Software testing

Software testing:

It is the process of verifying and validating weather the developed software is properly working or not and all the client requirements satisfied or not.

We can conduct the software testing in 2 ways:

1.Manual testing

2.Automation testing

1.Manual testing:

In this method we will follow only manual approaches to conduct the software testing.

- We don't use any tools or program

2.Automation testing:

In this method we use a tool or programing language to

conduct the software testing.

It is a program or collection of programs used to perform some activity in a computer system.

There are two types of software.

1.System software

2.Application software

System software will interact with the hardware components of a computer system. They will developed by using languages like C,C++, etc.

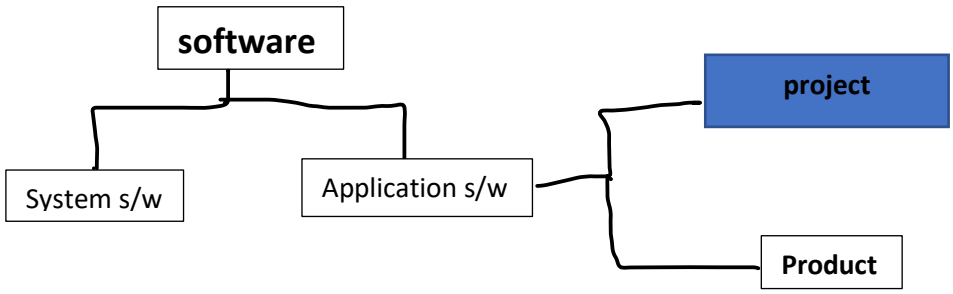
- They will be used by hardware engineers or system administrators

Ex: Display drives, sound drives.

Software testing

Application software are GUI tools [Graphical User Interface] developed for non-technical people called clients or end users. They will be implemented by using technologies like '.net ,java '.

- The application software are further divided into 2 categories. They are
  1.project
  2.product
- A **project** is a application software developed based on specific client requirements only that client can use software.
- A **product** is a application software developed based on market needs. Anybody can use the product.

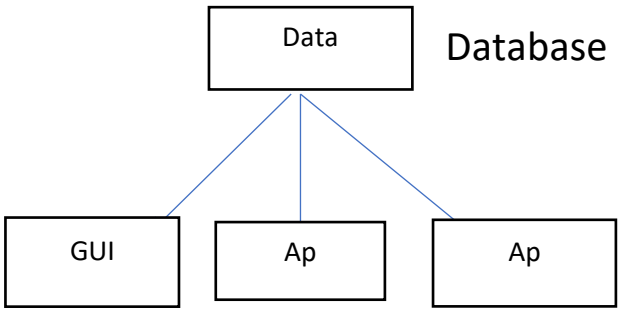


Ex: display drives, Ex: .net, java etc.

Sand drives etc.

Client server architecture:

Most of the real time projects will be implemented based on client server architecture model. In this model there will be a main computer called server which is connected with multiple other computers called clients. Server



Database s/w ex: SQL server, oracle server, my SQL server

Front end

Client 1 client 2 client 3 ex: .net, java.

Software testing

Project/product=1FE+1BE

Ex: .net proj=c++.net +SQL server

(or)

ASP .net+ SQL server

Java proj=java +oracle/my SQL

- The server is used to store the data in a organized manner by using data base software like SQL server ,oracle etc.
- In order to access and modify the data each client system contains a GUI tool called “Application program”.
- They will be developed by using technologies like .net, java etc.
- A project (or) a product is a combination of 1 front end application and one backend data base.
- At first the backend developer will develop the data base and then the frontend developer will design the user interface and also communication between front end and back end.
- Before developing the software to the client the test engineer has to test the quality of the software. it can be done by using either manual testing(or) automation testing (or) both.

Roles and responsibilities of a tester:

1. Understand all the clients requirements available in the BRS (Business Requirement Specification) document.
   Ex: An e commerce client wants to have a website with lot of requirements and one of the requirement is
   “Provide a search facility to search for different products”.
2. Identify the test scenarios
   Test scenarios: the feature (or) the functionality to be tested is called as test scenario.
   Ex: for the above requirement the test scenario is

Software testing

"Verifying the product search process".

3. Preparing the test cases:
   Test case: Different conditions we are using to test a functionality is called a test case.
   Ex: For the above scenario the following are the test cases:
   Test case 1: checking product search with valid product details.
   Test case 2: checking product search with invalid product details.
4. Executing the test cases
5. Identifying the defects.
6. Preparing the defect reports and send them to the development team.
7. Preparing test scripts based on the test cases for automation testing.
8. Performing "Re-testing' and "Regression" testing.

SDLC PROCESS:

SDLC stands for "Software Development Life Cycle".

Every IT company has to follow a standard procedure to implement the software called as "SDLC" process. Then only that software is consider as a valid software project (or) a product.

Software Bidding:

It is a proposal given by a client to a development company about a new software to be implemented.

Kick-off meeting:

The CEO of the development company will conduct the meeting called as kick-off meeting to select a person as project manager. He will be the decision maker for the entire software implementation in the SDLC process.

People involved in SDLC process:

Software testing

The following are the different people involved in the project implementation based on SDLC process.

- Client
- CEO
- Project Manager (PM)
- Business Analyst (BA)
- System Analyst (SA)
- Technical Architect (TA)
- Developer
- Tester
- CCB (Change Control Board) team/production support team/maintenance team.

SDLC Phases:

The following are the different phases in SDLC process. Each phase is implemented by a responsible person by taking some input and delivers some output.

1. Requirement gathering
2. System analysis
3. System design
4. Coding/ implementation
5. Testing
6. Release/ Deployment
7. Maintenance

1.Requirement gathering:

In this phase the **Business Analyst** will visit the client location, understand current process using by the client, communicates with the client and client related people to collect all the requirements .Finally all these requirements are summarized in a BRS document.

Responsible person = Business analyst

Software testing

Input = User requirements

Output = BRS (Business Requirement Specification) document.

(or)

CRS (Customer Requirement Specification) document.

(or)

URS (User Requirement Specification) document.

2. System Analysis:

In this phase the **System Analyst** will study the BRS document, understand client requirements and identifies technically what is required to satisfy the user requirements. All this information is summarized in a document called **FRS** (or) **SRS**.

This document contains budget allocation, resources required, features to be implemented software and hardware to be used.

To prepare this document the **SA** (System Analyst) will conduct a special kind of study called as **Feasibility** study on the BRS document.

The following are the different feasibility studies conducted by the System Analyst.

- **Economical feasibility study = To know the budget allocation**
- **Resource feasibility study = To know the resource allocation**
- **Operational feasibility study = To know the functionalities to be implemented.**

Responsible person = **System Analyst**

Input = **BRS document**

Output document = **FRS (Functional Requirement Specification)**

(or)

SRS (Software Requirement Specification) document

Software testing

3. System Design: The **Technical Architect (TA)** will implement this phase by designing how the backend database, frontend application program should be implemented.

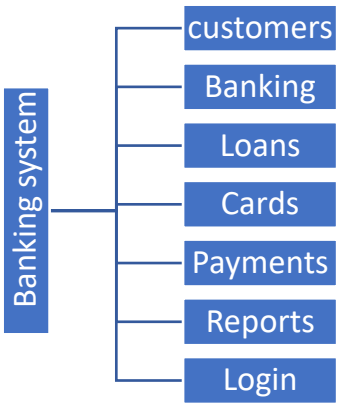
Based on the FRS document the TA will design **HLD** (**High Level Design)** and **LLD (Low Level Design)** documents.

The HLD document contains **main module** details and LLD document contains **internal details of each module**.

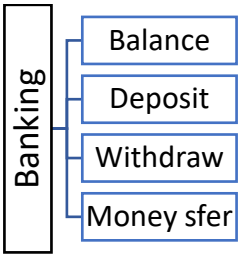
These documents are prepared based on the following diagrams:

- **ER** diagrams (**Entity Relationship) –** contains database design
- **DFD (Data Flow Diagrams)** – contains main and sub module design.
- **Flow charts** – contains execution flow of functionality.
- **Use case diagrams** – describes the classes and methods to be implemented.

EX: High level DFD of Banking system

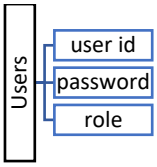


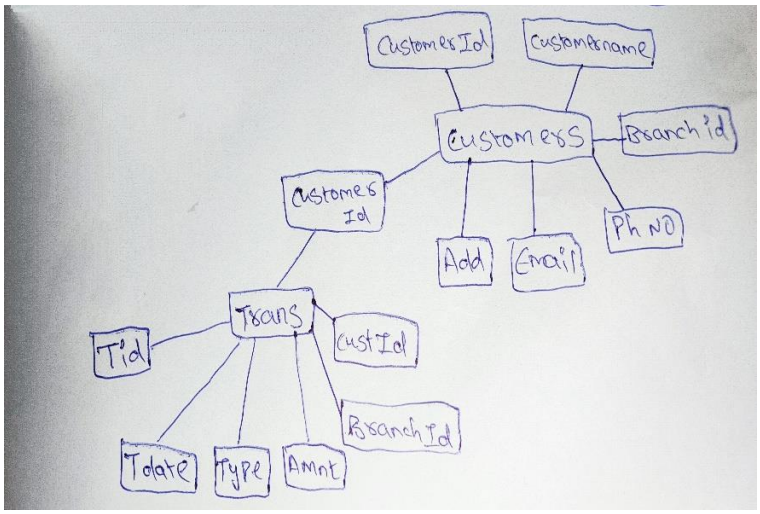
Ex: low level DFD of Banking module



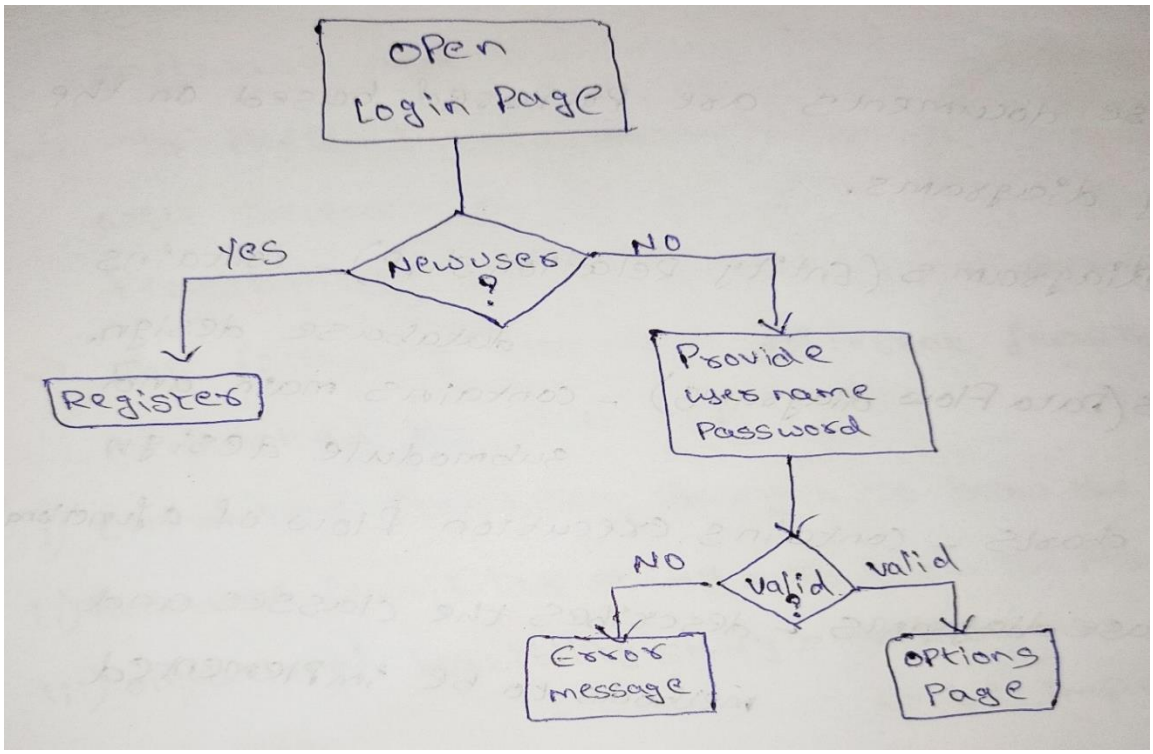
ER diagram:

Software testing





Flow chart of login process:



Software testing

Responsible person = Technical architect

Input = FRS / BRS document

Output = HLD (High level Design) document

LLD (low level Design) document

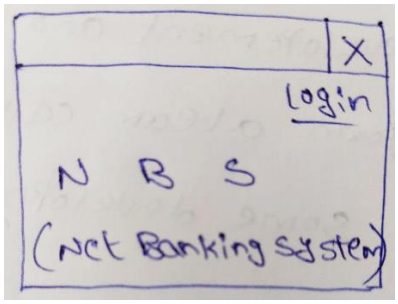
4. Coding/ Implementation:

In this phase the backend developer will implement the data base on the server system by using ER diagram.

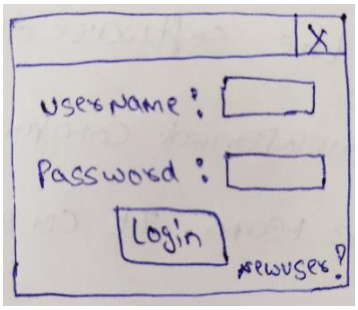
The frontend developer will use HLD and LLD documents to divide the project into different modules and sub modules also he will use flowchart to implement the design and logic for each functionality.

EX: Based on the previous flow chart for login process. The frontend developer will design the following phases.

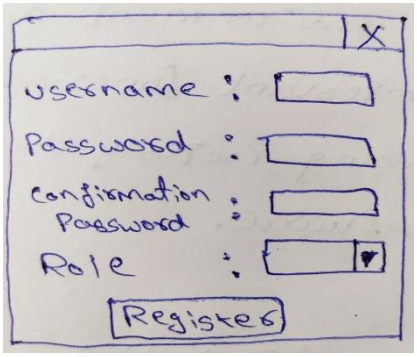
Home page



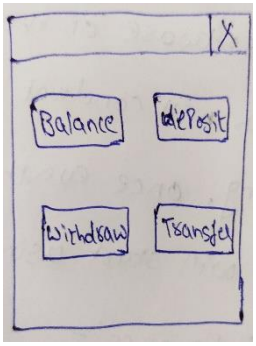
login page



Register page



Options page



Responsible person = Developer

Input = HLD, LLD document

Software testing

Output = Software product

5.Testing:

In this phase the tester will test the quality of the software. It means the tester will verify all the functionalities are properly working or not and all the clients requirements are satisfied or not.

The tester will use either manual testing or automation testing process the software.

Responsible person = Tester

Input = Software product

Output = quality software product

6. Release/Deployment:

After the software design development and testing .The development company will form a team called as "**Release team**". It consists of some developers and testers.

The developers of the release team will deploy the software on the client environment and the tester will conduct main important functionality testing. Once everything working properly then the client will start using the software.

7.Maintenance:

In this phase the **CCB** team or **production support** team or **maintenance** team will provide assistance to the client in using the software.

In this team there will be some technical architect developers and testers.

There are 2 types of maintenance

1. Corrective maintenance

Software testing

2. Enhancive maintenance

1. Corrective maintenance:

While using the software if there is any issue in a particular functionality then the client will **give a request to the maintenance team.**

The technical people of the maintenance team will identify the problem area and inform to the developers. The developers will fix the issue and the testers will test weather the issue is resolved or not.

Finally a file in the form of an update will be sent to the client. The client will run the update to get the solution to the problem.

NOTE:

- A solution to a specific problem is called as **HOT fix**.
- A set of HOT fixes is called as **SERVICE PACK**.

2.Enhansive maintenance:

When there is no issue with the existing software but the client wants to have some enhancements to the existing software .he will **give a change request to the maintenance team**.

The technical people of the maintenance team will analyze the cost of enhancement and report to the management. After getting permission from the management developers will add extra features, testers will test them and finally an update file is created and sent to the client. The client will return the update to get the enhancement.

SDLC Models:

Based on the availability of requirements we can make use of different types of SDLC models to implement the software. all the models internally follows SDLC process.

Software testing

Every model will follow a process and it will have some advantages and disadvantages.

The SDLC models are

- Water fall model
- Incremental model
- Prototype model
- Spiral model
- V- model
- Agile model

1.Water fall model:

it is the first and oldest model used to implement software project.

When all the client requirements are clear and small in size then we this water fall model.

In this model the software will be implemented as phase by phase. after one phase is completed then only the next phase will be started.

Advantages:

1. It is easy to implement .
2. Since the requirements are simple we can implement the software very fastly.

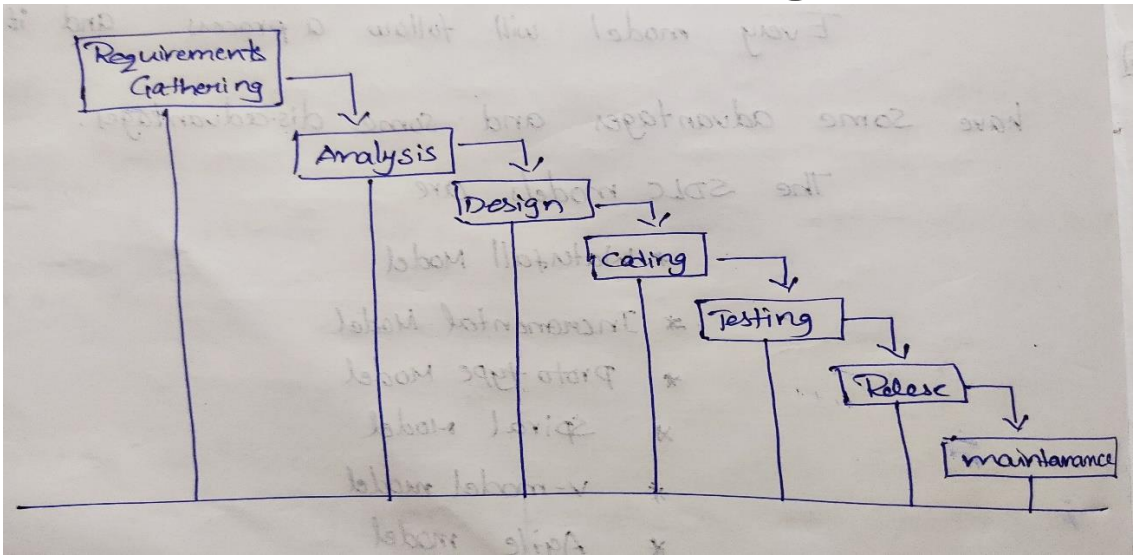
Disadvantages:

1. Resources wastage will happen.

2.Changes to the existing system is difficult

3. Testing is happening in the later stages. So cost of fixing the defect is high.

Software testing



2.Incremental model:

When the client requirements are clear but huge in size then we use this incremental model.

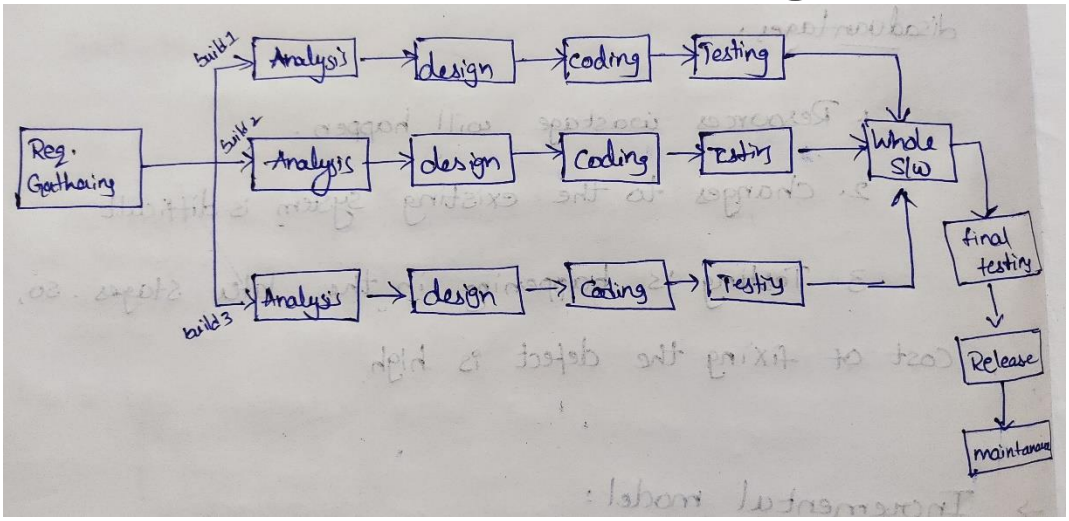
In this model the software will be implemented as build by build.

After gathering all the requirements they will be divided into different parts called as **BUILDS.**

For each build some set of people are allocated and each build will be implemented parallelly.

After all the builds are developed finally it will be integrated as a whole software on this final testing should be conducted followed by release and maintenance.

Software testing



Advantages :

1. The resource wastage can be avoided.
2. The burden on individuals can be reduced.

Disadvantages:

1. It will take longer period of time for the end user to get the final version of the software.
2. Changes to the existing system is difficult.
3. Testing is happening in the later stages.

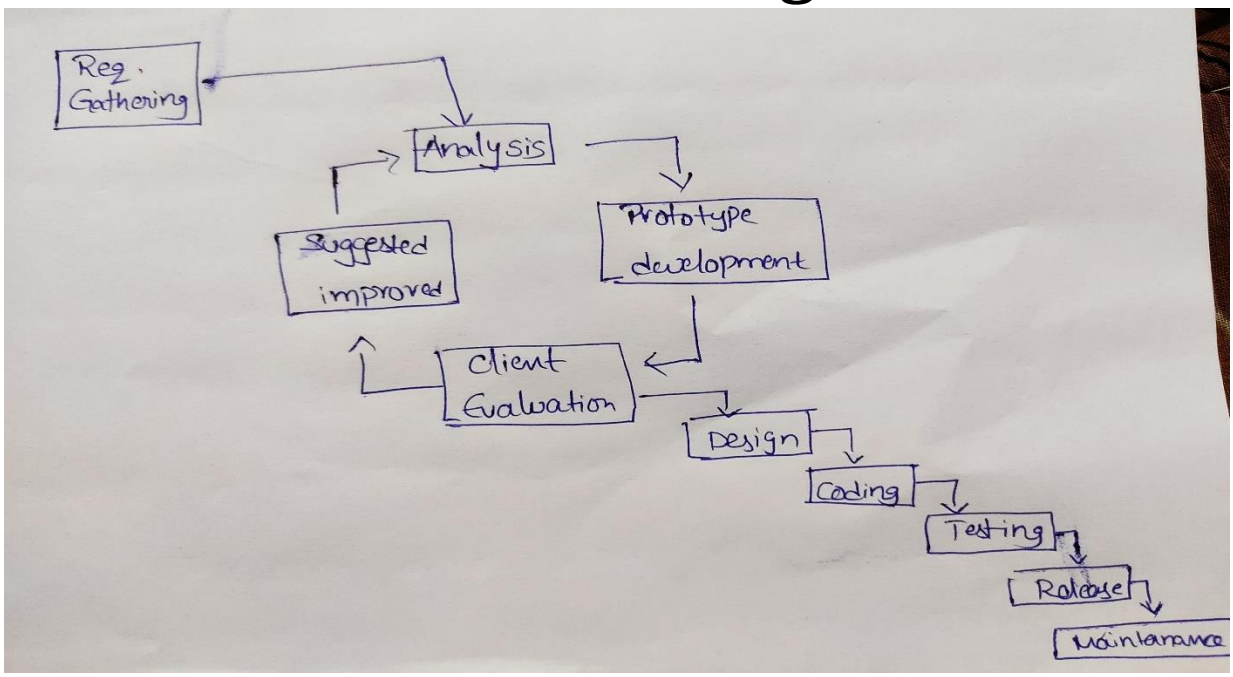
3. Proto type model:

when the client requirements are not clear then we use this prototype model.in this model the actual software will be implemented based on a sample software called Proto type.

Here at first the development company will prepare a prototype and send to client evaluation. After the suggestive improvements from the client the prototype will be modified and again it will be send for client evaluation this cycle will continue. Until satisfies with the prototype.

After the Clients conformation the actual design, coding, testing, release and maintenance will be perform.

Software testing



Advantages:

1. It is not required to perform changes to the existing system. Because after Clients confirmation only the actual design, coding, etc. are happening.

Disadvantages:

1. It is costly when compared to other models.
2. Testing is happening in the later stages.

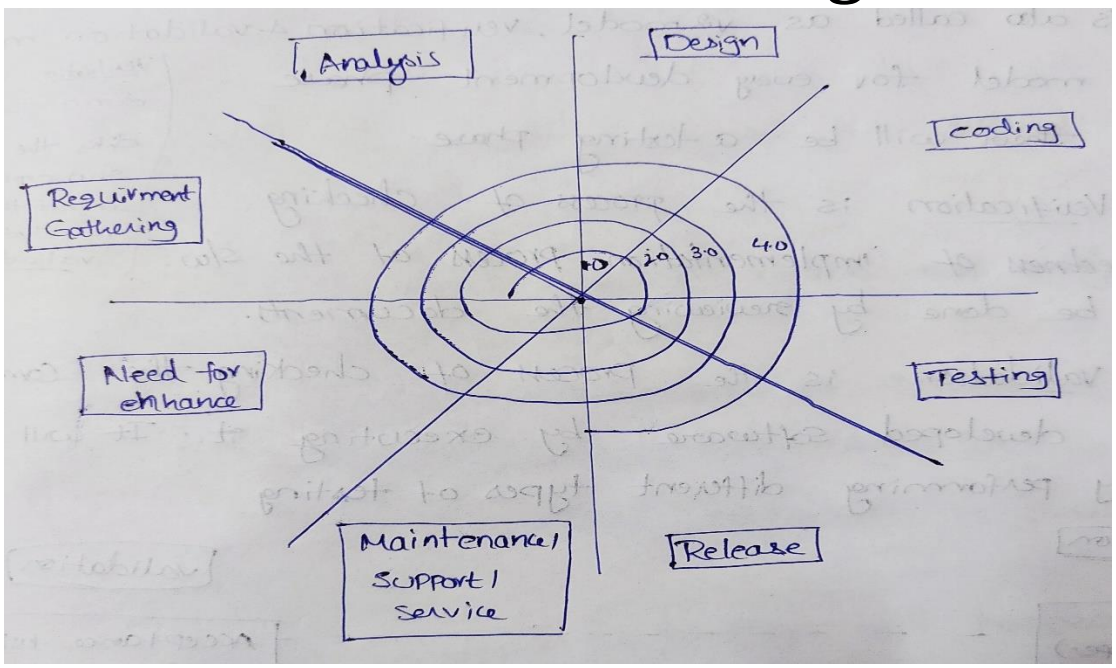
4.Spiral model:

This model is used to implement Product based software.

In this model **version by version** the software will be implemented.

Out of all requirements few requirements are made as one version and it will be implemented by following SDLC process. After checking the feedback from customers and with some other features next version will be implemented and soon….

Software testing



Advantages:

1. A version can be released fastly to the customers with a set of features.
2. There will be **NO** release time pressure.

Disadvantages:

1. We don't know the end date.so we have to work for indefinite time.
2. Maintaining the resource for a longer period of time is difficult.
3. Testing is happening in the later stages. So cost of fixing the defect is high.

5.V-Model/ V square model:

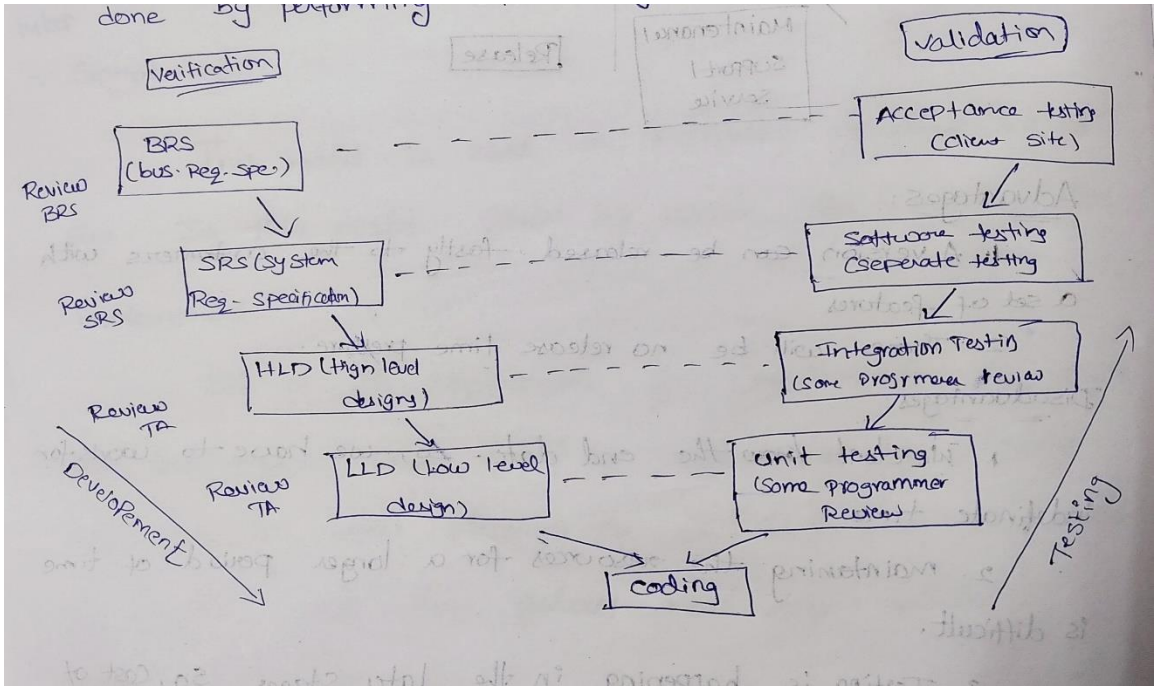
It is also called as **Verification** and **Validation** model.

In this model for every development phase parallelly their will be a testing phase.

Software testing

Verification is the process of checking the correctness of implementation process of the software. It will be done by reviewing the documents.

Validation is the process of checking the correctness of developed software by executing it. It will be done by performing different types of testing.



Advantages:

Testing is happening in the earlier stages which will reduce the cost of fixing the defect.

Disadvantages:

Changes to the existing system is difficult.

6. Agile model:

Agile stands for **fastness.**

It is the mostly widely used SDLC model to implement the projects.

It contains all the advantages of previous model.

Some of the advantages of agile are

Software testing

- **The client collaboration is available at every stage.**
- **Lot of meetings will happen to get the best results.**

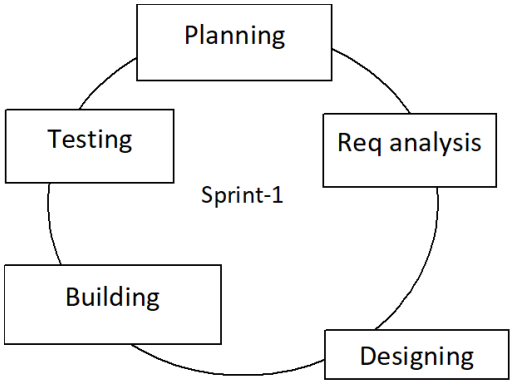
In the agile model the following are the frequently used terminology.

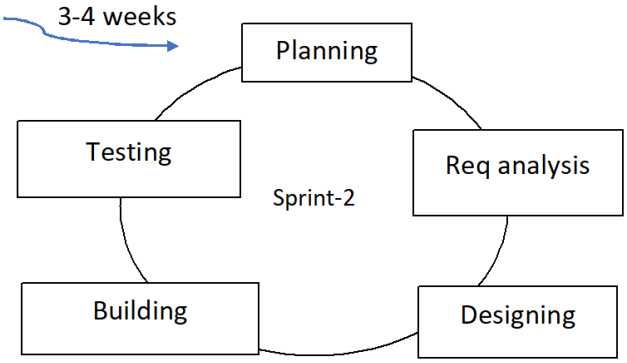
- **User stories** = the client requirements in the agile model are called as user stories.
- **Epic =** it is a large user story that can be divided into different small user stories.
- **Sprint =** it is a small part of the software.

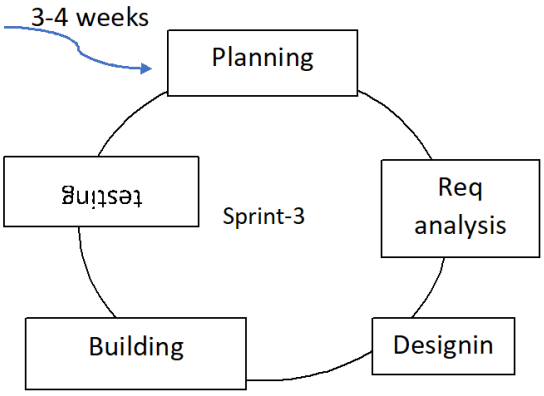
In the agile model the software will be implemented as **Sprint by Sprint.** All the user stories are divided into different equivalent sprints. In such a way that each sprint implementation will take around **3-4 weeks.** In every sprint implementation planning analysis, design, coding and testing will be followed.

Once a sprint implementation is completed the client can use the sprint without waiting for other sprints.

Software testing







Software testing

Agile scrum model:

We can make use of different **sub models** in agile to implement the project. One of the important frequently used sub model is **agile scrum model**.

The following are the different peoples involved in the agile scrum model.

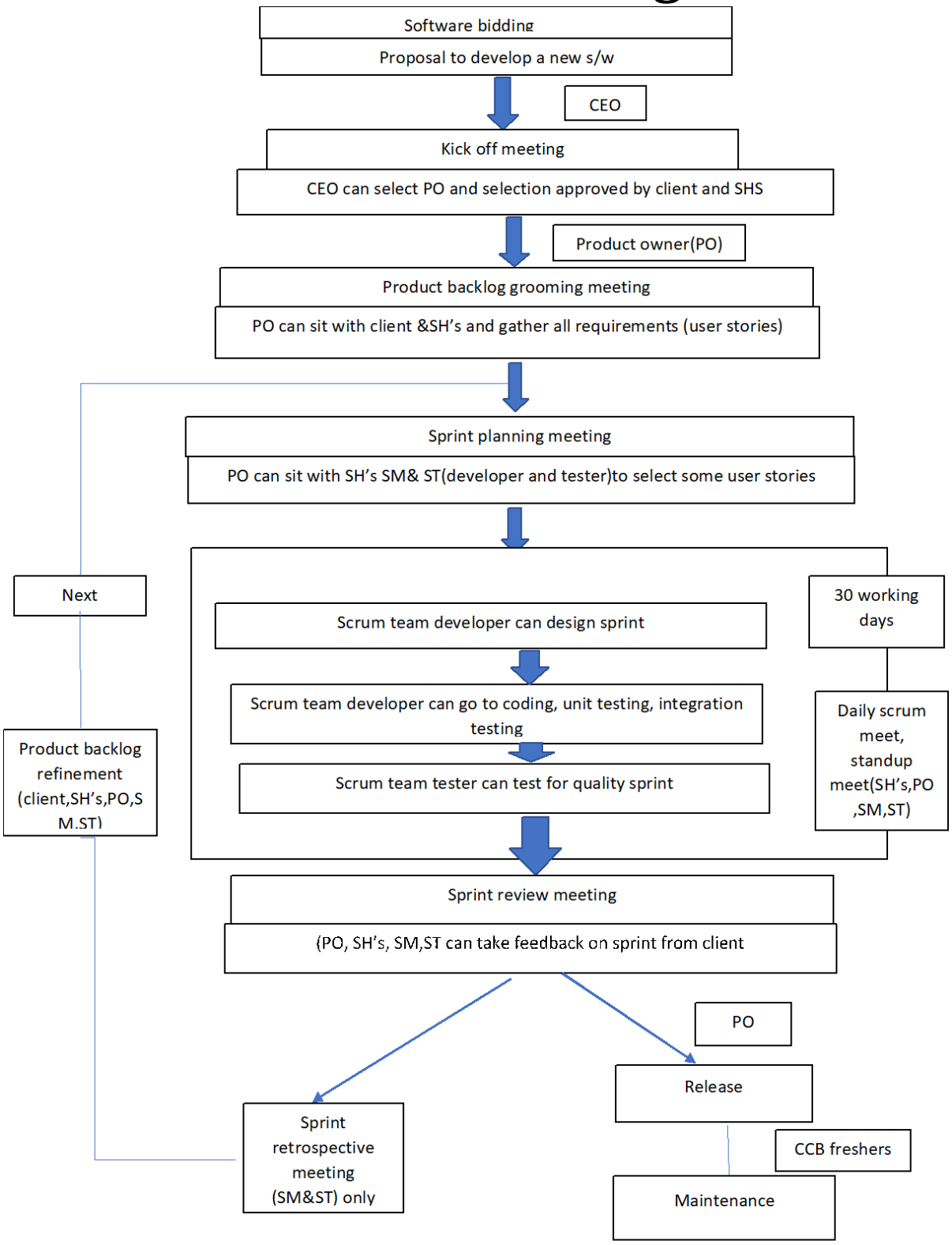
- Client
- Stake holder (SH) – is a **technical person** recruited by the client for technical assistance. He will be the **clients representative with the development company.**
- CEO
- Product owner (PO) – is the decision maker in the agile scrum model. **He is responsible for entire project implementation in agile.**
- Scrum master (SM) – is a **facilitator** who is responsible to organize different meetings in the agile scrum model.

  He maintains a scrum team that consists of some **developers** and **testers** called as **Scrum team developer** and scrum **team tester**.
- CCB (Change Control Board) team.

Agile Scrum Architecture Model:

In order to implement a project using agile scrum model. We can follow this architecture model.

Software testing



Software testing

i. The client will give a proposal to develop a new software called as **Software Bidding.**
ii. The CEO will conduct KICK OFF MEETING to select the **Product Owner** and whose selection will be approved by the **Client** and **Stake Holder.**
iii. The **product owner** (PO) will conduct **PBL** (**Product Backlog**) meeting to collect all the user stories from the client and stake holder.
iv. The PO will prepare a PBL document with all the user stories.
v. The **Scrum Master** (SM) will conduct sprint planning meeting to select some user stories for the current sprint.
vi. The PO will prepare **SBL** (**Sprint Backlog**) document with selected user stories.
vii. The **scrum team developer** will start design the sprint.
viii. The scrum developer also performs coding, unit and integration testing.
ix. The **scrum team tester** will test the quality of the software.
x. This design, coding, testing is a **30 days** work.
xi. Every day a special meeting called as **Daily Scrum Meet** or **Stand Up Meeting** will be organized by the scrum master to know the status of the sprint.
xii. After the sprint implementation sprint review meeting will be organized by the scrum master to get the feedback on the sprint from the client.
xiii. If the feedback is **positive** product owner will release the sprint to the client and CCB team will maintain it.
xiv. Where as if the feedback is **negative** scrum master will organize sprint retrospective meeting to fix the issue.
xv. The product owner will organize PBL refinement meeting to find the enhancement to the existing sprint.

Agile scrum components:

Software testing

The following are the different components available in the agile scrum model.

1. Jobs
2. Ceremonies
3. Artifacts

1.Jobs:

These are the roles and responsibilities of different of people of agile model.

Client roles:

i. Responsible to give software bidding involves in the stake holder recruitment.
ii. Involves in the kickoff meeting to approve the product owner selection along with stake holder.
iii. Responsible to deliver all the user stories to the product owner.
iv. Responsible to give the feedback on the sprint in the sprint review meeting.
v. Involves in the PBL refinement meeting to know the enhancement.

Stake holder role:

i. Involves in the kick off meeting to approve the PO selection.
ii. Responsible to deliver the client user stories to the product owner in the PBL grooming meeting.
iii. Involves in the sprint planning meeting in the selection of some user stories for the current sprint.
iv. Involves in the daily scrum meet to know the sprint status.
v. Involves in the sprint review meeting and PBL refinement meeting.

Software testing

Product owner role:

i. PO is the decision maker.
ii. Optionally PO can recruit a **subject matter expert (SME)** if any technically assistance is needed on a particular domain.
iii. He is responsible to conduct PBL grooming meeting to gather all the user stories from the client and stake holder.
iv. Involves in the sprint planning meeting to select some user stories.
v. Involves in the daily scrum meet.
vi. Involves in the sprint review meeting.
vii. Responsible to release the sprint when the client feedback is positive.
viii. Involves in the PBL refinement meeting.
ix. Responsible to divide the epic into small user stories.
x. Responsible to prepare PBL and SBL documents.
xi. Provide the responsibilities to the scrum master.

Scrum Master roles:

i. He is the facilitator.
ii. Responsible to organize most of the meetings such as sprint planning, daily scrum meet, sprint review, sprint retrospective and PBL refinement meetings.
iii. Responsible to maintain scrum team consisting of scrum team developers and scrum team testers.
iv. Take care about the scrum team from outside people.

Scrum team developer role:

i. Involves in the sprint planning meeting to understand user stories for the current sprint.
ii. Responsible to design the sprint.
iii. Responsible in coding the sprint.
iv. Responsible to perform unit and integration testing.
v. Responsible to fix the bugs identified by the testers.

Software testing

vi. Involves in the daily scrum meet, sprint review, sprint retrospective and PBL refinement meetings.

Scrum Team Tester role:

i. Involves in the sprint planning meeting to know the user stories for the current sprint.
ii. Understanding the user stories.
iii. Identifying the test scenarios.
iv. Preparing the test case documents.
v. Performing the test execution.
vi. Identifying the test defects.
vii. Sending defect reports to development team.
viii. Preparing test scripts in case of automation testing.
ix. Performing retesting and regression testing.
x. Involves in the daily scrum meet, sprint review, sprint retrospective and PBL refinement meeting.

2.Ceremonies:

These are the different meetings that will be organized by different people for different purpose.

Ceremony/meeting	Organized by	purpose	People involved	Duration
1.kick off meeting	CEO	To select a PO	CEO,CLIENT,SH	No time bound
2.PBL grooming meeting	PO	To collect all the user stories	PO,CLIENT,SH	No time bound
3.sprint planning meeting	SM	To select some user stories	PO,SH,SM,STD,STT	No time bound

Software testing

4.Daily scrum meet		To know the sprint status	PO,SH,SM,STD,STT	20-30 min
5.sprint review meeting		To get feedback from client	PO,SH,SM,STD,STT,CLIENT	2-3 hrs
6.sprint retrospective meeting	SM	To fix and test the issue identified in the review	SM, STD,STT	2-3 hrs
7.PBL refinement meeting	SM	To know the enhancement to the sprint	PO, SH, SM, STD, STT, Client	2-3 hrs

3.Artifacts:

These are the different types of documents that can be prepared by different people in the agile model.

Artifact (doc)	Prepared by	Contains
1.PBL doc	PO	List of all user stories
2.SBL doc	PO	Selected user stories
3.HLD, LLD doc	STD	Main and sub module details
4.unit, integration test case doc	STD	Unit, integration test results
5.Test case doc	STT	Functionality results
6.defect report doc	STT	Details of identified defects
7.test summary doc	SM	Total test process
8.sprint burn down chart	SM	Defects during the timeline

Software testing

Software testing:

It is the process of verifying and validating weather the developed software is properly working or not and all the client requirements satisfied or not.

The IEEE definition for software testing is “testing is the process of exercising or evaluating system or system component by manual or automated means to ensure that its satisfies specified client requirements”.

In general software testing can be conducted in 2 ways they are

1.Manual testing

2.Automation testing

1.Manual testing:

In this method we will perform the software testing only by following manual approaches.

- We don’t use any tools or program to conduct the testing.

2.Automation testing:

In this method we use a tool or programing language to

conduct the software testing.

So as a test engineer we have to test the quality of the software.

Quality:

Quality is measured by different people in different ways.

In a development company prospective if the developed software is satisfying all the client requirements then it is called as a quality software.

In the clients perspective if the developed software is fit for his purpose then it is called as quality software.

The following are the different factors that are effecting the quality of the software.

- Failures
- Budget
- Delivery time
- Reliability

Software testing

- Maintainability

When the developer software is not having any failures it is developed with in the client expected budget, it is delivered with in the client expected delivery time, it is working in any kind of circumstances and it is easy and less cost to maintain then that software is called as a quality software.

Quality management:

In the software product the quality will be achieved with the help of this quality management.

Quality management is the process of preventing the defects in the process to ensure that there are no defects in the final product.

It is implemented in two ways they are:

1.Quality assurance (QA)

2.Quality control (QC)

1.Quality assurance (QA):

It is an activity that will be applied on a process, identify the weakness in each process and suggest the improvements.

2.Quality control (QC):

It is an activity that can be applied on a product identify the weakness and suggest the improvements.

Difference between QA and QC:

Quality assurance	**Quality control**
1.it is process oriented	**1.**it is product oriented
2.it is a preventive approach	**2.**it is a detective approach
3.it can be done at any stage of SDLC	3.it will be performed only after software product development.
4.it will be done through reviews	**4.**it will be done through testing

Software testing

Types of mistakes:

During the software development different people will identify different types of mistakes in the software.

They are

1.Error:

These are the mistakes identified by the developer at the time of coding of the software.

2.Defect:

These are the mistakes identified by the tester during software testing.

3.Bug:

After identifying the defects the tester will send those defects in the form of a defect report to the development team. If the development team accept the defect then that is called as a bug.

4.Failure:

After releasing the software to the client while using the software if the client identifies any mistakes in the software then it is called as a failure.

Principles of software testing:

While performing the software testing we have to keep in mind about the following principles. If we follow these principles then we will get best test results.

Principle 1: testing shows presence of defects.

Principle 2: exhaustive testing is impossible.

Principle 3: early testing.

Principle 4: defect clustering.

Principle 5: pesticide paradox.

Principle 6: testing is context dependent.

Principle 7: absence of error fallacy.

Software testing

Principle1:

Always we have to keep in mind that there are some more defects in the software which are to be identified. We should never say that the software is not having no defects.

Principle2:

In general we will follow three approaches in conducting the software testing they are

1. Exhaustive testing
2. Ad – hoc testing
3. Optional testing

1.Exhaustive testing:

in this method we have to test the specification with all possible values.

Ex: In a banking system money transfer process transfer amount field must contain a value between 1000 to 50000.

The above specification in exhaustive testing will be tested with values like 1000 ,1001,1002 up to 50000.it is almost impossible to perform.

2.AD – HOC testing:

in this method we will test the specification with random values. When there is no sufficient time to conduct the software testing then we use this method.

EX: The above example in ad-hoc testing can be tested with values like 500, 20000, 60000 etc. it is not recommended because it is not complete testing.

3.Optimal testing:

This is the method to be used in software testing here we will test the specifications with all possible conditions.

EX: the above example in optional testing will be tested as follows

Optimal test	test data	Valid/invalid
Min-1	999	Invalid
Min	1000	Valid
Min+1	1001	Valid
Max-1	49999	Valid

Software testing

Max	50000	Valid
Max+1	50001	Invalid

Principle3:

It specifies that how much early it is possible to start the testing that much early it should be done. this will reduce the cost of fixing the defects.

Principle4:

Suppose a small part of the software is having more defects rather than remaining large part of the software perform the testing on that small part where more defects are clustered.

EX: in a banking software there will be different modules but most of the problems might occur in the banking module only. So do test on that part mostly when compared to remaining modules.

Principle5:

Even though we are preparing g multiple test cases to test similar kind of functionality we don't get different test results. So don't duplicate the test cases.

EX: suppose that in an application we have admin login page and employ login page. To test the two login pages we no need to write to two different test cases. We can write a single test case to test both login procedures.

Principle6:

Based on the type of application we are testing context dependent We have to follow different approaches in conducting the testing so testing is context dependent.

EX: when we are testing a desktop based application we will perform platform compatibility test where as if we are testing a website we have to conduct both platform compatibility and browser compatibility test.

Principle7:

Even though the software is not having any failures we can not say that it is a quality software along with not having failures. Then only it can be called as a quality software.

Testing Types:

In a border way software testing is of two types. They are

Software testing

1.Static testing

2.Dynamic testing

1.Static testing:

Testing the software at specification or implementation level is called as static testing.

At the time of specification we will contain only different types of documents. So in this method we have to test those documents for their correctness. That is why static testing is also called as document testing.

It will be performed by reviewing the documents.

2.Dynamic testing:

Testing the software by executing it is called as dynamic testing. It will be performed with different levels of testing.

Reviews:

Examining the correctness or completeness of a document is called as review.

The following are the different people who involves in the review process

- **Author –** the person who prepares the document
- **Reviewer(inspector) –** the person who is examining the document.
- **Moderator (inspection leader) –** the responsible person for entire review process. He maintains a set of reviewers.
- **Scribe (recorder) –** the person who will do the recording of all communication happening in the review process.

Types of reviews:

The following are the different types of reviews that can be performed in static testing.

1.Informal reviews

2. walkthrough

3. Technical reviews

4. formal reviews

Software testing

1.Informal reviews:

These are also called as **peer to peer** reviews.

It is a oral communication between the team members. This review meeting is to help each other. It is not a prescheduled meeting and it is not documented.

2.Walkthrough:

Step by step explanation of a document by a author to the reviewers is called as walkthrough.

It is a scheduled meeting and it has to documented.

3.Technical reviews:

These reviews will be done by the technical people called developers. The moderator will schedule a meeting between the developers to decide what kind of architecture model to be used. What are the classes and methods to be implemented etc.

So it is a schedule meeting and it has to be documented.

4.Formal reviews:

This is the most structured review process. All the people such a author, reviewer, moderator and scribe will involve in the formal review.

It is a pre scheduled review and it has to be documented.

The following are the different phases under the formal review

a. Planning
b. Kick off meeting
c. Preparation
d. Review meeting
e. Re – work
f. Follow-up

a. Planning:

After competition of the document preparation the author will give a formal request to the moderator about his document to be reviewed.

The moderator will prepare a checklist about what has to be reviewed in the document.

Also the moderator will schedule the review of the document with one or more reviewers.

Software testing

b. **Kick off meeting:**
   it is the step by step explanation of the checklist by the moderators to the reviewers. It is an optional phase.
c. **Preparation**:
   in this phase the reviewer will start examining the document and prepares the set of questions to be asked to the author.
d. **Review meeting:**
   it is the most important phase in formal review. It contains different sub phases. They are
   i. **logging**:-
   in this phase the scribe will document entire communication that is happening in the review process.
   ii. Discussion:
   in this phase the reviewer will start asking the questions with the author and the author will explain about the reviewer questions.
   iii. Decision:
   the author will decide whether the modification has to be performed or not.
e. **Rework:**
   Suppose more than 70 % of the document has to be changed. then the author will decide to re design the document.
f. **Follow up:**
   It is the responsibility of the moderator to follow up the review process. There by he can understand delays in the review. So that the work allocation can be done with the other reviewers.

Difference between inspection and audit:

Inspection is a scheduled review. It is performed within the organization by the higher management people.

Where as an audit is a sudden review which will be conducted by the third party organizations. Generally when company applies for standard certification. This audit will happened.

Levels of testing:

The dynamic testing will performed with the help of different levels of testing. They are

I. Unit testing

Software testing

II. Integration testing
III. System testing
IV. User acceptance testing
V. Release testing
VI. Maintenance testing

1.Unit testing:

It is the process of testing **internal logic of each and every program in the project.** We must have programming language skills to conduct this test. Generally developers are the people who will conduct this test.

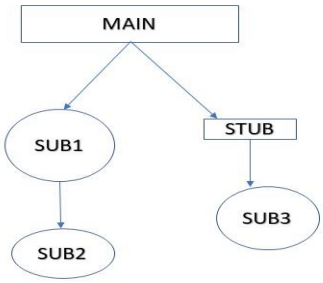
2.Integration testing:

In this test we have to test the **inter connectivity of the one program and other program.** In order to conduct this test we must have **programing knowledge**. Again developers are the people who will conduct this test. The following are the different approaches used by the developer to conduct the integration testing.

a) Top down approach
b) Bottom up approach
c) Hybrid approach
d) Big bang approach

a) Top down approach:

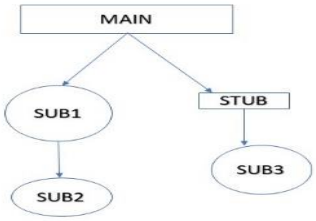
When the **main program is ready** and also **some sub programs are ready** but some **sub programs are not ready** we this approach. Here in place of under constructive sub program we use **dummy program** called **STUB.**



Software testing

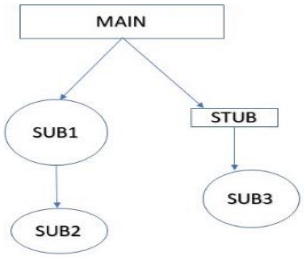
b) Bottom up approach:

When all the **sub programs are ready** but **main program are not ready** then we use this bottom up approach. Here in place of under constructive main program we will use a **dummy main** program called **drive.**



c)Hybrid approach:

When the **main program is not ready** and also some of **the Sub programs are not ready** then we use this hybrid approach.



d)Big bang approach:

when the main program and all the sub programs are ready and no program is under construction then we use big bang approach.

Note:

Developers will use **white box testing techniques** to conduct unit and integration testing.

Software testing

3.System testing:

In this test we have to **check the quality of the software** by **executing the developed software**. This test will be conducted in two ways. They are

a. Functional testing
b. Non-functional testing

a) <u>Functional testing:</u>

In this method we will test whether all the client requirements are satisfied or not. It is also called as **requirements testing**.

The following are the different tests that comes under functional testing.

a) GUI testing
b) Input domain testing
c) Error handling testing
d) Data base testing
e) Data volume testing
f) Inter system testing
g) Recovery testing

a) GUI testing:
In this we have to check the **navigations** when we select **menu item in a menu bar related page has to be displayed**. Also when we click on a button related operation is happening or not to be tested.

b) Input domain testing:
In this test we have to verify that **every input field in every page accepting only valid data or not.**
EX: In a application form page name field should be accept only alphabets. It not accept a numbers ,phone number field should accept compulsory 10 digits.

c) Out put testing:
Some times in the application based on **the input we provide in the filed another filed get the out put.**

Software testing

In output testing we have to verify that when we provide input in one filed. We have to check that related other field is displaying correct output or not.

EX: In the application form when we provide the date of birth correctly then age field should be display the correct age.

d) Error handling testing:

An error or exception is a deviation from the expected result.

In the programing environment we will get two types of errors. They are

1) Compiler error
2) Run time error

1.**Compiler error**: These errors will occur at the time of compilation of the program. They will be identified by the related language compiler until we fix these errors the program execution will not be started.

EX: Syntax error, type conversion errors.

2.Runtime error:

These error will occur at the time of execution of the program. They will occur because of the wrong programming logic implemented by the developer or mostly because of the incorrect input given by the user.

Once a runtime error occurs the program execution will be stopped. It is the responsibility of the program developer to provide necessary steps to be followed to continue the program execution. This process is called as error or exception handling.

As a test engineer in order to test this exception handling we have to test the functionality with valid data and ensure that proper result is displayed. Also we have to check the functionality with invalid data and ensure that related error message is displaying or not.

Note: the above four tests are called as frond end testing.

Software testing

v. Data base testing:

it is the process of performing and operation on the frontend application and verify whether it is correctly effecting data base or not.

vi. Data volume testing:

in this test we will check the capacity of the data base by adding lot of test data.

Note: the above two tests are called as Backend test.

Vii. Inter system testing:

in this test we have to test the inter connectivity between AUT (application Under Test) and third party service.

Ex: in amazon.in site to allow the customers to place orders and to the amount. It internally uses a third party payment service. While testing amazon site the inter system testing will be conducted as follows.

1.when we click on pay now button we have to redirect from amazon site to the payment service.

2. test all the payment options such as credit card payment, net banking, wallet payment etc.

3.after the payment process is completed we should redirect back to the amazon site.

Viii. Recovery testing:

In this test we have to verify that the software is working properly or not in any kind of circumstances.

For example while using the software if the system got restarted. The software should not be corrected and it should work in a normal way.

b.Non functional testing or expectations testing:

in this case we have to verify that the developed software is meeting the clients expectations or not. The following are the different tests we will conduct under non functional testing or expectations testing.

Software testing

i. Usability testing
ii. Compatibility testing
iii. Hard ware configuration testing
iv. Performance testing
v. Security testing
vi. Multi languity testing
vii. Parallel testing
viii. Compliance testing

i. Usability testing:

In this test we will verify the application is user friendly or not. It means the user should understand each and every option and also we have to check the uniqueness of look and feel through the application.

Also we have to check the alignments, spellings etc.

ii. Compatibility testing:

In this test we will verify how the application is working in different environments. This test will be conducted in 2 ways. They are

a. platform compatibility test
b. Browser compatibility test

a. Platform compatibility test:

In this test we have to check whether the software is working in different operating system or not.

b. Browser compatibility test:

In this test we have to check whether the software is working in different browsers or not.

Note:

case of desktop In based application we will conduct only platform compatibility test where as in case of a web application we can perform both platform compatibility as well as browser compatibility test.

iii. Hardware configuration testing:

In this method we have to verify that the connected hardware devices are working properly or not.

Software testing

iv. Performance testing:

In this test we will verify the speed and the capacity of the software. This test will be done in the following ways.

a. Load test
b. Stress test
c. Spike test
d. Endurance test

a. Load test:

In this method we will apply customer expected load and test the behaviour of the software.

EX: suppose that the client requirement is 1000 users should login to the application.

To test the above specification as a tester we will try to login with 1000 user accounts and test the software behaviour.

b. Stress test:

In this test we apply more than customer expected load and test the software behaviour.

EX: to test the above specification in stress test we have to login with 1000 users, 1100 users, 1200 users etc. and test the software behaviour.

c. Spike test:

In this test we will apply huge amount of load than the customer expected to know the maximum limit.

EX: to test the above specification in spike test we will login with 1000, 2000, 3000 users etc. to know the server crash point.

d. Endurance test:

In this method we will apply customer expected load for a longer period of time to verify any leakages.

EX: to test the above specification in endurance test we will login with 1000 users and keep on logging for 15

Software testing

mins, 30 mins, 60 mins etc. to verify any automatic disconnections are happening.

V. Security testing:

In this test we will check how secure is the software. This test will be done in the following ways.

a. Authentication test
b. Access control test
c. Encryption test

a. Authentication test:
in this method we have to verify that only valid user is accessing the application or not.

b. Access control test:
In this test we have to check the connected user is performing only permitted operations or not.

c. Encryption test:
In this test we to check whether the sensitive information like passwords, account numbers etc. are stored in a encrypted manner or not.

VI. multilanguity testing:

In this test we have to verify whether selected language is displaying in the application or not. This test will be performed only on those applications where multi language is supported.

VII. Parallel testing:
It is the process of comparing AUT (Application Under Test) with similar kind of software to know the strengths and weakness. In general this test will be conducted for a product based software.

VIII. Compliance testing:
it is also called as **standard testing**. Here we have to verify whether the **developer is following testing standards or not.**

Software testing

Note: testers will conducted this system testing by using "Black-Box" testing techniques**.**

4.User acceptance testing:

This test will be conducted by the client to ensure that whether the **software is acceptable or not.** This test will be conducted in the following ways.

i. Alpha testing / factory testing
ii. Beta testing / field testing
iii. Regulatory acceptance testing
iv. Contractual acceptance testing
v. Operational acceptance testing

i. Alpha testing / factory testing:

After the project development and testing is completed. Then the development company will requests the client to check the software. The client will visit the development company and test the software where it was developed and tested. This test is also called as factory testing.

ii. Beta testing / field testing:

After completion of alpha testing the client will test the software in his own environment where he will use the software to ensure that it is satisfying all this requirements. This test is also called as field testing.

NOTE: the **combination of alpha and beta** testing is called as **YELLOW BOX** testing techniques.

iii. Regulatory acceptance testing:

This test will be conducted by a regulatory authority only for machine critical software where human lives are involved.

In this test after the development company completes software development and testing the client we also

Software testing

test the software. After this a regulator authority will conduct a special test on the software to ensure that whether it is satisfying their standards or not.

iv. **Contractual acceptance testing:**
in the beginning of the project some kind of agreement will happen between the client and development company. This is called as contract document.
At the end of the software Implementation the client has to check the contract document to ensure that all the details mentioned in the contract document are satisfied or not. This kind of test is called as contractual acceptance testing.

v. **Operational acceptance testing:**
the system administrator of the client will conduct this test. In this test he will verify whether the software is working properly or not. In all kind of networks and also the side effects on already existing software after this software installation.

5.Release testing:

After the project development and testing is completed the development company will form a team called as **release team.** This team consists of some developers and testers. The developers of this team will deploy the software on the clients environment and testers of the release team will conduct main important test to ensure that whether the software can be used by the client or not.

6.Maintenance testing:

This will be conducted by the maintenance team or CCB team. This team consists of some technical people, developers and testers. The maintenance can be done in 2 ways. They are

i. Corrective maintenance
ii. Enhancive maintenance

Software testing

i. **Corrective maintenance:**
while using the software if the client finds an issue. He will send a request to the CCB team about the issue.
The technical people of the maintenance team will conduct **RCA (Route Cause Analysis)** to find the problem area and inform to the developers. The developers will fix the issue and testers will test it. Finally a solution will be send to the client in the form a update file.

ii. **Enhancive maintenance:**
even though the software is working properly some times the clients wants to have some extra enhancements to the existing software. Then the client will send a "**change request** to the CCB team.
The technical people of the CCB team will conduct impact analysis and prepares a document called as **impact analysis document**. This document will describe different kind of details to different people.
The management people can understand the cost of enhancement, the developers will know what are the new additions to be performed and testers will understand. What are the new test cases to be prepared.
The developer will implement the enhancement and tester will test them. Finally an update file is prepared and send to the client.

Testing methodologies (OR) test design techniques:

In order to perform different types of test we have to make use of different techniques called as testing methodologies or test design techniques. They will represent how to perform the test. They are

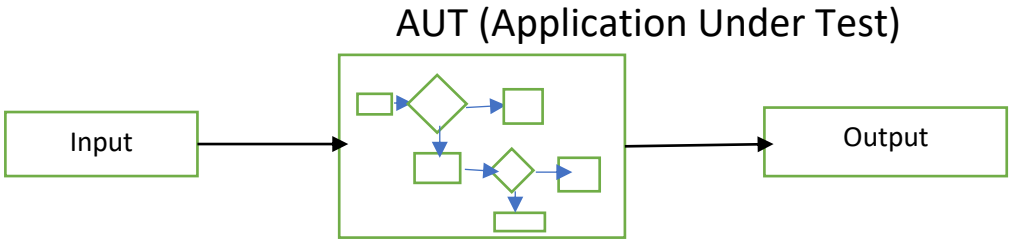
I. Structure based testing [white box testing techniques]
II. Specification based testing [black box testing techniques]

Software testing

III. Experienced based testing

I. Structure based testing [white box testing techniques]:

In this method we will perform the testing based on the program structure. In general developers will use these white box testing techniques in conducting the unit and integration testing.



WBT

The white box testing techniques are

a. Statement coverage testing
b. Branch / decision coverage testing
c. Path coverage testing

a. Statement coverage testing:

In this method we have to prepare a test case in such a way that most of the statements are covered. If more than 80% of the statements are covered then we can say that the given test case is the best one to be used.

The statement coverage percentage will be calculated as follows

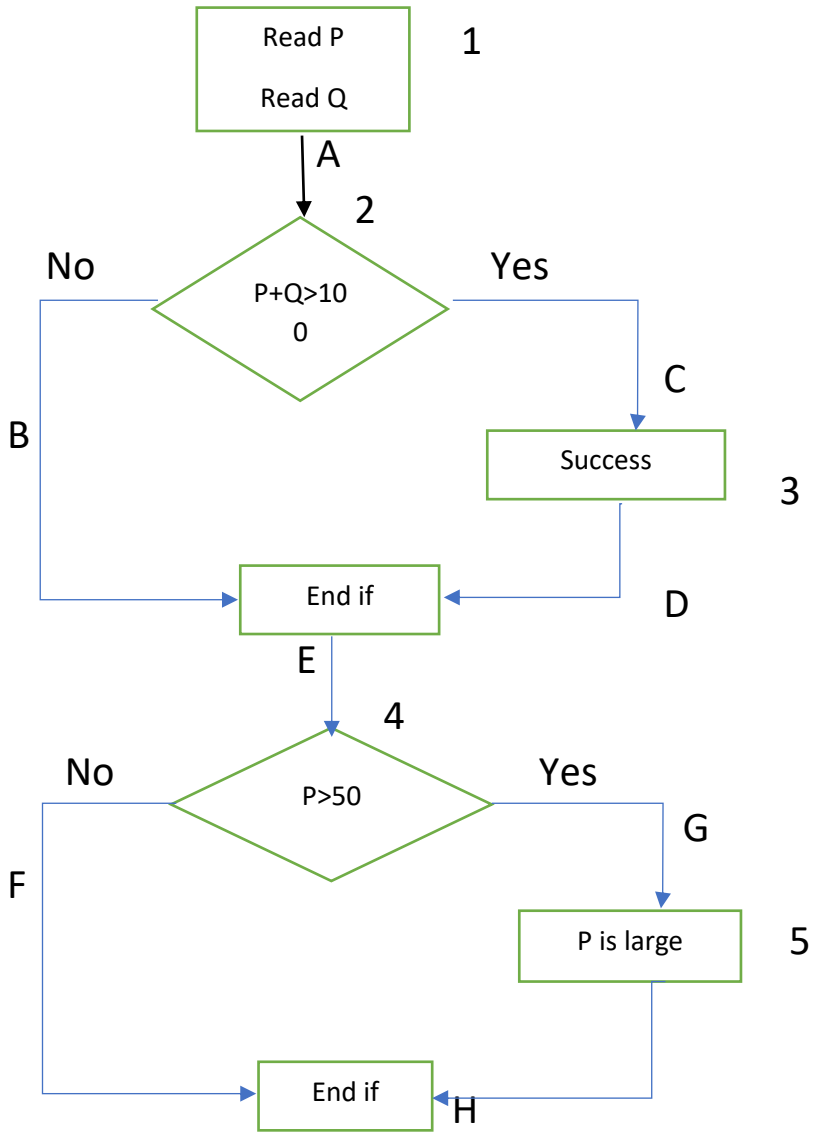
Statement coverage % = No of statements covered /total no of statements* 100

Ex:

1. Read P
2. Read Q
3. If ((P+Q)>100) then
4. Print “success”
5. End if
6. If (P>50) then
7. Print “P is large”

Software testing

8. End if



The above program in statement coverage testing will be tested as follows:

Test case: p=60; Q=50

Statement coverage % =8/8*100

= 100%

b. Branch / decision coverage testing:

in this method we have to prepare a test case in such a way that at least once all the true conditions and at least once all the false conditions must be satisfied.

Ex:

Software testing

The above program in branch coverage testing will be tested as follows:

Test case1: P=80, Q=30

1A-2C-3D-E-4G-5H

[Covers all TRUE conditions]

Test case2: P=40, Q=30

1A-2B-E-4F

[Covers all FALSE conditions]

c. **Path coverage testing:**

in this method we have to prepare a test case in such a way that all possible paths are covered.

EX: the above program in path coverage testing will be tested as follows:

Test case1: P=80, Q=30

1A-2C-3D-E-4G-5H

[Covers all TRUE conditions]

Test case2: P=40, Q=30

1A-2B-E-4F

[Covers all FALSE conditions]

Test case3: P=70, Q=20

1A-2B-E-4G-5H

[covers FALSE & TRUE conditions]

Test case4: P=20, Q=90

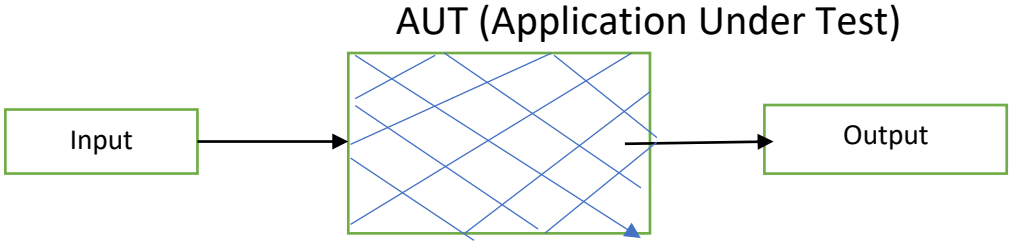
1A-2C-3D-E-4F

[Covers TRUE & FALSE conditions]

II. **Specification based testing:[black box testing techniques]**

In this method the testing will be performed based on user requirements or specification.

Here we will provide a input and verify that expected output is coming or not. We don't check the logic of the program.



Software testing

In general the testers will use the following Black Box testing techniques to conduct the system testing.

a. Equivalent partition testing (EP)
b. Boundary value analysis testing (BVA)
c. Decision table testing
d. State transition testing
e. Use case testing

a. Equivalent partition testing (EP):

in this method we have to divide the specification into different equivalent parts in such a way that each part is exhibiting similar kind of behaviour such as valid or invalid.

The partition can be done based on different data factors such as data type, data size, data range.

EX 1: Father Name field must allow only alphabets

[Data factor: Data type]

Partition 1	Partition 2	Partition 3	Partition 4
Numeric	Alphabets	Alpha – numeric	Special – char
Invalid	Valid	Invalid	Invalid

EX: phone number field must allow only 10 digits

[data factors: data type, data size]

i. Data type check

Partition 1	Partition 2	Partition 3	Partition 4
Numeric	Alphabets	Alpha – numeric	Special – char
Valid	Invalid	Invalid	Invalid

Software testing

ii. Data size check

Partition 1	Partition 2	Partition 3
< 10 digits	10 digits	>10 digits
Invalid	valid	Invalid

b. Boundary value analysis testing (BVA):

This method is used to test the specification based on the data range. It verifies all the possible conditions.

EX: transfer amount must contain a value between 1000 & 50000

[data factor: data type, data range]

1. Data type check

Partition 1	Partition 2	Partition 3	Partition 4
Numeric	Alphabets	Alpha – numeric	Special – char
Valid	Invalid	Invalid	Invalid

2. Data type range

BVA	Test data	Valid/invalid
Min-1	999	Invalid
Min	1000	Valid
Min+1	1001	Valid
Max-1	49999	Valid
Max	50000	Valid
Max+1	50001	Invalid

c. Decision table testing:

When the specification is based on more than one field combination then we use this decision table testing technique to test the validity

EX: 1

Candidate age must be between 25 and 30

Conditions:

For BC candidates 3 years of age relaxation

For SC / ST candidates 5 years of age relaxation

For PH candidates 6 years of age relaxation

Software testing

Category	Age	Valid/invalid
Gen	31	Invalid
BC	31	Valid
SC	36	Invalid
PH	36	Valid
ST	34	Valid

EX: 2

Interest rate for fixed deposit in a banking system is as follows

1-2 years 7%

2-5 years 8%

Above 5 years 10%

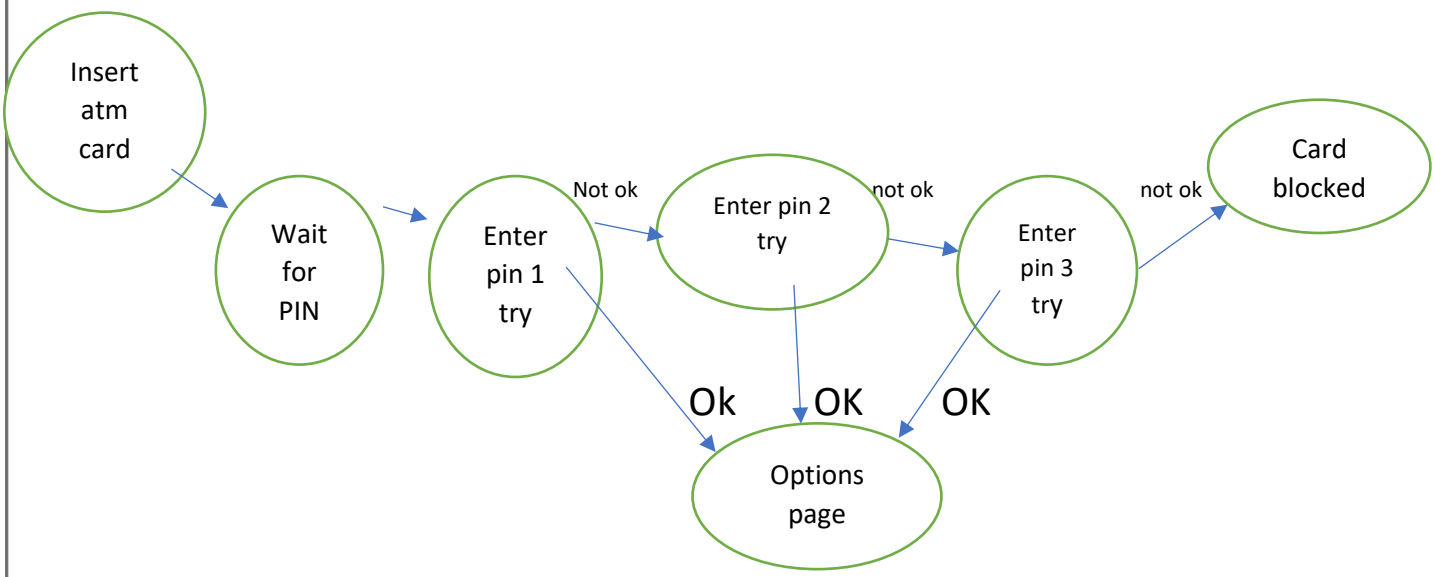
For senior citizens 0.5% extra

Age	Period	Rate of interest	Valid/invalid
25	4	8.5%	Invalid
70	4	8.5%	Valid
65	1	7%	Invalid
59	3	8.5%	Invalid

d. State transition testing:

When we are providing same input again and again repeatedly and if the state of the application is changing from one form to other form then we use this state transition testing technique.

EX: ATM PIN verification.



Software testing

e. Use case testing:

When there is a serious of interaction between the user and the system then we use this use case testing techniques. In this method we have main stream scenarios as well as alternative scenarios.

EX: ATM cash withdraw process.

Main stream scenarios:

1. USER: Insert ATM Card
   SYSTEM: Validates and ask for PIN
2. USER: Enter PIN
   SYSTEM: Validates PIN and ask to select language
3. USER: Select language
   SYSTEM: Validates and ask to select amount type
4. USER: Select account type
   SYSTEM: Validates & ask to enter amount
5. USER: Enter amount
   SYSTEM: Validates and release money

Alternative scenarios:

1. Suppose if user enters invalid pin
   System: Shows error message & ask to enter Correct pin
   USER: Enters correct pin
2. If user select invalid account type
   SYSTEM: Shows error message and ask to select correct account type
   USER: Select correct account type
3. If user enters incorrect amount (more than balance / more than daily limit)
   SYSTEM: Shows related message & again ask to enter correct amount
   USER: Enter correct amount

Software testing

III. **Experienced based testing:**

The test will be conducted by experienced test engineers

The following are the techniques that will be used in this method.

a. Error guessing testing
b. Exploratory testing

a. Error guessing testing:

In this technique with his experience the tester can guess the error and perform the testing on that part of the software.

b. Exploratory testing:

In the traditional testing method at first in the test design phase all the test cases are designed and after that in the test execution phase each test execution will be done. Where as in case of exploratory testing we don't prepare any test cases initially at the time of testing we will explore the functionality to be tested and spontaneously prepare the test case and immediately perform the test execution.

NOTE:

White box testing is also called as "**Glass box testing"** or "**clear box testing"**

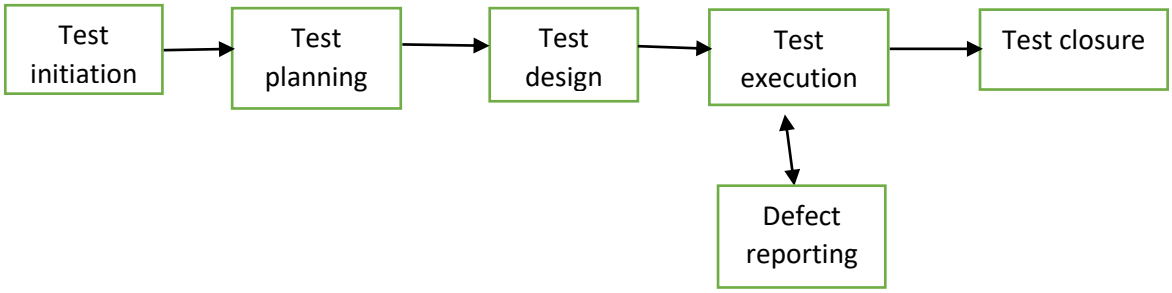
STLC:

It stands for ***Software test life cycle*****.**

this is the standard process followed by the testing department to conduct software testing.

It contains different phases and each phase is conducted by a responsible person by taking some input and gets an output in the form a document. It means in every phase the responsible person will prepare a specific document. The following are the different phases available in STLC.

Software testing



PM/PR	TL/SM	Test case	defect	TL/SM
Test Strategy doc	test plan doc	document	report	test summary doc

1.Test initiation:

In this phase the **project manager or product owner** will study the client requirements and define an approach to be followed to conduct the software testing in the form of **a test initiation or test strategy** document.

2.Test planning:

This phase will be conducted by the **scrum master or test lead**. Based on the test strategy document the scrum master or test lead will plan the test process in the form a **test plan document**. This document contains information about **what to test, who to test, how to test, when to test.**

3.Test design:

This is the most importance phase of STLC process. It is conducted by tester.

In this phase the tester will understand user requirements and they will prepare test case document by describing the tests to be followed to test the functionality.

Test case template:

	A	B	C	D	E	F	G	H	I	J	K
1	project name:										
2	module name:										
3	prepared by:										
4	designation:										
5	preparation date:										
6	reviewed by:										
7	reviewed date:										
8											
9	test case id	test scenario	test case description	severity	pre condition	test procedure					
10						step no	step description	test data	expected result	actual result	result
11											
12											

Software testing

[Example]

Prepared By	Narendar Reddy
Designation	Tester
Prepared Date	18-09-2021
Review By	
Review By	

Test case ID	Test scenario	TestCase Desciption	Priority/Severity	Test Setup	Test Procedure		Test Case/ Test date	Expected Result	Actual Result	Result
					Step No	Step desciption				
Login_01	Verifying Login Process	Checking with vaild username and vaild password	High Severity	IFMIS page is ready	1	Enter vaild usename	25007777777	Login page should be displyed		PASS
					2	Enter vaild Password	123456			

Note:

1. Test scenario is the feature or functionality to be tested.
2. Test case is the different conditions we are using to test the functionality.
3. The severity will describe the seriousness of the defect with respect to the tester. It contains values like High, Medium, Low.
4. The precondition specifies what is necessary to conduct the test.

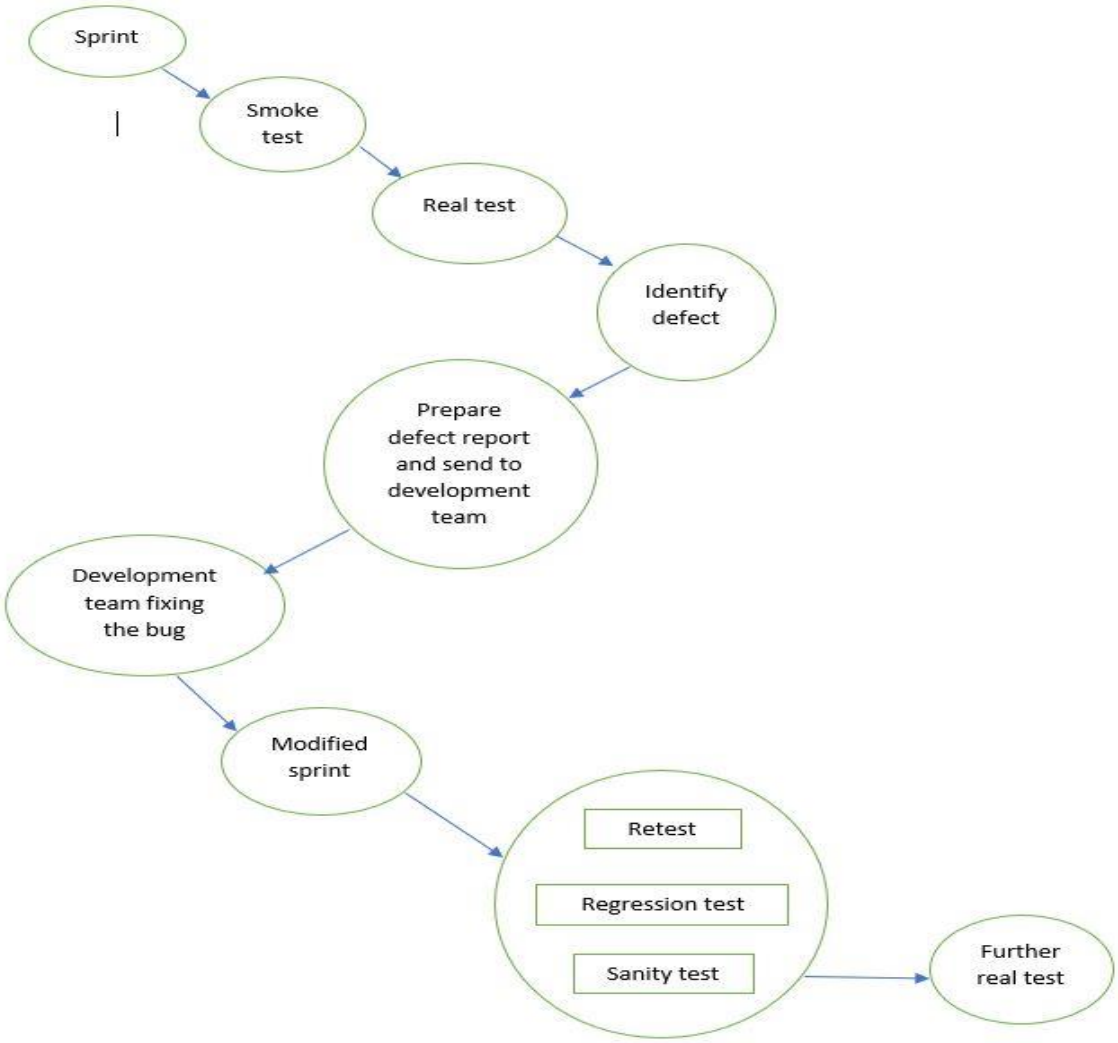
4.Test execution:

This phase is conducted by tester.

After preparing test case document for all the functionalities the tester has to take one by one test case and perform the test execution by following the steps in the test case. After the test execution he will fill the actual result field in the test case.

The test execution will be done with the help of different test in the following way.

Software testing



1. On the given sprint at first smoking test will be conducted.
2. Once the smoke test is passed then remaining tests one by one will be executed.
3. When the test execution is failed we have to identify the defect.
4. A defect report will be prepared and send to the dev team.
5. The dev team will fix the bug and modified software is send back to the tester.
6. On the modified software we have to perform confirmation testing such as **Retesting, Regression testing or sanity test.**

   Smoke testing:

   Testing the important functionalities out of all functionalities is called as smoke testing.

   Real testing:

   Testing the remaining functionalities after the smoke testing is called as Real testing.

Software testing

RE testing:

Testing the fixed functionalities is called as Re testing.

While testing a functionality if it is failed we will prepare a defect report and send to the dev team. Once the dev team fixed the defect (BUG) it is our responsibility to ensure that whether the bug is fixed or not.

Regression testing:

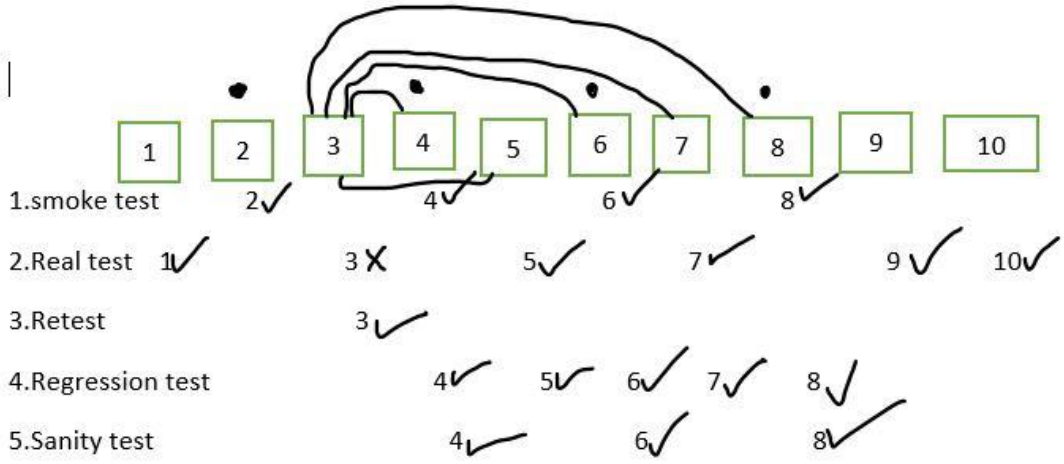
Testing the dependent functionalities based on the fixed functionality is called as regression testing.

This test is performed to identify any side effects on the dependent functionalities after the build modification.

Sanity testing:

Testing important dependent functionalities out of all dependent functionalities is called as Sanity testing.

When there is no sufficient time to conduct the regression testing then we perform this sanity testing.



5. **Defect reporting:**

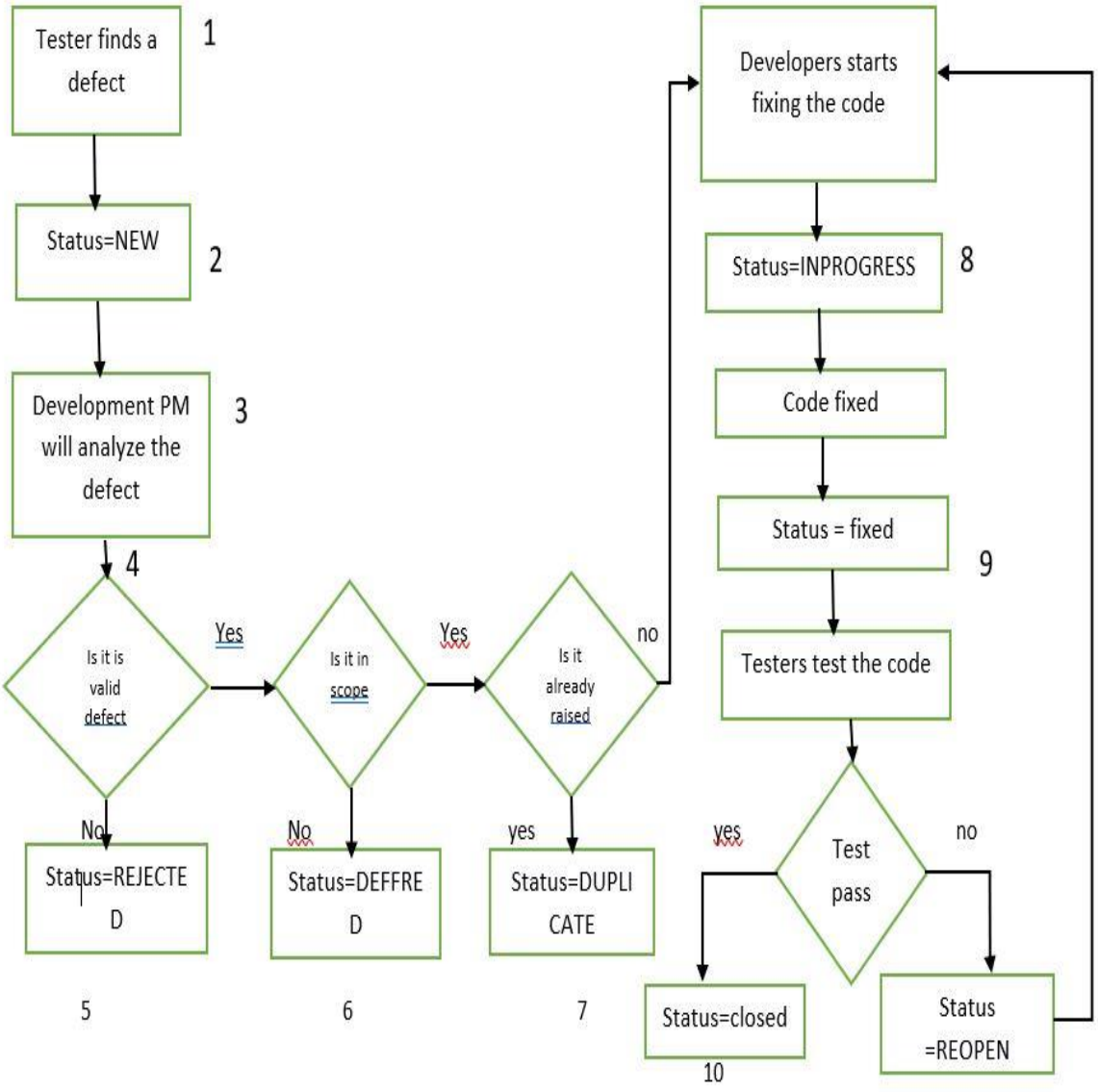
   After the test execution phase the tester observe the results and fills the actual result field in the test case document.

   When the **expected result is equal to the actual resu**lt then the result field will be made as **pass otherwise fail.**

Defect life cycle (OR) Bug life cycle:

The following are the different stages of a defect starting from the identification up to closing the defect.

Software testing



6. **Test closure:**

   In this phase we have to decide when we have to stop the testing.
   This phase is conducted by scrum master or test lead. He will prepare a test summary report or test traceability matrix by summarizing all the tests conducted and the result.
   In general after the following activities the testing will be stopped.

   i. When the estimated time is exceeded.
   ii. When all the selected test cases are successfully working.
   iii. When all the identify defects are fixed.