Testing Guide For Fresh SQA Engineers

What is MANUAL TESTING?

MANUAL TESTING is a process, in which all the phases of STLC (SOFTWARE TESTING LIFE CYCLE) like Test planning, Test development, Test execution, Result analysis, Bug tracking and Reporting are accomplished successfully and manually with Human efforts.

Why did U choose Testing?

➢ Scope of getting jobs is very very high.
➢ No need to depend upon any Technologies.
➢ Testing there for ever.
➢ One can be consistent throughout their life.

Who can do Testing?

Any graduate who is creative can do.

What exactly we want to get a job?

Stuff+communications+confidence+dynamism.

Why the Test engineers exclusively required in the software companies?

➢ One cannot perform two tasks efficiently at a time.
➢ Sentimental attachment.

Project: Project is something that is developed based on particular customer's requirements and for their usage only.

Product: Product is something that is developed based on the company specifications and used by multiple customers.

Note: The product based company will first have general survey in the market. Gather's clear requirements from different customers, and based on common requirements of so many customer`s. They will decide the specifications (Requirements).

Quality:

Classical Definition of Quality: Quality is defined as justification of all the requirements of a customer in a product.

Note: Quality is not defined in the product. It is defined in the customer`s mind.

Latest Definition of Quality:

Quality is defined as not only the justification of all the requirements but also the presence of the value (User friendliness).

Defect: Defect is defined as a deviation from the Requirements.

Testing: Testing is a process in which defects are identified, isolated, subjected for rectification and ensure that the product is defect free, in order to produce the quality product and hence the customer satisfaction. (or) Verification & Validation of software is called Testing.

Bidding: The project is defined as a request for proposal, estimation and signing off.

Kick off meeting: It is an initial meeting conducted in the software company, soon after the project is signed off, in order to discuss the overview of the project and also to select a project manager.

Usually Project managers, Technical managers, Quality managers, High level management, Test leads, Development leads and sometimes customer representatives will be involved in this meeting.

Note: Apart from this meeting any kind of startup meeting during the process can be considered as 'Kick off Meeting.

Project Initiation Note (PIN): It is a mail prepared by the project manager and sent to CEO of the software company as well as for all the core team members in order to intimate them, that they are about to start the actual project activities.

Software Quality:

Technical:

- Meeting Customer Requirements
- Meeting Customer Expectations (User friendly, Performance, Privacy)

Non-Technical:

- Cost of Product
- Time to Market

Software Quality Assurance: To monitor and measure the strength of development process, Organization follows SQA concepts.

Software Project: Software related problems solved by software engineers through a software engineering process.

Software Development Life Cycle (SDLC):

There are six phases in software development life cycle

1. Initial (or) Requirements phase
2. Analysis phase
3. Design phase
4. Coding phase
5. Testing phase
6. Delivery and Maintenance phase

I. Initial (or) Requirement phase:

Tasks: Interacting with customer and gathering the requirements.

Roles: Business analyst –BA, Engagement manager - EM

Process:

➤ First of all business analyst will take an appointment from the customer, collect the template from the company and meet the customer on appointed date, gather the requirements with the support of that template and comes back to the company with the requirement document.

➤ The engagement manager go through the requirements, if he find any extra requirements then he will deal with excess cost of the project.

➤ If at all he finds any confused requirements, then he will ask the concern team to build a prototype, demonstrate that prototype to the customer, gather's the clear requirements and finally hand over the required documents to the BA

Proof: The proof document of Initial phase is Requirement's Document (RD).

These documents are called different Names in different companies:

FRS: Functional Requirement Specifications

CRS: Customer (or) Client Requirement Specifications URS: User Requirement Specifications

BRS: Business Requirement Specifications BDD: Business Design Document

BD: Business Document

Some companies will maintain two documents. One is for overall business flow information and second is detailed functional information, but some companies will maintain both this information's in a single document.

Template: Template is pre-defined format, which is used for preparing a document very easily and perfectly.

Prototype: Prototype is a roughly and rapidly developed model which is used for demonstrating to the client in order to gather clear requirements and also to build the confidence of a customer.

Ex: Power Point slide show.

II. ANALYSIS PHASE:

Tasks:

- ➤ Feasibility study
- ➤ Tentative planning
- ➤ Technology selection and Environment confirmation
- ➤ Requirement analysis

Roles: System Analyst – SA, Project Manager – PM, Technical Manager - TM

Process:

a. Feasibility study:

It is detailed study conducted on the requirement documents, in order to confirm whether the given requirements are possible within the given budget, time and available resources or not.

b. Tentative planning:

In this section resource planning and time planning will be temporarily done.

c. Technology selection & Environment confirmation:

The list of all technologies required for accomplishing the project successfully will be analyzed, the environment suitable for that project will be selected and mentioned here in this section.

d. Requirement analysis:

The list of all the requirements that are required by the company to accomplish this project successfully will be analyzed and listed out clearly in this section.

Note: Requirements may be Human Resources, Software's, and Hardware's.

Proof: The proof document of the Analysis phase is System Requirements Specifications (SRS).

III. DESIGN PHASE:

Tasks:

➢ High level designing
➢ Low level designing

Roles:

➢ High level design is done by chief Architect(CA)
➢ Low level design is done by Technical Lead(TL)

Process:

➢ The chief architect will divide the whole project in to modules by drawing some diagrams using unified modeling language (UML).
➢ The Team lead will divide the modules into sub modules by drawing some diagrams using the same UML.
➢ In this phase they will also design GUI part of the application, as well as PSEUDO CODE also developed

Proof: The proof document of this phase is Technical Design Document (TDD).

LLD: Ex: DFD-Data Flow Diagram, E-R Diagram, Class Diagram, Object Diagram.

PSEUDO CODE: PSEUDO Code is a set of English instructions, which will make the developer's more comfortable, while developing the actual source code.

IV. CODING PHASE (WHITE BOX TESTING):

Tasks: Programming (or) Coding

Roles: Programmers (or) Developers

Process: The developers will develop the actual source code by following the coding standards and also with the support of Technical Design Document.

Example's for Coding standards:

➢ Proper Indentation (left margin)
➢ Color codling's
➢ Proper Commenting

Proof: Proof document of the Coding phase is Source Code Document (SCD).

V TESTING PHASE (BLACK BOX TESTING):

Tasks: Testing

Roles: Test Engineer's.

Process:

➢ The Testing department will receive the requirement document and the test engineers will start understanding the requirements.
➢ While understanding the requirements, if they get any doubts they will list out all the doubts in Requirement clarification Note and sent it to the author of the requirement document and wait for the clarification.
➢ Once the clarification is given and after understanding all the requirements clearly they will take the test case template and write the Test cases.
➢ Once the first build is released they will execute the test cases.

- ➤ If at all any defects are found they will list out all the defects in the defects profile and send it to the development department and will wait for the next build.
- ➤ Once the next build is released they will re execute the required test cases.
- ➤ If at all any more defects are found they will update the defect profile. Send it to development department and will wait for the next build.
- ➤ This Process continuous till the product is defect free.

Proof: The proof of Testing phase is Quality product.

BUILD: A finally intigrated all modules set .EXE form is called Build.

TEST CASES: Implementing the creative Ideas of the Test Engineer on the application for testing, with the help of requirement document is known as TEST CASES.

VI. DELIVERY AND MAINTENANCE PHASE:

Delivery:

Tasks: Hand over the Application to the client

Roles: Deployment engineers (or) Installation engineers.

Process: The Deployment engineers will go to the customers place, install the application in to the customers environment and handover the original software to the client.

Proof: The final official agreement made between the customer and company is proof document for Delivery.

Maintenance:

Once the application is delivered. The customer will start using it, while using if at all they face any problems then that particular problem will be created as tasks. Based on the tasks corresponding roles will be appointed. They will define the process and solves the problem .This process is known as Normal Maintenance. But some customers may request for continuous Maintenance, in that case a team of members will be continuously working in the client site in order to take care of their Software.

Where exactly testing comes in to the picture?

Which sort of testing we are expecting?

How many sort's of testing are there?

There are two sorts of Testing.

1. **Unconventional Testing**
2. **Conventional Testing**

Unconventional testing: It is sort of Testing in which the Quality Assurance people will check each and every out come document is according to the company standards or not right from the Initial phase to the end.

Conventional testing: It is sort of Testing in which one will check the developed applications or its related parts are working according to the exceptions or not, from the Coding phase to the end. Usually Quality Control people will do Conventional testing.
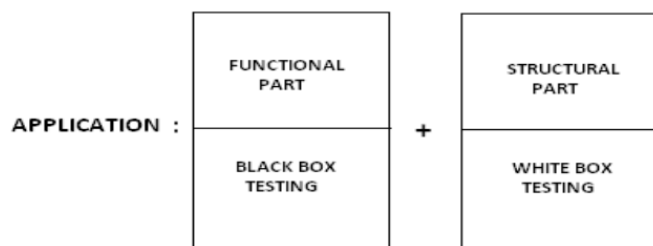
Testing methodology (Testing Techniques):

Basically there are 2 methods of Testing:

1. **Black Box Testing**
2. **White Box Testing**

Note: One more derived method is Grey Box Testing

BLACK BOX TESTING:

➤ It is method of testing in which one will perform testing only on the function part of the application without having the knowledge of structural part.
➤ Usually the Black Box Test engineers will perform.

WHITE BOX (or) GLASS BOX (or) CLEAR BOX TESTING:

➤ It is a method of testing in which one will perform testing on the structural part of the application.
➤ Usually the White Box Tester's or Developer's will perform.

GREY BOX TESTING:

It is method of testing in which one will perform testing on both the functional part as well as structural part of on application.

➤ Usually the Test engineer's who has the knowledge of structural part will perform.

LEVELS OF TESTING:

There are 5 levels of Testing:

1. Unit level testing
2. Module level testing
3. Integration level testing
4. System level testing
5. User acceptance level testing

1. UNIT LEVEL TESTING:

Unit: Unit is a smallest part of an application (Program).

In this stage the white box testers will test each and every program and combinations of programs in order to confirm whether they are working according to the expectations or not. They test the structural part of a module.

2. MODULE LEVEL TESTING:

Module: Module is defined as a group of related features to perform a major task in an application.

In this stage the Black Box test engineer's will test the functional part of a module.

3. INTEGRATION LEVEL TESTING:

In this stage the developers will develop interfaces (Linking Prg's), in order to integrate the modules. The White Box testers will test whether the interfaces are working fine or not. Developers will integrate the modules by following any one of the following approach:

TOP-DOWN APPROACH: In this approach parent modules will be develop first and then related child modules will be integrated.
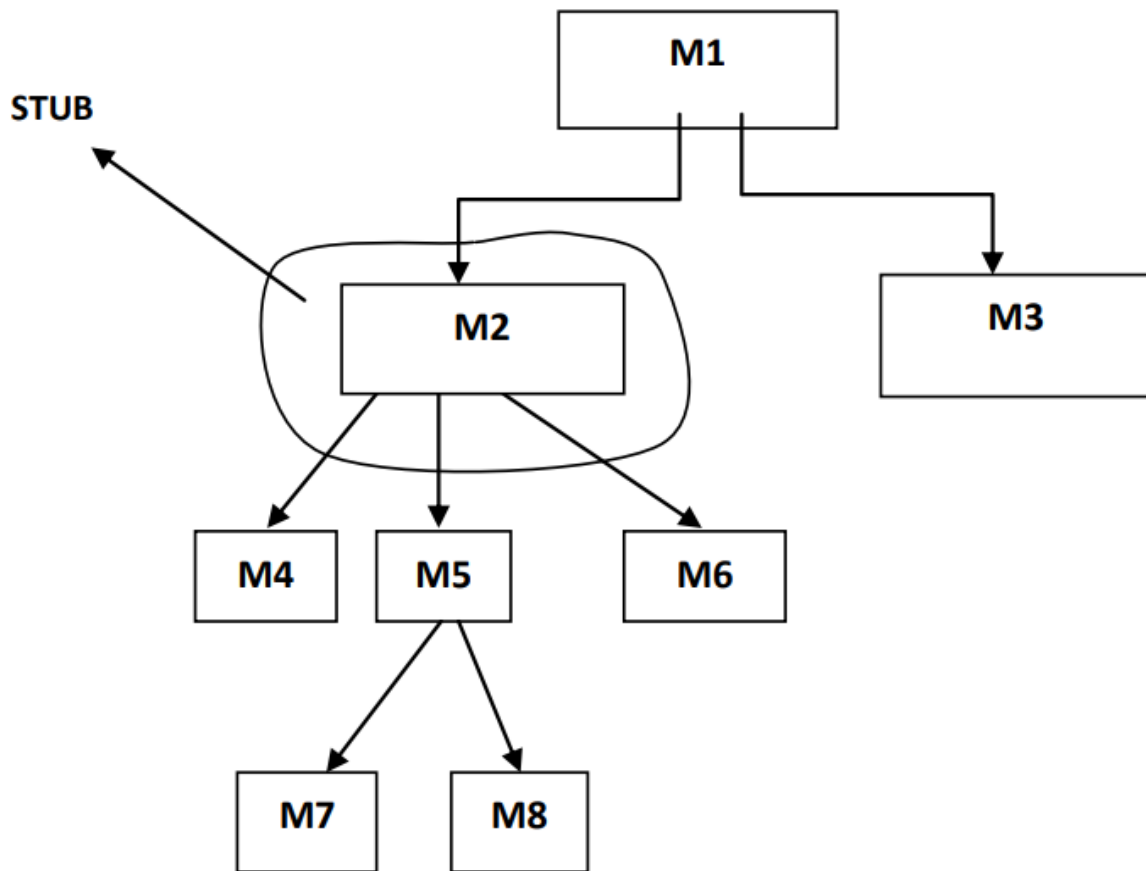
STUB: While integrating the modules in the Top-Down approach, if at all any mandatory module is missing then that module is replaced with a temporary program known as STUB.

BOTTOM-UP APPROACH: In this approach child modules will be developed first and then integrated back to the corresponding parent modules.

DRIVER

M1

M2

M3

M4

M5

M6
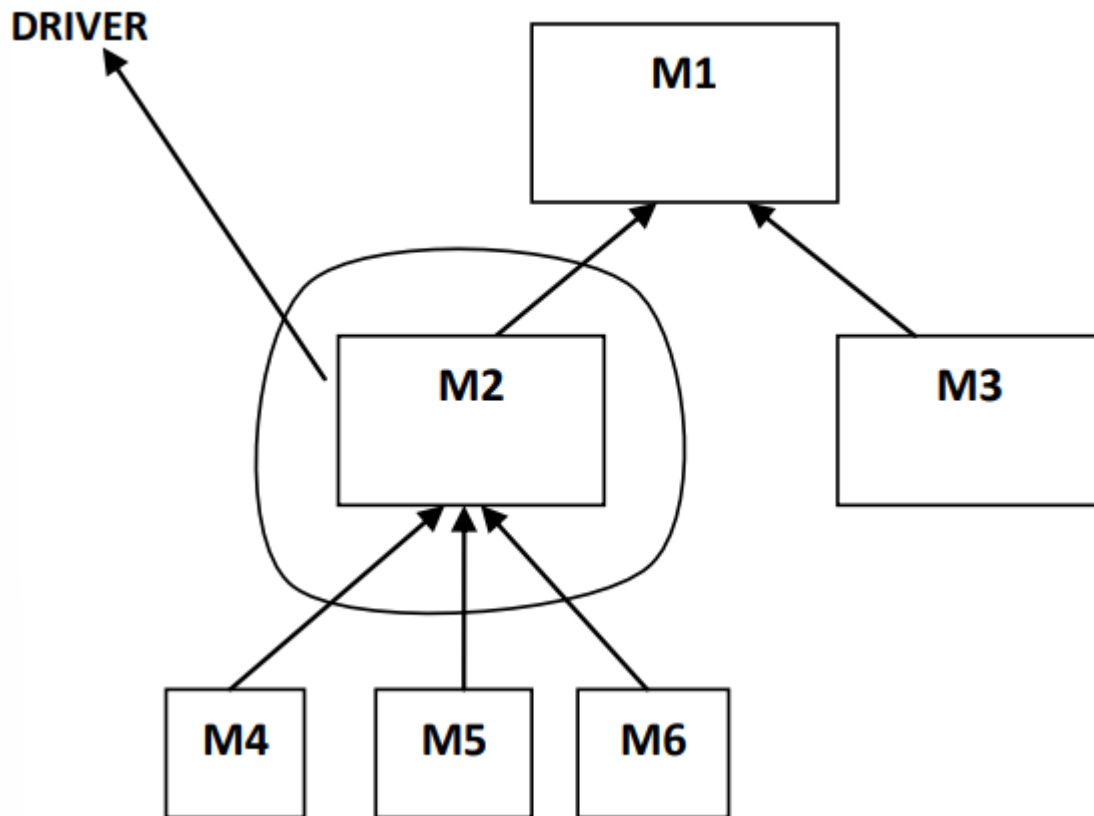
DRIVER: While integrating the modules in Bottom-Up approach, if at all any mandatory module is missing then that module is replaced with a temporary program known as DRIVER.

SANDWICH (OR) HYBRID APPROACH: This is a mixture of both Top-Down and Bottom-Up approach.
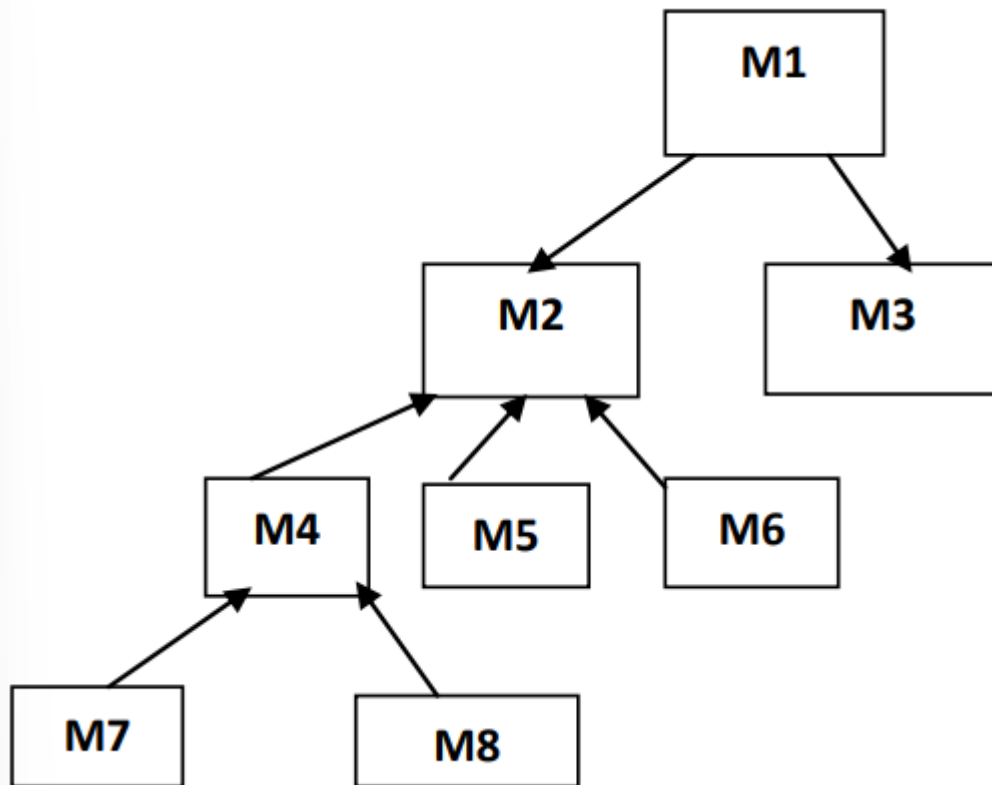
BIG BANG APPROACH: In this approach one will wait till the modules are developed and will integrate them at a time finally.
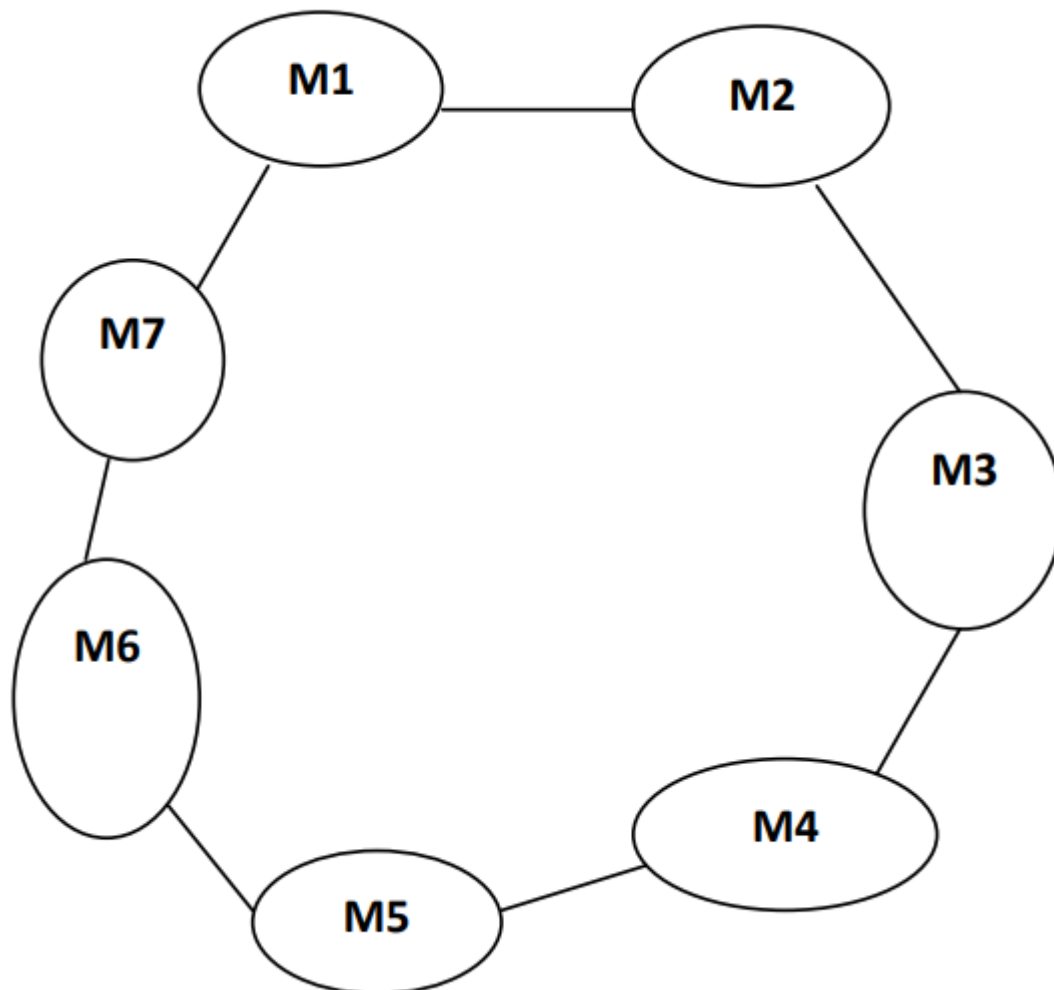
4. SYSTEM LEVEL TESTING:

In this level the Black Box test engineers will conduct so many types of testing like load testing, performance testing, stress testing, compatibility testing, system integration testing etc.

These type of Testings are also conducted:

1. Usability Testing

2. Functionality Testing

3.   Performance Testing

4. Security Testing

During Usability Testing, testing team validates User Friendliness of screens.

During Functionality Testing, testing team validates Correctness of Customer Requirements.

During Performance Testing, testing team estimates Speed of Processing.

During Security Testing, testing team validates Privacy to User Operations.

SYSTEM INTEGRATION TESTING: It is a type of testing in which one will perform some actions at one module and check for the reflections in all the related areas.

 Ex:
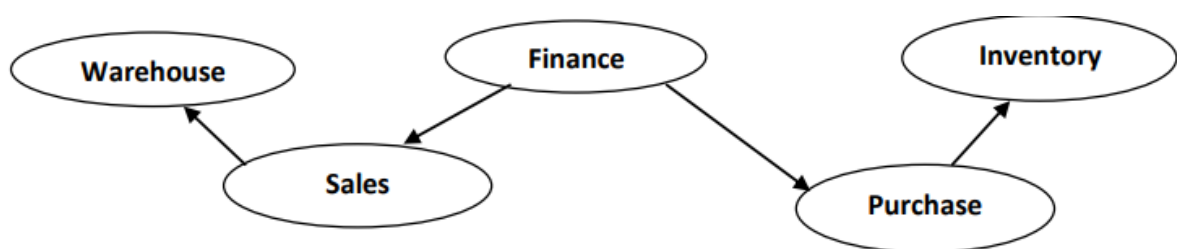
5. USER ACCEPTANCE TESTING:

In this stage the Black Box test engineers will test once again the user desired areas in the presence of the user in order to make him to accept the application.

ENVIRONMENT: Environment is defined as group of hardware components with some basic software's which can hold the business logic, presentation logic and database logic.

(Or)

Environment is a combination of Presentation layer, Business layer, and Database layer which can hold presentation logic, business logic and database logic.

TYPES OF ENVIRONMENTS:

There are 4 types of environments:

1.   STAND-ALONE ENVIRONMENT (OR) ONE-TIER ARCHITECTURE.
2.   CLIENT-SERVER ENVIRONMENT (OR) TWO-TIER ARCHITECTURE.
3.   WEB ENVIRONMENT (OR) THREE-TIER ARCHITECTURE.
4.   DISTRIBUTED ENVIRONMENT (OR) N-TIER ARCHITECTURE.

1. STAND-ALONE ARCHITECTURE:

In this environment all the three layers that is presentation layer, business layer, database layer will be available in the single tier. When the application needs to be used by a single user at a time then one can suggest this environment.

Ex: Personal Computer.

PL
BL
DBL

2. CLIENT-SERVER ENVIRONMENT:

In this environment two tiers will be there. One is for clients, other is for servers. Presentation layer and business layer will be available in each and every client; database layer will be available in the server.

Whenever the application need to be used by multiple users sharing the common data in a single premises and wants to access the application very fastly and there is no problem with security. Then one can suggest client- server environment.

Ex: LAN.

```
┌───────┐        ┌───────┐
│  PL   │        │       │
├───────┤   +    │  DBL  │
│  BL   │        │       │
└───────┘        └───────┘
```
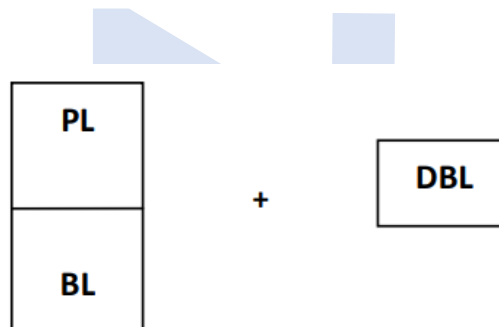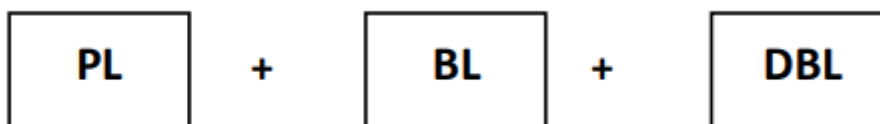
3. WEB ENVIRONMENT:

This environment contains three tiers. One is for clients, middle one is for application server and the last one is for database servers.

Presentation layer will be available in client, Business layer will be available in the application server and Database layer will be available in the database servers.

Whenever the application needs to be used all over the world by limited number of people then this environment can be suggested.

Ex: WAN.

```
┌───────┐     ┌───────┐     ┌───────┐
│  PL   │  +  │  BL   │  +  │  DBL  │
└───────┘     └───────┘     └───────┘
```

4. DISTRIBUTED ENVIRONMENT:

This environment is similar to web environment but number of application servers are introduced in individual tiers. In order to distribute the business logic, so that load will be distributed and performance will be increased.

Whenever the application needs to be used all over the world by huge number of people then this environment can be suggested.

Ex: yahoo.co.in, yahoo.co.uk, yahoo.co.us....etc.

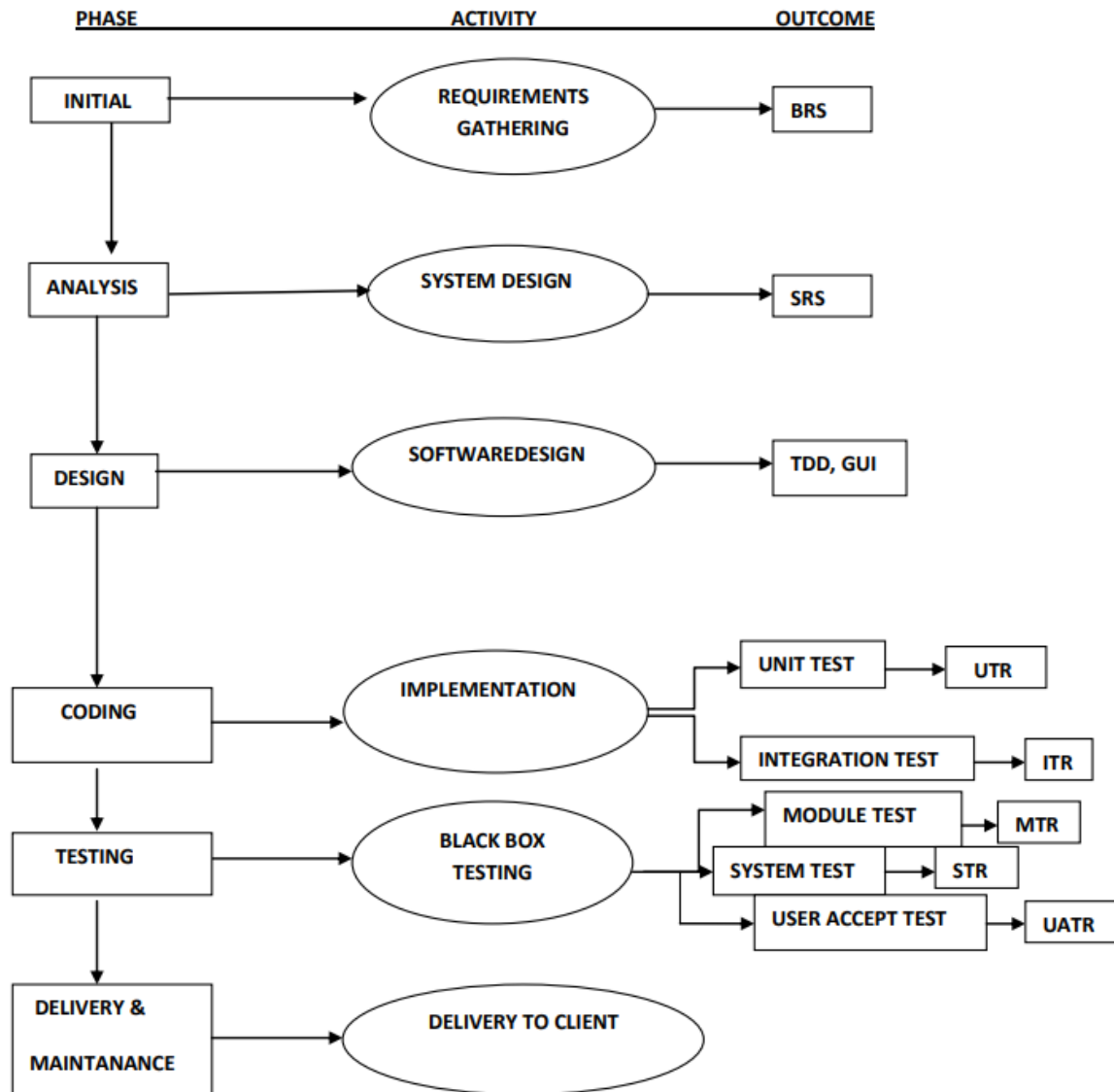| PL		BL	BL	BL			DBL
	+	AS	AS	AS	+		
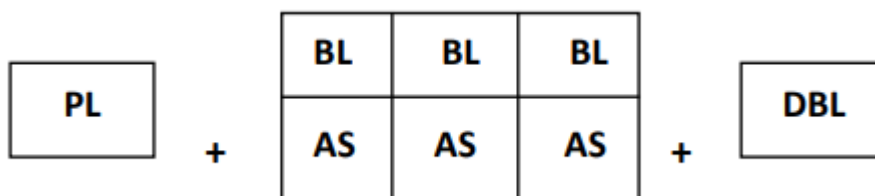
AS – APPLICATION SERVER

BL – BUSSINESS LOGIC

DATABASE: It is a base (or) a place where on can store and retrieve the data

SOFTWARE PROCESS DEVELOPMENT MODELS:

1. WATER FALL MODEL

2. PROTOTYPE MODEL

3. EVOLUTIONARY MODEL

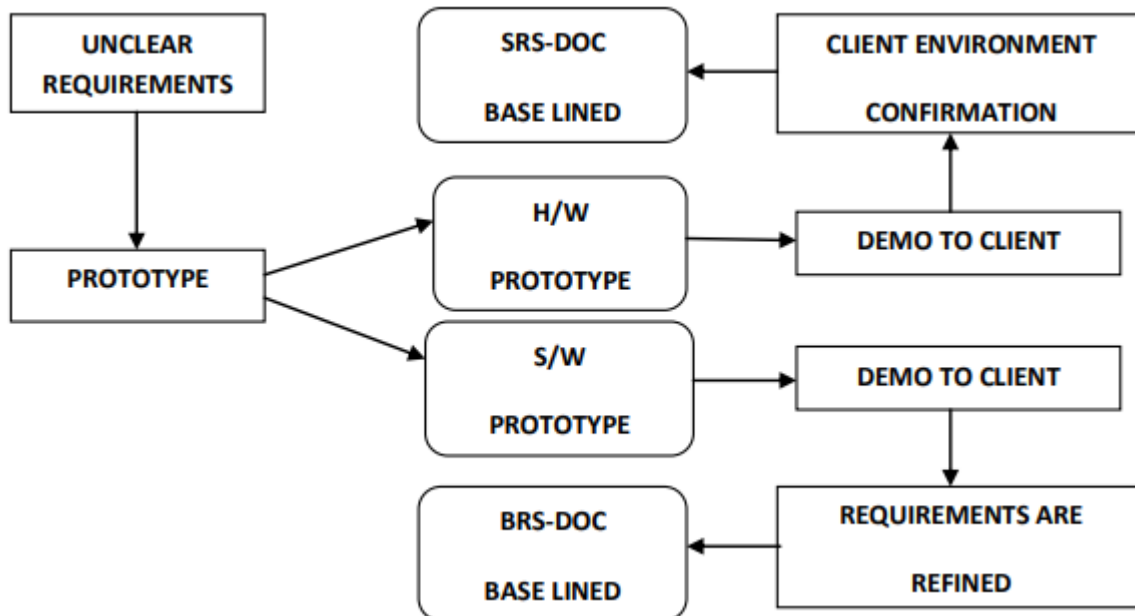4. SPIRAL MODEL

5. FISH MODEL

6. V-MODEL

1. WATERFALL MODEL:

ADVANTAGES:

1. It is a simple model.
2. Project monitoring and maintenance is very easy.

DISADVANTAGES:

Can't accept the new requirements in the middle of the process.

2. PROTOTYPE MODEL:

ADVANTAGES:

Whenever the customer's are not clear with their requirements then this is the best suitable model.

DISADVANTAGES:

a. It is not a full fledged process development model.
b. Prototype need to be build on companies cost.
c. Slightly time consuming model.
d. User may limit his requirements by sticking to the PROTOTYPE.
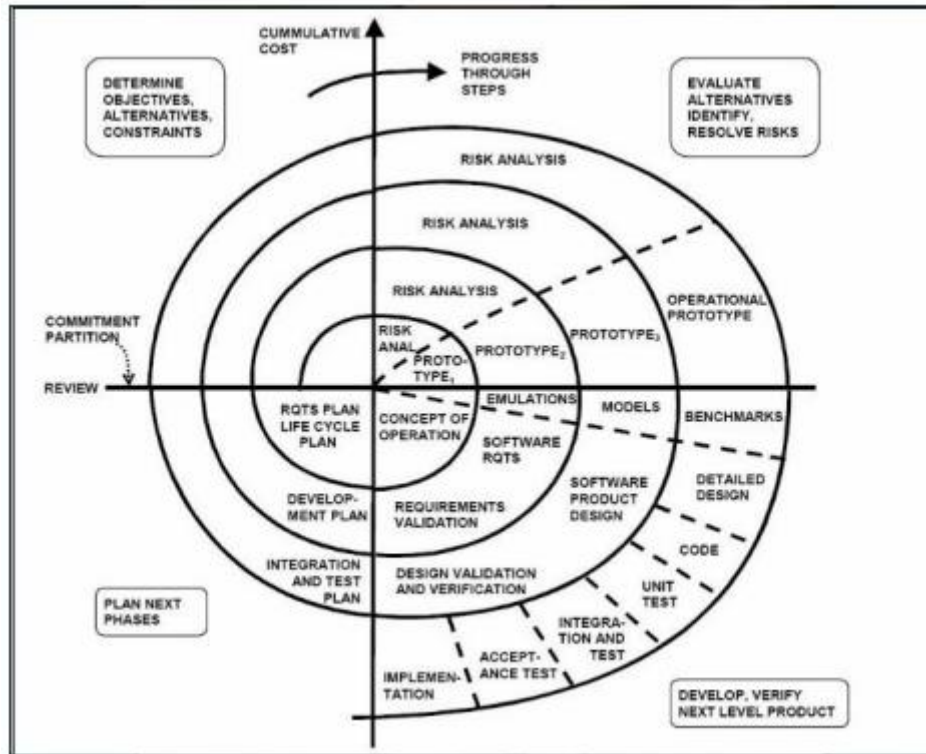
3. ENVIRONMENT MODEL:
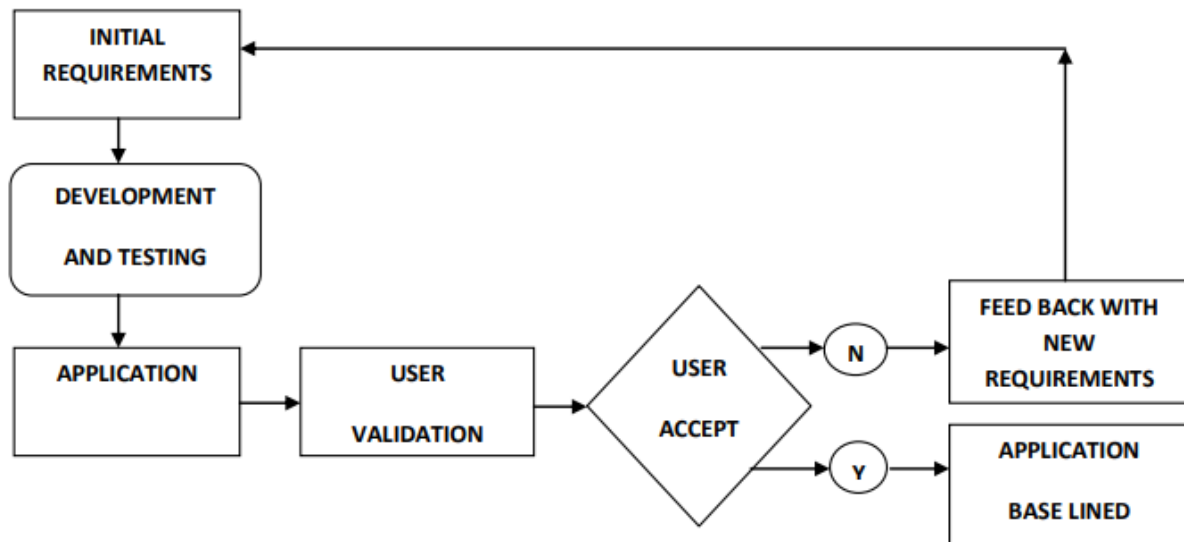
ADVANTAGES:

Whenever the customers are evolving the requirements then this is the best suitable model. Adding the new requirements after some period of time.

DISADVANTAGES:

1. Project monitoring and maintenance is difficult.
2. Can't define the deadlines properly.

4. SPIRAL MODEL:

Ex: Risk based scientific projects, Satellite projects.

ADVANTAGES:

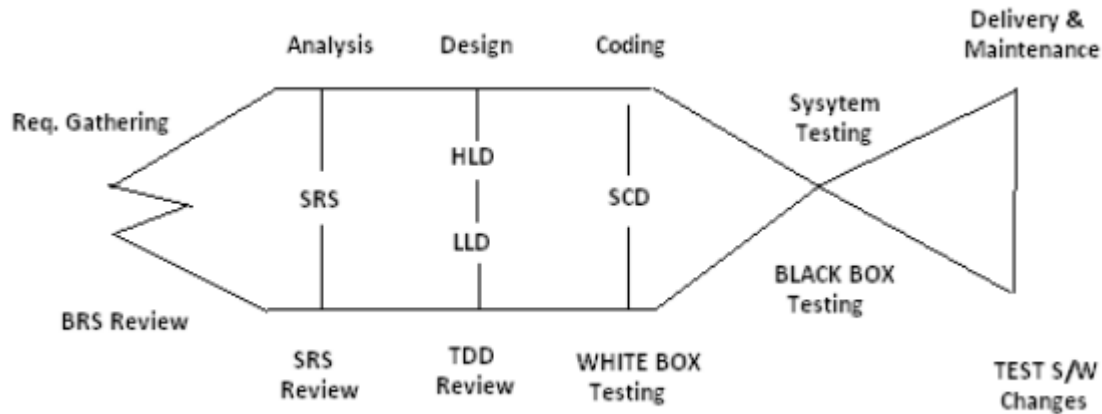Whenever the project is highly risk based this is the best suitable model.

DISADVANTAGES:

1. Time consuming model.
2. Costly model.
3. Risk route cause analysis is not an easy task.

NOTE: Cycles depend upon Risk involved in the project and size of the project, Every cycle has 4 phases, except the last phase.

5. FISH MODEL:

ADVANTAGES:

As both verification and validation are implemented the outcome will be a quality product.

DISADVANTAGES:

1. Time consuming model.

2. Costly model.

VERIFICATION:

Verification is a process of checking each and every role in the organization in order to confirm weather they are working according to the company's process guidelines or not.

VALIDATION:

Validation is a process of checking, conducted on the developed product or its related parts in order to confirm weather they are working according to the expectations or not.
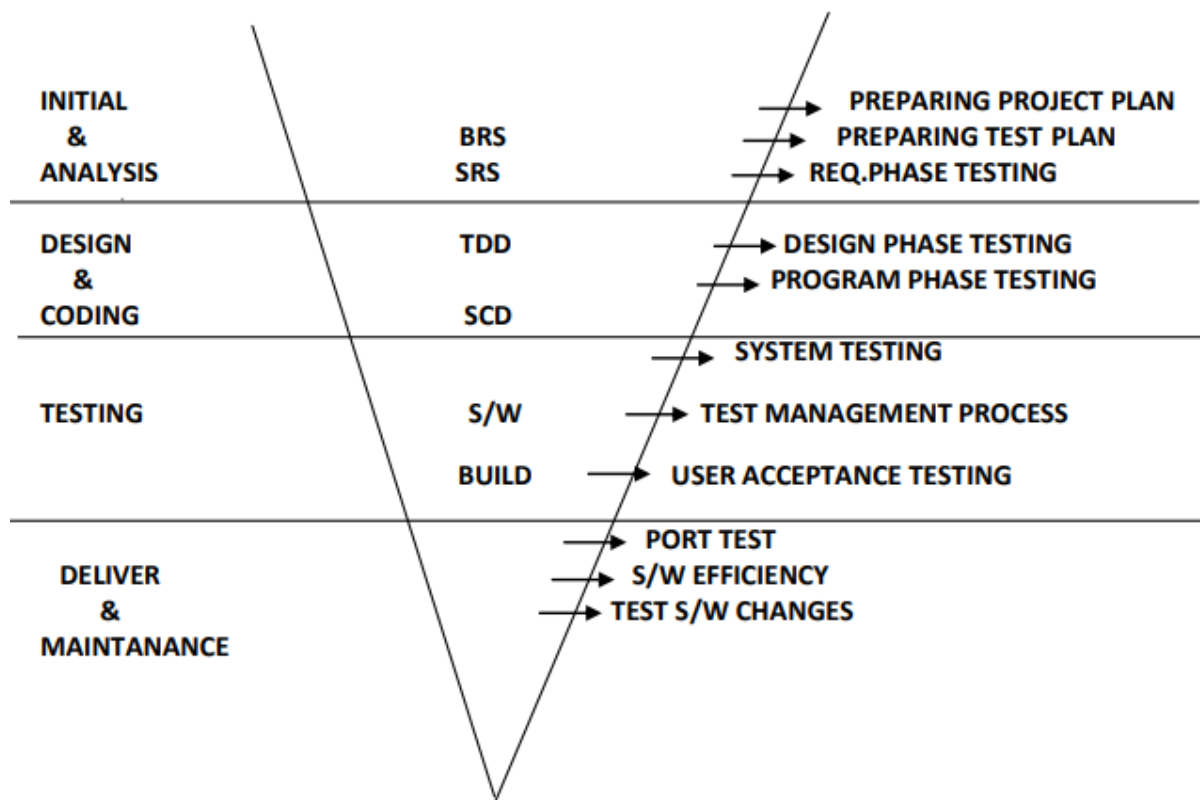
VERIFICATION:

Quality Assurance people (Reviews, Inspections, Audits, Walk through).

VALIDATION:

Quality Control people (Testing).

6. V-MODEL:

"V" Represents Validation and Verification. DRE=0-1(Range).

DRE=A/A+B.

DRE = Defect Removal Efficiency.

A = Defects found by the Testing Team. .

B = Defects raised by the Customer. DRE=80/80+20=80/100=4/5=0.8 (Good Software).

GOOD SOFTWARE :0.7-1 POOR SOFTWARE :Below 0.7

ADVANTAGES:

As verification, validation, test management process is maintained. The outcome will be a quality product.

DISADVANTAGES:

1. Time consuming model.
2. Costly model.

AGILE MODEL: Before development of the application, where testers write the test cases and gives to the development team, so that it can be easy for developers to develop defect free Programs.

TERMINOLOGY:

IF A DEVELOPER FINDS A MISTAKE IN CODE, WHILE DEVELOPING OF AN APPLICATION IT IS CALLED AN ERROR. IF A TESTER FINDS A MISTAKE IN A BUILD, WHILE TESTING IT IS CALLED A DEFECT (or) ISSUE.

IF A DEFECT IS ACCEPTED BY DEVELOPER TO SOLVE IT IS CALLED A BUG.

IF A CUSTOMER FINDS ANY MISTAKES, WHILE USING THE APPLICATION IT IS CALLED A MISTAKE (or) FAULT (or) FAILURE.

A mistake in code is called ERROR. Due to errors in coding, test engineers are getting mismatches in application called DEFECTS. If defect accepted by development to solve called BUG.

TYPES OF TESTINGS

1. BUILD ACCEPTANCE TEST/BUILD VERIFICATION TEST/SANITY TESTING:

It is type of testing In which one will perform overall testing on the released build, in order to confirm whether it is proper for conducting detailed testing or not.

Usually during this type of testing they check the following:

Whether the build is properly installed or not

Whether one can navigate to all the pages of application or not

Whether all the important functionality are available or not

Whether all the required connections are properly established or not

Some companies even called this as SMOKE TESTING, but some companies will say that before releasing the build to the testing department, the developers will check whether the build is proper or not that is known as SMOKE TESTING, and once the build is released what ever the test engineers is checking is known as BAT or BVT or SAINITY TESTING (BAT: Build Acceptance Test, BVT: Build Verification Test).

2. REGRESSION TESTING:

It is type of testing in which one will perform testing on the already tested functionality again and again.

Usually we do this in 2 scenarios:

➢ When ever the tester's identify the defects raise to the developers, next build is released then the test engineers will check defect functionality as well as the related functionality once again.

> When ever some new features are added, the next build is released to testing department team. Then the test engineers will check all the related features of those new features once again this is known as Regression Testing

Note: Testing new features for the first time is known as New testing it not the Regression testing. Note: Regression testing starts from 2nd build and continuous up to last build.

3. RETESTING:

It is type of testing in which one will perform testing on the same funcatnality again and again with deferent sets of values, in order to confirm whether it is working fine or not.

Note: Retesting starts from 1st build and continuous up to last build. Note: During Regression testing also Retesting will be conducted.

4. ALPHA TESTING:

It is type of user acceptance testing conducted in the software company by the test engineers just before delivering the application to the client.

5. BETA TESTING:

It is also a type of user acceptance testing conducted in the client's place either by the end users or third party experts, just before actual implementation of the application.

6. STATIC TESTING (Look and Feel Testing):

It is a type of testing in which one will perform testing on the application or its related factors without doing any actions.

EX: GUI Testing, Document Testing, Code Reviews etc...,

7. DYNAMIC TESTING:

It is a type of testing in which one will perform testing on the application or its related factors by doing some actions.

Ex: Functional Testing.

8. INSTALLATION TESTING:

It is a type of testing in which one will install the application in to the environment, following the guide lines provided in the deployment document(Installation Document), in order to confirm whether these guide lines are really suitable for installing the application into the environment or not.

9. PORT TESTING:

It is a type of testing in which one will install the application in to the original client's environment and check weather it is compatible with that environment or not.

10. USABILITY TESTING:

It is a type of testing in which one will test the user friendliness of the application.

11. COMPATABILITY TESTING:

It is a type of testing in which one will install the application into multiple environments, prepared with different configurations, in order to check whether the application is suitable with those environments or not.

Usually these types of testing will focused in product based companies.

12. MONKEY TESTING:

It is a type of testing in which one will perform abnormal actions on the application. Intentionally, in order to check the stability of the application.

13. EXPLORATORY TESTING:

EXPLORING: Having basic knowledge of about some concept, doing some thing and knowing more about the same concept is known as Exploring.

It is a type of testing in which the domain experts will perform testing on the application with out having the knowledge of requirements, just by parallel exploring the functionality.

14. END TO END TESTING:

It is a type of testing in which one will perform testing on the end to end scenarios of the application.

EX: Login---> Balance Enquiry ---> Withdraw ----> Balance Enquiry ---> Logout.

15. SECURITY TESTING:

It is a type of testing in which one will check whether the application is properly protected or not. To do the same the BLACK BOX TEST Engineers will perform the following types of Testing:

1. AUTHENTICATION TESTING: In this type of testing one will enter different combination of user names and passwords and check whether only the authorized people are able to access application or not.

2. DIRECT URL TESTING: In this type of testing one will directly enter the URL's of secured pages, in order to check whether the secured pages are directly access or not with out login to the application.

3. FIRE WALL LEAKEGE TESTING(or) USER PRIVILLAGES TESTING: It is a type of testing in which one will enter in to the application as one level of user and will try to access

beyond the user limits, in order to check whether the fire walls are working properly or not.

16. MUTATION TESTING:

It is a type of testing in which one will perform testing on the application are its related factors by doing some changes to them.

17. SOAK TESTING/REALIABILITY TESTING:

It is a type of testing in which one will use the application continuously for a long period of time, in order to check the stability of the application.

18. ADHOC TESTING:

It is a type of testing in which one will perform testing in their own style after understanding the requirements clearly.

Note: Usually in the final stages of the project, This type of Testing can be encouraged.

19. INPUT DOMAIN TESTING:

It is a part of Functionality Testing. Test engineers are maintaining special structures to define size and type of every input object.

20. INTER SYSTEM TESTING:

It is also known as end to end testing. During this test, testing team validates whether our application build co-existence with other existing software's are not?

21. PARALLEL TESTING:

It is also known as comparative testing and applicable to software products only. During this test, testing team compare our application build with competitors products in the market.

22. PERFORMANCE TESTING:

It is an advanced testing technique and expensive to apply because testing team have to create huge environment to conduct this testing. During this test, testing team validates Speed of Processing. During this performance testing, testing team conduct load testing and stress testing.

23. LOAD TESTING:

The execution of our application under customer expected configuration and customer expected load to estimate performance is called Load Testing.

24. STRESS TESTING:

The execution of our application under customer expected configuration and un interval load's to estimate performance is called stress testing.

25. STORAGE TESTING:

The execution of application under huge amounts of resources to estimate storage limitations is called storage Testing.

26. DATA VOLUME TESTING:

The execution of our application under customer expected configuration to estimate peak limits of data is called data volume testing.

27. BIG BANG TESTING/INFORMAL TESTING/SINGLE STAGE TESTING:

A testing team conducts single stage testing, after completion of entire system development instead of multiple stages.

28. INCREMENTAL TESTING/FORMAL TESTING:

A multiple stages of testing process from unit level to system level is called incremental testing. It is also known as formal testing.

SOFTWARE TESTING LIFE CYCLE (STLC):

STLC contains 6 phases:

1. Test Planning.
2. Test Development.
3. Test Execution.
4. Result Analysis.
5. Bug Tracking.
6. Report.