

* Longforms

* DEVOP	Development and operations
* BA	Business Analyst
* BRS DOCUMENT	BUSINESS REQUIREMENT SPECIFICATION
* SRS DOCUMENT	System Requirement Specification
* FRS DOCUMENT	Functional Requirement Specification
* FFD	Functional Flow Diagram
* FR	Functional Requirement
* HLD	High Level Design
* LLD	Low Level Design
* SQA	Software Quality Assurance
* SDLC	Software Development Life Cycle
* WBT	White Box Testing
* BBT	Black Box Testing
* GBT	Gray Box Testing
* FED	Front End Developer
* BED	Back End Developer
* DBA	Database Administrator
* TRM	Test Responsibility Matrix
* RTM	Requirement Traceability Matrix

Q. What is software testing?

→ To check the correctness and completeness of the functionalities of software application with respect to clients requirement / user specifications is called as software testing.

* SQA (Software Quality Assurance)

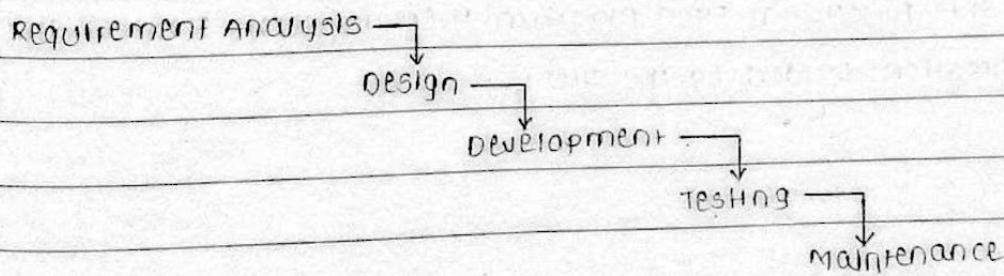
- SQA is a set of procedures that should be followed while developing product or services to assure of good quality to the customers.
- There is a process to follow the software quality assurance they are as follows :-

FACTORS :-

- ① To meet the customers requirement :- They will check what domain and features client want.
- ② To meet the customers expectation :- privacy and performance
- ③ Costing of project :- They will decide the cost of project.
- ④ Time period :- They will decide the time of project.
- ⑤ Escalation / penalty :- If the time limit exceeds then company have to pay money to client.
- ⑥ Maintenance :- They will provide all types of services after deploying the project.

* SDLC Model (Software Development Life Cycle)

- SDLC is a process which includes various phases that are followed to develop the software application.



① Requirements / Information gathering (done by BA)

- BA is responsible person for information gathering.
- BA will collect all the business requirement from the client.
- BA will prepare one document is called as BRS (Business requirement specification).
- BRS is having overview of project.

② Analysis (done by BA & Domain Expert)

- In this phase BA will prepare one document and that document is known as SRS (System requirement specification) with the help of BRS.
- SRS is also known as FRS (Functional requirement specification)
- SRS / FRS is having a detailed information of project.
- SRS / FRS is having some stages to be follow that are known as factors

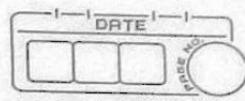
③ Functional Requirements :-

- In this requirement the detailed description of each functionality is given / decided by the client.

For e.g. :- In password functionality for creating / generating password following requirements will be mandatory. (like one upper letter capital letter, small letters, numbers, special characters, combination of characters, numbers & special characters, password size 8 digit to 16 digits)

④ Functional flow Diagram :-

- In this functional flow diagram there is detail flow of the features or application decided by the client.



for e.g. :-

sign up into the application

↓
Display product search option on home page

↓
select product

↓
Add to Cart product

↓
Buy now product

↓
Select Address

↓
Payment option

↓
Order placed.

④ Screenshots / Snapshot :-

- snapshots are prepared by BA with the help of IRISE tool.
- It is just visualization of the functionalities of the application.
- These snapshots are in static mode.
- snapshots are required for taking the approval from the client side to start the development.

⑤ Design (Done by Architecture and UI developer)

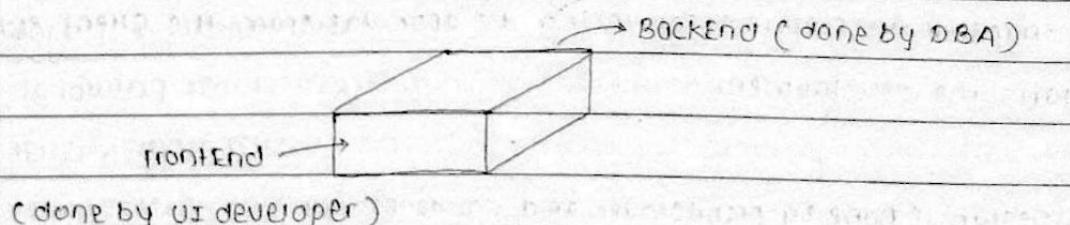
- In this application design is prepared with the help of SRS.
- Main module design by the Architecture
- Sub module design by the UI developer.
- Based on the snapshots UI developers develop the UI of the application.
- There are two types of design as follows :-
 - a) High Level Design [HLD] - done by Architecture
 - b) Low Level Design [LLD] - done by UI developers.

Q. Difference between High Level Design & Low Level Design.

→ HIGH LEVEL DESIGN [HLD]	LOW LEVEL DESIGN [LLD]
* High level design contains design of working of main module.	* Low level design contains static logic of every sub module.
* HLD includes relation and dependency of main module.	* LLD is developed according to main module.
* HLD is developed by architecture developer.	* LLD is developed by UI developers.

④ Development / Coding (done by developer)

- In this phase of development, coding is done by developer.
- Developers built the functionalities of the application with the help of programs.
- programs is a set of instruction to perform specific task for the user.



- In this phase of development there are two types of developers working simultaneously front end developers and back end developers.
- The development done with respect to SRS document.

(5) TESTING (Done by Tester)

- TO CHECK THE CORRECTNESS AND COMPLETENESS OF SOFTWARE WITH RESPECT TO SRS.
- TESTER SHOULD ENSURE THAT APPLICATION IS DEFECT FREE.
- TESTER WILL FOCUS ON POSITIVE AND NEGATIVE SCENARIOS.
- TESTER WILL FOCUS ON THE QUALITY OF PRODUCT, PERFORMANCE AND USER INTERFACE.
- THERE ARE THREE TYPES OF TESTING

(i) WHITE BOX TESTING (WBT) (Done by Developers)

- WHITE BOX TESTING IS DONE BY THE DEVELOPERS.
- WHITE BOX TESTING IS ALSO CALLED AS UNIT TESTING.
- CODING KNOWLEDGE IS REQUIRED FOR CONDUCTING WHITE BOX TESTING.
- THEY WILL FOCUS ON POSITIVE SCENARIOS ONLY.

(ii) BLACK BOX TESTING (BBT) (Done by Tester)

- TESTER WILL PERFORM BLACK BOX TESTING WITH RESPECT TO THE USER REQUIREMENTS.
- IN THIS TESTING CODING KNOWLEDGE IS NOT REQUIRED.
- THIS IS ALSO CALLED AS SYSTEM AND FUNCTIONAL TESTING.
- BLACK BOX TESTING IS DONE BY TESTER.
- IN WHICH WE WILL FOCUS ON POSITIVE AS WELL AS NEGATIVE SCENARIOS.

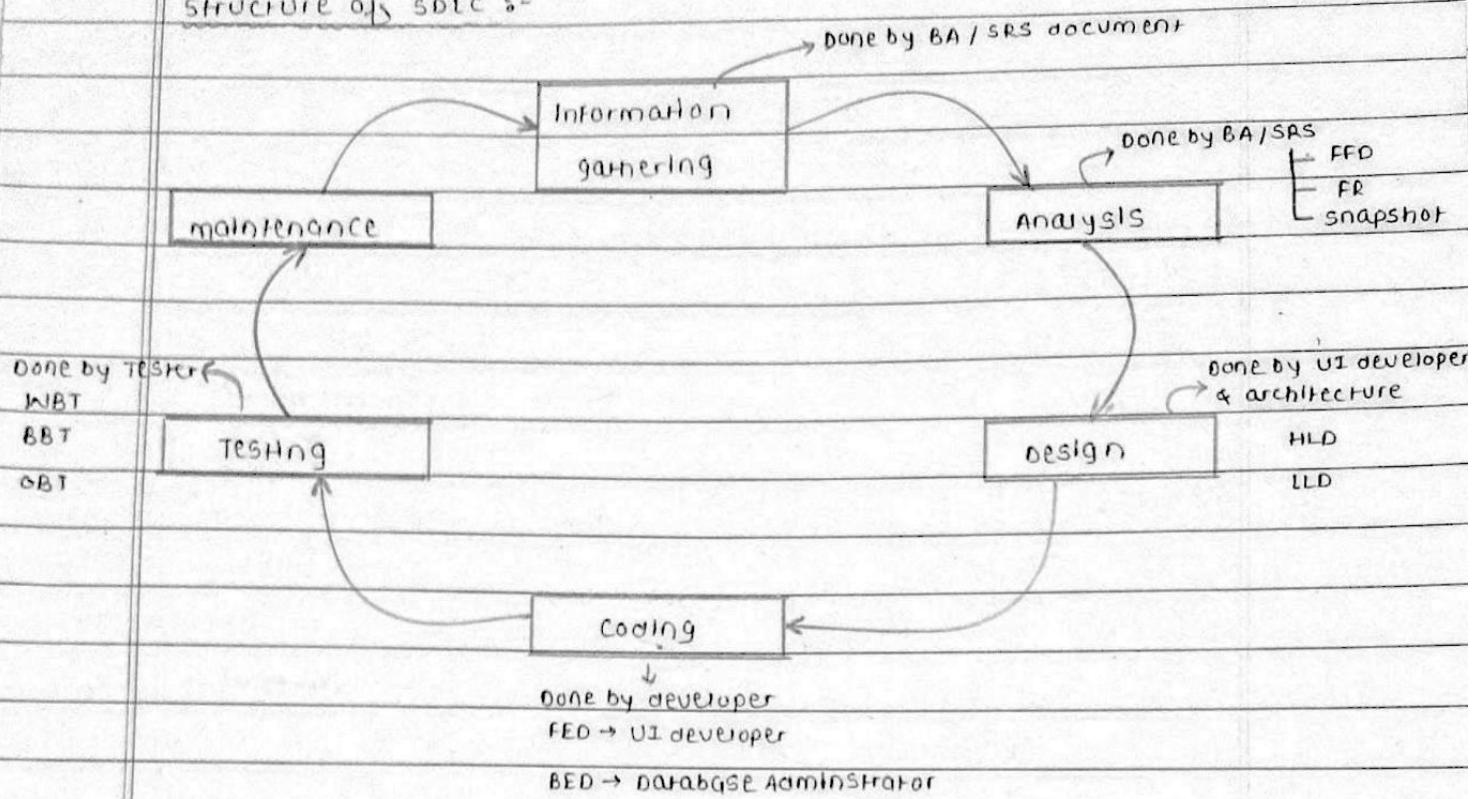
(iii) GRAY BOX TESTING (GBT) (Done by Tester) (Devops)

- IT IS A COMBINATION OF WHITE BOX AND BLACK BOX TESTING.
- IN THIS TESTING CODING KNOWLEDGE IS REQUIRED.
- THEY WILL FOCUS ON POSITIVE AND NEGATIVE SCENARIOS.

(6) MAINTENANCE

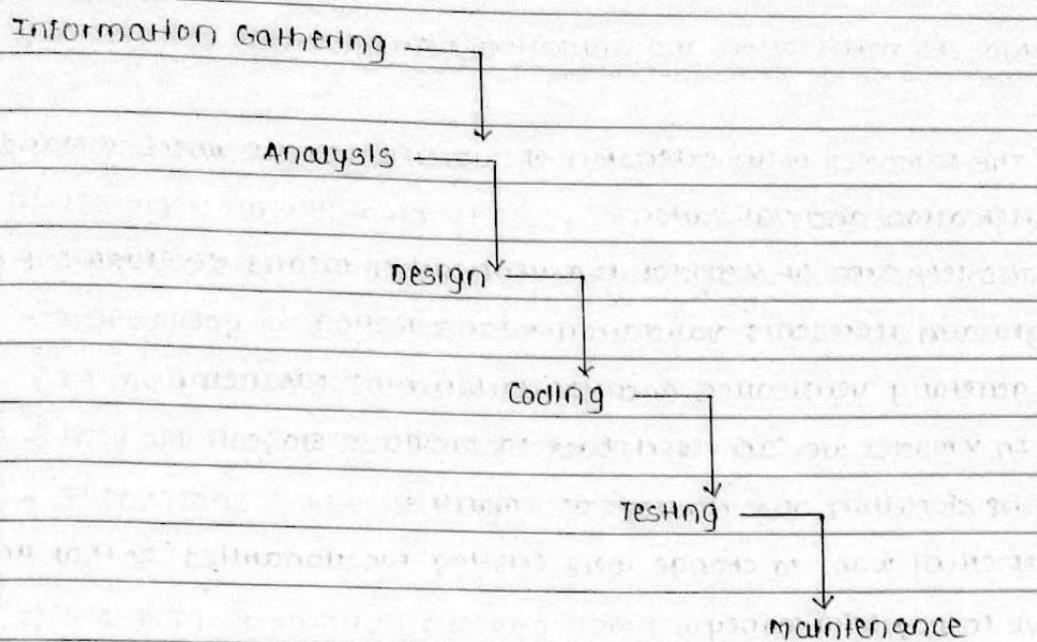
- IN THIS PHASE COMPANY WILL PROVIDE ALL KIND OF SERVICES AFTER DEPLOYING THE PROJECT.
- IT INVOLVES TECHNICAL AS WELL AS NON-TECHNICAL SERVICES (SUPPORT).

STRUCTURE OF SDLC :-



* Waterfall Model :-

- Waterfall Model is a step by step implementation of SDLC.
- In waterfall model without completion of first stage you can't proceed to the next stage.
- This model is generally used in product base organization.
- General time duration for waterfall model is of 3 months.
- We can't revert back to the previous stage.
- Waterfall model is a planned driven module.
- If testers found any bug in any stage then that bug will get fixed in next cycle or in next release.



Q. Difference between BLACK BOX TESTING and WHITE BOX TESTING.

Black Box Testing	White Box Testing
① The process of giving input to the system and checking the output of the system, without knowing about how the system generates the output is called as black box testing.	① It is a software testing method in which the internal structure / design and the implementation of the item being tested is known to the tester.
② This type of testing is carried out by testers.	② This type of testing is carried out by software developers.
③ Implementation knowledge is not required to perform black box testing.	③ Implementation knowledge is required to perform white box testing.
④ We can check positive and negative scenarios.	④ Here developers checks only positive scenarios.

* V-Model (verification and validation) :-

- THE V-MODEL IS AN EXTENSION OF WATERFALL MODEL WHERE V STANDS FOR VERIFICATION AND VALIDATION.
- THE LEFT ARM OF V MODEL IS A VERIFICATION MEANS DEVELOPMENT AND RIGHT ARM REPRESENTS VALIDATION MEANS TESTING.
- GENERALLY VERIFICATION AND VALIDATION RUNS PARALLELLY.
- IN V MODEL WE CAN REVERT BACK TO PREVIOUS STAGE.
- THE DURATION OF V-MODEL IS OF 3 MONTHS.
- IF CLIENT WANT TO CHANGE THERE EXISTING FUNCTIONALITIES AT THAT TIME CLIENT HAVE TO PAY EXTRA MONEY.

Verification

(Development)

Validation

(Testing)

Information gathering				1) ASSESSMENT OF DEV. PHASE 2) TEST PLAN PREPARATION 3) REQUIREMENT OF PHASE TESTING
Analysis				
Design				1) DESIGN PHASE TEST 2) PROGRAM PHASE TEST 3) DESIGN TEST CASE
Coding				
Integration				1) SANITY TESTING 2) SYSTEM & FUNCTIONAL 3) USER ACCEPTANCE TESTING 4) DOCUMENTATION
Maintenance				1) DRC 2) RFC (REQUIREMENT FOR CHANGE) 3) REGRESSION TESTING 4) POST MORTMEN TESTNG

1) ASSESSMENT OF DEVELOPMENT PHASE (PROJECT MANAGER AND TEAM LEAD)

- Project manager and team lead are involved in this phase.
- Testing team will make strategy for testing the software / Application.
- They will decide which methodology is going to be used in our testing.
- Methodology means which type of testing is going to be used in project.
(e.g. Automation, manual or API testing)
- They will finalize TRM (Test Responsibility Matrix).
- TRM means "who is going to perform?" and "what to perform?"

2) TEST PLAN PREPARATION (TESTING TEAM LEAD)

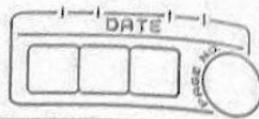
- TRM is implemented in this phase.
- Project manager is responsible for TRM implementation.
- Test plan will be prepared with the help of project manager.
- Project manager will prepare the testing team and allocate work to them.
- In this phase test estimation is created, it means how much time we require for complete test.

3) REQUIREMENT OF PHASE TESTING

- In this phase whatever test estimation is created, it will be finalized in this phase.
- How many testing cycle is needed in this project will be decided.
- Time period for particular cycle is decided.

e.g :-

JIO is continuously introducing new features / Applications like Jiotv, Jiomoney, Jopay and many more, so these modules are like new features so for that requirements get estimated.



2) Design and Coding

1) Design phase testing

2) program phase testing

3) Design test cases

1) Design phase testing (Developers are involved)

- In this design phase testing code level testing is done by developers
- The code testing is done on every module of the application.
- Code testing is also called as unit testing.
- This phase is also known as white box testing.

2) Program phase testing (Developers are involved)

- Program phase testing is also called as code level or unit level testing.
- Whatever the code is written with respect to the requirements that will be tested by the developers.
- In this phase testing starts from small unit of programs.
- Developers test the application with all the positive scenarios.

3) Design Test Cases (Tester)

- Test Case design are done by the testers
- For test case design we need SRS document in this phase.
- Test case design done with the help of scenarios.
- It is divided into two different parts.
① positive test cases ② negative test cases.

For e.g :- Suppose there is SBI login page and in that there is a password field.

① Positive test case :- The password should accept 2 capital letters, small letters, special character and digits, maximum limit of 8 characters, then tester design test case and entered password like "vinit@@2"

② Negative test case :- Testing with negative scenario means invalid data like "vinit2".

3) Integration

- 1) sanity testing
- 2) system and functional testing
- 3) user acceptance testing
- 4) test documentation

3) Integration (Build Information) (done by Developers)

- TO confirm that the individually tested modules can work together to deliver the intended functionality.
- IF we want to add any new module or functionality in old/existing system then at that time we perform integration.

1) Sanity Testing (Done by Tester)

- Tester is responsible for sanity testing.
- sanity testing is the process of validation of V-Model
- sanity testing is done for checking core functionality testing.
- In sanity testing only high risk / critical errors are get documented.
- After the build installation firstly sanity testing done.
- Without build installation we are not able to do sanity testing.

2) System and Functional testing (BBT, Done by Tester)

- BLACK BOX Tester are responsible for this testing.
- system and functional testing is done after the completion of sanity testing.
- In this phase tester checks the functionalities of the application as per the SRS document.
- Testers test the application with respect to positive and negative scenarios.
- Defects are documented in this phase.
- In this testing product gets validated.

3) User Acceptance Testing (Done by Tester & users)

- user acceptance testing is done after the completion of system and functional testing.

- AFTER deploying the product to the user environment , tester and users performs UAT testing .
- product has UI , design , colour , functionalities as per user requirements .
- when user validate it as correct then product send to the final production .

4) TEST DOCUMENTATION

- TEST Documentation is documentation of artifacts created before or during the testing of software . It helps the testing team to estimate testing effort needed , test coverage , resource tracking , execution progress , etc .

4) Maintenance

- 1) DEFECT REMOVAL EFFICIENCY (DRE)
- 2) REQUIREMENT FOR CHANGE (RFC)
- 3) REGRESSION TESTING
- 4) POST MORTUM TESTING

1) DEFECT REMOVAL EFFICIENCY (DRE)

- DEFECT REMOVAL EFFICIENCY measures how well the development and testing team does at removing defects before they enter production .
- There are two phases in DRE

- ① DEFECT FOUND BY TESTER
- ② DEFECT FOUND DURING USER ACCEPTANCE TESTING .

① When tester found defects , some defects get solved and some get cancelled so only solved defects get into the consideration .

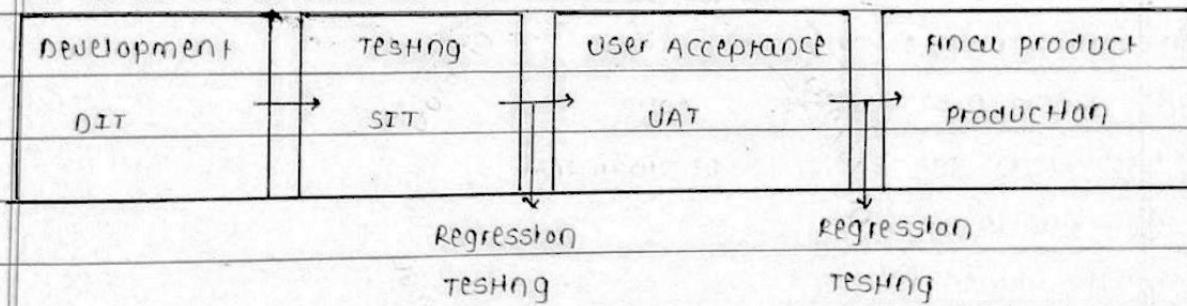
② In user acceptance testing , whenever customer test system in test environment , he could find defects then defects take into the consideration .

2) Requirement for change (RFC)

- IF customer want changes in his product at the time of release then customer is able to send the request for change.
- These changes are mentioned in the SRS document at the end of the document there is a special section for change request.
- Customer have to pay extra money for the change request.

3) Regression testing

- AS defects are fixed or new functionalities are added into the application, it becomes necessary to check that there is no impact of these changes on the previously working functionality.
- It is carried out to determine whether the changed component has affected the functionality of the unchanged component.
- This basic functionality testing is called as regression testing.
- AS we know there are four test environment



4) Post mortem testing

- In post mortem testing they will check all the functionality of the application.
- It is done by the white box tester.
- After completion of testing, product is ready for production. In the meantime product does not produce expected output then white box tester should have to check all the modules in details.

Advantages

- simple and easy to use
- works well for small projects where requirements are easily understood.
- testing activities like planning , test designing happen before coding.
- this model saves a lot of time.

Disadvantages

- very rigid and least flexible
- if any changes happen in midway , then the test documents along with requirement documents has to be updated.



Agile Methodology

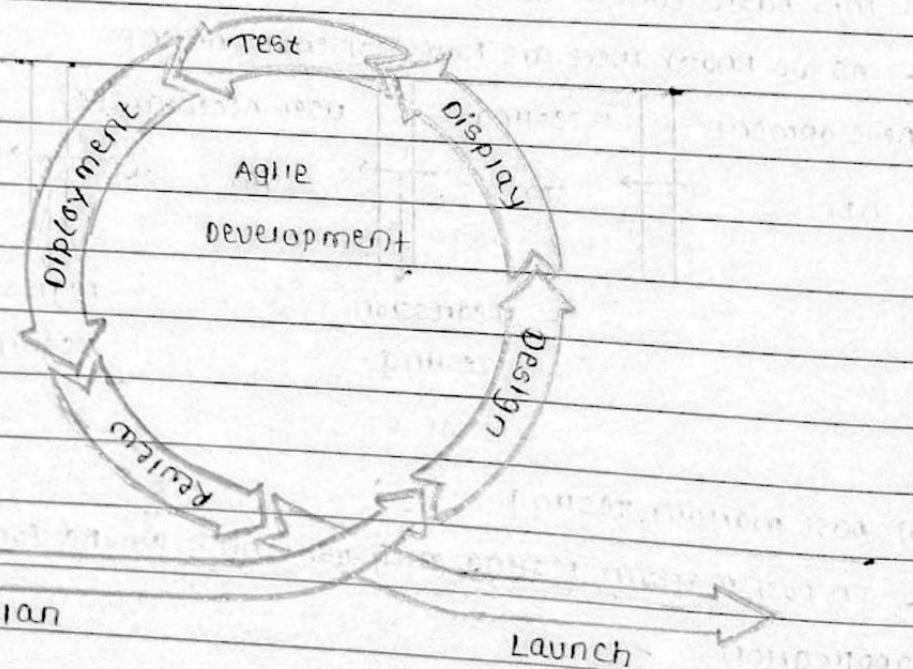


Diagram: Scrum Agile development



so basically agile is used to implement SDLC model. As we know, we have three different modules to implement the SDLC like waterfall model, V-Model and agile model. but we have seen some drawbacks in waterfall model and V-model so that's the reason agile is introduced.

Agile is having four different types those are Scrum agile, Kanban, DSDM and Crystal. Most of the companies preferred Scrum agile because it provides the more facilities. Scrum agile follows sprint wise release and the duration of sprint is 3 weeks. Sprint is also known as cycle. In agile, Anytime change request is accepted at any stage of development and that change request cannot affect the other modules, for that change request client don't have to pay extra money to the company. Agile is user driven methodology, it is not a planned driven because in agile we have set some milestones and we planned for it in a sprint.

Architecture of Agile First one is Stake Holders

1. Stake Holder is a main member of the top most body. Stake Holder is also called as customer or client. Stake Holder gives a bunch of requirements to the project owner regarding the project which he wants to a develop, also he can make a request change at any stage. After getting all the requirements from the stake holder he will prepare one document that document is nothing but a project backlog.

2. So project backlog means all the requirements regarding project is present in this document and this document is created by project owner. So once project backlog is done Estimation is takes place.

4. Estimation is a sprint planning meeting duration of this meeting is generally 3-4 hours. Scrum master and project owner are the host of this meeting. The basic purpose of this meeting is to sort out the requirements among the all requirements. We have to sort out requirements which we have to needed for developing a module. This requirements are also called as user stories and this user stories will be explained by the project owner, means what exactly we have to do. This things are discussed in this meeting.

Scrum master allocates these user stories to us on the basis of three factors knowledge, efforts and complexity.

1. Knowledge means each & every member of the team should have a knowledge of the project domain.

2. Efforts means they will decide how much efforts will required to test that particular application.

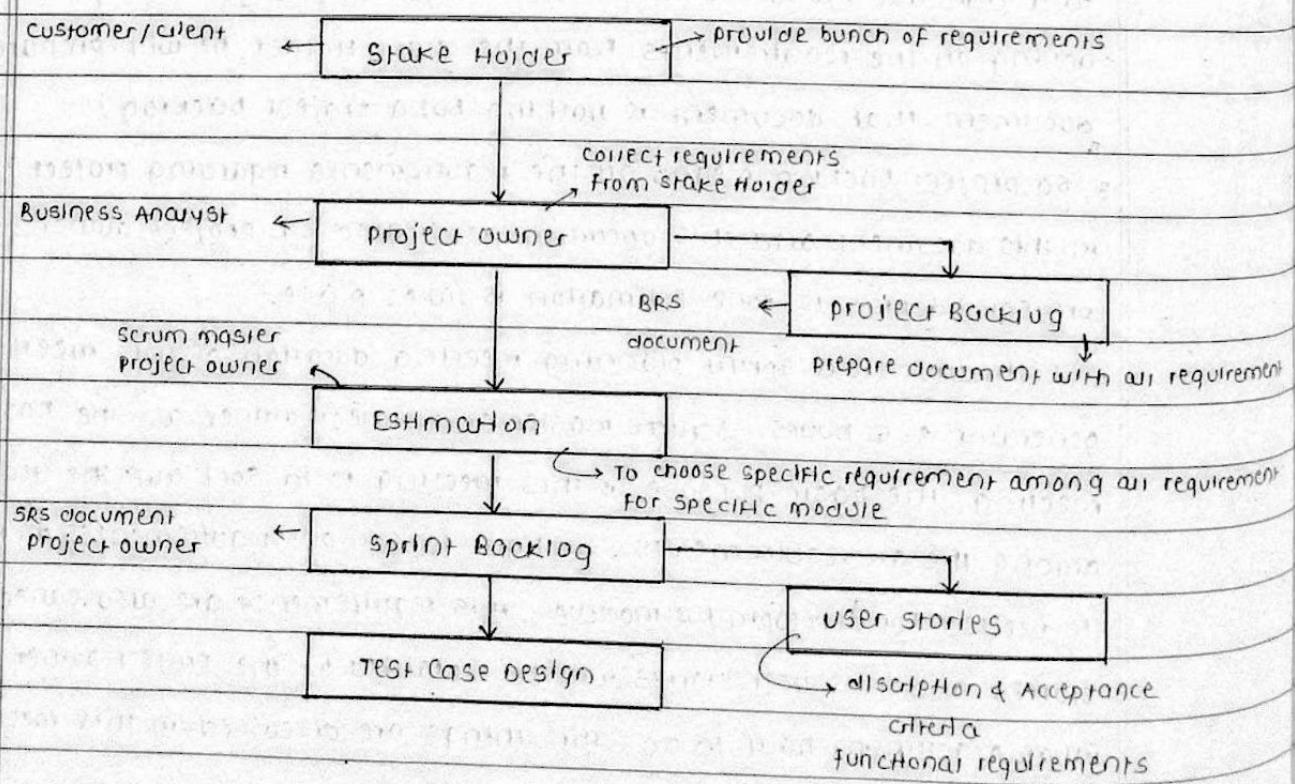
3. Complexity means how much time, cost, resources and efforts will be required in sprint.

4. Whatever user stories are sorted here these are present in sprint backlog.

Sprint backlog is a collection of user stories and it contains detailed information of requirements which are required for the development of module. and sprint backlog is created by the project owner.

5. User stories basically it contains three factor those are description, acceptance criteria and snapshots. User stories is the functional requirement of the module. each & every member of the team should get tasked through the user stories.

6. Once you will get a user stories then we can starts test cases means we can design test cases according to the user stories. This is a architecture of agile.



Agile is having some advantages those are as follows.

1. Daily standup call / scrum meeting
2. Checkpoints
3. Implementation of automation
4. Sprint wise delivery

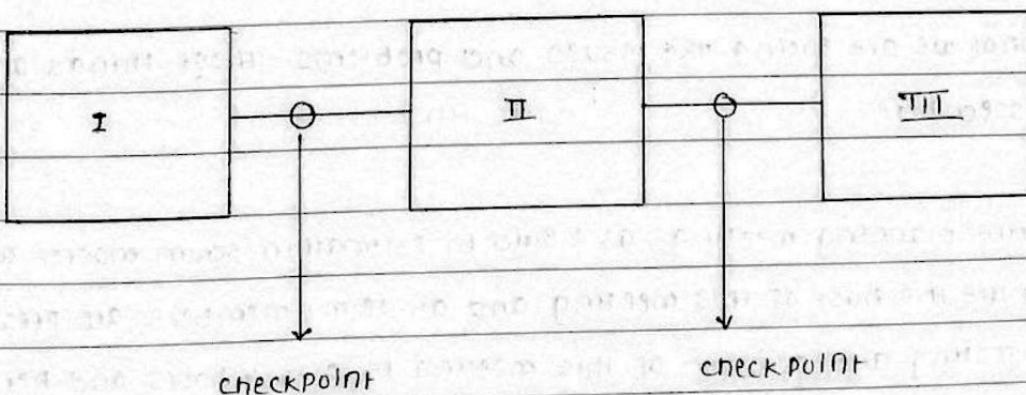
So, let's discuss one by one.

First one is daily standup call / scrum meeting. Duration of this meeting is generally 15 min. and scrum master is a host of this meeting, all team members are present in this meeting and basic purpose of this meeting is to check the status of the project and mainly focuses on 3 questions in this meeting.

1. What we did yesterday?
2. What we are going to do today?
3. What are the issues we are facing?

This is what we are discussing in this meeting.

Second advantage of agile is checkpoints suppose any bug or any problem occurred at the time of production then we have to check a checkpoints with the help of AVAS tool. So basically I have three modules between the two module there is a checkpoint. I will check the first checkpoint if it is okay then I will check the second checkpoint. If we found any bug in that then we can revert back to the developers, so this is a advantage of Agile.



Third advantage of agile is implementation of automation. In the market 70% is manual testing and 30% is automation testing. So in agile we can also implement automation. Automation requires low cost, less resources and giving high accuracy. On the basis of this we can implement automation in agile.

Fourth advantage of agile is sprint wise delivery means in agile we follows sprints. On the basis of sprints we can able to release one or two modules so this is the advantage of agile.

Agile is having some ceremonies those are as follows

1. Daily standup call / scrum meeting
2. Sprint planning meeting
3. Sprint Review meeting
4. Retrospective meeting

So, let's discuss one by one.

1. Daily standup meeting / scrum meeting; as I said already, the host of this meeting is scrum master and duration of this meeting is 15 minutes. The basic purpose of this meeting is to check the status of the project, and mainly focuses on the 3 questions

1. What we did yesterday?
2. What we are going to do today?
3. What we are facing like issues and problems, these things are discussed here.

2. Sprint planning meeting, as I said in estimation scrum master and project owner are the host of this meeting and our team members are present during this meeting and duration of this meeting is 3 to 4 hours and here scrum master allocates user stories to the each & every team member based on the three factors like knowledge, efforts and complexity.

3. Sprint Review meeting , host of this meeting is scrum master and all team members are present here and client also present in this meeting . duration of this meeting is 30 min and main purpose of this meeting is give a demo on whatever work we have done in a particular sprint these things are discussed here and we give the accomplished of our project and client gives immediate feedback to us in this sprint review meeting .

4. Retrospective meeting this is the last ceremony of agile . scrum master is a host of this meeting and duration of this meeting is 60 min. all the members of the team are present here , the purpose of this meeting is giving a feedback to our team members and also we can say that they can give us a feedback and also discussed on what bugs we have successfully resolved , what amount of work is done and how much work is remains and what are the problems for that , what is the creative solution for that , these things are happened in the retrospective meetings . this is all about the agile .

* BRIEF OF MANUAL TESTING

- 1) Requirement Analysis
- 2) Test plan creation
- 3) Test case creation
- 4) Test Case Execution
- 5) Defect logging
- 6) Defect fix and re-verification

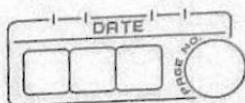
* Sanity Testing

- sanity testing is conducted after receiving the build from development team before starts the black box testing.
- sanity testing is also called as zero level testing or tester acceptance testing and also called as build verification testing.
- In sanity testing we will check core functionalities but here we don't log any defect except showstopper or blocker.
- the duration of sanity testing is 1 day (one day).
- the main purpose of sanity testing is to check that our build is stable or not.
- showstopper or blocker means 404 error or server not found or page not found or where our testing team gets stuck at that time we are not able to work further that time will log that showstopper or blocker (defect) under the sanity testing.
- we will test the random functionalities under the sanity testing.
- It's done after a test cases creation.
- whenever we are performing sanity testing we will focus on some factors like core functionalities, tab validation, link validation, page validation and GUI validation.

1) Core functionalities

- In core functionalities we will check the icons, buttons, check boxes are working or not.

For Example :- If there is close button or submit button, then we will check that buttons by clicking on them suppose, if I click on close button at that time it should get closed. or if I click on submit button, whatever data is present there it should get submitted. If in any case these functions are not behaving as we expected at that we will log that defect under the sanity testing.



2) Tab validation

- we will check the functioning of tab under the tab validation.
- by using tab we will check that we are able to enter the values in the text boxes or not.
- so, this will comes under the tab validation.

3) Link validation

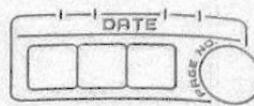
- In the software or applications pages are connected via inter linking and in this validation we have to focus on these pages.

For example & suppose there is a phone pay application, first of all we will see the home page with multiple options. If I want to make payment then I have to click on scan button after clicking on it scanner should gets open, so in that case developer provides links of that pages under the icons that links get validate or not we will check under the link validation.

4) Page validation

- page validation means navigation testing or navigation validation.
- In this process we click on forward or backward arrow or next or previous button at that time page should navigate next or previous this comes under the page validation.

For example & suppose, I am giving an online exam. In that exam there are 40 questions and each question is having different page. I will start my exam from 1st question after solving the 1st question I have to move on 2nd question at that time there are two possible ways are present first one is next button and second one is question number two. After clicking on next button or question number 2 that page should navigate to the second question. If I want to jump on 15th question then I will click on that question number 15 so it



should be redirected to the question number 15 or not. will check here this is comes under the page validation.

5) GUI TESTING

- GUI STANDS FOR GRAPHICAL USER INTERFACE
- IT DETERMINES HOW USER FRIENDLY THE APPLICATION IS FROM LOOK AND FEEL PERSPECTIVE.
- WE WILL CHECKS THE PAGE IS DISPLAYING CORRECTLY OR NOT, IMAGE SHOULD BE CLEARED, COLOURS, ICONS, NO BLUR IMAGES ARE PRESENT IN OUR PAGES AND ETC.
- THIS VALIDATION OF VISUALIZATION IS CALLED AS GRAPHICAL USER INTERFACE TESTING.

* System and functional

System and functional testing is comes under the black box testing. In this testing we will focus on positive and negative scenarios and for conducting this testing there is no need of coding knowledge. There are different types present under the system and functional testing are as follows.

- 1) FUNCTIONAL TESTING
- 2) USABILITY TESTING
- 3) SECURITY TESTING
- 4) PERFORMANCE TESTING

1) FUNCTIONAL TESTING

- FUNCTIONAL TESTING COMES UNDER THE SYSTEM AND FUNCTIONAL TESTING.
- THE MAIN PURPOSE OF FUNCTIONAL TESTING IS TO CHECK THE INTERNAL FUNCTIONALITY DEPENDS UPON EXTERNAL FUNCTIONALITY.
- THERE ARE TWO DIFFERENT TYPES OF FUNCTIONAL TESTING ARE AS FOLLOW

- 1) FUNCTIONAL COVERAGES
- 2) NON-FUNCTIONAL COVERAGES

i) Functional coverages

- Behavioral coverage
- Input domain coverage
- Error handling coverage
- Back end coverage
- Selection level coverage
- Calculation based coverage

These coverages are comes under the functional coverages and functional coverages comes under the functional testing.

a) Behavioral coverage

- In this coverage we test the behaviour and the property of the object.

For example :- Suppose there is an facebook signup page on that page there are multiple objects are present like first name, last name, email id, mobile number. In that objects user wants to add their first name at that time whenever user is clicking on that text box at that time text box should get highlighted this is a behaviour of that text box. property means whatever value user is entering in that text box is known as the property of that object.

b) Input Domain coverage

- In this coverage we will mainly focuses on size of input and type of input.
- For this coverage we have two methods like BVA & ECP.

i) BVA (Boundary Value Analysis)

- In boundary value analysis we will check with maximum and minimum values providing to the criteria of the input.

For example :- Suppose there is password field and it should contain 8 to 15 characters long with alphanumeric values, special symbols, no space this is the criteria of password then we will check this password with minimum and maximum values.
First we will check $\min - 1$, $\min + 1$

$8 - 1$ $8 + 1$

Invalid Valid

$\max - 1$ $\max + 1$

$15 - 1$ $15 + 1$

Valid Invalid

A to Z → Valid

a to z → Valid

0 to 9 → Valid

Space → Invalid

This kind of testing we are conducting in boundary value analysis.

a) ECP (Equivalence class partitioning)

- EQUIVALENCE CLASS PARTITIONING IS USED TO REDUCE THE NUMBER OF TEST DATA TO A MANAGEABLE LEVEL AND MAINTAIN REASONABLE TEST COVERAGE

For Example :- If I want to buy a car phone from any site with limit of 10,000RS. So the valid and invalid test data is 1 to 10,000RS. In between 1 to 10,000RS mobiles will get separated and above 10,000RS will get separated into the other side.

i) 1 to 10,000

ii) Above 10,000

So, this happens under the equivalence class partitioning.



c) Error handling coverage

- In this coverage whenever user entering any input in in-correct way so it will display the correct message to the user.

For Example :- suppose there is a mobile number field and user will enter only 9 digits instead of 10 digits then at that time system will generate the message and that message will be "please enter valid mobile number" for password it is also the same if user enters invalid password then system generates message "please enter valid password" or "password is wrong".

d) Back End coverage

- In this coverage user is able to fetch the information from database or whatever information user enters that information is stored in database or not we will check this under the back end coverage.

e) Selection level coverage

- In this coverage we will check the flow of our application according to the functional flow diagram . Flow always should be in a correct way if flow is going to be disturbed then it will generate unexpected output this we will check under the selection level coverage.

f) Calculation based coverage

- In this coverage we will check that if our application is having any calculation like addition, subtraction, multiplication, division, at that time we perform calculation based coverage.

For Example :- suppose there is a banking application and user wants to credit some amount in account it should display that amount means how much balance he is having in the account . It should be addition of credited amount and previous amount which is already present in his account . So in this way calculation based coverage works.

2) Non-functional coverage

- Non-functional coverages comes under the functional testing and functional testing comes under the system and functional testing
- the main purpose of non-functional testing is to check external functionalities.
- there are different types of testing are present in non-functional coverages

a) Recovery testing / reliability testing

b) Compatibility testing

c) Intersystem testing

d) Globalization testing

e) Garbage testing / sanitation testing

a) Recovery testing

- Recovery testing is also called as reliability testing

- In this testing we will check the software is able to recover or not from abnormal to normal mode

For Example :- If I want to work on any browser in that case

suddenly connection gets interrupted and after some time connectivity comes in so again I will open that browser and check that the previously opened page is recovering or not this comes under the recovery testing.

b) Compatibility testing

- In this compatibility testing we will check that the build is compatible with user expected platform or it is able to support operating system and browsers with user expected platform.

Compatibility testing have two different types

- Software compatibility (Compatibility testing)
- Hardware compatibility (Configuration testing)

For conducting the compatibility we have two types of testing

- Forward compatibility testing
- Backward compatibility testing

1) Forward compatibility testing

- If build is correct but we are having problem with OS / browser
Then it is forward compatibility testing

For Example :- there is a game with size of 2GB. If we are installing this game on a device whose having 4 or 6 GB RAM then it will work without any problem but at the same time we install that game on device whose having 2 or 3 GB RAM that game will get install but it will not run on that device because operating system is not compatible

2) Backward compatibility testing

- If OS / browser is okay but build do not work properly then it is backward compatibility testing

- There are two compatibilities in compatibility testing

- a) OS compatibility testing
- b) Browser compatibility testing

a) OS compatibility testing

- In this OS compatibility testing we will check our build on different types of operating system to check that software is compatible with different OS or not.

b) Browser compatibility testing

- In this testing we have to check that software is compatible with different browsers or not.
- Generally we testers are involved in browser compatibility testing.
- In this testing it includes two subtypes:
 - 1) Cross browser compatibility testing
 - 2) Version browser compatibility testing

1) Cross browser

- In this testing we test the build on different browsers like chrome, firefox, internet explorer etc.

2) Version browser

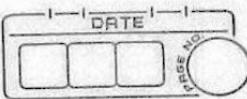
- In this testing we test the build on different, different versions of some browser.

For example at chrome 80, 81, 82, 83, 84, 85, 85.5, 85.6 etc.

c) Intersystem testing

- In this testing we will check our application is able to share the information with other application.
- the data communication between two application will happen with "XML".

For Example - Suppose dominos in that application there is a google map after ordering. Someone comes to deliver that order but in the meantime dominos will share the location of that delivery person to us by using google map in the same application. It means dominos load some data to the google map so we can able to see where is our delivery or we can say that we can able to track our order. This testing comes under the intersystem testing.



d) Globalization testing

- In that testing we have to check that our software / application is supporting to the many languages or not.
- there are three different types are present

1) Local

2) Global

3) International

1) Local

- Localization testing is conducted to check that for particular culture / language.

For Example :- Marathi, Telugu, Tamil, Bengali, Punjabi

2) Global

- Global means it should support English language.

3) International

- International language means to check that our application should support all the international languages.

For Example :- French, Germany, Chinese, Japanese, Hindi

e) Garbage Testing / sanitation testing

- In this testing we have to check extra features means suppose developer wants to add some additional features which is not mentioned in the requirements so in that case we will perform garbage testing.

2) Usability Testing

- It is a part of BBT (BLOCK BOX TESTING)
- In usability testing we will check user friendliness of software.
- Test scenarios for usability testing are prepared by project owner.
- Testers will execute those scenarios.
- If we found any defect then definitely we are going to log a defect.
- Usability testing is carried out to check how easy it is for a new user to carry out basic task of application.
- It also checks for quicker access to the one who uses application frequently and consistently.

Example :-

1) Check for use of correct icons and corresponding tools, HPS.

2) Check for dropdown values being sorted correctly.

- There are two types of usability testing they are as follows.

1) GUI testing

2) Manual support testing

1) GUI testing (Graphical User Interface)

a) Easy of use

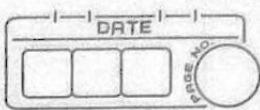
- On one click next action should happen immediately.
- Even physically challenged people can access the application smoothly and independently.

b) Speed of processing

- In this type we will check how quickly application respond to users action.

2) Manual support testing (Regular Expression testing)

- It is a process of checking context sensitivity to the users manual input.



- In this testing module or tabs respond to users manual action mention in users request.
- Regular expressions are sometimes more than one objects have some logical name or physical description so this we called it as "Regular Expression."

Example 8-

- 1) In this application, we enter pickup and drop location so, when we click on pickup and if we enter 'E' keyword at that time it suggest all the locations starting with 'E'.

3) SECURITY TESTING

- Security testing determines whether an application is capable of identifying related risks and averting possible attack.

4) PERFORMANCE TESTING

- Performance testing determines how a system performs in terms of responsiveness to the above requirements under a particular workload.

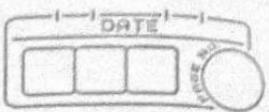
- Performance testing checks whether an application provides stipulated output in a stipulated time. It is carried out after functional testing.
- There are four types of performance testing

1) LOAD TESTING

2) STRESS TESTING

3) ENDURANCE TESTING

4) VOLUME TESTING

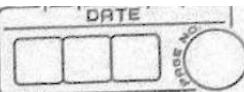


* RE-TESTING

- RE-TESTING IS CARRIED OUT BEFORE REGRESSION TESTING.
- IT IS USED TO DETERMINE WHETHER THE IDENTIFIED DEFECT IS SUCCESSFULLY REMOVED OR RESOLVED AND ALSO KNOWN AS CONFIRMATION TESTING.
- IT IS THE METHOD OF RE-EXECUTING SAME BUILD / APPLICATION WITH MULTIPLE TEST DATA.
- IN RE-TESTING WE WILL CHECK PREVIOUSLY FAILED TEST CASES ARE EXECUTED AGAIN IN THE NEXT CYCLE TO ENSURE REMOVAL OF THE EARLIER EXISTING DEFECT.
- RE-TESTING MOSTLY HAPPENS TWO TIMES
 - 1) BEFORE WE LOGGED DEFECT
 - 2) AFTER DEVELOPER SOLVE DEFECT
- WE PERFORM RE-TESTING WHEN DEVELOPER SOLVE DEFECT IN WHICH WE CHECK THAT SYSTEM IS WORKING FINE FOR THE TEST CASES AND FOR SOME DATA.

* REGRESSION TESTING

- AS DEFECTS ARE FIXED OR NEW FUNCTIONALITIES GET ADDED INTO THE APPLICATION, IT BECOMES NECESSARY TO CHECK THAT THERE IS NO IMPACT OF THESE CHANGES ON THE PREVIOUSLY WORKING FUNCTIONALITY.
- REGRESSION TESTING IS CARRIED OUT TO DETERMINE WHETHER THE CHANGED COMPONENT HAS AFFECTED THE FUNCTIONALITY OF THE UNCHANGED COMPONENT.
- FOR REGRESSION TESTING ONLY PASSED TEST CASES ARE TO BE CONSIDERED.
- REGRESSION TESTING IS DONE AFTER UAT AND BEFORE UAT.
- WHILE PERFORMING THE REGRESSION TESTING WE WILL FOCUS ON POSITIVE SCENARIOS, NEWLY ADDED SCENARIOS, FAILED SCENARIOS IF TIME PERMITS THEN ONLY WILL FOCUS ON MEDIUM AND LOW PRIORITY SCENARIOS.



* UAT (User Acceptance Testing) (End-to-End testing)

- UAT is done after the system and functional testing and UAT is conducted by customers and end-users.
- UAT is the final stage of testing before system goes to the final production.
- Customers or end users of the system perform acceptance testing based on the user requirements specification to confirm that application is behaving as expected by them or not.
- This is also called as End-to-End testing.

Let's see what exactly happens in UAT.

- After the completion of system and functional testing will start the UAT.
- But before starts the UAT we need to perform re-testing and regression testing and after performing UAT will again perform re-testing and regression testing.
- User will decide how many user stories they want to get tested from the tester after deciding the user stories tester checks all test cases with respect to the user stories.
- End user or customer can change the test data.
- After performing the UAT customer send the application for production.

- UAT is having two different types

1) Alpha Testing

2) Beta Testing

1) Alpha Testing

- Generally alpha testing happens in service based companies for service based application.
- Alpha testing is conducted in controlled environment.

2) Beta Testing

- Beta testing happens in product based application companies.

- Beta testing is conducted in uncontrolled environment.
- Uncontrolled environment means developer and testers are worked in same environment and customer and user are worked in different environment.
- Product based companies having their own product to develop so they are directly giving to the end user for accessing it and get feedback from them.

* TESTING TERMINOLOGIES

* ADHOC TESTING

- In general, every testing team conducts planned testing but testing team adopts informal testing sometimes due to some challenges or risks. This informal testing is known as Adhoc testing.
- When we are having strong domain knowledge and we have test cases but do not have test case data in this situation we do Adhoc testing.
- We will focus on positive and negative scenarios and log defects from small to large under Adhoc testing.
- The main purpose of this testing is to find defects by random checking.

* MONKEY TESTING

- Due to lack of time, the testing team concentrates on some of the main activities (like positive scenarios) in the software build for testing, this style of testing is known as monkey testing.

* EXPLORATORY TESTING

- When we are having test cases and test data but due to lack of knowledge on project domain at that time we prefer Exploratory testing.
- We do step-by-step execution of test cases with respect to SRS document.
- We will focuses on positive and negative scenarios in this testing.

* Severity and priority

Severity

- Severity of a defect is related to how severe a bug is
- It is a parameter which is used to show the impact of a particular defect on the software
- Severity is categorized into four types

Critical / Showstopper

Major / High

Moderate / Medium

Minor / Cosmetic

1) Critical severity / Showstopper

- This defects indicates complete shut-down like behaving abnormally, hanging-up, getting 404 errors, or we can say that where our testing team gets stuck. These defects comes under the critical severity.

2) Major severity / High

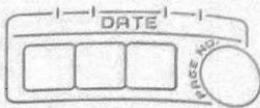
- Here, we are not getting expected result or half of the functionalities are working and remaining part is not working when the system is functional so, this is highly severe bug and comes under major severity.

3) Moderate severity / Medium

- Here, we are getting expected result but system causes some undesirable behaviour when the system is functional, this comes under the moderate severity.

4) Minor severity / Cosmetic

- It won't cause any major break down to the system like spelling mistakes, grammatical errors, color mismatch, blur images so, this comes under the minor severity.



Priority

- Priority of a defect is related to how quickly a bug should be fixed and deployed to live servers
- Priority is categorized into three types

High

Medium

Low

1) Low priority

- Low priority defects can be released as it is, if needed, we can fix this defect once the more serious defect has been fixed.

2) Medium priority

- Medium priority defects should get fixed if time permits us or it can wait until a new version is created.

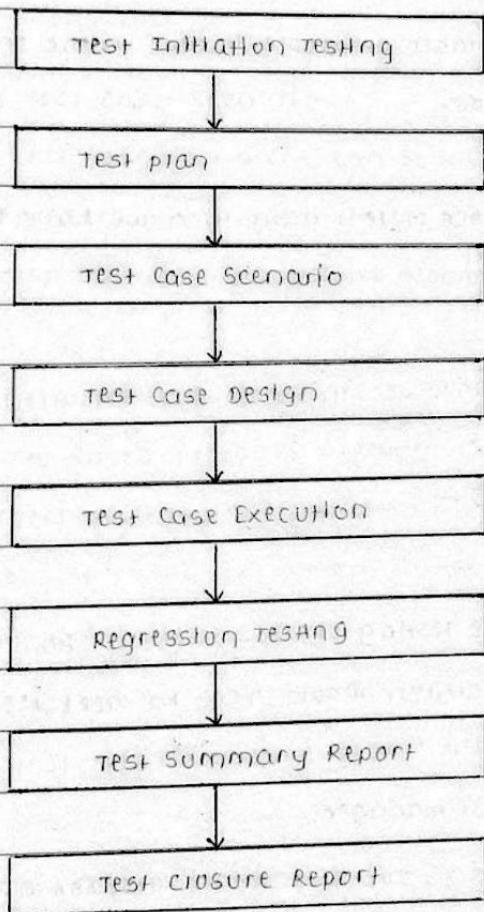
3) High priority

- High priority defects must be resolved as soon as possible, as it affects the system severely before the release.

When a defect is of high severity, most likely it will also have a high priority. Likewise, a low severity defect will normally have a low priority as well.

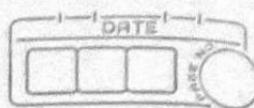
* Software testing life cycle (STLC)

- STLC stands for software TESTING LIFE CYCLE .
- STLC is a sequence of different activities performed by the testing team to ensure the quality of the software or the product.
- As soon as the development phase is over, the testers are ready with test cases and start with the execution.
- It defines the series of activities which should be carried out systematically to help the evaluation of software application .



i) TEST INITIATION TESTING

- Test Initiation testing includes requirements of project , scope of project and risk involve in the project .



a) Requirement of project

- In this requirement of project they will decide what is the domain of upcoming project like banking, telecom, pharma, insurance, E-commerce etc.

b) Scope of project

- Scope of project means selection of test strategies to test functionality of modules includes in the project like manual, automation, API, database, security, performance.

c) Risk involve in the project

Less resources - If there are less number of people involve in the project then each person would get extra work to do.

Less test data - If there is less number of test data then we have to perform exploratory testing on the basis of domain knowledge.

Lack of knowledge - If new person is involved in team then knowledge transfer(kt) is required for them.

2) Test plan

- Test planning is the first step of the testing process. In this phase we identify the activities and resources which would help to meet testing objectives.
- Test plan is done by test lead or test manager.
- Test plan includes resource allocation, job allocation and estimation and team is going to finalise under test plan.

a) Job allocation

- Jobs are getting allocated on the basis of scope of project means who is working on what (technology like, manual, automation etc).

b) Resource Allocation

- Resources include human effort, equipment and all infrastructure required for accurate and comprehensive (detailed) testing.
- This part of the test plan decides the measure of resources (number of testers and equipment) the project requires.

c) Estimation

- Test Estimation is a management activity which approximates (calculate) how long a task would take to complete.
- In simple words, estimation means start date and end date of the project.

3) Test Case Scenarios

- Test case scenarios can be defined as a top view of functionalities and it represent what needs to be tested in an application.
- Test Engineer / Testers are involved in the test case scenarios.

4) Test Case Design

- Test cases provides a detailed procedure that helps to test a particular aspect or feature of an application in details.
- based on SRS document or requirements test case scenarios and test case design are created.
- When we write scenarios we only write positive scenarios and at the time of test, we test positive and negative scenarios.

5) Test Case Execution

- After creating test case scenarios and test case design we execute them under the test case execution.
- While executing test cases, if we found any defect at that time we assign that defect to the developer.
- After resolving the defect developer send us corrected system or build.

6) Regression Testing

- AFTER RECEIVING THE CORRECTED SYSTEM OR BUILD THEN WE DO REGRESSION TESTING TO CHECK THE PREVIOUSLY FAILED TEST CASES.
- ALSO WE CHECK THAT NEWLY ADDED FUNCTIONALITIES UNDER REGRESSION TESTING.
- WE DO TESTING USING HIGH PRIORITY TEST CASES AND IF TIME PERMITS THEN WE DO EXECUTION OF LOW PRIORITY TEST CASES AS MEDIUM PRIORITY OR LOW PRIORITY TEST CASES.

7) Test Summary Report

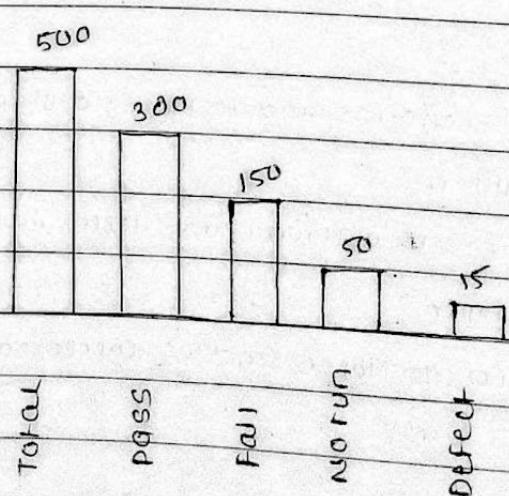
- WE ARE RESPONSIBLE TO CREATE TEST SUMMARY REPORT
- WE SEND THIS REPORT TO OUR LEAD

EXAMPLE 8- SUPPOSE PHONEPAY, COMPANY DEVELOPED A BUS BOOKING PROJECT MODULE

	TEST CASES	PASS	FAILED	NO RUN	DEFECT
DAILY	50	35	15	0	7
MONTHLY	500	300	150	50	15

8) Test Closure Report

- TEAM LEAD IS RESPONSIBLE TO MAKE TEST CLOSURE REPORT
- IN TEST CLOSURE REPORT LEAD WILL CHECK WHETHER ALL PROCESSES ARE CORRECT OR NOT.
- TO MAKE TEST CLOSURE REPORT TEAM LEAD USE DASH BOARD TOOL.
- THIS REPORT IS IN FORM OF GRAPH.
- TEAM LEAD WILL SEND THIS REPORT TO THE PROJECT MANAGER.



* TEST PLAN

* Agile Test Plan

Sprint Planning and Grooming

- WEEK 1
 - user story Analysis
 - prepare test scenario
 - prepare test cases

- WEEK 2
 - TEST CASE REVIEW
 - sanity TESTING
 - TEST CASE EXECUTION / BBT / SYSTEM AND FUNCTIONAL TESTING
 - RE-TESTING - Defect log

- WEEKS
 - RE-TESTING - Defect fix
 - REGRESSION TESTING - IF CR IS THERE
 - DOCUMENTATION - SUMMARY REPORT AND CLOSURE REPORT
 - END OF SPRINT

- Agile test plan is prepared by testing team lead.
- In agile sprint is of 3 weeks process
- So agile test plan is divided into 3 weeks.

- For week 1 - Sprint planning and grooming session
- user story analysis
 - prepare test scenarios
 - prepare test cases

- Before starting of each sprint, there is a sprint planning meeting, held in presence of product owner, development team, testing team and scrum master.

- In Sprint planning meeting , there is one session involved that is grooming session .
- Grooming session is activity where product owner explain each and every user story of the product .
- After completing the grooming session we starts analyzing user stories means which user stories are required for particular sprint , reviewing that user stories and asking doubts about these user stories to project owner .
- We know that user stories means all the requirements so , on the basis of user stories we starts preparing test scenarios or deciding the test scenarios .
- Once we will done with the test scenarios then we starts writing test cases according to our test scenarios or for selected user stories .
- So , this is work plan for first week .

FOR WEEK 2 - TEST Review process

Sanity Testing

System and Functional Testing

Defect log process

- In this week we starts with test review process after the completion of test case preparation .

- There are 4 types of review

i) Self Review

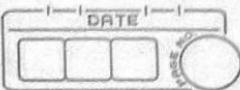
ii) Peer Review

iii) Internal Review

iv) External Review

i) Self Review

- Self Review means review done by own to check whether , according to the test scenarios , the test cases are prepared properly or not .



ii) Peer Review

- Peer review means review done by our team like team lead.
- Team lead go through all the test cases that we have prepared and on that basis gives us feedback or suggestions.

iii) Internal Review

- Internal review is done by our team members of our project like (product owner, development team, design team and testing team)

iv) External Review

- External review is done by client / stakeholder
- Most of the time external review is done at UAT
- External review is also known as "Walk through".
- Testing team lead is host / chair person of external review and he will prepare MoM (Minutes of meeting) document.
- After completion of test review process and before starts executing test cases we need to perform sanity testing to check that our build is stable or not.
- After performing sanity testing will starts executing test cases that means will perform system and functional testing.
- At the time of executing test cases some test cases are going to be passed and some are going to be failed. When our test cases are going to be failed at that we log the defect under defect log process.
- In this process of defect logging, defect goes from its lifetime means from getting defect to resolving defect.
- So, this is work plan for second week.

for week 3 - Re-testing

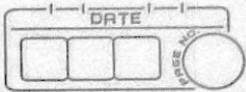
Regression Testing

Test Summary Report

Test Closure Report

End of Sprint

- In this week we starts with testing to check that previously found defect gets resolved or not.
- When we accept change request or new functionalities get added at that time we need to perform regression testing to check the impact of existing functionalities or we can say that the impact of new functionalities or change request won't harm our previously working functionalities.
- Once we done with testing then we need to prepare one report and in that report we have to mentioned the count of all test cases , passed test cases , failed test cases , defects logged and etc . after completing that report we have to send it to the team lead.
- After receiving the summary report team lead will prepare one more report and that report nothing but a Test Closure Report to check that everything is correct or not .
- And then finally the End of sprint.



* V-Model Test plan

- Duration of V-model is of 3 months
- Test plan of V-model is prepared by project manager and testing team lead.

INPUT OF TEST PLAN

a) SRS

- This document contains each and every detail of the requirements of a project taken from the client.

b) TRM

- This document is prepared by testing team lead and project manager
- It contains "who is going to perform" in this project, "what to perform" in this project

PROCESS OF TEST PLAN i) Team Formation

- According to SRS/FRS and TRM, team lead do some task like -

a) Resource Allocation

- resources are get selected by team lead. who are working on what is get decided.

b) Job Allocation

- Jobs are get allocated on the basis of SRS by team lead. that means, who is going to perform which testing in the project.

c) Estimation

- start date and end date (time) of task.
- Estimation is given by each team member to team lead. that means, how much time they require to perform their own task.

ii) RISK Involved in the project

- There are different type of risk occurred in the project like
 - 1) LACK OF resources
 - 2) LACK OF knowledge
 - 3) LACK OF test data
 - 4) LACK OF requirements
 - 5) LACK OF communication between testing team & development team
 - 6) LACK OF communication between business analyst and team
 - 7) LACK OF software requirement
 - 8) LACK OF defect tracking.

iii) Prepare Test plan

- Team lead is responsible to create a test plan
- Team lead will analyse the requirements of project
- Then according to this requirements will starts preparing test cases and decide "which features are need to be tested" and "what to be tested".
- Also team lead focus on which type of testing are required to test particular functionality.

iv) Review Test plan

- Test plan is reviewed by project manager and team lead
- They check whatever we have planned is correct or not.
- Project manager focus on 3 factors

a) BRS base coverage

- Check that test plan is prepared as per BRS or not.

b) TRM base coverage

- Check whatever we have mentioned in test plan is according to TRM or not.

c) RISK base coverage

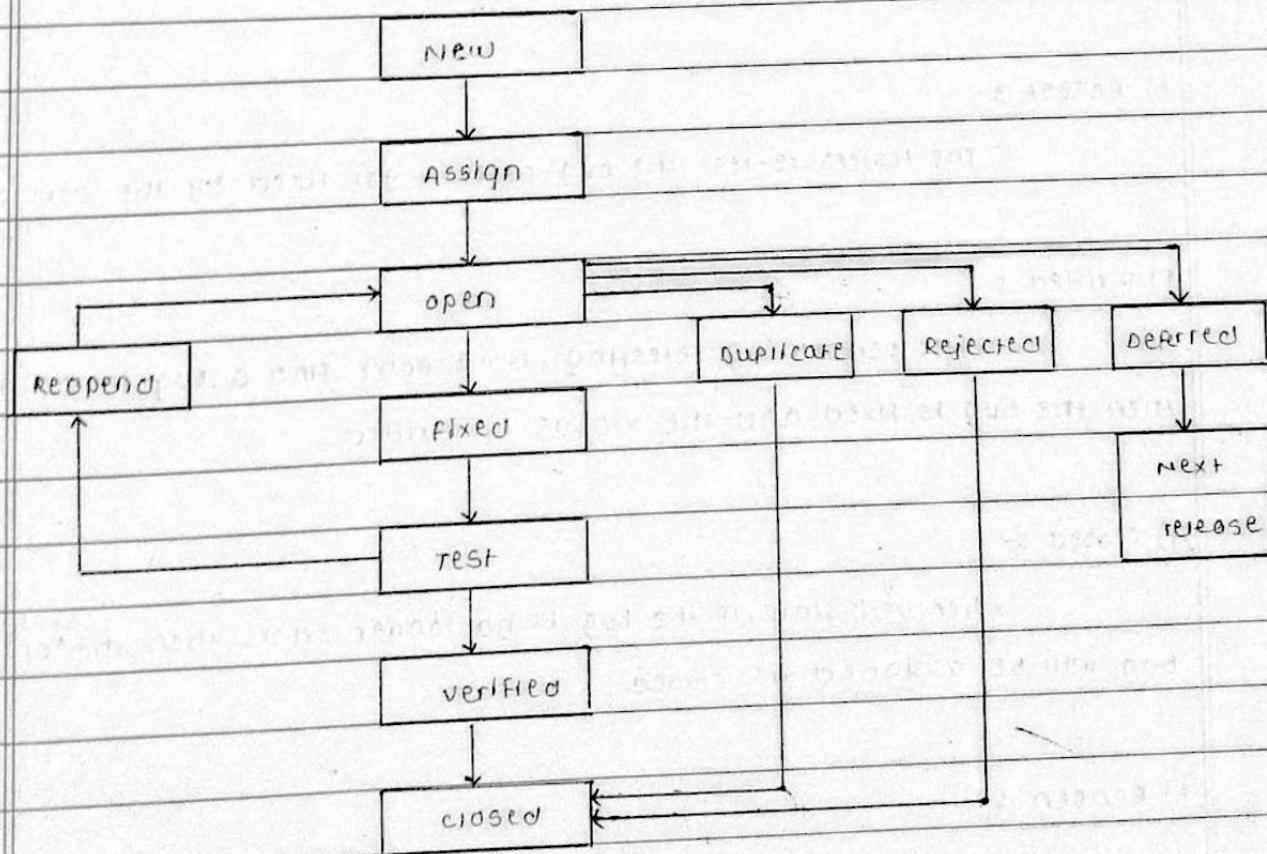
- RISK AND THEIR SOLUTION OCCUR IN PROJECT.

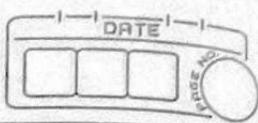
In this way test plan is get finalized by project manager.

Output

- Before starts the execution of test cases , we perform sanity testing to check that our build is stable or not.
- After sanity testing , we perform system and functional testing while performing this testing if we found any defect then we log that defect.
- Once we done with the system and functional testing then we perform UAT , if user validate it as correct then product send to final production.
- This is all about the v-model of test plan ..

* Defect Life Cycle





a) NEW :-

the reported defect has a status 'NEW'.

b) Opened :-

After assigning bug to developer, development team starts analyzing and work on that defect then at that time status of bug is opened.

c) Assigned :-

New bug is assigned to the developers by us. Once the defect is assigned then the status of the bug changes New to Assigned.

d) Fixed :-

When a developer makes the necessary changes in code and verify the change, then the status of the bug will changed as fixed and passed to the testing team.

e) Retest :-

The testers re-test the bug after it got fixed by the developers.

f) Verified :-

While performing retesting, if we don't find a bug in the software then the bug is fixed and the status is verified.

g) Closed :-

After verifying, if the bug is no longer exists then the ^{status} of the bug will be assigned as closed.

h) Reopen :-

If the defect remains the same after the retest, then tester will assign status for that bug is reopen, and again that bug goes through the life cycle to be fixed.



i) Duplicate :-

IF THE SAME DEFECT IS already reported by any other tester ,
THE DEFECT CAN BE CLOSED BY CHANGING ITS STATUS TO DUPLICATE.

j) Deferred :-

FIX IN THE NEXT VERSION , THE STATUS IS DEFERRED .

k) Rejected :-

IF THE DEVELOPER FEELS THAT THE BUG IS NOT GENUINE , DEVELOPER
REJECTS THE BUG. THEN THE STATE OF THE BUG IS CHANGED TO REJECTED .