Software

collection of computer programs that helps us to perform a task.

Types of software:

1. System software
   Ex: Device drivers, Operating systems, Servers, Utilities, …
2. Programming software
   Ex: compilers, debuggers, interpreters,...
3. Application software
   Ex: Web Applications, Mobile Apps, Desktop Applications …

What is software testing?

- Software Testing is a part of software development process.
- Software Testing is an activity to detect and identify the defects in the software.
- The objective of testing is to release quality product to the client.

Manual Testing

- Manual testing is the process carried out to find the defects in software
- In this process tester plays an important role as end user and verify all the features of the application to ensure that the behavior of the application is working as per user requirements
- It is not necessary to have any programming knowledge and tool for manual testing
- As the software testing fundamental always says “100% automation is not possible” So manual testing is very important

Software Development Life Cycle(SDLC)

- SDLC is the process consisting series of planned activities to develop or alter the software products
- It is a procedure to develop software

➢ SDLC contains six phases they are

1. Requirement Study or Initial phase
2. Feasibility Study
3. Design phase
4. Coding phase
5. Testing phase
6. Installation/Delivery & Maintenance phase

Requirement Study or Initial phase

- Tasks : Interaction with the customer and gathering the requirements
- Roles : Business Analyst(B.A) or Product analyst(P.A) or Engagement Manager(E.M)
- Process : First of all the business analyst will take an appointment from the customer, collects the templates from the company meets the customer on appointed day, gather the req. with the help of template and comes back to the company with the requirements documents.
- Once the requirement document has come to the company the engagement manager will cheek whether the customer gives any extra requirements or confused requirements. In case of extra requirements he deals the excess cost of the project. In case of confused requirements he is the responsible for prototype demonstration and gathering the clear requirements.

- Proof :The proof document of this phase is Requirements Document. This is called with different names in different companies.
- *FRS* (Functional Requirements Specification)
- CRS (Customer Requirement Specification)
- URS (User Requirement Specification)
- BDD (Business Design Document)
- BD (Business Document)
- BRS (Business Requirement Specification
- Some companies may maintain the overall business flow information in one document and the detailed functional requirement information in other document
- Once CRS id ready B.A will convert it to SRS(software requirement specification) document

2.Feasibility Study Phase :

- Tasks : It is detailed study of the requirement in order to check weather the product is able to develop or not.
- Roles : System Analyst, Project Manager HR, Finance team and team manager
- Process : In this phase following activities are performed :

1 .Technical Feasibility : Check the enough technology we have or not

2. Financial Feasibility : Check weather we have financialy enough or not

3. Resources Feasibility : Check weather we have enough resources for development

3. Design

- Tasks:
- High level designing : HLD gives the overall System Design .This is very useful for the developers to understand the flow of the system. For this the entry criteria are the requirement document that is SRS. and the exit criteria will be HLD.
- Eg : dataflow diagrams ,flow charts..
- Low level designing - During the detailed phase, the view of the application developed during the high level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created.
- The entry criteria for this will be the HLD document. And the exit criteria will the program specification and unit test plan (LLD).
- Roles : High-level designing is done by the chief Architect & Low level designing is done by the Technical Lead
- Note : Developers use LLD for coding

4. Coding phase:

- Tasks: Developing or Programming
- Roles: Developers or Programmers
- Process :Developers will actual code by using the technical design document (LLD) as will as following the coding standards like proper indentation colour coding , proper commenting and etc…
- Proof : The proof document of this phase is source code

5. Testing

Task : Testing

Roles: Test engineers

Process :

- First of all the test engineers will collect the requirements document and try to understand all the requirements
- While understanding if there are any confusions then they will list out all of them in a review report.
- They will send the review report to the author of the requirements document for clarification.
- Once the clarifications are given and after understanding all the requirements clearly, they will take the test case template and write the test cases.
- Once the first build is released then they will execute the test cases
- Ifat all any defects are found. They will list out all of item in a defect profile template.
- They will sent the defect profile document to the development department and then will be waiting for the next build to be released.
- Once the next build is released then they re-execute the test cases

- Once then next build Is released then they re-execute test cases.
- If at all any defects are found they will update the profile document and sent it to the development department and will be the waiting for the next build to be released .
- This process continuous till the product is defect free.
- Proof :The proof of the testing phase is “Quality Product”.
- Bug - free
- Delivered on time
- Within budget
- Meets requirements/expectations
- Maintainable

6. Installation/Delivery & Maintenance Phase:

- Task : Installing the application in the client's environment
- Roles : Deployment/Installation engineer

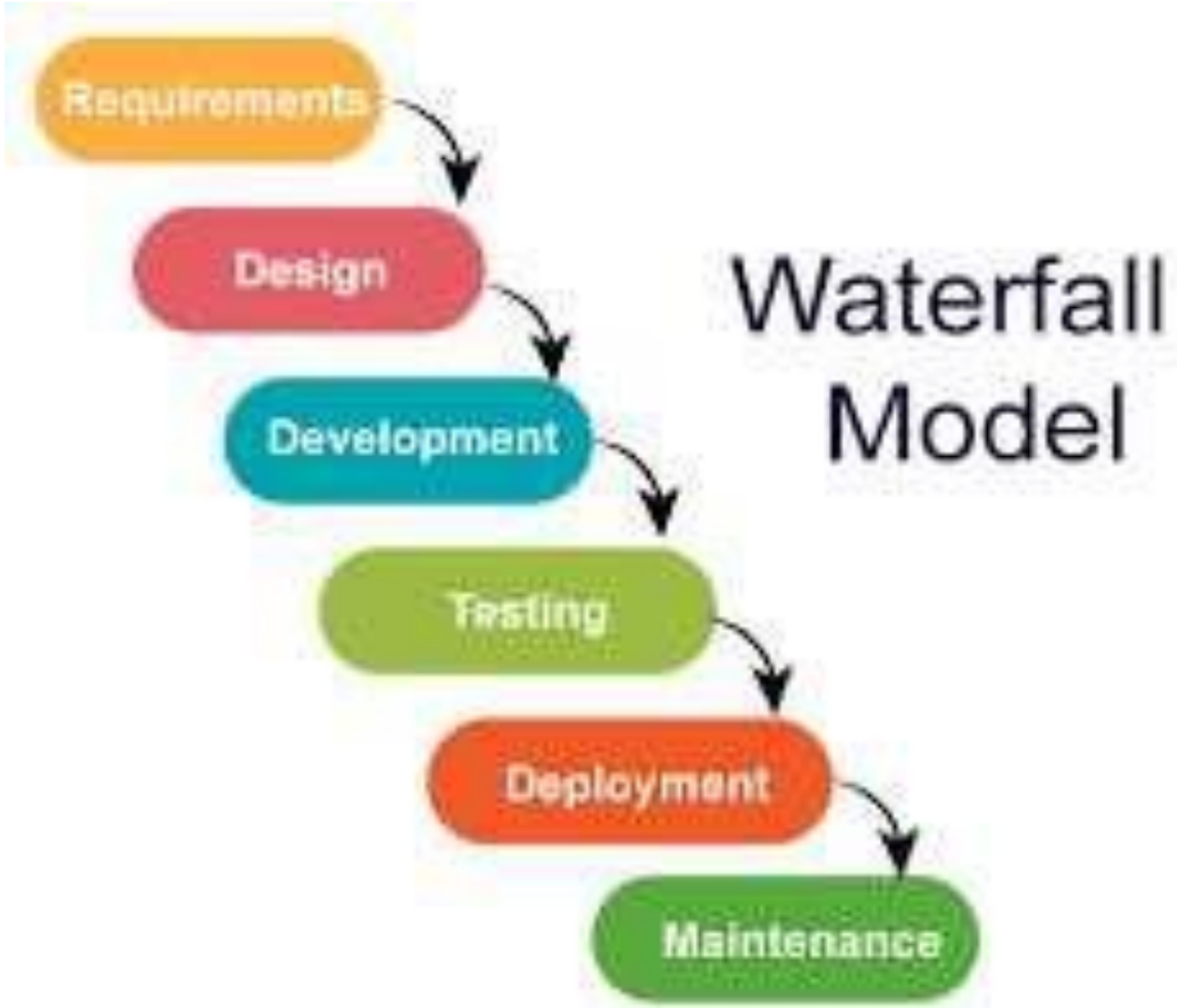
Process: Deployment/Installation engineer will be going to the clients place and install the application in their environment with the help of the guidelines provided in deployment document.

Maintenance:

In this phase :

- Bugs identified by customer will be reported to developers and will be fixed by developers.
- Changes to existing software will be implemented

WaterFall Model :



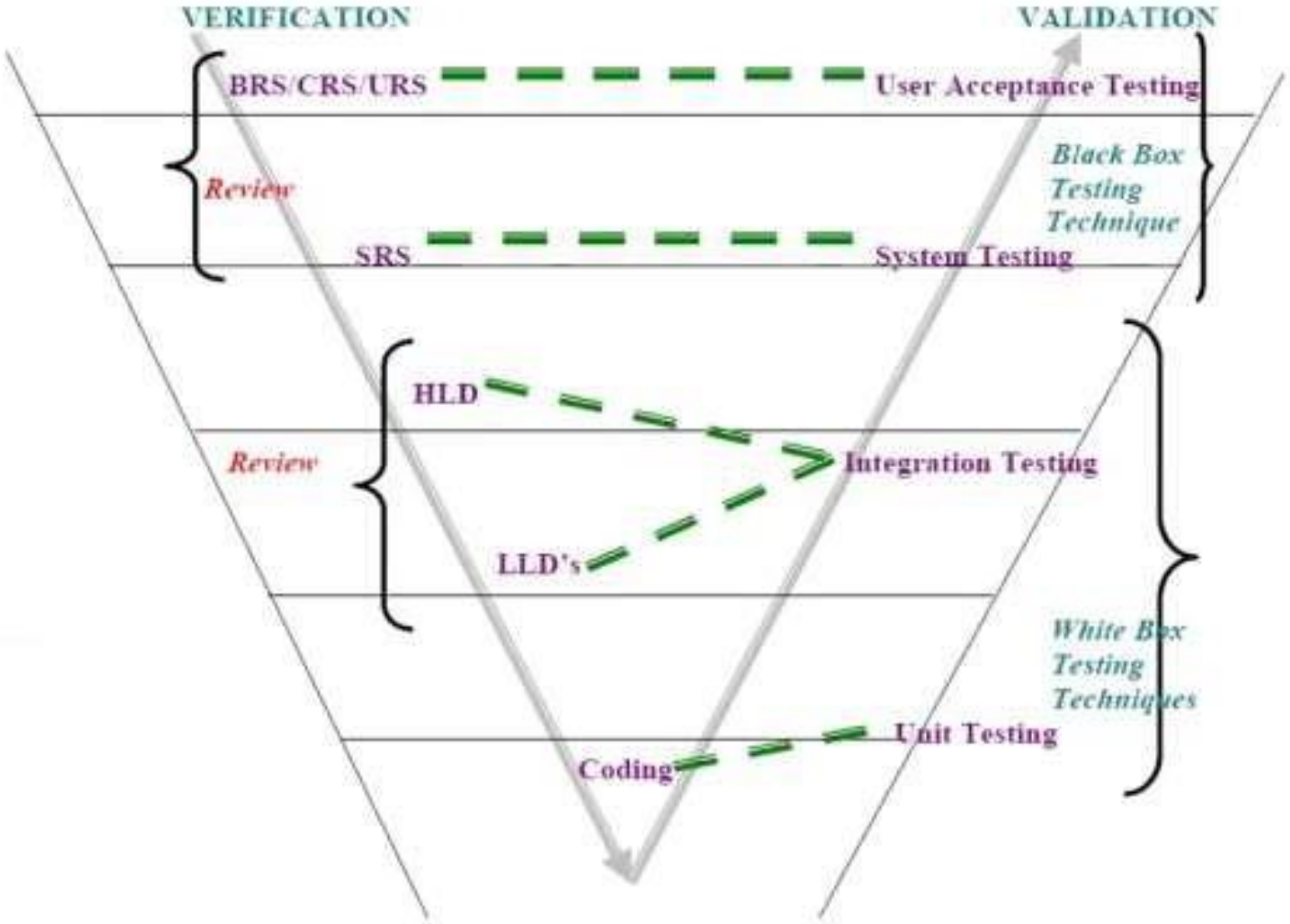
- Requirement Elicitation and analysis :All possible requirements of the software to be developed are captured in this phase and documented in a requirement specification document.
- Design :The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps m specifying hardware and system requirements and also helps in defining overall system architecture.
- Implementation Coding : With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase, Each units is developed and tested for its functionality which is referred to as Unit Testing.
- Integration and Testing : All the units developed in the implementation phase are integration into a system after testing of each unit. Post integration the entire system is tested for any faults and failures

- Installation of System : Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
- Maintenance : There are some issues which come up in the client environment. To fix those issues patches are released. (Patch is a piece of software which has only modified programs) Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.
- Advantages of waterfall model :
- This model is simple and easy to understand and use;
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.
- Each phase are well documented

- Disadvantages of waterfall model :
- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the Concept stage. In other words backtracking is not possible
- Requirements are freezed,Not suitable for the projects where requirements are at moderate to high risk of changing.
- Testing activity starts after coding.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model long and ongoing projects.
- When to use the waterfall model:
- This model is used only when the requirements are very well known, clear and fixed.
- product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

V and V Model /V model/Verification & Validation Model/Fish Model

V-Model



- Verification —Are we building product right?
- Validation — Are we building right product?
- In this model testing starts from initial phase of software development.
- The V-model typically consists of the following phases :
- In this model following activities takes place :
- Stage — Initially customer requirements(CRS) are converted to SRS parallel testers check CRS to find is there any incomplete requirement or ambiguous requirement. Also testers plan for acceptance testing by writing acceptance test cases.
- Stage2 — Using SRS high level design is obtained by developers parallely testers check if SRS matching with CRS and also plan for System testing by writing system test cases.
- Stage3 -- In LLD actual software components are designed and parallely testers plan for integration testing by writing integration testcases.

- Stage4 — Using LLD coding is performed by developers and parallel testers plan for the functional testing by writing functional test cases.
- Stage5 – Developers test each and every part of code by creating unit testcases.
- Verification : The process of evaluating the work products(not the actual final product) of a development phase to determine whether it meets specified requirements for that phase.
- Verification checks whether we are building the right product.
- Focus on documentation
- Verification typically involves testing documents
  - Static testing
    - Review
    - Walkthrough
    - Inspection

- Validation : The process of evaluating the work product at the end of the development phase to determine whether it satisfied the business or customer req.
- Validation checks whether we are building the product right
- Takes place after verifications are completed
- Focus on software
- Validation typically involves actual software testing
  - Dynamic testing
    - Unit testing
    - Integration testing
    - System testing
    - UAT

- **Static Testing**
- Testing the project related documents
  - Review
  - Walkthrough
  - Inspection
- **Dynamic testing**
- Testing the actual software
  - Unit testing
  - Integration testing
  - System testing
  - UAT

- **Review**
- Conduct on documents to ensure correctness and completeness.
- Any time by any one
  - Requirements reviews
  - Design reviews
  - Code reviews
  - Test plan reviews
  - Test cases reviews
  - Defect reviews …
- **Walkthrough**
- It is a informal review.
- Author reads the documents or code and discuss with peers.
- It’s not pre-planned and can be done whenever required.
- Also walkthrough does not have minutes of the meet.

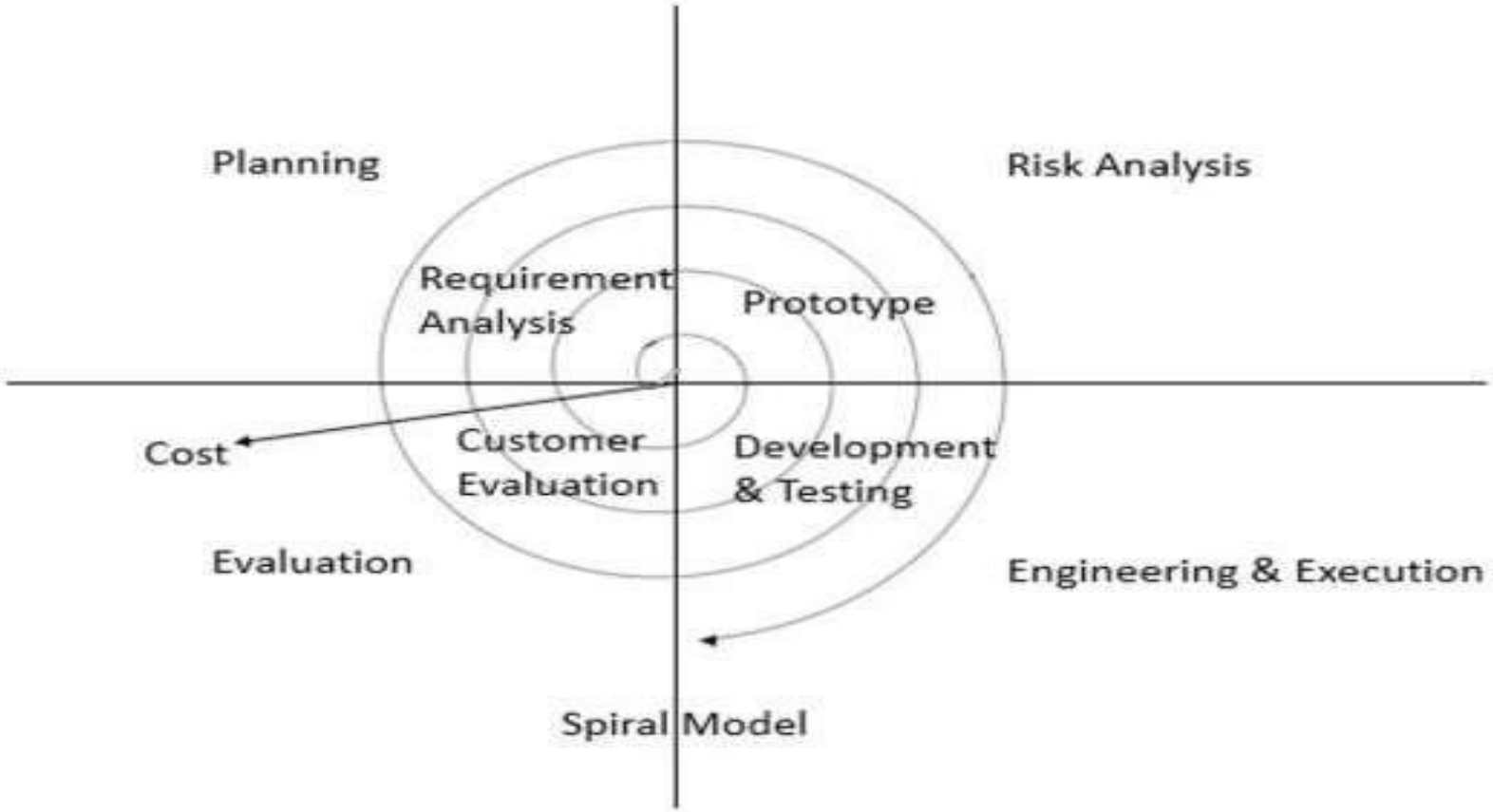
- **Inspection**
- Most formal review type
- 3-8 people will involve in the meeting
  - ✓ reader
  - ✓ writer
  - ✓ moderator
  - ✓ concerned
- Has proper schedule intimated via email to the concerned developer/tester.

- QA vs QC

QUALITY ASSURANCE	QUALITY CONTROL
QA is process related.	QC is the actual testing of the software.
QA focuses on building in quality.	QC focuses on testing for quality.
QA is preventing defects.	QC is detecting defects.
QA is process oriented.	QC is product oriented.
QA for entire life cycle.	QC for testing part in SDLC

- Advantages
  - Testing is involved in each and every phase
  - Testing starts in early stages
  - Avoid the downward flow of defects
  - Simple and easy to use
  - Works well for small projects
- Disadvantages
- Documentation is more
- Initial investment is more
- Very rigid and least flexible
- When to use the V model
- Used for small to medium size project where the requirements are clearly defined and fixed

- Spiral Model



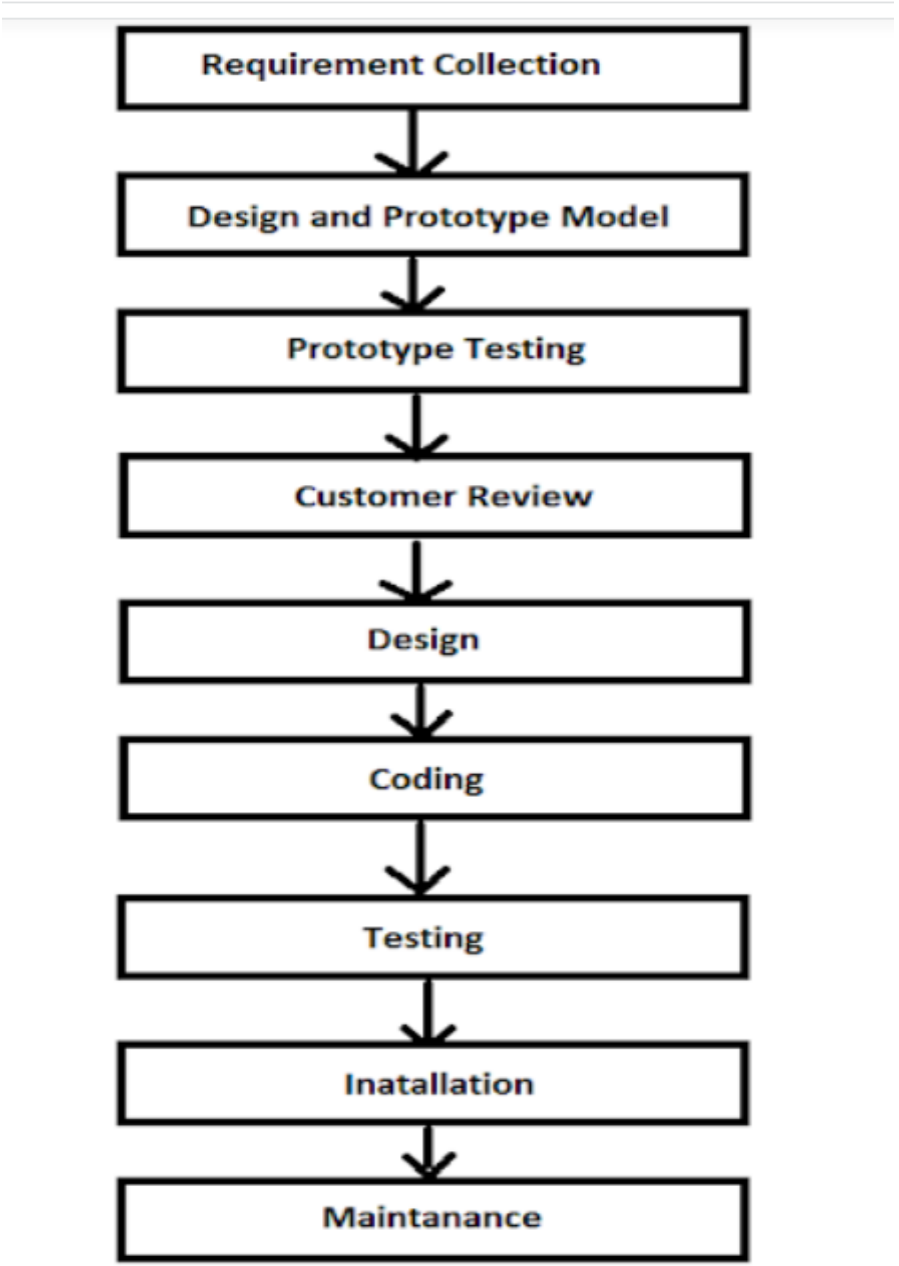
- Here the software is developed in small increments, It is also called as risk assessment model.

- Assume software is made up 3 modules(x y and z) initially requirements of x module is collected
- design is obtained. Coding of module x is performed by developers. After coding testing of performed in this stage identified errors are reported to developers and fixed. This completes
- In the second cycle module y is developed and so on...
- Spiral model is risk assessment model.
- Risk Analysis in spiral model: In the risk analysis phase, a process is undertaken to identify risk and alternate solutions,
- A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.

- Advantages of Spiral model :
- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date. In other words changes are allowed.
- Disadvantages of Spiral model :
- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project's success is highly dependent on the risk analysis phase.
- Doesn't work well for smaller projects.
- Testing activity starts after coding.

- When to use Spiral model :
- When costs and risk evaluation is important
- For medium to high-risk projects
- Rights reserved
- Long-term project commitment unwise because of potentlal changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

- Prototype Model :



- Blue print of the software
- Prototype is a dummy model. It is represented in the form of screenshots or images.
- Initially collect requirements from customer based on the requirements design & develop prototype.
- Perform prototype testing here we test all components exists or not.
- Once prototype is tested it will be shown to customer if any feedback provided implement feedback in
- prototype by changing requirements/design.
- Once customer agrees prototype then original software is developed and tested.
- Original software testing involves testing if components are working or not.

Note —Prototype model involves 2 types of testing prototype testing and actual testing.

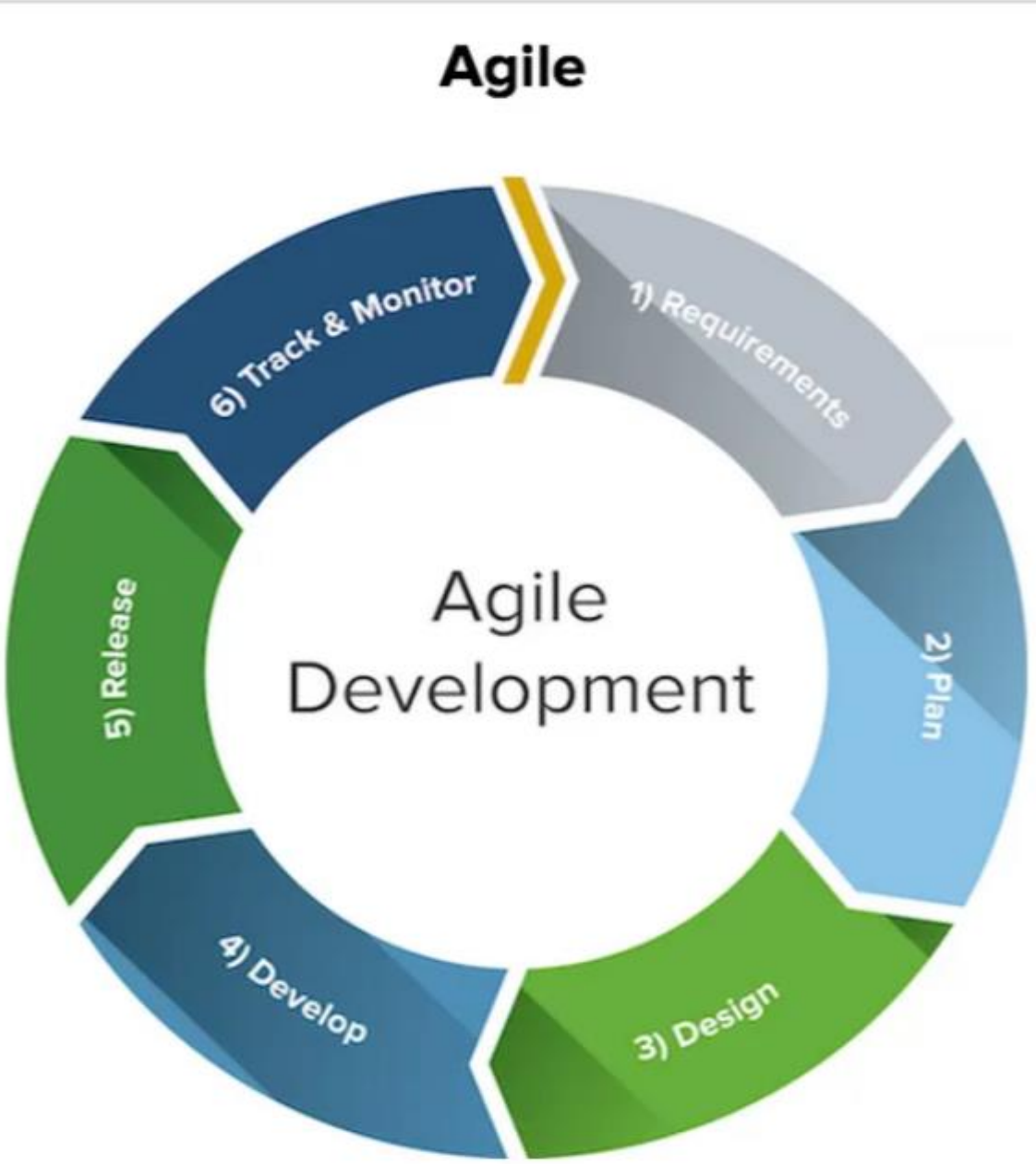
- Disadvantages of Prototype model
- Leads to implementing and then repairing way of building systems.
- Practically. this methodology may increase the complexity of the system as scope of the may expand beyond original plans.
- High cost
- When to use Prototype model :
- Prototype model should be used when the desired system needs to have a lot of interaction with the end users,
- This model is used when customer and developer is new to market,
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

• Agile Model :

- Agile simply means Quick
- Agile is one of the different Software Development Life Cycle Models (i.e. SDLC Models) available in the market
- **Why Agile SDLC Model became popular in the market?**
- Most of the Software projects use Agile SDLC Model for the below reasons:
- These days most of the Application like Amazon, Flipkart etc. adapt to the changes in the market very quickly.
- Traditional SDLC Models won't adapt to changes quickly and hence are not suitable for these days applications.
- Agile SDLC Model become popular in the market, for the medium and large sized Application which evolve over a period of time by quickly adapting to the market needs.
- Agile SDLC Model develops the Software over multiple iterations in an incremental manner

- **Agile Methodologies**

1. **Scrum**
2. **Kanban**
3. **Lean**
4. **Extreme Programming**
5. **Crystal**
6. **FDD**
7. **DSDM**



- **Agile Scrum Methodology**
- Under Agile, Scrum Methodology is the number one in the market
- Agile is not equal to Scrum
- Scrum follows Agile principles and process

- **Sprint**
- Software is developed in iterations known as Sprint
- Software will be developed in an incremental manner with few set of requirements for every iteration
- Sprint duration is 1 to 4 weeks
- Ideal sprint duration is 2 weeks and is followed by most of the projects in real time

- **Scrum Roles**
- Product Owner
- Scrum Master
- Development Team
- **Product Owner**
- Knows the Business well
- Interacts with the customers
- Activities performed by Product Owner in the Project is
- Creates Epics

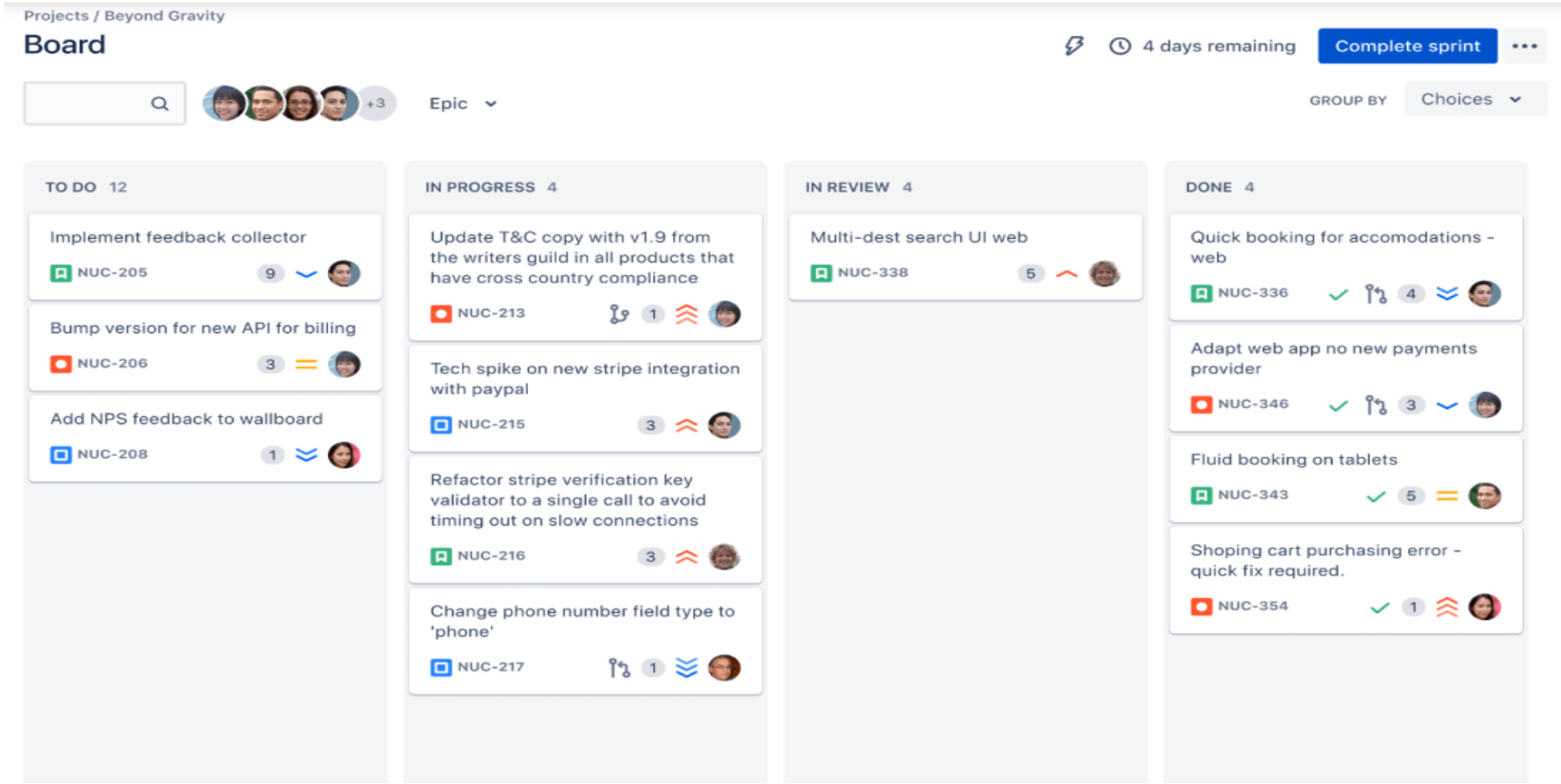
1. EPIC is a large requirement, which can be broken into smaller requirements known as User stories

- **User Stories**

Epic	User Story ID	User Story	Acceptance Criteria
TN_Epic_001: For a new e-commerce website to launch, the highest Business value will be when a new user is able to register, login, search and purchase an item from the website.	US_001	As a first-time visitor to the e-commerce website, I want to register my account, so that I can login to the application.	Ensure the first-time visitor is able to: - Navigate to Register Account page - Fill his First Name, Last Name, E-Mail, Telephone, Password, Password Confirm, Newsletter (Yes or No), Privacy Policy checkbox and Register Button - Create account by providing the above details - Login automatically on creating the Account
	US_002	As a registered user, I want to login to the website, so that I can see my account details etc..	Ensure the Registered User is able to: - Navigate to Login page - Provide credentials into Email and Password fields and login - Finding all the below Account details after login: - Edit Account Information - Change Your Password - Modify your Address Book Entries - And others
	US_003	As a registered user, I want to logout from website, So that no one else can access my account.	Ensure the Registered User is able to: - Logout from the Application using the logout option provider in the Menu options - Other users should not able to access My Account functionality of the User

- Creates User Stories
- User Stories are smaller requirements broken down from a larger Requirement known as Epic
- Maintains the Stories in Product Backlog and priorities them
- Created list of stories by Product Owner will reside in Product Backlog by default
- Creates a Sprint
- Create an Iteration known as Sprint for a duration which ranges from 1 to 4 weeks
- Sprint Planning Meeting
- Conducts Sprint Planning Meeting with Development Team
- Brings up the list of Prioritized Stories
- Estimates the stories with Development Team
- Discusses and Finalizes stories that can be delivered by the Development Team in the Sprint
- Assigns the finalized stories to the Sprint Backlog
- Starts the Sprint

- **Development Team**
- Starts working on the User Stories assigned to the Sprint
- Scrum Board
- To Do
- In Progress
- Done



- Developers in the Team, perform the below activities for each and every User story added to the Sprint:
- Understand the Requirements to be developed
- Design the Requirements
- Code the Requirements
- Unit Testing and Integration Testing of Requirements
- Fix the bugs reported against the User Stories
- And others
- Testers in the Team, perform the below activities for each and every User story added to the Sprint:
- Understanding the Requirements for deriving the Testable Requirements
- Create Test Cases
- Execute Test Cases
- Re-test bugs which are stated as fixed by developers
- Perform other Testing activities like Automation, Testing Types etc.
- And others

- **Scrum Master**
- Monitors and drives the Development Team to finish their Sprint goals
- Conducts Daily Stand-up Meetings (i.e. Daily Scrum Meeting)
- Meeting lasts for 15 Minutes every day
- Development Team will attend this meeting and will be standing during this meeting
- Every team member will provide their current work status during the meeting
- What are we working now?
- What is blocking our work?
- What are our pending tasks?
- Will we be able to finish our work on time?
- Scrum Master Uses Burndown Chart to track the progress of the Development Team
- Burndown chart gives the clear picture of how much work is remaining in the Sprint

- **Sprint Review Meeting**
- Developed and Tested Stories will be shown as Demo to the Product Owner
- Product Owner has to review and accept these stories
- **Sprint Retrospective Meeting**
- End of the sprint
- What went well?
- What went wrong?
- Improvements needed for next sprints
- **Meetings**
- Sprint Planning Meeting
- Daily Scrum Meeting
- Sprint Review Meeting
- Sprint Retrospective Meeting

- Testing Methodologies or Testing Types;
- There are three methods of testing
- A) Black-Box -resting
- b) White-Box Testing
- c) Gray-Box Testing
- (a) Black-Box Testing: If one performs testing only on the functional Of an application Without having any structural(code) knowledge then that method of testing is known as Black-Box testing,
- usually the test engineers perform this testing.
- (b) White-Box Testing : If one performs testing on the structural(code) part of an application then that method of testing is known as white box testing,
- usually the developers or white box testers Perform it.

- (c) Gray-Box Testing; If one performs testing on both the functional part as well as the structural(code) part of an application then that method of testing is known as grey box testing , usually the test
- engineers with code knowledge perform this testing. In other words it is a combination of both black box and white box testing.
- Black box testing types/ Levels of Testing
- Different levels of testing are:

1.Unit Testing

2.Functional/Component Testing

3. Integration Level Testing

4. System Level Testing

5. User Acceptance Testing (U.A.T)

1.Unit Testing : A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method or property.

Example :
```
public class Basics {
    public int compare(int n1, int n2) {
    if (n1 > n2) return 1;
    return -1;
     }
    }
```

2. Functional Testing : Also called component testing. Testing each and every component rigorously against requirement specifications is known as functional testing,

- Functional testing will be tested with valid data first, If it works for valid data then test for invalid data.
- Do both positive and negative testing.
- Don't perform under testing or over testing.

- Functional testing falls in to two categories :
- Positive functional testing : This testing carry exercising the application's functions with valid input and also verifying that the outputs are correct.
- Negative functional testing :This testing involves exercising application functionality using a combination of invalid inputs, so that unexpected operating conditions and by some other “out-of-bounds" scenarios.

3. Integration Testing :

It is a level of testing in which the developers will develop some interfaces to integrate the modules and test whether the interfaces are working fine or not. It is a white box testing usually developers or white box tasters perform. Here we test dataflow between two modules.

- The developers may follow one of the following approaches while integrating the modules.

a. Top-down approach

b. Bottom—up approach

c. Hybrid or Sandwich approach

d. Big bang approach

a. Top-down approach :- In this approach one will develop the parent modules first and then integrate them with the related child modules

b. Bottom-up approach : - In this approach one will develop the child modules first and integrate them to the parent modules

c. Hybrid approach Or Sandwich approach :-This is a mixed approach of both the top down and bottom up approaches

d. Big bang approach :- In this approach one will wait till all the modules are ready and finally they will integrate all the modules at a time.

- STUB While integrating the modules in top down approach if at all any mandatory module is missing then that module is replace with a temporary program known as STUB
- DRIVER While integrating the modules in bottom up approach. If at all any mandatory module is missing then that module is replaced with a temporary program known as DRIVER

Note : Big bang approach is type of Non incremental integration testing.

Top down and bottom up approach is type incremental integration testing.

- Drawbacks of Non increamental integration:
- We might miss data flow between some of the modules
- If you find any defect we cant understand the root cause of defect

4. System Testing: -

It is a level of testing in which one will install the complete application in to the testing environment and then perform testing on it. It is end to end testing where testing environment is similar to production environment, Here, we go through all the features of software and test if the end business functionality works.

- System testing focuses on below aspects.
  - Graphical User Interface Testing (GUI)
  - Functional Testing
  - Non-functional Testing
  - Usability Testing

- GUI Testing :

- Graphical User-interface Testing is a process of testing the user interface of an application.
- A graphical user interface includes all the elements such as
  - Menus
  - Checkbox
  - Buttons
  - Colors
  - Font-sizes
  - Icons
  - Content
  - Images …

- Testing the size, position, width, height of the elements.
- Testing of the error messages that are getting displayed.
- Testing the different sections of the screen.
- Testing of the font whether it is readable or not.
- Testing of the screen in different resolutions with the help of zooming in and zooming out.
- Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.
- Testing the colors of the fonts.
- Testing whether the image has good clarity or not.
- Testing the alignment of the images.
- Testing of the spelling.
- The user must not get frustrated while using the system interface.

- Testing whether the interface is attractive or not.
- Testing of the scrollbars according to the size of the page if any.
- Testing of the disabled fields if any.
- Testing of the size of the images.
- Testing of the headings whether it is properly aligned or not.
- Testing of the color of the hyperlink. [active, hover, visited]
- Testing UI Elements like button, textbox, text area, checkbox, radio buttons, drop-downs, links etc.
- Usability Testing :
- During this testing, validates application provided context sensitive help or not to the user.
- Checks how easily the end users are able to understand and operate the application is called usability testing.

- Functional Testing:
- Functionality is nothing but behavior of application.
- Functional testing talks about how your feature should work.
  - Object Properties Testing
  - Database Testing
  - Error Handling
  - Calculations/Manipulations Testing
  - Links Existence & Links Execution
  - Cookies & Sessions
- Object properties testing
- Check the properties of objects present on the Application.
- Ex:
  - Enable
  - Disable
  - Visible
  - Focus
  - Multi select

- Database/backend testing :
- DML Operations are checked
  - Insert
  - Update
  - Delete
  - Select
- Table & column level validations
  - column type
  - column length
  - number of columns
- Relation between the tables
  - Normalization
- Functions
- Procedures
- Triggers
- Indexes
- Views

- Error handling Testing :
- Tester will verify the error messages while performing incorrect actions on the application.
  - Error messages should be readable.
  - User understandable or simple language.
- Ex: On login fail,
  - Incorrect data / Something went wrong
  - Invalid user. Please try again.
- Calculations/Manipulations testing :
- Tester should verify the calculations.
- Ex:
  - Total bill amount
  - Discount amount
  - Total balance after a transaction
  - For different sets of input data, check the calculations

- Links Existence & Execution :
- Links existence:
  - Where exactly the links are placed
- Links execution:
  - Links are navigating to proper page or not
- Types of links
  - Internal links (navigation inside same page)
  - External links (navigation to external target page)
  - Broken links (no target page)
  - Cookies &Sessions
- Cookies
- Are temporary files created by browser while browsing the pages through internet.

- Sessions: are time slots created by the server.
- Session will be expired after some time.

Note : Testing environment is similar to production environment means hardware, software and data in both environments should be same.

When we can start system testing :

l) Minimum number of feature ready

2) Basic functionality of all the modules must be working

3) Testing environment should be similar to production environment..

6.User Acceptance Testing :

It is the level of testing in which one will perform the same system testing in the presence user in order to make him accept the application.

- Acceptance testing can be performed by;

a) Test engineers of client side

b) Employees of client side

c) End users of client

- Alpha testing—Acceptance testing performed at developer site.
- Beta testing —Acceptance testing performed at customer site.

• Non-functional Testing :

- Once the application functionality is stable then we do Non-functionality testing.
- Focus on performance, load it can take, and security.
  - Performance Testing

  Testing speed of application

    - Load Testing
    - Stress Testing
    - Volume Testing

- Load testing : Increasing load on the application gradually and checking the speed of the application.
- Stress testing : Suddenly increase/decrease load on the application and check its speed
- Volume testing : Check for quantity of data that can be handled by the application
- Security testing :Test for how secure is the application

- We will concentrate on the following areas of the application

a. Authentication Testing

b. Direct URL Testing(Uniform Resource Location)

c. Fire Wall Leakage Testing.

a) Authentication Testing: -

It is a type of testing in which one will enter different combinations of usernames and passwords and try to access the home page in order to check whether only the authorized persons are accessing the application or not.

b) Authorization / Access control :

- Testing the application by giving authorization to users or services permission to access some data or perform a particular action.

c) Direct URL Testing:-

- It is a type of testing in which one will enter the URL of an unauthorized page directly in the browser and try to access that page in order to check whether it is been accessed or not.

d) Firewall Leakage Testing: -

- It is a type of testing in which one will enter as one level of user and try to access the other level of Unauthorized pages in order to check whether the firewalls are working properly or not.

- Recovery testing:
- Check when the system changes from abnormal to normal
- In order to check how fast and better application can recover after crash or hardware failure
- Compatibility testing :
- It is a type of testing in which one may have to deploy the application into the multiple number of environments prepared with different combinations of environmental components in order to check whether the application is compatible with all that environments. This type of testing is usually done to products.
- Installation testing :
- Check installation steps related screens are clear to understand.
- Check for easy screen navigations
- Simple steps not complex
- Check proper Un-installation procedure

- **What is build?**

- Build is an developed software or a build is the process of converting source code files into standalone software artifact(s) that can be run on a computer or
- Compressed zip folder

- **What is STLC**

- **STLC Stands for Software Testing Life Cycle**

- It is process of different procedure to Test the Software.

- **Phases involved in STLC are:**

- **Requirement Study**
- **Write Test Plan**
- **Write Test Cases**
- **Traceability Matrix**
- **Test Execution**
- **Defect Tracking**
- **Prepare Test Execution Report(TER) / Summary Report**

- 1. **Requirement Study** : Tester should understand the requirements to generate scenarios. The requirements can be in the form of CRS, SRS and some other
- Requirement 2 types
- 1. Explicit Req.
- 2. Implicit Req.
- **Explicit Req**. : The requirements that are directly given by the customer.
- **Implicit Req.** : The requirements that are analyzed by business analyst feeling that they will add the value (user friendliness) to the application.
- **List of Explicit requirements:**

- Initially whenever login screen is invoked Login and Cancel buttons should be disabled
- Entering username and password the login button must be enabled
- Entering any information into any field the clear button must be enabled
- The tabbing order must be username, Password, Login and Cancel.

- **List of Implicit requirements:**

- Initially whenever Register/login screen is invoked the cursor must be available in the username field.
- Entering invalid username, valid password and clicking on login button an error message should have displayed **“Invalid username please try again”**.
- Entering valid username, invalid password and clicking on login button an error message should have displayed **“Invalid username please try again”**.

- 2. Test Plan : It is document which describes the future activities in testing , how to perform testing an on application.

- **Test plan Templates standard fields are:**

1. Introduction/Objectives :

   - In this section the purpose of the document is clearly described here

2. Scope :

   - **Features to be tested :** The list of all the features to tested in this phase clearly mention here,

- Register
- Login & Logout
- Forgot Password
- Search
- Product Compare
- Product Display Page
- Add to Cart
- Wish List
- Shopping Cart
  - **Features not to be tested :** the features that are not planned for testing in this phase.
- Out of scope features
- Low risk features

3. Test Strategy
   - Test strategy is defined as an organizational level term,which is used for testing all the projects in the organization.
   - What are the various testing levels, testing types, and test design methodologies will use in the project

4. Assumptions

•

5. Risks

- The list of all the risks and the corresponding solution plans will be listed out here in this section.
- For Ex:
- Unable to deliver the software within deadline.
- Employees may leave company in the middle of the project.
- Unable to test all the features within time.
- Lack of experience.

6. Backup Plan/Mitigation

- What not to be tested has to be planned in case of imposed deadlines
- Proper training needs to be provided.
- Severity & Priority based testing.

7. **Roles & Responsibilities**

ABC	Test Manager	• Escalations
EFG	Test Lead	• Create the Test Plan and get the client signoffs • Interact with the application, create and execute the test cases • Report defects • Coordinate the test execution. Verify validity of the defects being reported. • Submit daily issue updates and summary defect reports to the client. • Attend any meeting with client.
HIJ	Senior Test Engineer	• Interact with the application • Create and Execute the Test cases. • Report defects
XYZ	Test Engineer	• Interact with the application • Execute the Test cases. • Report defects

8. **Schedule**

Following is the test schedule planned for the project –

Task	Time Duration
• Creating Test Plan	Dec 13th, 2022 to Dec 15th, 2022
• Test Case Creation	Dec 16th, 2022 to Dec 26th, 2020
• Test Case Execution	Dec 27th, 2020 to Jan 27th, 2021
• Summary Reports Submission	Jan 28th,2021

9. Tools

- What are all the tools used for testing project will mention here.
- Bug tracking tool : Jira
- Automation tool : Selenium java
- Test management tool : Testlink (used to store test related items such as test case, test plan etc.)

10. Environment / Test Bed

- Windows 10 – Chrome, Firefox, IE 11 and Edge
- Mac OS – Safari Browser
- Linux Ubuntu OS – Firefox
- Android Mobile OS – Chrome
- iPhone Mobile OS – Safari

11 Entry & Exit Criteria

- The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

- **Requirement Analysis**
- Entry Criteria:
- Once the testing team receives the Requirements Documents or details about the Project
- Exit Criteria:
- List of Requirements are explored and understood by the Testing team
- Doubts are cleared
- **Test Planning**
- Entry Criteria:
- Testable Requirements derived from the given Requirements Documents or Project details
- Doubts are cleared
- Exit Criteria:
- Test Plan document (includes Test Strategy) is signed-off by the Client

- **Test Designing**
- Entry Criteria:
- Test Plan Document is signed-off by the Client
- Exit Criteria:
- Test Scenarios and Test Cases Documents are signed-off by the Client
- **Test Execution**
- Entry Criteria:
- Test Scenarios and Test Cases Documents are signed-off by the Client
- Application is ready for Testing
- Exit Criteria:
- Test Case Reports, Defect Reports are ready

- **Test Closure**
- Entry Criteria:
- Test Case Reports, Defect Reports are ready
- Exit Criteria:
- Test Summary Reports

1. Test Automation

- What are all the frameworks are used in this project will list out here

1. Deliverables

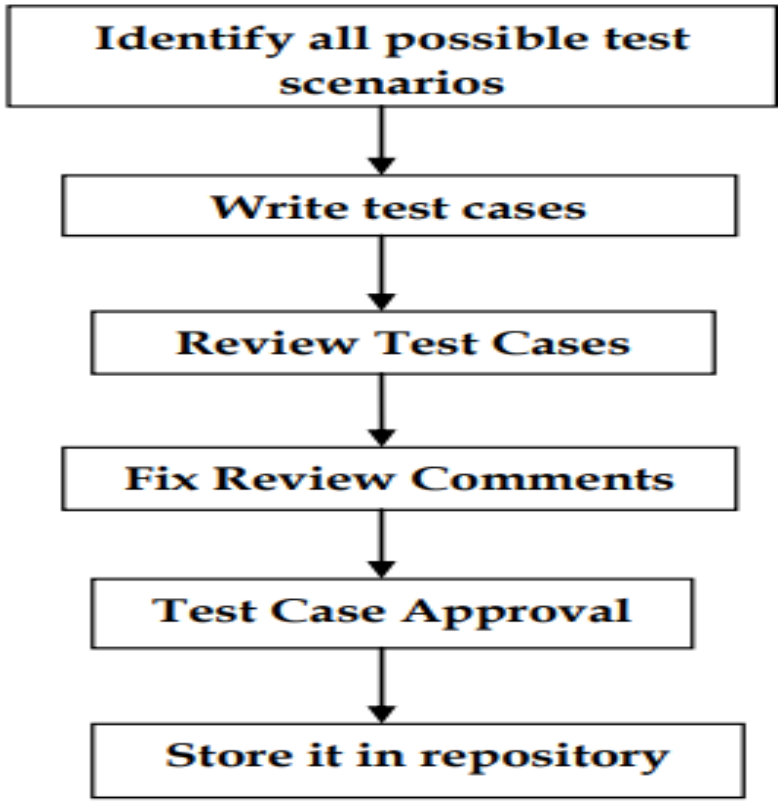
The following are to be delivered to the client:

Deliverables	Description	Target Completion Date
Test Plan	Details on the scope of the Project, test strategy, test schedule, resource requirements, test deliverables and schedule	Dec 15th, 2022
Functional Test Cases	Test Cases created for the scope defined	Dec 26th, 2022
Defect Reports	Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis.	NA
Summary Reports	Summary Reports – Bugs by Bug#, Bugs by Functional Area and Bugs by Priority	Jan 28th, 2023

- **Test Case Procedure**
- **The following steps to write test case.**
- **Test Case :**
- **A test case is document with all possible scenarios.**
- **It is collection of input and output test cases are written by**

Understanding requirements

- **We use different techniques to write test cases.**
- **Why we need to write test case**



1. **To Have better test coverage – cover all possible scenarios**

And document it, so that we need not remember all the scenarios.

2. **To have consistency in test case execution – seeing the test case and testing the product**
3. **To avoid training every new engineer on the product.**
4. **To depend on process rather than on a person.**

When do we write test case

Once requirements are available developer start developing and during this time parallelly testing team start writing test cases,once it is done they send it to test lead who reviews it and approve test cases Once developers finish developing product is given for testing , the test engineer then looks at the test cases and starts testing the product.

Test Design Technique / Black box test design technique

1) Boundary value analysts(BVA)
2) Error guessmg
3) Equivalencc class partitioning

1) Boundary value analysts(BVA)

- It's widely recognized that input values at the extreme ends of input domain cause more errors in system. more application errors occur at the boundaries of input domain. 'Boundary value analysis' testing technique is used lo identify errors at boundaries rather than finding those exist incenter of input domain
- Test cases for input box accepting numbers between 1 and l000 using Boundary value analysis ;
- 1) Test cases wi th test data exactly as the input boundaries of input domain i.e. values I and 1000 in our case.
- 2) Test data with val ues just below the extreme edges of input domains i.e. values 0 and 999.
- J) Test data with values just above the extreme edges of input domain i.e. values 2 and I00l.
- 4) test data with nominal value ie 500.

2) **Equivalence class partitioning :**

In this method the input data is divided into different equivalence data classes. This methos os typically used to reduce the total number of test cases to a finite set of testable test cases, still covering maximum req.

E.g : If you are testing for an input box accepting numbers from 1 to 1000 then there is no use in writing 1000 test cases for all 1000 valid input numbers plus other test cases for invalid data.

Using equivalence class partitioning method above test cases can be divided into three sets of input data called as classes. Each test cases is repetitive of respective class.

So in above example we can divide our test cases into three equivalence class of some valid and invalid inputs.

1. **One input data with valid input data, pick single value from range 1 to 1000 as valid test.**
2. **Input data class with all values below 1.**
3. **Input data with any value greater than 1000.**

3) Error Guessing.

The error guessing is a technique where the experienced and good testers are encouraged to think of the situation in which the software may not be able to cope.

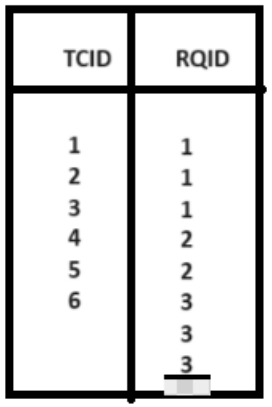
The success of error guessing is very much dependent on the skill of the tester, as good testers know where the defects most likely to be.

- **Traceability Matrix :**

Traceability matrix is a document which maps between requirements and test cases or defects and test cases.

We write traceability matrix to make sure that every req has at least one test cases.

Below figure shows that for the specified req no. which testcase is assigned for requirement no. “1” test cases id is “1”



TCID	RQID
1	1
2	1
3	1
4	2
5	2
6	3
	3
	3

Types of Traceability Matrix

1. **Forward Traceability Matrix :**

Traceability matrix is a document which maps between requirements and test cases or defects and test cases

1. **Backward Traceability Matrix**

Traceability matrix is a document which maps between test cases and requirements or defects and test cases

- **Test Execution :**
- **Test Engineer will do following during the test execution**

1. **He will perform the action that is described in the procedure column**
2. **He will observe the acutal behavior of the application**
3. **He will document the actual behaviour under the acuatl result column**
4. **Compare both expected result and actual result and enter the status**

After test execution report is created.

Sample test execution report :

Module Name	Total Test Cases	Total Executed	Total Pass	Total Fail	Pass %	Fail %
Register	200	150	110	40	73%	27%
Login	100	75	55	20	73%	27%
Logout	300	200	158	42	73%	21%
Total	600	425	323	102	76%	24%

- ***Defect Tracking life Cycle :***

Bug/defect tracking is process in which the defects are identified, isolated and managed

Mistake performed while writing code called as defect.

Defect can occure due to missing req. or extra req or wrong req.

When defects are identified by testers he will be reporting this defects to developers for fixing

The Communication can be done 2 ways :

1) Report defect in word document on daily bases and email to developer
2) Report using bug tracking tool

Bug tracking tools are software used ti communication defect information between testers and developers

Tools : Mantis,bugsilla,Jira etc…

Sample Defect reporting template :

1. Defect id :

Is is unique id given for every defect.

2. Defect Summary :

What exactly the defect is clearly mentioned here

3. Steps to Reproducibility :

The list of all the steps that are followed by the test engineer to identify the defect will be listed out here

4. Submitted / Author :

The name is the test engineer who has submitted the defect will be mentioned here

5. Date of Submission

The date on which defect is submitted

6. Version No :

Corresponding version no is mentioned here.

7. Build No :

Corresponding build no is mentioned here.

8. Assigned to

The development lead will fill the developers name for whom the defect is assigned.

9. Severity

How Serious the defect is defined in terms of severity

Note : Severity is The defect or bug is how much impact to the customer business, three types of sevarioty

Minor, Major, blocker or critical

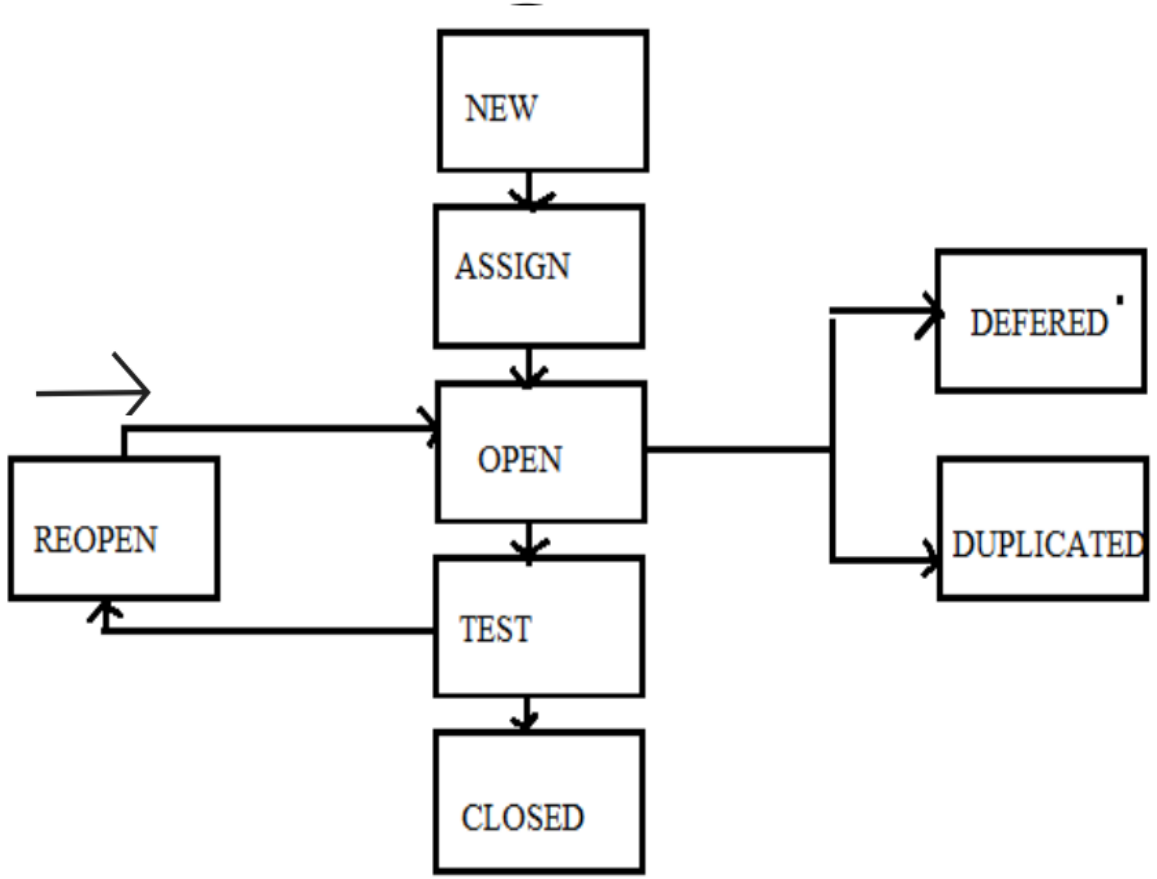
Screenshot:

Specified defect screenshots should be attached here.

Defect Life Cycle or Bug Life Cycle :

Customer gives requirements – developers are developing the s/w – testing team is writing test cases looking at the requirements Developer develops the product – test engine er starts testing the product – he finds a defect – now the TE must send the defect to the development team.

He prepares a defect report – and sends a mail to the Development lead saying "bug open".



Development lead looks at the mail and at the bug – and by looking at the bug – he comes to know to which development engineer developed that feature which had a bug – and sends the defect report to that particular developer and says "bug assigned".

The development engineer fixes the bug – and

sends a mail to the test engineer saying "bug fixed" –

he also "cc mail" to the development lead.

Now the TE takes the new build in which the bug is fixed and re-tests it again – and

if the bug is really fixed – then sends a mail to the developer saying

"bug closed" and also "cc mail" to the development lead.

Every bug will have an unique number.

If the defect is still there – it will be sent back as "bug reopen".

Note : PRIORITY

It is the importance to fix the bug or how soon the defect should be fixed or which are the defects to be fixed first.

Types of priority

1. **High**
2. **Low**
3. **Medium**

➢ **What is the procedure to write testcases :**

- **Requirement Study :**

  In this stage understand the software by reading requirements.

- **Generate Scenarios – In this case tester should think in what all ways that the end user will use software and should document all scenarios.**

- **Write Test Cases :**

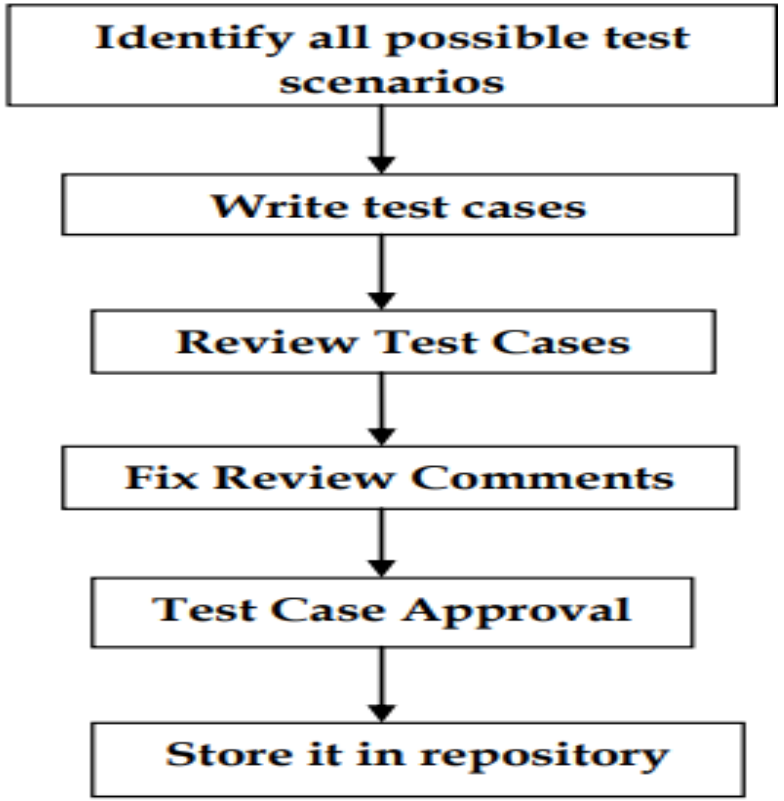
  1) Convert all the identified scenarios to test cases
  - **Group Scenarios related to 1 feature**
  - **Prioritize**
  - **Write Test cases**

  2) Use Test cases design techniques to write test case.

  3) Use Standard test cases template to write testcases

- **Review test case and fix comments :**

  Test case is written by one tester will be reviewed by other tester(Lead). and check is there any deviation in testcases as per req.



- **Test case Approval : Testcase is written for the modules will be send to test lead or project manager where they will approve it to execute while testing software.**

- **Test case repository :**

  Is shared folder created using version control.

- **NOTE**

For which and all types of testing do we write test cases ?

- **Smoke Testing** – here, we are testing basic features only. So we can pull out some test cases which has all the basic features – so we don't have to write test cases for this.
- **Functional Testing** – yes, we write test cases Integration Testing – yes, we write test cases
- **System Testing** – yes, we write test cases Acceptance Testing – yes, customer may write test cases
- **Compatibility Testing** – we don't write because the same test cases as above are used for testing on various platforms
- **Adhoc Testing** – we dnt write because they are "on the fly" or "random" ideas/scenarios. However, if we find critical bug, then we convert that scenario into test case.
- **Performance Testing** – we may not write test cases because we do this using tool.