special character

### **From account number :**

- Verify by entering 9 digits numbers
- Verify by entering 10 digits numbers
- Verify by entering 11 digit numbers
- Verify by entering invalid 10 digit numbers
- Verify by leaving blank
- Verify by entering alphabet number
- Verify by entering special character
- Verify by entering alphabet number and special character
- Verify by entering alphabet number , special character , numerical value
- Verify by entering number in decimal

### **To account number :**

- Verify by entering 9 digits numbers
- Verify by entering 10 digits numbers
- Verify by entering 11 digit numbers
- Verify by entering invalid 10 digit numbers
- Verify by leaving blank
- Verify by entering alphabet number
- Verify by entering special character

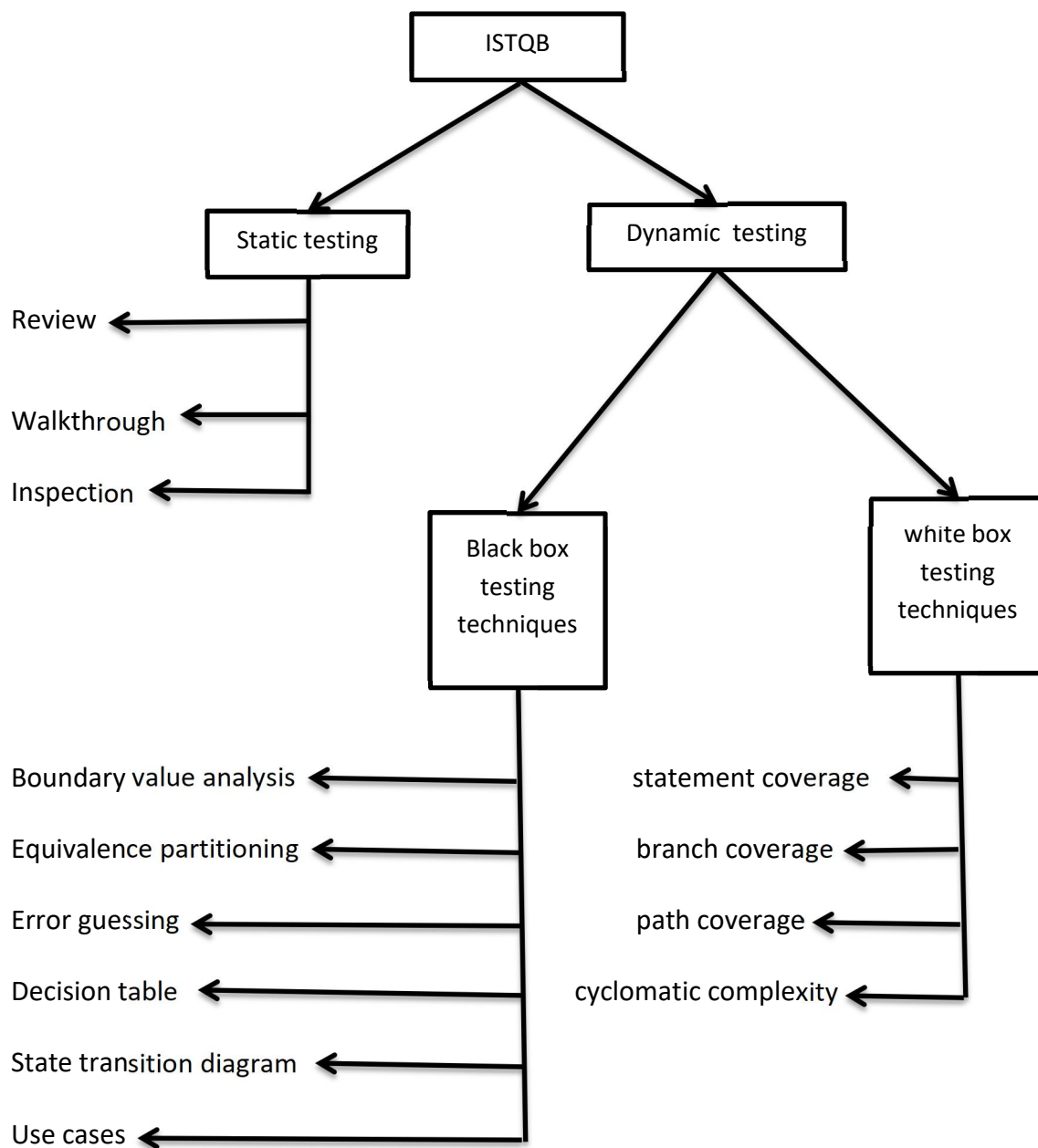
- Verify by entering alphabet number and special character
- Verify by entering alphabet number , special character , numerical value
- Verify by entering number in decimal

#### **Amount :**

- Verify by entering the value 100
- Verify by entering the value 5000
- Verify by entering the value 99
- Verify by entering the value 5001
- Verify by entering the value 99.99
- Verify by entering the value 5000.001
- Verify by entering the value 000
- Verify by entering the value 50
- Verify by entering the value 6000
- Verify by entering the value 2500
- Verify by entering the value 100.00
- Verify by entering the value hundred
- Verify by entering the value nooru
- Verify by entering 100 only
- Verify by entering rs.100
- Verify by entering 50 + 50
- Verify by leaving blank
- Verify by entering special character only ( @ , # , + , - , < , > )
- Verify by entering alphabet number , special character , numerical value

## International software testing qualification board (ISTQB)

It is certificate for software testing engineers



### Static testing :

Testing which is performed on the document is called static testing. There is no execution here checking requirement , test case , RTM documents is static testing

## **Types of static testing :**

### **Review :**

Finding the mistakes in the document is called review mistakes can be something which is missing or extra or wrong or confusion.

#### **In review we have types :**

1. **Self review** : review done by ourselves
2. **Peer to peer review** : review done by other who is equal to us
3. **Manager review** : review done by manager

### **Walk through :**

Explaining a document to a set of people who is unaware of it

**Example** : explaining about requirement by BA to project team members

### **Inspection :**

It is kind of auditing process which is done on the document for a software company we have recognition based on CMMI.

CMMI - capability maturity model integrated . it has 5 levels. CMMI level 1 , CMMI level 2 , CMMI level 3 , CMMI level 4 , CMMI level 5

The highest level of CMMI is level 5

CMMI is the recognition by SEI is software engineering institute . some of the level 5 company are capgemini , infosys , wipro , cognizant

### **Dynamic testing :**

Testing which is done on application is called dynamic testing (I.e) on the source code (or) user interface of an application there in this testing execution take place

Example : FT , IT , ST etc comes under dynamic testing under dynamic testing we have BBT technique and WBT technique

### **Black box testing / behaviour testing / closed box testing :**

Testing UI (user interface) or GUI (graphic user interface) of an application

It is done by test engineer

## Black box testing techniques :

Boundary value analysis :

- It is a technique where we can apply boundaries for some component in web page
- This can be applied for functional testing
- That is for the text field which is having requirement

**Example :** we can apply BVA for username , password , mobile number , email address etc.

**(EX-1):**

Requirement : text field with 10 - 20 characters

10	20
X	Y

X - 1	X	X+1	Y-1	Y	Y+1
9	10	11	19	20	21
-	+	+	+	+	-

**(EX-2):**

Requirement : amount text field with 100 - 5000 numbers

99,100,101                      4999,5000,5001 --> (3 value BVA )

**(EX-3):**

Requirement : 50 to 100 text field

49.9	50	50.1	99.9	100	100.1 (decimal value)
49.99	50	50.01	99.99	100	100.1 (decimal value)

## Equivalence partitioning :

- It is the technique where it is used for a requirement which has range of values
- It is mainly used for components like text fields

**Example :** user name , password , Email ID , address , comments etc.

**EX:** requirement says username should accept between 5-10 character like 3 , 7 ,15 here we have one valid value and two invalid value.

One valid value will be within the range , two values should be outside the range . it is used in functional testing.

## Error guessing :

It is the technique where will apply for components like text fields , text area , button , radio button , drop down , check box.

Error guessing is used in functional testing

**Example :** requirement says user name should accept between user to 7 to 15 character (only alphabet)

## Decision table :

It is a table which helps us to understand the complex requirement in easy way

**Example :** ( ISTQB - question no.10)

## State transition diagram :

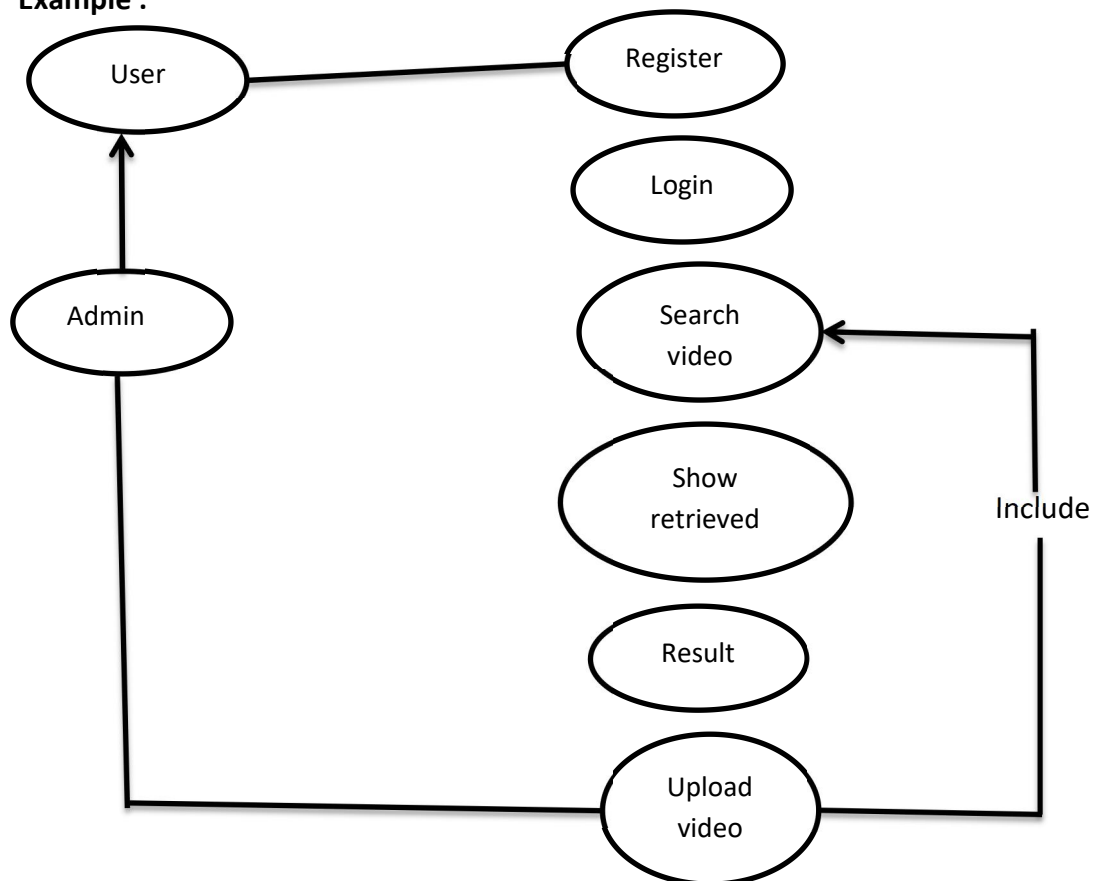
It is a technique which explain requirement in the form of a diagram

**Example :** ( ISTQB - question no.14)

## User cases :

It is a diagram which explains requirement in the form of actors & actions

**Example :**



**Note :** we do testing based on use case diagram that is use cases



## **White box testing techniques :**

Statement coverage , branch coverage , path coverage are the techniques which are available to cover all the statement , branches and path in a program

This is usually used by developer while doing WBT

It is also called as ( types of white box testing )

100% PC = 100% BC = 100 % SC --> true

100% SC not = 100% BC not = 100% PC = true

100% SC = 100 % BC = 100% PC = false

### **Types of WBT :**

#### **I. Statement testing :**

Testing each and every statement in a program including the rules , syntax , logics

#### **II. Condition testing :**

Testing each and every statement like if condition , else condition , switch condition etc.

#### **III. Path testing :**

Testing each and every path from starting till last line of the program is called as path testing

Example : ISTQB question paper 1 ( question no 27 )

Minimum tests required for statement coverage and branch coverage

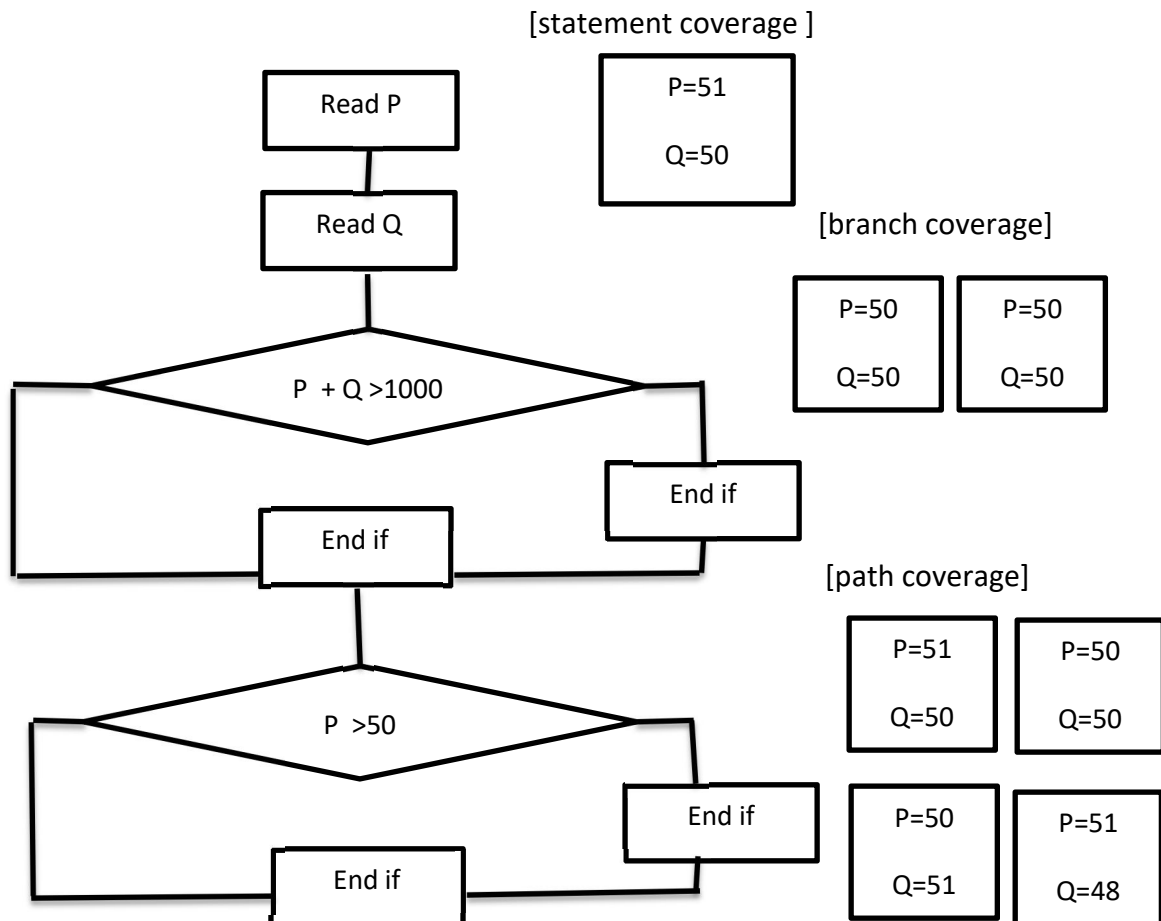
Read P

Read Q

if P + Q > 100 then

Print " large "

End if



### Cyclomatic complexity :

It is a technique which is mainly used to reduce the complexity of the source code

If the code is high complexity testing become high complex

Formulas to measure cyclomatic complexity are

Formulas - 1

$$cc = (c - n) + 2$$

E-Edges

N-nodes

Formulas - 2

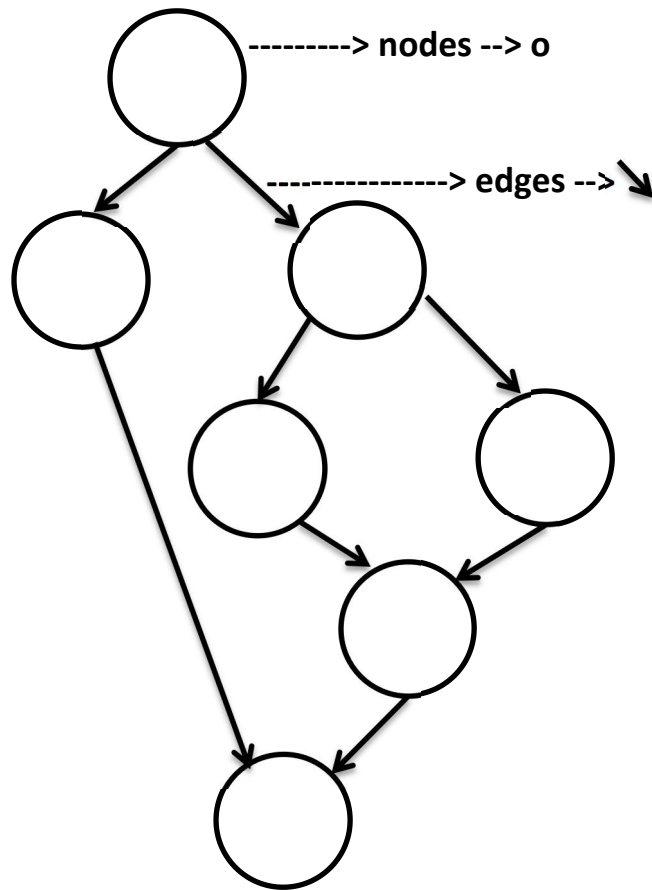
Numbers of predicate nodes + 1

Formulas - 3

Number of regions + 1

Formulas - 4

Number of paths



**Formula 1:**

$$Cc = (E - N) + 2$$

$$= (8 - 7) + 2$$

$$Cc = 3$$

**Formula 2:**

$$Cc = \text{no. of predicate nodes} + 1$$

$$Cc = 2 + 1$$

$$Cc = 3$$