MANUAL TESTING

Terminologies used in software industry

1) **REQ**: requirement
2) **Defect**: it is a deviation in the APPLICATION with respect to requirement. It is Formal name.
3) **Bug**: it's a informal name of a defect.
   Whenever a defect is found by a test engineer it should be raised to the developers. Once the developers accept the defect, it is termed as BUG.
4) **Error** : Mistake in the source code is known as error which is done by Developers.
5) **Issue** : Problem faced by the customer or the end users.
6) **Failure** : Multiple issues will lead to failure.

When the end users face lot of issues that will lead to the failure.

QA and QC

QA: is quality assurance.

It is a process oriented.

QC: is a quality control.

It is a product oriented.

QA	QC
It is a procedure that focus on providing assurance that quality requested will be achieved.	It is a procedure that focus on fulfilling the quality requested.
QA aims to prevent the defect	QC aims to identify the defect.
It is a method to manage quality verification	It is a method to verify the quality validation
It does not involve executing the programs.	It always involves executing the programs.
It is a preventive technique.	It is a corrective technique.

RTM: requirement traceability matrix.

KT: knowledge transfer.

MANUAL TESTING

DOMAIN: It is a categorization of the software application into various fields. Each software company will work for various domains.

Finacle: it is a banking software used by the banks.

Servers and environments

We have usually 3 types of servers.

1. Development server.(used by developers)
2. Testing server or QA server(used by test engineers)
3. Production server(end users)
   Eg .www.dev.facebook.com
   www.qa.facebook.com
   www.facebook.com

We have 3 different servers because developers, testengineers, end users should use the application and work independently.

When developers are working for 1 feature. That feature is given to testing team, during that time developers can start working for another feature.

Due to this

1. Everybody can work independently.
2. Dependency will not be there.
3. If any problems occurs it can be handled separately.
4. Disturbance will not be there for end users.

Stake holders

A person who directly (or) indirectly involved in a software application (or) project is a stake holder.

SDLC(SOFTWARE DEVELOPMENT LIFE CYCLE)

It is a procedure to develop the software application.

It consists of different stages or phases.

***CRS**: customer requirement specification*

***BRS**: business requirement specification.*

***SRS**: system/software requirement specification.*

***BA**: business analyst.*

MANUAL TESTING

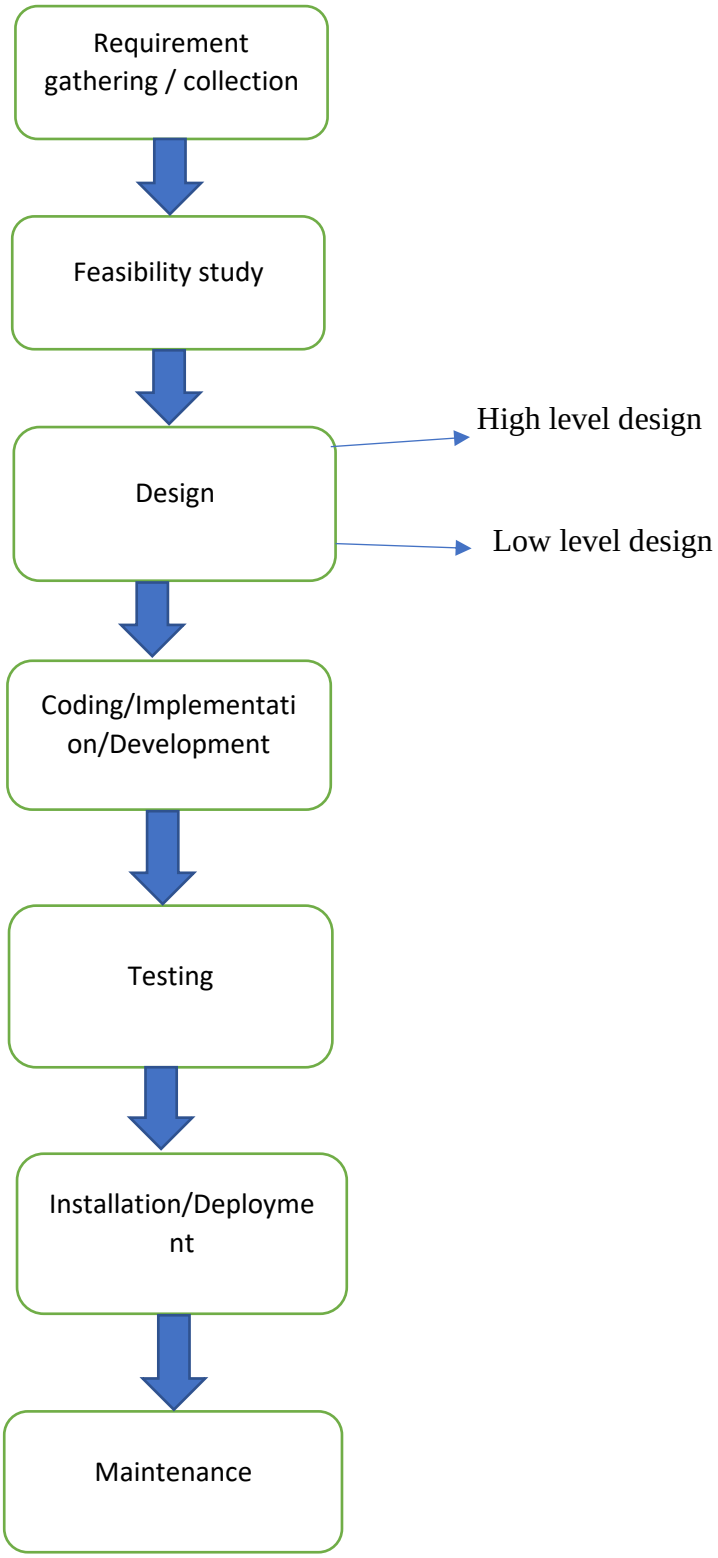
***HR**: human resource.*

***HLD**: high level design.*

***LLD**: low level design.*

***TE**: test engineer.*

***PM**: project manager*



MANUAL TESTING

Requirement Gathering/Collection:

Usually customer or client gives a requirement in the form of CRS/BRS. And it is converted into SRS by BA. BA is business analyst. BA acts like bridge b/w customer and company.

BA should be good in

1. Communication skills
2. Domain expert
3. Convincing skills.
4. Technical expert.
5. Analyzing skills

BA collects all the requirement from the customer and gives to Software Company.

This is Requirement Gathering/Collection.

Feasibility study:In this phase BA,HR,ARCHITECT,FINANCE,PM will discuss all the information that is needed about the Project.

BA will explain about the Requirement.

HR will think for needed resource.

Finance will take care about budget.

Architect will think which technology we should use for project.

PM will take the final decision

and gathers all the data information to proceed for the next stage or Phase.

Design: once everything is fine, we go for design. In design phase, we have HLD and LLD. Usually design phase is done by architect or senior developers.

High level design: It is just like a blue print. It shows the external architecture of the Application.

Low level design: It shows the internal architecture of the Application.

Both HLD and LLD are the documents.

Coding:Once the design is done,developers starts writing the code for application by looking at design and requirement.

This is done in Development Server. All developers will be involved in this.

Testing: Once coding is done, developers will give application to test engineers. All test engineers start testing the application and find defects. If any defects are found in software application, it again goes back to the developers. They will fix it and again given back to Test Engineers.Like this process continues, till Application meets the Requirement.

Once everything is completed, we will go for next phase called installation.

MANUAL TESTING

Installation/Deployment: Making the software application available for end users is called installation. i.e Moving the Software from Testing Server to Production Server.

It is done by the software company once testing is completed.Customer/Client should approve for installation.

It can be done by Installation Team/ Testing Team/Development Team.

Maintenance: Once software is provided to customer or end users, if they face any problem, a support has to be provided, that is maintenance.

Initially it will be free. Later it will be paid.

Free service will be provided based on the agreement b/w customer and company.

During maintenance, defect fix will be handled and changes will be taken care.

Changes can be Adding, Modifying, Deleting a feature.

TYPES OF SDLC MODELS

Water fall model

It is one of the oldest traditional models.

When we go for water fall model?

1. When the **customer freezes the requirement**
2. For any **short-term project**.
3. For developing **simple application**.

What are the advantages of waterfall model?

1. We can expect **stable application**.
2. There will be **no disturbance for the team members** if the requirement does not change.

Drawbacks of waterfall model?

1. Testing happens after coding.
2. **Requirement** and **design** are **not tested.**
3. Developers used to do testing before. (currently it is done by test engineers)
4. If **requirement changes** it leads to **lot of REwork**.

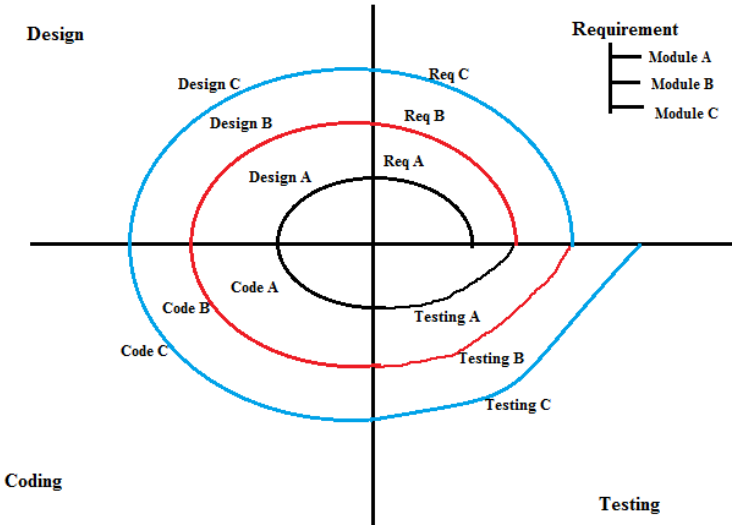
Why developers are not involved in testing?

1. Developers always focus **how to build** the software not to **break the software**.
2. They always be **over confident** what they do.
3. They **does not like to find their own mistake**.
4. Since it **is not easy** to find **their own mistake** , even though mistake are there they will **hide it.**

MANUAL TESTING

Spiral model / ITERATIVE MODEL

This is also one of the traditional model.



When do we go for spiral model?

1. When the customer gives requirement part by part.
2. When there is a lot of dependency b/w the modules.

What are the advantages of spiral model?

1. Customer can see the application partially and can get confidence.
2. Requirement changes can be done.

What are the drawbacks of the spiral model?

1. Requirement and design are not tested.
2. Testing happens only after coding.
3. Developers used to do testing.
4. If there is any requirement changes it may delay the projects.

Requirement changes are of two types.

1. **Major**: whenever there is a major changes we work only on that changes.we cannot work on the new requirement.
2. **Minor**: whenever there is a minor change we can work on that changes and also we can work on the new requirement.

MANUAL TESTING

V and V model

V and 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 and V is done only by test engineer.

Verification is QA process.

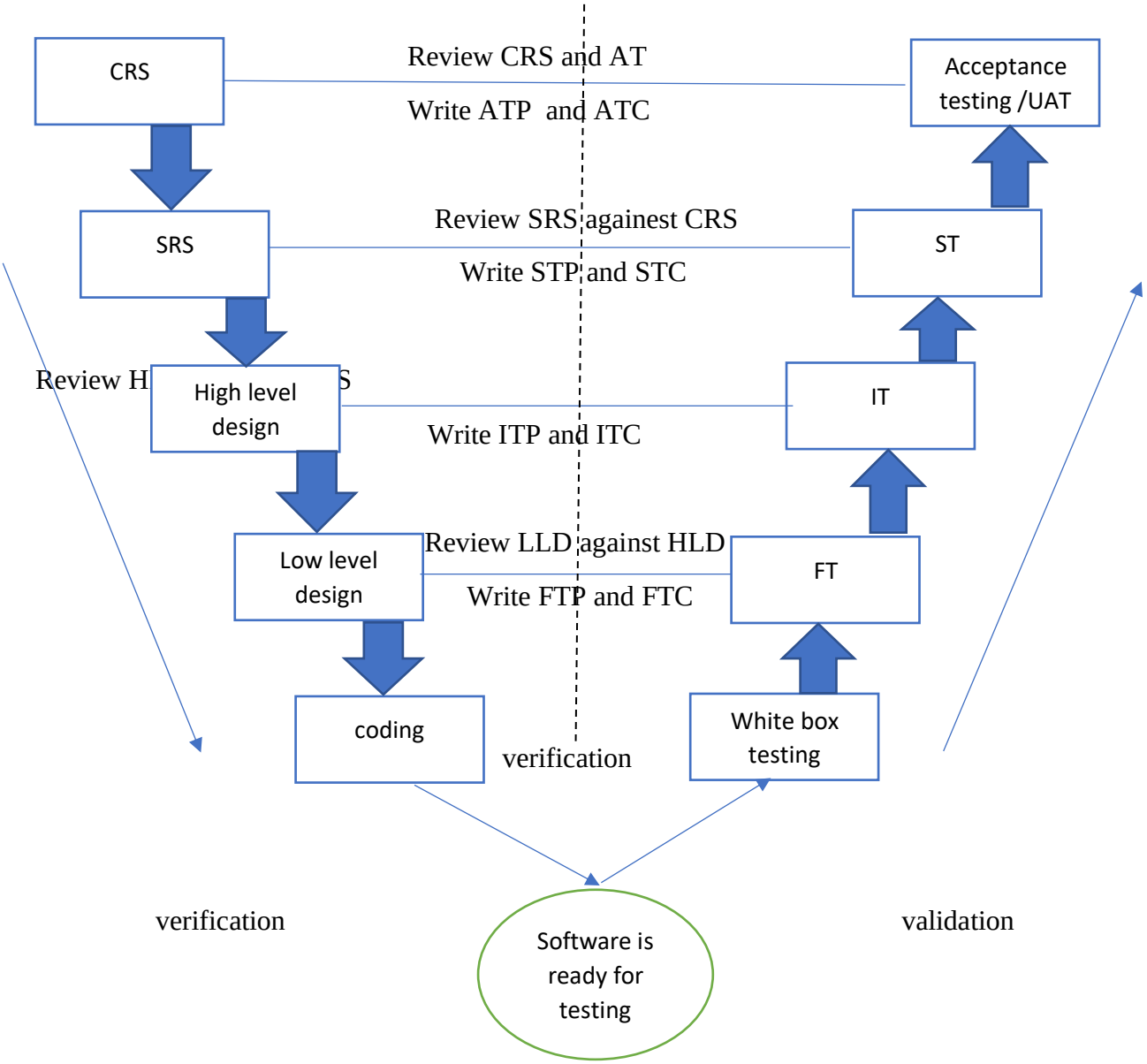
Validation is a QC process.

When do we go for V and V model?

1. When the customer needs high quality products
2. For complex applications

   Ex: Banking, health care applications, space applications, airlines applications, navy applications.

3. For Long term projects. (more than one year).



MANUAL TESTING

Advantages of V and V model

1. Testing is started at initial stage.
2. Requirement and design are tested
3. The downward flow of defects are less.
4. Requirements changes can be done.
5. Quality will be high compared to other models.
6. Rework will be less.

Drawbacks of V model

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

Explain about V and V model?

V and V means verification and validation model. It is one of the best model in SDLC. In this model the development and testing are done parallelly the left side of model is done by developers and the right side of the model is done by the test engineers.

When the customers give the requirement of 100 pgs documents in the form of CRS. It is converted in the form of SRS by BA. At same time, review of CRS is done by test engineers. If there is any mistakes it goes back, if not it will continue the process. The SRS will be reviewed against the CRS to find the defects. Parallelly they prepare the test plan and test cases.

Once the documentation of development process is done, with the design and coding, the software is ready for testing.

First testing is white box testing. This testing is done by the developers, then there will be FT it is done by the TE. At the same time the execution of the test cases are also done.

After FT the IT is done, later ST and then AT is done by the 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.

Prototype model

Prototype is a dummy model that is a non-working application.

When we go for prototype model?

1. When the customer is not clear about requirement.
2. When the software company is new to the domain, then they will go for prototype model.
3. When the developers are new to the technologies.It is a experimentation.
4. When the customer and software company are new to the business.

Advantages of prototype model

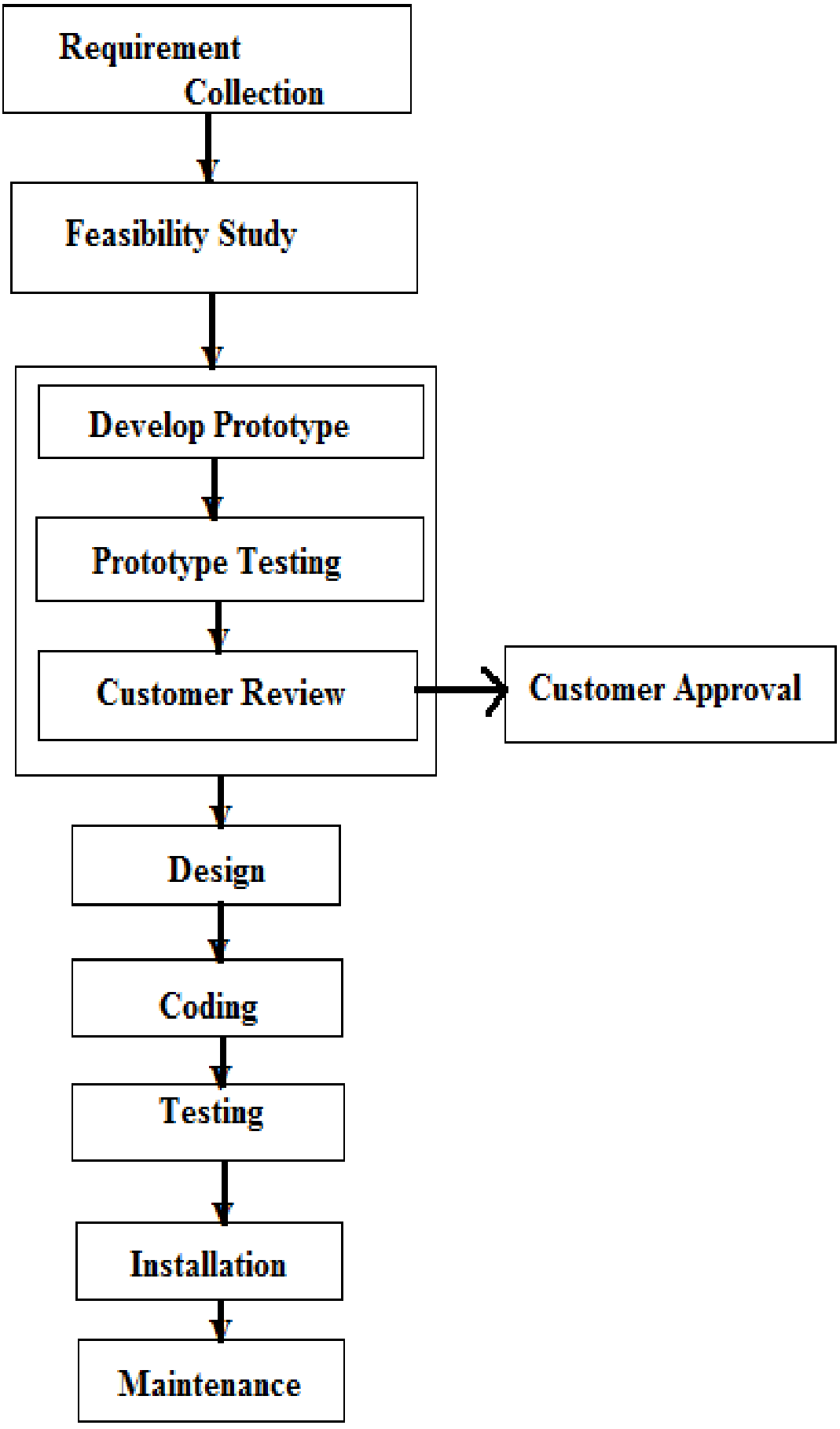
1. Initially customer can get to know what he gets on.
2. Initially itself developers will also come to know what they should deliver on last day.
3. Requirement changes can be done initially itself.
4. Initially investment is very less.

Drawbacks of prototype model

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

AGILE

MANUAL TESTING



MANUAL TESTING

SOFTWARE TESTING

- Testing the functionality of an application with respect to given REQ.
- Checking the application with the intent of finding the defect.
- Checking the behavior of an application to see whether it meets customer REQ or not.
- Testing the process of QA and QC.

MANUAL TESTING

Testing the software without using any tool and ensure its working fine.

AUTOMATION TESTING

Testing the software by using the tool to ensure it is working fine.

(the tool can be selenium)

Examples

> Televisionremote: if we operate by using the remote it is a automation testing
> if we do without remote it is manual testing.

Why testing is important?

Every software is developed for business purpose if testing is not done, end users may find the defects while using the applications.

This will spread negative impact in the market and number of users who uses application will be reduced. This will become loss for the investor on business. To avoid all those things s/w testing has to be done before it is released to end users.

SCENARIOS

Testing the application in all the possible ways is scenarios.

It can be +ve scenarios or -ve scenarios.

Testing the application with valid or expected data is called +ve scenarios. It is also called positive testing.

Testing the application with invalid or unexpected data is called -ve scenarios. It is also called negative testing.

To identify the defects we should find the scenarios first.

Scenario is not a defect or defect is not a scenario.

+ve scenarios and -ve scenarios are not a defect.

MANUAL TESTING

Few cases we may get confused, whether it is a +ve scenario or -ve scenario. In this case we can judge whether it is a defect or not a defect based on REQ.

DRAWBACKS OF MANUAL TESTING.

1. It is a time-consuming process.
2. It is a repetitive in nature (Because of REGRESSION TESTING)
3. Turn around time is more in manual testing, so we go for automation testing.
4. Need more Man Power.

RETESTING AND REGRESSION TESTING

RETESTING

Testing the defect fixed which is done on an application is called retesting.

REGRESSION TESTING

Testing the impacted areas of an application :

1. After the defect fixed.
2. After changes done in application.
   Changes can be adding, modifying and deleting a feature.

   Why Regression Testing has to be done ?

   When defect is fixed or New feature is added, old/existing features might stop working. To ensure existing features are still working fine, we have to do REGRESSION TESTING.

   Since regression is repeatedly done, many company will go for AUTOMATION. We can also do Retesting and Regression Testing Manually.

TOOLS USED IN INDUSTRIES

Functional testing tools

1. Selenium(can be used with core java ,python,perl…)

2. QTP/UFT(quick test professional, unified functional test) can be used with VBS(visual basic scripting, java script)

   Selenium is a free and is an open source.
   Thought works is a company that came up with the selenium tool.

   Qtp/uft is a paid and **licensed** is with HP.
   (HewlettPackard). Before it was with Mercury.

QTP is older version, UFT is latest version.

3. Winrunner.
4. Silk test.
5. Test partner.

The tools which we are using are

1. Functional testing tools
    1. Selenium : selenium is a tool which is in demand and it is a open source and free tool. The best combination is that selenium is used by core java. Thought works is the first company to come with the selenium.
    2. Qtp/uft: it is powerful tool and it is a paid and licensed tool.
       The best combination is that qtp is used by vbs.

Performance testing tools

Testing an Application by applying LOAD and check RESPONSE TIME.

1. Load runner
2. J meter.
3. QA Load
4. Silk performer.
5. Neo load

Load runner is a licensed tool. It is with HP.

Test management tools

1. QC/ALM(quality center /application life cycle management)
   It is a licensed tool. It is with HP. It is PAID TOOL
2. JIRA.(best tool for AGILE model)
3. OTM(Oracle Test Manager)
4. Test link

Using the above tools we can manage the testing activities like below.

Ex :QC/ALM. Below are the modules of QC/ALM.

Req-Add requirement/Edit req/Delete req

Test Plan-Write test cases

Test Lab -Execute test cases.

Defect - Report/Raise/Log a defect.Track a defect.

MANUAL TESTING

What are the test management activities?

1. Going through the req given by the customer and adding to the test management tool.
2. Write test cases by identifying scenarios
3. Execute test cases and write results.
4. Find defect and report a defect. Track a defect.

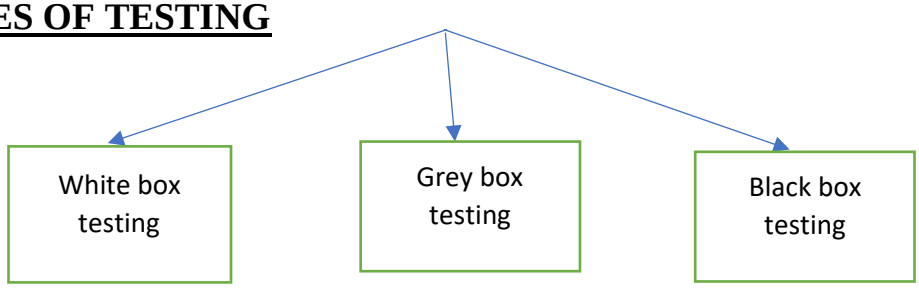
Defect or Bug Tracking tools

1. Bugzilla: It is a open source and free.
2. QC/ALM(quality center /application life cycle management)
   it is a licensed tool. it is with hp.
3. JIRA.(partially free)

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

We can only report the defect and track the defect.

TYPES OF TESTING



Unit testing Functional testing

Glass box testing Behavioral testing

Structural testing Closed box testing

Transparent testing

Open box testing

White box Testing : Testing the **source code or internal structural** of an application is called **White box** testing.

It is usually done by **developer.**

When situation demands, even Test Engineer can do this if we know coding.

Black box Testing: Testing the **user interface (UI) or graphical user interface (GUI)** of an application is known as black box testing.

MANUAL TESTING

Here we don't worry about internal structure of an application. It is done by **test engineers.**

Grey box Testing: Testing the source code and user interface of an application parallelly together is called grey box testing.

It is done by a person who is aware of both testing and source code of an Application. Means he can be a developer or test engineer. OR he can be SDET(Software Developer Engineer in TEST).

White box testing tools / Unit testing tools

Junit: when the source code is written in java to test that we use junit.

Nunit: when the source code is written in .net to test that we use nunit.

Selenium is a tool to test the black box testing. It cannot be used for WBT.

WBT

manually

1. Check each and every code(LOC) Manually without using a tool Ex .check the syntax and logics etc

automation

1. Write a pgm to develop
2. Write a pgm to test the developed pgm
3. Run the test pgm in a tool

BBT

manually

1. Check each and every Functionality without using a Tool on ui.

automation

1. Write a pgm(TEST SCRIPT)(using selenium, qtp)
   To use selenium core java is used
   For qtpvb is used.
2. Run the test pgm in the tool.

MANUAL TESTING

Difference b/w scenario and test cases.

Scenario	Test cases
It will tell all the possible ways we can test the application	It is step by step procedure to test the application.
Scenarios says “what to test”	Test cases says “how to test”.
Scenario is a high-level document.	Test case is a low-level document.
Scenario doesn’t have a navigation steps.	Test cases have a navigation steps.
Scenarios doesn’t say where the exact defect is present.	Test cases will tell where the exact defect is present.
Scenarios take less time to write.	Test cases take more time to write.

Template is a format used to write a test cases.

Test cases

A test case is a document which contain all the possible scenarios and explains step by step.

When do we write test cases?

When the developers are developing the application, we will write a test case.

There are two stages

1. Test case preparation/writing
2. Test case execution.

Under the test case preparation, we have to fill following sections like

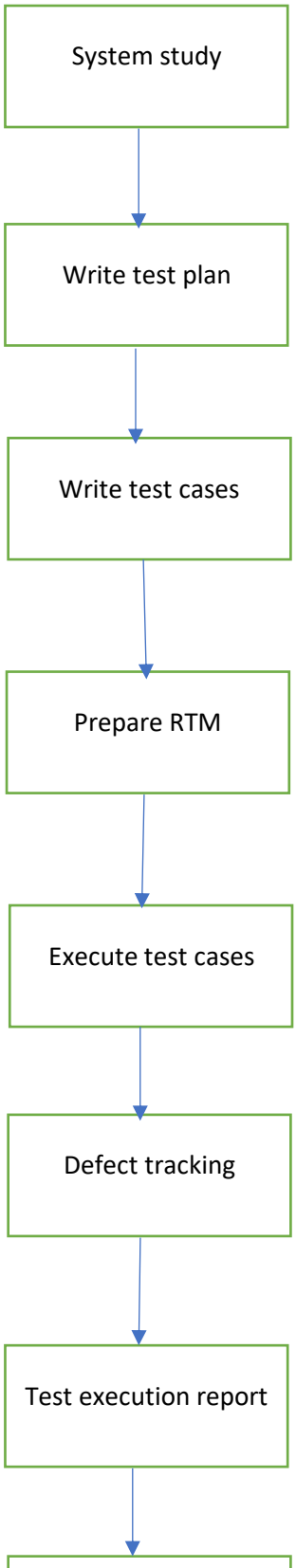
HEADER, BODY, FOOTER sections.

MANUAL TESTING

Software testing life cycle (STLC)

It is a procedure to test the software application.

It has a different stages or phases like



MANUAL TESTING

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

Write test plan: It is a document which is prepared for FUTURE testing activities.

It is done by the test lead or test manager. Because the plans will be done by experienced people.

Write test cases: 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 scenarios and then converted into the test cases. To write the test case, we need requirement and test case template or tools (qc/alm, JIRA)

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 cases.

Execute test cases : Once the requirement is given to the test engineer, he writes test cases for the application. After the developer gives the developed application, then the test cases are executed and it is compared with expected result and actual result. If the expected result and actual result 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 cases.

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 the defects.

These defects are raised/logged/reported to the developers.

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

Test execution report: Once we execute the test cases, 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.

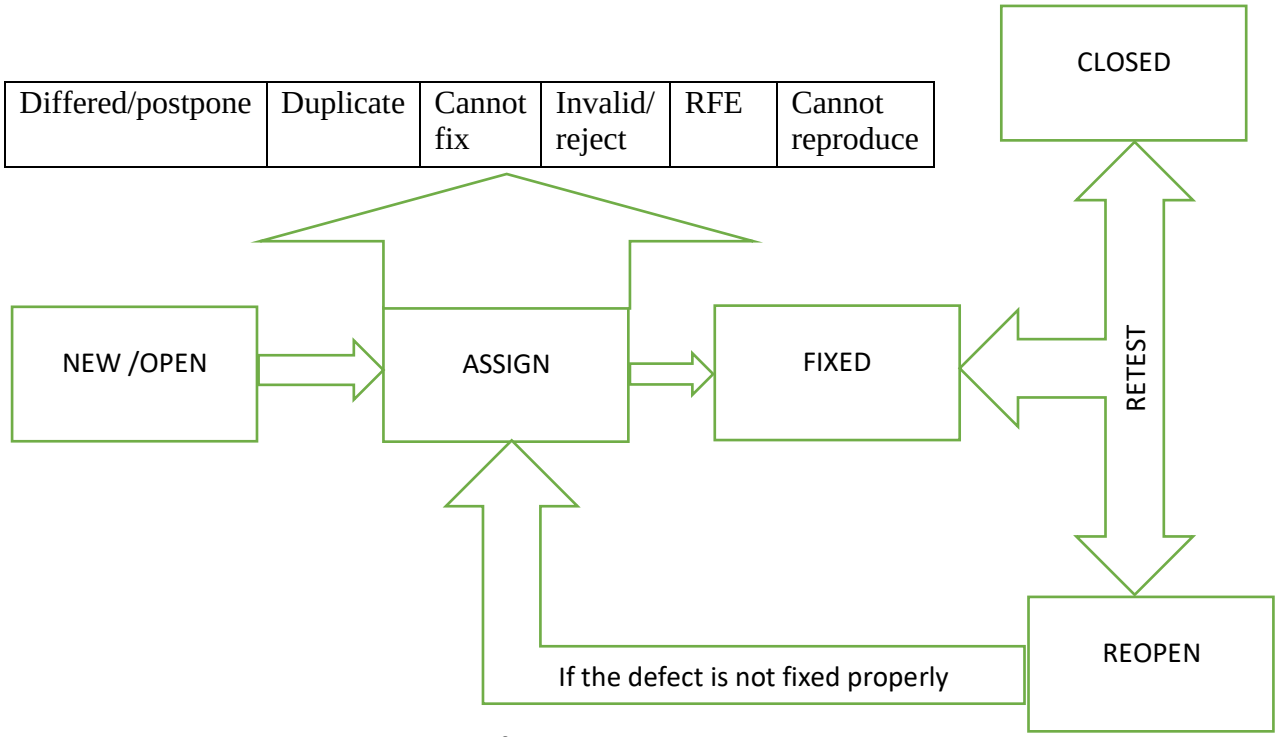
Till here customer involvement will be there. According to the customer, this is the end of the project.

Retrospective Meeting/Postmortem Meeting:

MANUAL TESTING

In this meeting, all the team members will gather together like PM, TL,DL, DEV,BA, TM and they discuss about the good and improvements needed once the project is done. This will be very useful for next release or next project.

DEFECT LIFE CYCLE OR BUG LIFE CYCLE



When we are executing the test cases if the expected result and actual result are not same, we 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 concerned developer or development lead (If we don't know the correct developer), then the status is **ASSIGNED.**

Then developer will reproduce the defect and if it is Reproducible, it will be accepted. Then he starts fixing the defect in the development server and it has to be installed to Testing Server. Then he should change the status to **FIXED.**

The test engineer starts retesting the defect fixed 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 whole process is called DEFECT LIFE CYCLE / BUG LIFE CYCLE / STATUS OF DEFECTS.

What is deferred/postpone?

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

Ex : Whenever there is a major defect and minor defect. Developer have 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 or postpone**.

What is duplicate status?

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

To avoid this DUPLICATE status, check whether the same defect is already raised or not in the Repository(Ex : BUGZILLA, QC/ALM, JIRA) before raising any defect.

Can't be reproduce /unable to reproduce:

When the test engineer finds the defects, but developer cannot be able to reproduce that defect. So he will change the status as **"cannot be reproduced".**

Ex :mobile hang .. etc.

Reasons for can't be reproduce

1. Installation problem
2. Improper defect report.
3. Inconsistent bug. (A bug which occurs sometimes and does not occur sometimes)

Can't be fixed

Whenever the developer is unable to fix the defect 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 fix the defect.

Ex : Whenever Technology does not Support to fix that defect, then developer will change the status to **"can't be fixed"**

Finally what to do for this defects will be decided by Managers / Customer.

What is Invalid/Rejectstatus:

Whenever the test engineer raises the defects but developer will not accept the defect as VALID and he changes the status to **INVALID / REJECT.**

Reasons for invalid :

Due to the misunderstanding of the requirement.

1. If test engineers misunderstand requirement
   New/open → assign → invalid → closed

2. If developer misunderstand the requirement
   New/open → assign → invalid → assign/reopenfixed → closed

Request for enhancement (RFE)

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

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

Types of severity

1. Blocker / Showstopper
2. Critical
3. Major
4. Minor
5. Trivial (this defect is negligible)

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

1. High
2. Medium
3. Low

OR

1. Urgent
2. High
3. Medium
4. Low

OR

1. P1
2. P2
3. P3

Who will set severity and priority?

Test engineer will set severity and priority.

Developer or managers can discuss about this if they want to change it they have to provide proper justification.

Priority is very much important for the developer to decide which defect has to be fixed first

If priority is not there, developer may fix the easy defects and they leave all the important defects

As a project point of view, the important part of defects has to be fixed earlier. To manage these things, we have severity and priority for each and every defect.

PRIORITY CAN CHANGE ONCE OTHER DEFECTS ARE FIXED AND CLOSED.

TEST CASE TEMPLATE

Test case name / ID	
Project name	
Module name	
Requirement number	
Test data	
preconditions	
Test case type	
priority	
Test case environment	
Test case description	
Test steps	

Step NO	Description/Action	Input DATA	Expected result	Actual result	Status	comments

Author	
Reviewer	
Approved by	
Approved date	

Header part

Test data: it is the data we need to execute the scenarios.

Pre conditions: it is the action which we should do before executing the scenarios.

Test case type: in this section we should write which type of testing we are going to use.

Priority: assigning the priority to each and every test cases will help us in prioritizing the defects

Test case description: summery of test cases.

DEFECT REPORT

When ever the test engineer finds the defect it has to be raised or logged or reported to developer. For this we have to create a defect report.

When we create the defect report in tools like BUGZILLA, or by using the excel file or word documents

Template for defect report

Defect ID	This will be auto generated if we use the tools
Build no	It is the number of build where defect has found
Test case no	It is a number of the test case where we found this defect
Status	
severity	
Priority	
Test environment	
Module no	
Reported by	
Brief description	
Test data	

Steps to reproduce

Expected result	
Actual result	
Attachment of screen shot	

MANUAL TESTING

ACCEPTANCE TESTING:

It is a testing which is mainly done to check the business scenarios of the application which is done by the customer.

(or)

It is an end to end testing which is done by the customer to ensure the application is fit for business scenarios or not.

Why acceptance testing is done?

1. This is mainly done to get a **confidence for customer** before he releases the product to the end users.

2. Because of business pressures, software company might be releasing the Application to the customer with some defects. To ensure this is not there, customer will do acceptance testing.

3. If the product is released to the end user without checking the business scenarios, it will affect the customer business. To avoid this, acceptance testing has to be done.

4. We may forget some of the business scenarios to test and those scenarios would be tested by the customer.

Note:

If the customer finds the defect during the acceptance testing. That is negative to test engineers.

So before giving application to customer, we should think all those business scenarios and finds defects.

Alpha testing and Beta testing are types of acceptance testing

Difference between alpha and beta testing

Alpha testing	Beta testing
Alpha testing is performed by tester who are usually an internal employee of an organization.	Beta testing is performed by the client or end users who are not an employee of an organization
Alpha testing is performed at the developer site	Beta testing is performed at the client location or end user of product
Reliability and security testing are not performed in depth in alpha testing	Reliability and security and robustness are checked during beta testing
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 testing environment	Beta testing doesn't require lab environment or testing environment because software is

	available for end users and is said to be real time environment
Long execution cycle may require for alpha testing	Only few weeks execution are required for beta testing
Critical issues or fix are addressed by developers immediately in alpha testing	Most of the issues or feedback are collected from beta testing will be implemented in future version of the product
Virtual environment	Real time environment
Done at Offshore	Done at Onsite
Alpha testing is done under controlled environment	Beta testing is done under uncontrolled environment

Approaches of acceptance testing.

Approach no 1 : **BA or IT Engineers of customer** will do acceptance testing at customer place. (BETA)

Approach no 2: **Employees or test engineer of customer** will do acceptance testing at customer place. (BETA)

Approach no 3: **Test engineers of software company** will do acceptance testing at customer place. (BETA)

Approach no 4: **Test engineer or BA of software company** will do acceptance testing under the control of customer at developer place. (ALPHA)

Hot fix / incident management system

When the software company releases the product to customer/end user, if the end user/customer finds any defects in the software, customer will raise an incident to software company and software company should fix the defect as early as possible and give back to the customer. This situation is called hotfix.

If any problem is find by the customer or end user in the production server, it will be raised as the incident to the software company. Every incident is raised in the form of ticket and every ticket will have a priority.

The commitment between software company and customer is called service level agreement.

(SLA)

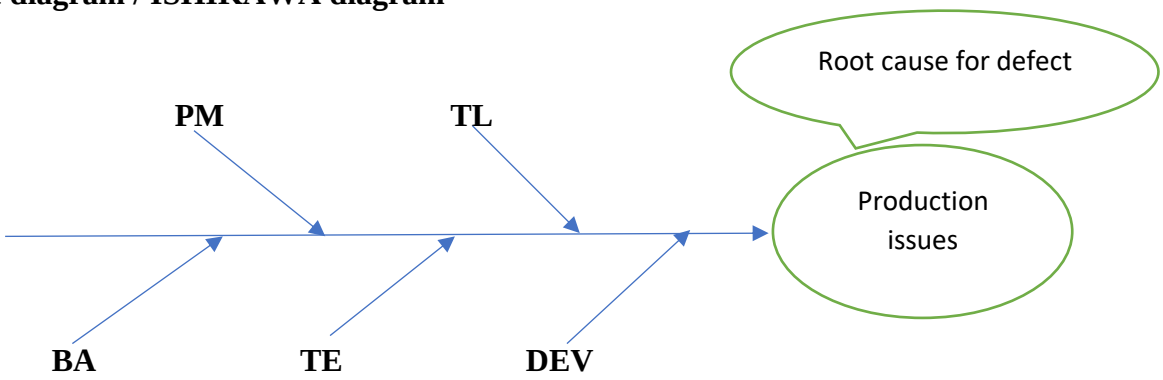
Ex:

PRIORITY of Ticket	Time given to fix
P1	– 3hrs
P2	– 10 hrs
P3	– 1 day
P4	– 3 days

MANUAL TESTING

P5 – 10 days

Fish bone diagram / ISHIKAWA diagram



Whenever there is a production issue all the team members will gather together, and they discuss about reason for production issues that is documented in the form of fish bone diagram. The main purpose of this is to find the root cause of the production issue.

SHORT TERM RELEASE/INTERIM RELEASE

Between two planned releases of an application, some times customer gives a unplanned release which is done in short duration. It is called short term release.

REAL TIME PROJECT (OR) HOW ARE THE SERVERS USED (OR) BUILD PROCESS

Usually we have 3 servers like development server, testing/ QA server, production server. Customer gives the requirement then the developer start writing the source code for developing the application in the local system, this source code will be saved in the repository(Ex: github,vct, cvt). There white box testing will be done by developers. The source code will be compiled and compressed, then we will be getting a file called BUILD. All this will be happened in development server.

The build has to be installed from development server to testing server. It will be done by the installation team or testing team or development team, depending upon project to project.

Once build is installed to the testing server, testing team will perform different type of testing depending upon the content of the build. ex. For new features, FT, IT, will be done for old features regression testing. for defect fixes retesting will be done.

While doing the testing, we find the defects that will be raised to the developers. Development team will reproduce the defect and they will fix it in the local system and it is saved in the repository again. Same process will be repeated for every build, until we get the final stable build.

Once the testing is completed, we will give the application to the customer for acceptance testing. Customer will do acceptance testing in the testing server or customer will have their own server called user acceptance testing server (UAT Server).

MANUAL TESTING

Once everything is fine application is released to end users that is for production server. This is called production release OR Go-LIVE.

What is release?

Starting from gathering the requirement, developing the application, testing the application, finally releasing it to the end users is called release.

That is called production release or GO-LIVE.

What is a build?

The compiled and compressed format of source code is called build.

A build contains different formats like compress and archive.

Compress format:

1. Zip
2. Multiple files can be made in single file.
3. **Size of the file will be reduced.**

Archive format

1. .Jar(java archive), .war (web archive), .tar (tape archive) , .gz(Gun Zip)
2. Multiple files are made in a single file.
3. **Size of the file will be almost same.**

What does build contain?

A build contains

1. New features
2. Old features
3. Defect fixes

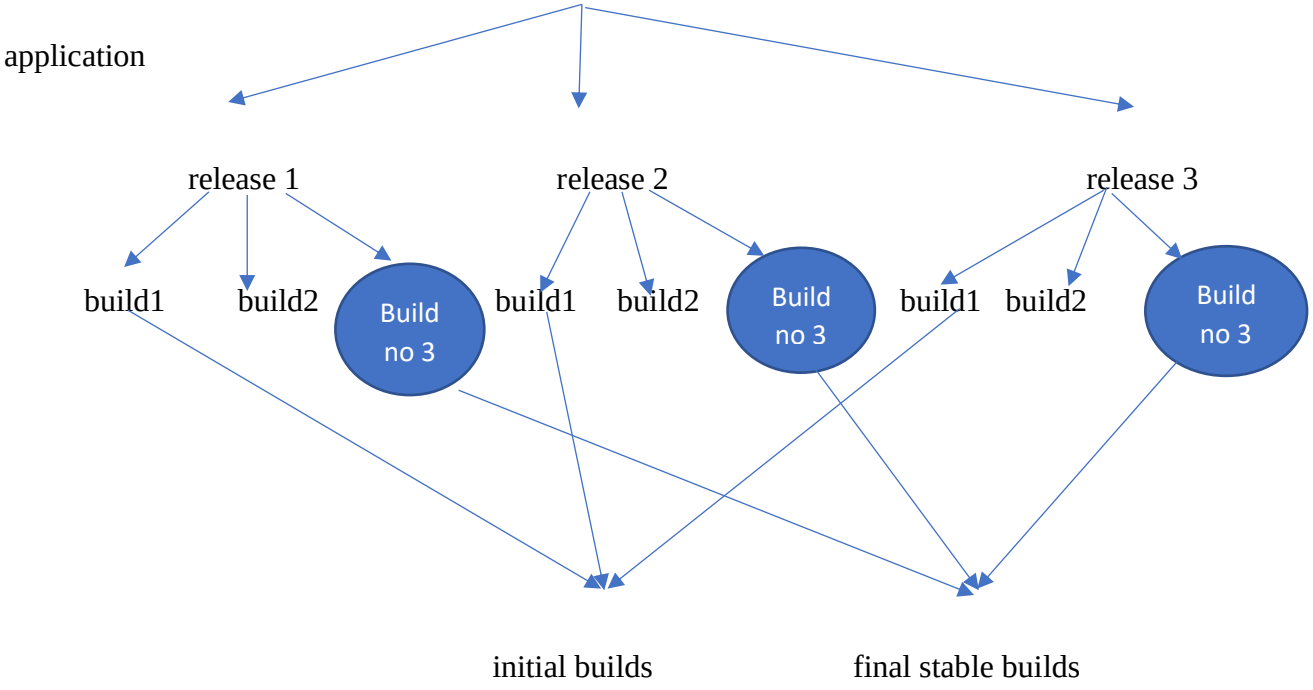
It depends upon the contents which developer will add inside a build, and it varies from build to build.

What is test cycle?

It is effort or time spent to test the application once the build is given. The duration of each test cycle can be days or weeks or months, depending upon the build and application size.

MANUAL TESTING

Retesting and regression testing with respect to build



A single Application can have single Release or Multiple release.

A Release can have multiple Builds.

The final stable build will be released to production server.

The **first build of first release**, **we cannot do retesting and regression testing**.

For all other remaining builds, we can do Retesting and Regression testing based on the situations.

TEST STRATEGY

Test strategy is the approach to test the software application.

Based on the situation, one approach should be selected.

TEST PLAN

Based on that selected approach, we plan how to take it forward.

How do you a define a format of good test case?

MANUAL TESTING

A good test case should have the header, body and footer sections.

It should contain the attributes which are easily understandable.

Ex in body section it should contain

1. Step no
2. Description
3. Input
4. Expected result
5. Actual result
6. Status
7. Comments

If the test case contains all the possible scenarios that can be called good test case.

What is a good test case?

a good test case is a test case which is easily understandable to every test engineer.

It should cover all the scenarios with respect to requirement.

A good test case should be written by applying like boundary value analyses, equivalence partitioning, error guessing.

A test cases has to be reviewed by another test engineer and find mistakes in it. Those mistakeshave to be corrected then it becomes a good test case.

What would you do if you have a large suite and execute it in very less time?

First, we have to execute the basis features of an application that is smoke testing.

Difference b/w functional and non-functional testing?

Functional testing	Non-functional testing
Functional testing Integration testing System testing	Performance 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 People.	Checking for multiple users 1. Multiple platforms 2. Multiple languages

What is negative testing?

Testing the invalid or unexpected data then it is called negative testing.

How do you ensure that the testing having the good coverage?

We prepare RTM document.

MANUAL TESTING

Apply test case technique.

SMOKE TESTING (OR) BUILD VERIFICATION TESTING (OR) CONFIDENCE TESTING

It is testing the basic features of an application before we do thorough testing like functional, integration, system testing.

Why we do smoke testing?

Test engineer will get confidence on basic features, that is working fine or not.

If we find the defect in the basic features only, that create a good impact on the testing team.

If there are any blocker or critical defect found in the basic features, developer will get more time to fix the defect.

When we do smoke testing?

Once the build is given by the developer, we start doing smoke testing first.

For every build given by the developer we start with smoke testing first we assure basic feature are working fine.

When time is less for testing team to do the testing, we will do smoke testing and if the customer is fine, we will release to customer.

Difference b/w formal and informal smoke testing?

Formal smoke testing	Informal smoke testing
We have documentation(TEST CASE)	There is no Documentation
We have a proof	There isNO proof
It has a procedure	There is no proper procedure

Difference b/w smoke testing and sanity testing

Smoke testing	Sanity testing
It is testing the basic 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 the unstable builds	It is done on the stable builds
Only positive testing is done	Both positive testing and negative testing are done
	It is the subset of regression testing.

MANUAL TESTING

ADHOC TESTING (OR) MONKEY TESTING (OR) GORILLA TESTING

Testing the application **randomly** without following any formal documents like requirement, test cases etc.

Why we do adhoc testing?

An application can be used by different set of people or end users,

like:

1. Children
2. A person with happy mode
3. A person with depressed mode.
4. A drunken person
5. Illiterate
6. Mad

When we are not able to get the defect through our regular testing like FT, IT, ST we can think about adhoc testing so that we can get more defects.

This is the best testing for gaming application. Because while playing games usually children will click where ever they want without their knowledge.

This testing is **mandatory for gaming applications**, for other applications we will do testing based on customer request or based on situation.

When we are not getting much defects with functional, integration and system testing think about adhoc testing we will definitely get defects.

All the defects we found through adhoc testing need not to be fixed by the developer unless it is crashing the applications.

PERFORMANCE TESTING OR SCALABILITY TESTING

Testing the response time of an application by applying load for an application is known as performance testing.

Load: number of users who uses the application.

Response time: time taken to get the expected screen based on the user actions.

Types of performance testing

Load testing:

MANUAL TESTING

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

Ex: if the 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 designed no of users, that can be around 10-20% greater than the load.**

Ex : Req-100000 users ------ response time should be within 2sec.

Stress testing 1,10,000 users --------- response time 2sec

Volume testing

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

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

Soak testing

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

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

Can we do performance testing manually?

Yes, but there are some drawbacks, like:

1. **Too much cost** will be involved because of multiple human resource and multiple devices, a huge place for gathering everybody.
2. There will be **no accuracy in the results**, we human beings we do not perform all the actions at same time because of this result may vary.

So, we go for automation (Ex: J METER, LOAD RUNNER etc)

RELIABILITY TESTING

Testing the functionality of an application **continuously for some duration of time**. This can be done by **single users**. It is to check the consistency of an application.

Ex: Mobile should function properly with one user for long time.

MANUAL TESTING

Standalone applications: it does not depend on the server. It is installed separately

Ex calculator, calendar, word.

Client server application: using an application by server by using internet connection.

Web applications:fully depends on internet connection.

Ex. Chrome---gmail--- facebook

TYPES OF APPLICATIONS

Generally, we have three types of application

1. Standalone applications
2. Client server application
3. Web applications

Standalone applications:

This kind of applications can be installed, accessed and used without any dependency on server or internet.

Ex notepad, calculator, word, calendar.

This is the fastest applications with respect to response time.

Client server applications:

In this case we have software in two categories.

1. Client software
2. Server software

Client software will be installed by the end users usually from play store.

Server software will be available at the company location.

To use client software to interacting with each other, we need server software. To connect to the server software, we need internet.

In client server application static elements will be already stored in the mobile , only dynamic elements will be accessed through the internet.

This is the fastest compared to the web application and slower compared to standalone application.

Ex whatsapp, facebook, gmail, redbus app.

Web application:

MANUAL TESTING

In this we access the application through the browser. Here static and dynamic element will load together.

Ex: we can access gmail or fb any other application by entering the url into the browser.

We need internet connection compulsory.Here the browser will act as client.

COMPATIBILITY TESTING

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

Why do we do compatibility testing?

1.To ensure that application is working for **multiple platforms** because there might be **different types of users.**

2.To check whether the application is **working in all platforms** or not.

Software Compatibility Testing :

Ex : In Different O.S like :

- Windows(w7, w8, w8.1, w10 etc)
- Linux(ubunto,federo etc)
- MACINTOSH(Tiger, Lion, Leopard,snow leopard, mountain lion, sierra etc)

All above is for PC

- Android (mobile)
- Alpha, beta, cupcake, donut, éclairs,foreo, gingerbread, honey comb, icecream sandwich, jellybean, kitkat, lollypop, mar……, noughet, oreo, pie,
- IOS(mobile)

Hardware Compatibility Testing :

- **Ex : In different Processor like – Intel, AMD**
- **In different Mother Board like – Intel, ASUS**
- **In different RAM**
- **In different ROM**
- **In different Hard disk**
- **In different Graphics card**

MANUAL TESTING

Mobile compatibility testing:

When we test an application for different operating system and also for each different versions like kitkat, windows, ios and for each **brand** of mobile

(ex:Samsung, redmi, nokia etc)
For each brand, for different **models** like A series, M series, Galaxy for SAMSUNG - Xperia c4, c5, z5 for SONY - v15, v7, v19 for VIVO

Browser compatibility testing:

Ex:
Chrome
Mozilla Firefox
Internet Explorer
Opera
UC
Netscape Navigator
Duckduckgo
Safari

What kind of bugs we find in compatibility testing?

1. Look and feel changes
2. Object overlapping (Login and cancel button sitting on one another)
3. Certain images will not displayed in certain browsers
4. Scroll bar issues
5. Alignment issues/problems
6. Scattered content
7. Certain buttons, links and components may work in one browser and may not work in another browsers

GLOBALIZATION TESTING

Testing an application which is developed for multiple languages is globalization testing.

When the language is changed, the translated content may not be proper because a machine translator could not understand the exact meaning of the word displayed. This may happen because of below reasons:

1. Machine does not have feeling
2. It cannot understand exact meaning.
3. It cannot understand exact grammar.
4. It cannot understand exact spelling.

All these are reasons for wrong translation. So, we go for human translator.

If a person is very good with multiple languages local and international, we could perform good globalization testing.

Types of globalization testing

1. I18N: Internationalization
2. L10N: Localization

Internationalization

Testing the application whether it displays the right content at the right place in the right language is called I18N testing.

Ex :

Localization

Testing the application with respect to the local culture or local standard to that country or state or region is localization testing.

Rs	$
Dd/mm/yyyy	mm/dd/yyyy
Pin code	zip code
(560097)	(CA 12345) , (SJ 25634)

MANUAL TESTING

USABILITY TESTING

GPS

VPS

Touch screen

Keypad

OLA

Pick up , drop , type of cab, confirm

Your name, mother name , father name, age, aadhar card etc

UBER

Pick up , drop , type of cab, confirm

M1 marks, 1st sem , btech – pass , yofp, backlogs?

Mobile

Long press, PIN, password, pattern, finger print, facelock, iris, voice,

ACCESSIBILITY TESTING or ADA Testing or 508 ACT rule

Testing whether the application is suitable for physically challenged people or not.

Ex : RGB (red green blue) colour should not be used in the application.

Every feature or component in an application should be accessed by mouse and also by keyboard.

RGB color, blind people will not be able to view that.

Mouse and keyboard: anybody who is injured with wrist, they can still use the application using the keyboard.

MANUAL TESTING

Tools used for accessibility testing

1. In focus
2. Wave

RECOVERY TESTING

Testing the application whether it is able to recover from crash state or not.

AESTHETIC TESTING:

Testing the beauty of an application is called aesthetic testing.

USABILITY TESTING

Testing the application, whether it is user friendly or not

Who should do usability testing?

1. End users is the best
2. Customer or client 2nd best.
3. Others project team members.
4. Nobody is there, test engineer has to do usability testing.

Test strategy:

Strategy is an approach to do any type of task.

TEST PLAN:

Test plan is a document which is prepared for future testing activities.

It is usually prepared by the test lead or manager.

Test engineer can also involve – if there is any support needed for test lead or manager.

Test plan contain different sections or attributes like:

1. **Objective:** this section tells about the purpose of preparing the test plan.
2. **Scope:** **This will say the** limitations.
   This section will tell about
   a) Features what to be tested. (Ex: GMAIL)
   b) Features what not to be tested. (Ex: FB,INSTAGRAM)
3. **Schedule and milestone:**
   This section will tell which activity has to done first and which activity has to done next.
   It is just like a time table of the project.

   Ex:

MANUAL TESTING

REQ - wtc – Etc - AT(milestone) – PR(milestone)

4. **Entry and exit criteria : T**his section tells about when to enter and when to exit each type of testing.
5. **Defect tracking:** This section will tell about whenever a defect is found how to track the defect and which is the tool, we are using to track the defect. And what are the terminologies we are using to raise the defect.
6. Assumptions: This section will tell about what are the assumptions we have during this project.

   Ex : All employees will be there in the office daily
7. Risk: This section will tell about what are the possible risk happen during the project.

   Ex : All employees are not in office. Leave/Quit
8. Contingency plan or mitigation plan: this section will tell about how to over come the risk which occurs during the process.

   Ex : Arrange regular KT among the employees when everyone are there in office, before Risk occurs.

9. **Roles and responsibilities**

   This section will contain what are the roles and responsibilities for team members.

   a. **Ex: Roles and responsibilities for test engineers**.(**Manual test engineer**)
   Going through the requirement and understanding the requirement and identifying the scenarios and writing the test cases, prepare RTM, and execute test cases when the build is given, find the defect and raise the defects.
   b. For automation test engineer:
   We should perform above task plus we should write test scripts and execute test scripts by using the selenium tool. Manage test script.
10. **Environment: T**his section contains which platform we are using for testing purpose ex. Window 10,8,7 chrome browser,
11. **Deliverables:** This section will tell what are the documents that should be prepared for the project. **EX:** Test plan, Test cases, RTM, Defect reports etc.

12.Graphs and metrices: This section contains the graphs and metrices that are prepared for project.

MANUAL TESTING

RTM (REQUIREMENT TRACEABILITY MATRIX)

It is a document which is prepared to test that every requirement having at least one test case or not.
Mapping from requirement to test case to automation test script to defect. This is called **forward traceability matrix.**
Mapping from defect to automation test script to test case to requirement this is called **backward traceability matrix.**
Mapping from test case to requirement and test case to automation test script to defect this is **bidirectional traceability matrix.**
Mapping from test case to automation test script to defect can be **traceability matrix.**

FUNCTIONAL TESTING / Field level Testing / Component Testing

Testing each and every component individually and thoroughly with respect to requirement is called functional testing.

INTEGRATION TESTING:

Testing the data flow between two or more dependent modules is called integration testing.

SYSTEM TESTING:

Testing the application end to end just like a real user is called system testing.

It is a done on a Testing Server, which is similar to Production server.

System Testing Scenarios

Ex 1 : Sign UP -> Login-> Compose->Sent Items-> Logout

Ex 2 : Login -> Inbox->Read Email->Reply-> Sent Items->Logout

Ex 3 : Login ->Settings->Reset Password-> Save New Password-> Logout-> Login(new password)->Logout

BEFORE MARRIAGE

9 am - 9 pm - 10 defects

MARRIED

AFTER MARRIAGE

10 am – 5 pm - No defects Build

MANUAL TESTING

Defect seeding:

Injecting the defect in the application by developer to check the efficiency of test engineer is called defect seeding.

This is done based on the request of manager.

Build 01

Login-A - Defect raised (MASKING defect)

Compose-B - MASKED DEFECT

Build 02

Login – A -Retesting

Compose-B – Defect

Defect masking:

Whenever one defect is hiding another defect that is defect masking.

Defect leakage:

The defect which is not found by the testing team but it is found by the customer or end users it is called defect leakage.

For a good testing team defect leakage percentage should be 0%.

Defect Triage Meeting/Call

Manager arranges a meeting with all the development team and testing team and discuss about the existing defects which are not fixed because the defect age should not be increased. And defect status should be moved for further level until the defect status is closed.

This meeting is known as defect triage meeting. It is the main part of agile meeting.

APART FROM CLOSED STATUS , ALL ARE EXISTING DEFECTS.

(New,Assign,Deferred,Fixed,Unable to Reproduce,Duplicate,Reopen,RFE)

Main intention of this is to CLOSE the defects as early as possible.

MANUAL TESTING

Defect Density:

Defect density = Number of defects / Total size of the project **(LOC/Duration)**

Ex :

DD = 100 / 2000 **(loc)**

DD = 100 / 6 months **(duration)**

Number of defects means the defect which are considered as valid defect not all the reported defect.

Latent Defect : It is a defect which is found once application is released to Production server and that defect is there in Production server for long time.

SIT :

(System Integration Testing)

Testing the data flow between 2 or more systems

FB

Sit

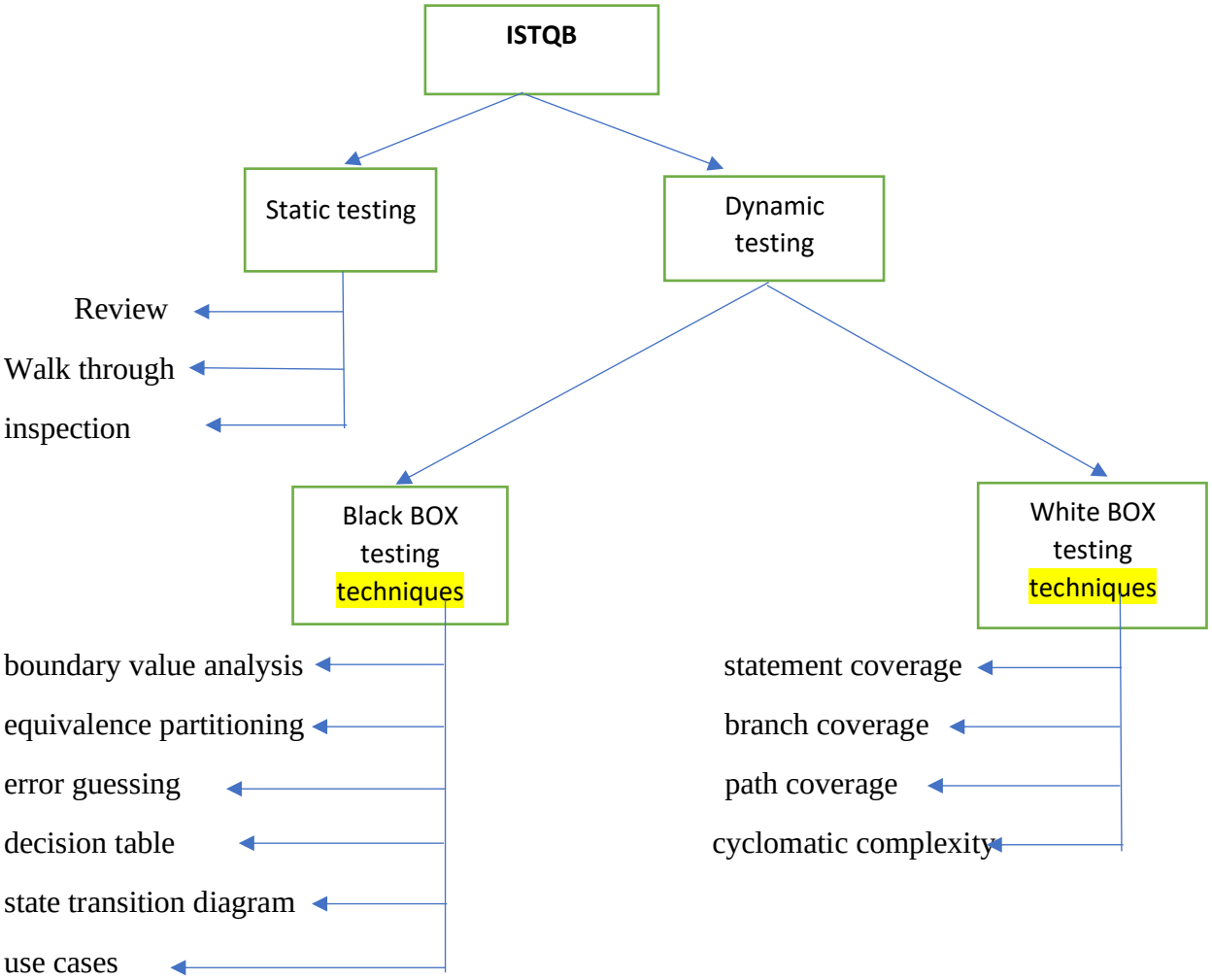
GMAIL

MANUAL TESTING

ISTQB

International software testing qualifications board

It is a certification for software testing engineers.



STATIC TESTING:

Testing which is performed on the document is called static testing.

There is no execution here.

Types of static testing:

Review:

Finding the mistakes in the document is called review.

In review we have types

MANUAL TESTING

1. Self-review: review done by ourselves
2. Peer to peer review: review done by others who is equal to us.
3. Manager review: review done by managers

Walk through:

Explaining about the document to the set of people who are not aware of that.

Inspection:

It is the kind of **auditing** process which is done on the documents for a software company.

CMMI: capability maturity model integrated. (FOR SOFTWARE COMPANIES)

It has five levels CMMI 1, CMMI 2, CMMI 3, CMMI 4, CMMI 5.

Highest level is CMMI Level 5

SEI: Software Engineering Institute

CMMI is the recognition given by SEI

ISO : International Organization for Standardization.

It is for Hardware.

DYNAMIC TESTING:

Testing which is done on the application is called dynamic testing, that is on source code or user interface of an application

Here in this testing, execution takes place.

Black box testing techniques

Boundary value analysis:

Testing the boundaries of certain components or element present on web page is called boundary value analysis

Ex testing on text field which accepts only 3-10 characters.

Equivalence partitioning:

It is the technique where we come up with given range of requirement. Here it contains one valid and two invalid data.

Error guessing:

MANUAL TESTING

It is the technique where we come with all the possible error message for different scenarios . It is error guessing

Here we have more of Negative scenarios.

Decision table:

It is the table which helps us to decide how many numbers of test cases are needed for a given set of requirements.

The formula for this is **2^n**= no of test cases or scenarios.

Where n is number of conditions or components

State transition diagram:

It is a technique which is used to come up with the flow of an application in a easy manner.

This can be used in system testing and integration testing.

For atm machine this diagram helps a lot.

Use cases:

Use case testing is the functional black box testing technique that helps us testers to identify the scenarios that excises whole system on each transition basis from start to finish.

Characteristics of use case testing

Use case can capture interaction between actor and system.

Actors represent user and interaction is that each user take part into.

Test cases based on use cases and are referred as scenarios.

Capability to identify the gaps in the system which would not find by testing individual components is isolation.

Very effective in defining the scope of acceptance testing.

White box testing techniques:

Path testing: testing all the independent paths in a program is called path testing.

100% **statement coverage**is not equal to 100% branch coverage is not equal to **path coverage.**

100% **branch coverage** is not equal to 100% **statement coverage**.

100% **path coverage** is equal to 100% **branch coverage** + 100% **statement coverage**

Cyclometric complexity

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

If the source code is complex then testing also become complex.

MANUAL TESTING

Formula1: (e-n) +2

Formula 2: number of predicate nodes+1

Formula3: number of regions + 1

Formula 4: number of paths

Conditional testing:

Testing all the conditional statements like **if, else, switch to**, is called conditional testing.

Loop testing:

Testing all the looping statement like **for, while, do while** is called looping testing.

RETESTING

Testing the defect fixed or bug fixed of an application is called retesting.

REGRESSION TESTING

Testing the impact areas of an application after

1. After the defect fixed.
2. After changes done in application.
   Changes can be adding modifying and deleting a feature.

Why we do regression testing?

To make sure that existing features are still working properly after the changes is done in the application or whenever a defect is fixed for an application.

When do we do regression testing?

1. After the defect fixed.
2. After changes done in application.
   Changes can be adding modifying and deleting a feature.

How do we do regression testing?

Test engineer will write test case manually the test cases which has been executed again and again will be categorized and a separate document will be maintained as regression test cases

Whenever we have to do regression testing, we will start testing by using the test cases.

Since the regression testing is repetitive in nature better, we will always go for automation testing.

IN AUTOMATION REGRESSION TESTING TYPES:

Full execution:

Executing all the test scripts is called full execution.

Batch execution:

Executing the partially test scripts is called batch execution.

MANUAL TESTING

Types of regression testing

Unit regression testing:

Whenever a new feature is added for existing module and the impact is only for that particular module, we can also call it as unit.

When we do regression testing for that unit, we do not do regression testing for any other unit that is called unit regression testing.

Regional regression testing:

Whenever there is a change in one module and that module is having dependency with other few modules not all the modules, then we do regression testing for all the dependent modules, we call them as regional regression testing.

Full regression testing:

Whenever there is any change in the application and because of that change if we have an impact on the whole application then we do complete testing again for whole application. This is called full regression testing.

Difference between QC and QA

Quality analysis	Quality control
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 includes dynamic testing

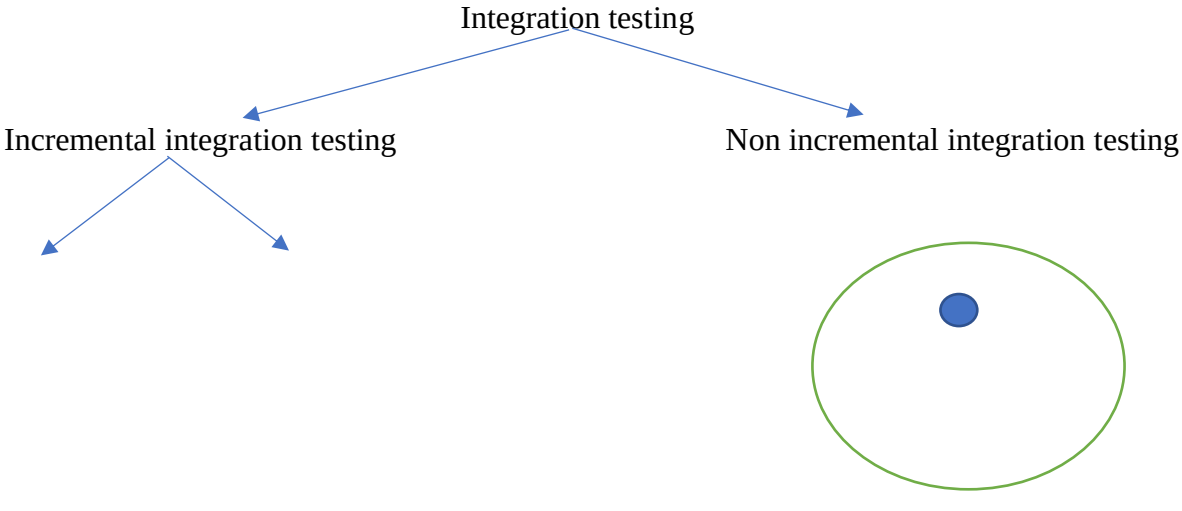
INTEGRATION TESTING

Testing the data flow between two or more dependent modules is called integration testing.

Ex: In gmail, if we send a mail, the sent mail should be displayed in sent items.

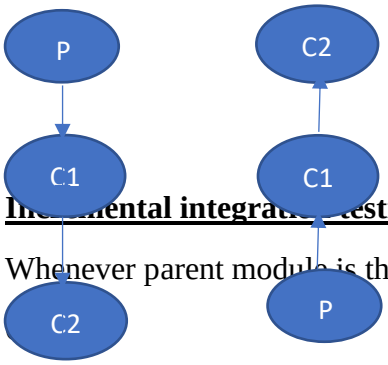
Checking whether it there in sent items or not is a integration testing.

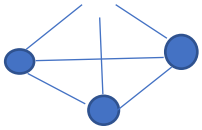
Types of integration testing



MANUAL TESTING

Top Down Bottom Up





Incremental integration testing:

Whenever parent module is there, child module is added and we do integration testing.

Whenever child module is there, parent module is added and we do integration testing between these modules.

Non incremental integration testing:

Whenever we cannot find which is a child and which is a parent module, even then we do integration testing between them, this is called non incremental integration testing.

STUBS

DRIVERS

Stubs and drivers

Dummy Modules

STUB

Acts like a child module when CHILD MODULE IS NOT PRESENT

DRIVER

acts like a PARENT module when PARENT MODULE IS NOT PRESENT

UNIT REGRESSION TESTING

REGIONAL REGRESSION TESTING

Acceptance Testing

Cake – cool cake, chocolate, 10 kgs,

MANUAL TESTING

Happy birthday to Nikki

whole

hole

A - 1 year PARENT

B - 1 year - Data Base - STUB - CHILD

B – 1 year CHILD

A – 1 year – DRIVER – PARENT

Ex : Machine Critical Applications(SPACE, SUBMARINE, AIRFORCE, NAVY, ROCKETS, SATELLITES, SCANNING)

What is STUBS and DRIVERS ?

OrOne module is ready and another module is not ready, how can we do integration testing ?

Stubs:

It is a dummy module which acts like a child module.

Whenever there is no child module available, stubs will receive the data and acknowledges the data.

Drivers:

It is a dummy module which acts like a parent module.

Drivers will send or trigger data and send data to child module.

Stubs and drivers are mainly used in critical projects related to space, medical, Submarines, defence etc.