- **Software:**
  - Software is a collection of specialized program which takes user input and generate desired output. **OR**
  - A Software is a collection of computer programs that helps us to perform a task.

- **Types of Software:**

  1) System software

  - Ex: Device drivers, Operating Systems, Servers, Utilities, etc.

  2) Programming software

  - Ex: compilers, debuggers, interpreters, etc.

  3) Application software

  - Ex: Web Applications, Mobile Apps, Desktop Applications etc.

- **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.
  - It is process of checking the correctness and completeness of the software with respect to the client expectation.

- **Software Quality Depends Upon:**
  - The product should bug - free.
  - The product should delivered on time.
  - The product should be within budget.
  - The product should meets requirements and expectations.
  - The product should maintainable.

- **Difference between Product and Project:**
  - If software application is developed for specific customer based on the requirement, then it is called **as project.**

- If software application is developed for multiple customers based on market requirements, then it called **as product.**

- **Why Software has the Bugs?**
  - If there is a Miscommunication or no communication between development and testing team.
  - Software complexity
  - If no of programming errors that result software have more defects.
  - If the frequently changing requirements from client.
  - Lack of skilled testers.

- **What are the 3 pillars of any company?**
  - P-People.
  - P- Process.
  - P- Product.

- **What is prototype?**
  - Prototype is nothing but the visualization of functionality before development.
  - When the requirement came from client, business analyst prepare a prototype model to show them(client).
  - After approval from client they will work on it as design, development, coding and testing.
  - Then build or application is deploy to the client side.
  - It is made with the help of Rally Tool.

- **Manual Testing:**

  - **Defines:** Check/Validation developed application will work as per the requirements.

- Check completeness and correctness of the functionality by manual.

- **Software Quality Assurance:**
  - SQA process defines to measure and monitor the developed application.
  - SQA define software quality assurance.
  - It is communication between business analyst and client.
  - **SQA process parameters:**
    - **Clients/ customer requirements full fill :**
      - What is the purpose of the project?
      - Which type of application customer want? (Web Based+ Mobile Base+ Domain)
      - Domains: Banking/Telecom/Insurance/Pharma/Ecom/Travel etc.
    - **Clients/ customer exception (security & performances):**
      - **Security/Privacy:** Let's consider an example for the banking domain, Software like banking gathers a lot of customer data which is very sensitive, so, in that case, clients want to security/privacy with respect to the all data.
      - **Performance:** Software/Application should work properly under the extreme load, Load balancing working proper.
      - **Maintenance:** Need to take care, If any problem occurs after delivering the project, Bugs/Issues are handled on time with respect to their priorities
    - **Cost of the project:**
      - In MNC companies project costs are counted on an hourly basis. Customers have to pay this payment depending upon the resource utilization as well as time complexity.
    - **Time Delivery:**
      - At the time of resources gathering and documentation, the expected build duration of the project gets decided.
    - **Escalation:**
      - If the company exceeds the delivery time then company have to pay the penalty/fine, it is known as escalation.

❖ **Team Size:**

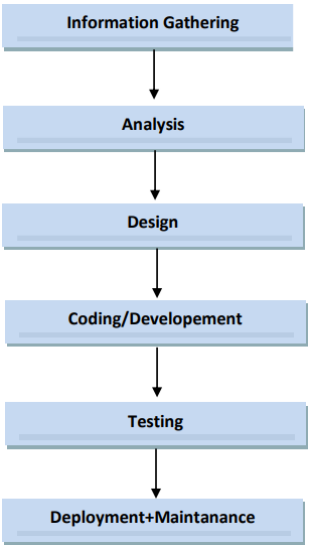
- **Project Team-** New features/ functionality/ Module. Ex. Paytm – Invest in stock.
    - Project team size is 24 to 26 people.
        - **Deliver manager (1) –** Project delivery to client.
        - **Project manager (1) -** Project task assign/ work need perform with the time/ team handling.
        - **Business analysis (1) -** BA interaction client and collect the requirement/ functionality/ New features.
        - **Designer/ Solution architecture (1) -** project application design.
        - **Developer (14 to 16) –** Developer will code for application.
        - **Tester (5 to 6) –** Tester will do testing on developer application.
- **Support Team –** Existing application issue/ defects/ end user quires/ Ticket Ex. Paytm- Recharge module
    - Support team size 9 peoples.
    - **Project manager/ Support manager (1) -** Support task assign/ work need perform with the time/ support team handling.
    - **Developer (5 to 6) –** Developer will code for application.
    - **Tester (1 to 2) –** Tester will do testing on developer application.

- **Who decide the process for software?**
    - If client have their own IT department then they will define the process or approach for software.
    - If not then our company will decide the process for software.
- **Types of Process Model for building the software?**
    1. Basic SDLC
    2. Waterfall Model
    3. V-Model
    4. Agile Methodology or model

❖ Basic SDLC or Software Development Life Cycle:

- This is the generic process for developing any software, Also it is the systematic approach to build the software as per the customer's requirement.

- **Phases of SDLC:**



- SDLC process involves development & testing of the application.

- **SDLC stage involved.**
    1. Information gathering
    2. Analysis
    3. Design
    4. Coding
    5. Testing
    6. Support/ Maintenance

- **Information gathering:**
    - Information gathering will be done by BA.
    - BA will collect the requirement from Client related to business of Client.

- In Information gathering stage BA will prepared a BRS documents.
- BRS – business requirements specification.
- BRS defines clients business related requirements.
- BRS documents will not get to tester.

- **Analysis:**
  - In Analysis stage BA will work.
  - In Analysis stage BA will collect the requirement from client.
  - BA will collect requirements related the functionality of application/ software.
  - In Analysis stage BA will prepared SRS (Software requirements specification).
  - SRS defines as functional requirements that will be implemented & system requirement that will be used.
  - SRS also called **FRS (functional requirements specification)/ CRS (Customer requirements specification).**
  - SRS will contains:
    - Functional requirements
    - Functional flow diagram
    - Use cases (1 specific requirement)
      - ✓ Description – Detail about the specific requirement.
      - ✓ Acceptance criteria – Does & don't about specific requirement/ Summary.
    - Screenshot/ Snapshot/ prototype
  - After completion of SRS documents BA will sent a mail these SRS documents to developer &tester.
  - Developer & tester will analysis/ understand/ study the SRS documents.
  - If documents is not clear then we will do the meeting with BA.
  - **Ex. Use cases-ZerodhaKite- Stock Module- Intraday buy & Sell form new window.**
    - **Description-** User has to buy or sell throw intraday window. For end user intraday added which can buy or sell share / stock. In intraday, user can hold stock/share within 1 day. I.e. within trading time as per your country.

- **Acceptance Criteria:**

1.	Intraday available in Stock module for all user.
2.	Intraday for active with trading day (trade cycle time – 9.15 to 3.15 pm).
3.	Throw Intraday user can 1st buy then sell OR 1st sell then buy.
4.	Intraday will provide margin for all stock/share.
5.	Intraday will provide margin for all future & options.
6.	After trade time, for Intraday error message will show – "Intraday not available you order will place after next trading day".

- **Design:**
  - In design stage designer or solution architecture will work.
  - Designer will prepared the design same as SRS documents.
  - Designer will prepared the HLD (high level design), LLD (low level design).

- **Coding:**
  - In Coding stage developer will work.
  - Developer will work on LLD (low level design).
  - LLD will implemented according to SRS/ as per Use cases testing.

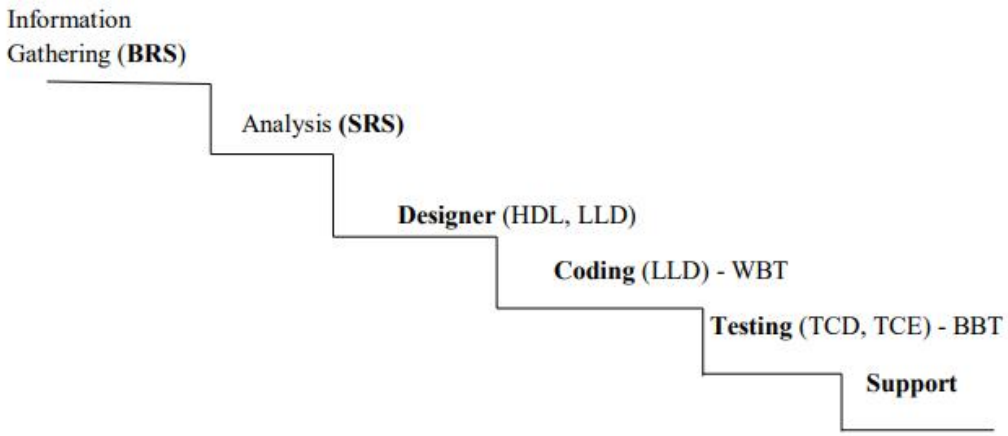
- **Testing:**
  - Tester will do the test cases design (TCD) & test case execution (TCE).
  - Tester will check functionality as per SRS documents OR as per use cases.
  - In TCE, if we found a defect/ bug, tester will raised to developer.
  - Developer will fix these defect/ bug.
  - At the end application/ software deploy/ delivery to the client.
  - Project team will give the warrant support for 1 month.
  -

- **Support/ Maintenance:**
  - After completion of project team warrant time/ period then project/ application went to support team.

- Existing application issue/ defects/ end user quires/ Ticket.

❖ Waterfall Model:

- Waterfall model/ process it is also sequential process.
- Sequential process after completion of one next stage will start with next stage.
- Ex. Until Developer will complete, tester will not start test testing.
- **Disadvantages of Waterfall Model:**
  - Requirement changes are not allowed.
  - If there is defect in Requirement that will be continued in later phases.
  - Total investment is more, because time taking for rework on defect is time consuming which leads to high investment.
  - Testing is start after coding so we have less time available to check the build.



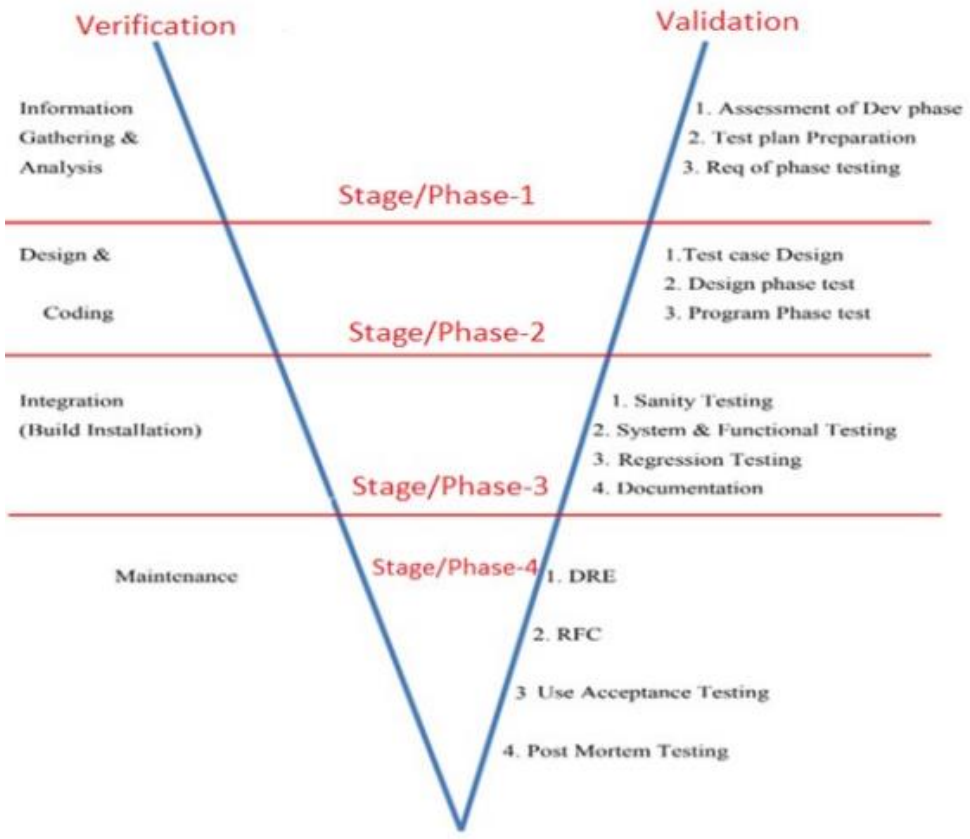
- **Advantages of Waterfall Model:**
  - Since Requirement changes are not allowed, chances of finding bugs will be less.
  - Initial investment is less since the testers are hired at the later stages.
  - Preferred for small projects where requirements are fizzed.

❖ V-Module:

- V-module stand for *Verification & Validation.*
- V-module / process Development (LCD) and testing (LCT) are doing parallel.
- Development & testing stages are work parallel.
- In V-module / process deployment process is around 3 month.
- V- Module it is Plan driven process (deployment process = 3 month)
- We can accept Change Request at any stage.
- In verification or Life Cycle development the phases are same as in Software Testing Life Cycle.
- But these stages are mapped with testing stages known as Life Cycle Testing or Validation.
  - **Life Cycle Testing:**
  - **In phase 1,**
    - **Assessment of development phase:**
      - ✓ In this step PM or TL involved.
      - ✓ They decide strategy &Methodology for the testing.
      - ✓ Methodology is nothing but which type of testing they preferred like manual, automation, API,Database etc.
      - ✓ Strategy is nothing but which type of programming language they preferred like java, ruby,python, junit, nunit etc.
      - ✓ They also take care of TRM.
      - ✓ They also take care of TRM it is nothing but test responsibility matrix.
      - ✓ TRM is the mapping of work & Resources.

    - **Test Plan Preparation:**
      - ✓ In this phase, PM and Test lead is involved.
      - ✓ In Test plan preparation, PM&QA lead is responsible for implementation of TRM.
      - ✓ They prepare test team & distribute the work to all team member.

- **Requirement of phase Testing:**
  - ✓ In this phase QA lead is involved
  - ✓ He decides how much test cycles are required and also decide time required for each test cycle.



- **In phase 2,**
  - **Test Case Design:**
    - ✓ In this phase QA are involved.
    - ✓ They prepared test cases & find out scenarios.
    - ✓ According to the requirements they consider positive as well as negative scenarios.
    - ✓ Test cases can be maintain with the help of excel based document or special tool like Test link.

- **Design Phase Testing:**
  - ✓ In this phase, Architecture & UI developer involved.
  - ✓ This testing ensure that build design is correct or not wrto requirements.
- **Program Phase Testing:**
  - ✓ In this phase, front end & back end developer involved.
  - ✓ This testing is also known as unit level or code testing.
  - ✓ From this testing they ensure that code is working as per expected or not.

- **In phase 3,**
  - **Sanity Testing:**
    - ✓ In this testing QAs are involved.
    - ✓ Here we check basic core functionality or build stability of the application.
    - ✓ If any critical issue or blocker they only we communicate with the developer, minor issue are not considered in sanity testing.
  - **System and Functional Testing (SIT):**
    - ✓ Here we execute test cases as per the step written in test case documents.
    - ✓ We perform functional & non-functional testing.
    - ✓ For each fail test case, we logged an issue.
    - ✓ InSIT, from small issue to big issue all are been logged and communicate with the developer.
  - **Regression Testing:**
    - ✓ In this testing QAs are involved.
    - ✓ Here we check the impact of newly added module with existing module.
    - ✓ In this testing, here we check only positive scenario's if time permits negative scenarios as well.

- **Documentation:**
  - ✓ Test Documents are created by the individual testers.
  - ✓ From this document we ensure that who did the testing on particular module & what was status of test case execution.

- ✓ After all individual documentation, Test lead create a document which is known as Test summary report, which contains all the test case execution records.

- **In phase 4,**
  - **Defect removal efficiency (DRE):**
    - ✓ It is a process which calculates at which level tester did the testing, while in BBT, regression and UAT.
    - ✓ **Formula of Calculate DRE:**
    - ✓ ***[defect found in SIT / (defect found in SIT + defect found in UAT)]*100***
    - ✓ Ideal value of defect removal efficiency is equal to one.
  - **RFC (Request for Change):**
    - ✓ If any change request come before the product release or during development of product we can easily accept that point should be highlight in BRS document.
    - ✓ As per the requirement client have to pay extra amount for that.
  - **UAT (User Acceptance Testing):**
    - ✓ After removing all open & logged defect found in SIT then product will move to UAT environment to perform UAT testing.
    - ✓ In UAT, here we check *end to end Functional flow of application.*
    - ✓ From UAT we ensure that the application is defect free & ready for production.
  - **Post Mortem Testing:**
    - ✓ If any unexpected behavior of the system before the product release at that time we communicate with the developer find the root cause and fix that issue at the same time.

- **In the Life Cycle Development:**
- **Phase_1:**
  - **Information Gathering and Analysis:**
    - ✓ BA collects the requirements from the client and create business

requirement specification document.

- ✓ BA converts business requirement specification document into the software requirement specification and send them to the team.
- ✓ SRS documents will contains the functional requirements, functional flow diagram, use cases and prototype.

- **Phase2 :**
  - **Design and coding:**
    - ✓ In design stage designer or solution architecture will work.
    - ✓ Designer will prepared the design same as SRS documents.
    - ✓ Designer will prepared the HLD (high level design), LLD (low level design).
    - ✓ In Coding stage developer will work.
    - ✓ Developer will work on LLD (low level design).
    - ✓ LLD will implemented according to SRS/ as per Use cases testing.

- **Phase_3:**
  - **Integration:**
    - ✓ It is performed by developer.
    - ✓ They merge or integrate new code with existing build of the application.
    - ✓ Once white box testing is done by developer, then they will add new code with existing build.
    - ✓ After successful integration they perform integration testing from this they ensure that code is successfully merge or not.

- **Phase_4:**
  - **Support/Maintenance:**
    - ✓ After deliver the product on client side, we need to take care the issue or defect are logged by client.
    - ✓ We have to handle that issue based on the priority of client.

- **What is Static Testing?**
  - Testing the project related documents is called *as static testing.*
  - It is also known *as verification.*
  - Verification checks whether we are building the right product.
  - This typically involves review, walkthrough, inspection.
  - Static testing / verification we will do/ check ***quality assurance.***
  - Static testing involves ***review, walkthrough, inspection*** etc.
  - **Review:**
    - **Requirement reviews:** business analyst will check their documents.
    - **Design reviews:** designer will check their design whether it is according to the requirements.
    - **Code reviews:** developer will check their code or logic.
    - **Test plan reviews:** Test lead will check the test plan.
    - **Test cases reviews:** The individual tester will check their test cases whether he covers all the functionality of the application or not.
  - **Walkthrough:**
    - It is 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.
    - At least 2 people are required to perform the review.
  - **Inspection:**
    - It’s a most formal review type.
    - In which at least 3- 8 people will sit in the meeting 1- reader 2-writer 3- moderator plusconcerned.
    - Inspection will have a proper schedule which will be intimated via email to the concerned developer/tester.

- **What is Dynamic Testing?**
  - It is also known *as Validation.*
  - Validation checks whether we are building the ***product right.***
  - Dynamic testing involves the actual testing of the software.
  - Dynamic testing takes *place after verifications* are completed.
  - Dynamic testing / validation we will do/ *check quality control.*
  - It focuses on the software.
  - Dynamic testing involves testing such as unit testing, integration testing, system and functional testing and user acceptance testing.
  - **Advantages:**
    - Testing is involved in each and every phase.
  - **Disadvantages:**
    - Documentation is more. Initial investment is more.
- **Difference between Quality Assurance and Quality Control?**
  - **Quality Assurance:**
    - QA focuses on building in quality.
    - QA is preventing defects.
    - QA is process oriented
    - QA for entire lifecycle.
    - It is nothing but the verification.
  - **Quality Control:**
    - QC is the actual testing of the software.
    - QC focuses on testing for quality.
    - QC is detecting defects.
    - QC is product oriented.
    - QC for testing part in SDLC.
    - It is nothing the validation

Questions on process model?

- **How will you receive the project requirements?**
  - The finalized SRS will be placed in a project repository; we will access it from there.
- **What will you do with SRS?**
  - SRS stands for software requirement specification.
  - SRS is used to understand the project functionality from business and functional point of view.
- **Is the testing team involved in SRS preparation?**
  - Business analyst prepare the SRS document by interacting with the client.
  - However a senior testing team member can also be involved in requirements collections along with the development team and the business analyst team.
- **How does your requirements document look like?**
  - It contains lots of use cases where each use case explains one or more functionalities.
- **How will you understand the requirements?**
  - If it is known domain by going through use cases I can understand the requirements.
  - If I have some queries, I will discuss them with business analyst for clarifications.
  - If it is new domain, first I will get domain training them I go through the use cases.
  - If the project requirements are very confusing, then (BA) can also walk through each use case.
- **How do u understand functionality without screens?**
  - We get wireframes in the use cases which helps a lot to understand the functionality.
- **What is wireframe?**
  - A diagram which stimulates the feel of the actual screen.
- **What is use case?**
  - Use case explains the step by step procedure of how a particular functionality of s/w is used by the end user.
  - **Use case contains sections such as.**
    - Use case id.
    - Use case name.

- Description.
- Flow of events.
- Alternative flow of events.
- pre, post conditions

- **What will be the problem without SRS?**
  - Without SRS we will not be able to understand the project features correctly.
  - Hence we will not able to test the project in depth and deliver the best quality product.
- **Where you involved in writing the use cases?**
  - I am aware of how use cases looks like and I can write if required.
  - But I have never got an opportunity to write the use cases because these are prepared *by requirements gathering team.*
  - Any how I have reviewed the use cases of certain functionalities and have given my inputs for betterment of the same.

- **What are the different sections present in SRS?**
  - Scope
  - Features
  - User characteristics
  - Software requirements
  - Hardware requirements
  - Performance requirements
  - Use cases
  - Security and reliability requirements
- **How long do u spend on understanding SRS?**
  - It depends on the familiarity of the domain and complexity of the project.
  - If it is a familiar domain, we can understand around 25 pages of the documentation every day. For a new complex domain, we manage around 15 pages per day.

❖ Agile Methodology:

- It is an iterative and incremental approach.
- Agile methodology follows iterative and incremental process.
- **Agile Process:**
  - Customers do not have to wait for long time.
  - We can develop, test and deploy the piece of software to the stakeholder with few features.
  - We can accept the requirement changes.
  - There is a good communication between product owner, stakeholder, and scrum master and development team.
  - Agile defines continuous development & continuous testing will happen in application/ build.
  - Agile process it is value driven process.
  - If any change request comes at any point of time then we will consider these change request.
  - When will accept these CR then we will check impact on current development, Testing & production.
  - If change request has more impact on development/ testing/ production then we will inform to client.
  - If change request has less impact on development/ testing/ production then we will develop & testing and we will deploy to client.
  - In Agile process deployment process in 2 week.
  - Agile subtypes/ framework/ methodology
    - **XP (extreme programming)** – Development & no testing.
    - **Scrum -** (Bunch of requirement then Sprint wise development & Testing & Delivery).
    - **Kanban –** Support team (tickets/ Existing issue/ bugs/CR).
    - **Lean -** Support team.
    - **FDD –** Future driven development.
  - I have worked in Scrum agile methodology.

- **Advantages of Agile Methodology:**
  - Requirement changes are allowed at any stage of development.
  - Release will be very fast.
  - Customers no need to wait for long time.

- Good communication between team.
- It is an easy model to adopt.
- Automation is possible is in agile process.
- Checkpoints are added after each module.
- Agile ceremonies is better advantage by using this we can track the project.

- **Disadvantage of Agile Methodology:**
  - If frequent changes in requirement/ US, we can't deployment/ delivery these US to client.
  - If your application/ module is depend on another application/ module, we can't deployment/ delivery these US to client.
  - Less focus on the design and documentation since we deliver the software very faster.
- **Terminologies used in agile:**
  - **Agile:** It is continuous process of development & testing.
  - **Sprint:** It is collection of requirement or bunch of user stories.
  - **Burn down Chart:** How much user story remaining with respect to time. In simple words, how much work remaining with respect to time.
  - **Burn up Chart:** How much user story completed with respect to time.
  - **Estimation:** Time slot provide to complete the work or User Story.
  - **Velocity:** Amount of work completed within sprint with respect to the estimation. It define sprint wise how much deployment of user stories to client.
  - **Epic:** It is nothing but the Main Module.

- What is **done** & **ready** Status?
  - **Done:** US is totally work completed for development & testing .US deployed to client then Userstory status is Done.
  - **Ready:** : US is totally work completed for development & testing .US is not yet deployed toclient then User story status is Ready.

❖ Agile Ceremonies:

- **Grooming Session:**
  - **Host:** Scrum Master
  - **Involvement:** scrum master, product owner, development team.
  - **Happens when:** before the sprint planning meeting.
  - **Duration:** 1-1.5hrs
  - **Discussion:** Here PO explains about the all user stories to team, those user stories which aregetting shortlisted for the sprint.

- **Sprint Planning Meeting:**
  - **Host:** Scrum Master
  - **Involvement:** scrum master, product owner, development team.
  - **Happens when:** before the sprint start.
  - **Duration:** 2-2.5hrs
  - **Discussion:** User Stories shortlisted here. Discussion about the user stories. Everyone sharetheir about the time, approach, strategy for development and testing. Based on this story points gets decided to each user stories.

- **Daily Standup Meeting:**
  - **Host:** Scrum Master
  - **Involvement:** scrum master, product owner, development team.**Happens when:** Everyday
  - **Duration:** 15-20min
  - **Discussion:** Here everyone provide their work update to the scrum master. If any blocker inwork then communicate with the team and get the solution.

- **Sprint Review Meeting:**
  - **Host:** Scrum Master
  - **Involvement:** scrum master, stakeholder, product owner, development team.
  - **Happens when:** at the sprint end.
  - **Duration:** 30-60 min
  - **Discussion:** Here we have to provide demo on what we completed within the sprint to the stakeholder. Then Stakeholder provide their review to the team .This is the time for team celebration for their accomplishment.

- **Sprint Retrospective Meeting:**
  - **Host:** Scrum Master
  - **Involvement:** scrum master, product owner, development team.
  - **Happens when:** at the sprint end.
  - **Duration:** 60 min
  - **Discussion:** Here we discuss about the previous sprint. What good things we have done .What bad things we have done .Here we set the goal for next sprint.

- ## Agile Architecture:
  - **Stakeholder:**
    - They are top most body in the company.
    - They have all the requirement related to the project.
    - They can ask for change request at any phase of development.
  - **Project Owner:**
    - Collect all the information from stakeholder.
    - Project owner create product backlog based on the requirements share by the stakeholder.
    - They select user stories according to the client priority.
  - **Sprint Backlog:**
    - It is prepared by the product owner.
    - In Analysis stage product owner will prepared sprint backlog.
    - Sprint backlog defies as functional requirements that will be implemented & system requirement that will be used.
    - Sprint Backlog will contains:
      - Functional requirements
      - Functional flow diagram
      - Use cases (1 specific requirement)
        - ✓ Description – Detail about the specific requirement
        - ✓ Acceptance criteria – Does & don't about specific requirement/ Summary
      - Screenshot/ Snapshot/ prototype.
    - After completion of Sprint Backlog documents product owner will sent a mail these documents to developer & tester.
    - Developer & tester will analysis/ understand/ study the sprint backlog documents.
    - If documents is not clear then we will do the meeting with product owner.
  - **Design:**
    - In design stage designer or solution architecture will work.
    - Designer will prepared the design same as SRS documents.
    - Designer will prepared the HLD (high level design), LLD (low level design).

- **Estimation:**
  - In Estimation project owner, development lead, test lead is involved.
  - Estimation is having 3 factors: (knowledge, efforts, Complexity)
  - **Knowledge:**
    - Whenever team formation is done, each team member should have knowledge about domain of the product.
  - **Efforts:**
    - High level management decides how much efforts & resources are required. As per dependencyof module. They select user story.
  - **Complexity:**
    - This is the important factor in Estimation for defining complexity. They consider time, cost, resources requirements, Bandwidth of team.

- **Coding:**
  - In Coding stage developer will work.
  - Developer will work on low level design.
  - Low level design will implemented according to SRS/ as per use cases.

- **Testing:**
  - Tester will do the test cases design & test case execution.
  - Tester will check functionality as per software requirement specifications documents or as per use cases.
  - While test case execution, if we found a defect/ bug, tester will raised to developer.
  - Developer will fix these defect/ bug.
  - At the end application/ software deploy/ delivery to the client.

- **Support/ Maintenance:**
  - After completion of project team warrant time/ period then project/ application went tosupport team existing application issue/ defects/ end user quires/ ticket.
  - Project team will give the warrant support for 1 month.

- **Difference between scrum and Kanban**

Scrum	Kanban/lean
Scrum will follow in project team.	Kanban will follow in support team.
Team member will defined work assigned	Team member not defined work assigned.
In scrum will work in sprint 2 week.	Kanban will work on CR
Sprint deployment in 2 week.	In Kanban CR & defects a work deployment 2-3 weeks
In scrum priority of work already decided	In Kanban priority of work not decided.

- **Scrum:**
  - Scrum is a framework through which we build the software product by following agile principles.
  - Scrum normally includes the group of people called as scrum team. Normally 8-9 peoples in scrum team.
    - Product owner
    - Scrum master
    - Development team
    - QA team

- **Product owner:**
  - Define the feature of the product.
  - Prioritize features according to the market value.
  - Adjust features and priority of every iteration, as needed.
  - Accept or reject the work results.

- **Scrum master:**
  - The main role is facilating and driving the agile process.

- **Development and QA Team:**
  - Develop and test the software.

- **Sprint:**
  - It is nothing but the period of time required to complete user story decided by product owner and team. Usually it is 2-4 weeks.

- **Story Points:**
  - It is rough estimation of user story is given by developer and QA in the form of Fibonacci series.

Questions on Agile Methodology?

- **What are the different artifacts in agile?**
  - Product backlog, sprint backlog, burndown chart, burn up chart etc.

- **How you will decide the estimation in Sprint planning meeting?**
  - In my project, PM will open the Agile Board (JIRA/HAPLM) in sprint planning meetingthen PM will ask to every developer & tester about the time/ story point for every US which assigned to you.
  - Estimation will defines depends upon the number of factors:
    - How much knowledge you have against user stories.
    - How much efforts will be required.
    - How much complexity of the user stories.

- **What is Sprint zero?**
  - For next sprint preparation whatever the extra effort will be required for preparation so it will called *as Sprint Zero.*

- **When are a delivery/ deployment in your project?**
  - In my project we are working in Scrum Agile methodology, where we will work for two week/ three week (Monday to Friday).
  - In my project we will Delivery/ deployment on – Saturday/ Monday.

- **Who will do the Delivery/ deployment?**
  - In my all project, developer will do deployment to all environment & to the client.

- **What is your approach if US is not completed (development/ testing) in current Sprint?**
  - In my project, PM will tell to developer & tester will work on Saturday & Sunday and complete the work (More chances work will be complete).
  - But still If still work will not completed in Saturday & Sunday then PM will prepared a US innext sprint – US= “Reaming Work In Last Sprint”

- **Different types of technology in your project**
  - Frond end is Dot net languages
  - Services is Java languages
  - Backend/ Database is SQL Server

- **What is your approach, when SIT environment is crash?**
  - If client want Deployment with the sprint and SIT environment is crash, then tester will access the UAT environment.
  - UAT environment access throw remote desktop.

- **What is your approach, defect is not fixed in current sprint? Or if we got defect in last 1hr in last day?**
  - If we got issue/ defects (high priority/ core functionality) in last of sprint in testing.
  - Developer & tester work extends (Friday) – try to fix & tester will test.
  - If in Friday, work is not completed so developer & tester have to work on Saturday & Sunday.
  - There will more chances to complete the user story.
  - Still the defect have not been resolve by developer
  - Developer & test & deployment to client if defects have not been fixed in Saturday & Sunday work.
  - Then we will inform to PM these defect will take more time. PM will inform throw Mail to client and these defects will be fixed in next sprint.
  - In next sprint PM will create US **– "Carried over Bug".**

- **What problem you have faced in your carrier? OR What problem you have faced in your last project?**
  - If we have less knowledge about project domain.
  - If we have less time for testing.
  - If we have less resources / less tester.
  - If developer is not communicate properly/ developer making silly mistake.
  - If we have problem in environments.
  - If we have frequent changes in requirements.

- **What is product backlog?**
  - The product backlog is a kind of bucket or source where all the user stories are kept.
  - This is maintained by the product owner.
  - The product backlog can be imagined as a wish list of the product owner who prioritizes it as per the business needs.

- During the planning meeting, one user story is taken from the product backlog, then the team does the brainstorming, understands it and refines it and collectively decides which user stories to take, with the intervention of the product owner.

- **What is sprint backlog?**
  - Based on the priority, user stories are taken from the Product Backlog as one at a time.
  - The Scrum team brainstorms on it determines the feasibility and decides on the stories to work on a particular sprint.
  - The collective list of all the user stories which the scrum team works on a particular sprint is known as Sprint backlog.

- **What is story points?**
  - Story points are a quantitative indication of the complexity of a user story.
  - Based on the story point, estimation and efforts for a story are determined.
  - A story point is relative and not absolute. In order to make sure that our estimate and efforts are correct, it's important to check that the user stories are not big. The more precise and smaller is the user story, the more accurate will be the estimation.
  - Each and every user story is assigned to a story point based on the Fibonacci series (1, 2, 3, 5, 8, 13&21). Higher is the number, the complex is the story.
  - **To be precise**
    - If you give 1 / 2 / 3 story point it means that the story is small and of low complexity.
    - If you give points as 5 / 8, it is a medium complex and
    - 13 and 21 are highly complex.

- **Explain the term 'increment'?**
  - When the team finishes the sprint, hopefully, they have completed everything they forecasted.
  - The sum of all the product backlog items which were completed in a sprint is called **increments.**
  - This new increment also has the value of increment of the previous sprints.

- **What is Sashimi?**
  - Sashimi in scrum methodology means every phase of the software development cycle in a sprint which includes requirement analysis, planning & design, development, testing, documentation is complete or not and the product is ready to be displayed, etc.

- **What are impediments?**
  - Any hindrance which prevents the smooth flow of work or due to which the team is not able to perform its task in a better way is what we call **'impediments'.**
- **What are the principles of agile testing?**
  - **Some major principles of agile testing are:**
    - Customer satisfaction.
    - Bug-free clean code.
    - Changes are welcome by customer.
    - Whole team, business people and developers work collectively.
    - Instead of lengthy documentation, focus on the essence.
    - It focuses on face to face conversation.
    - It promotes sustainable development.

- **What are the main roles in the scrum?**

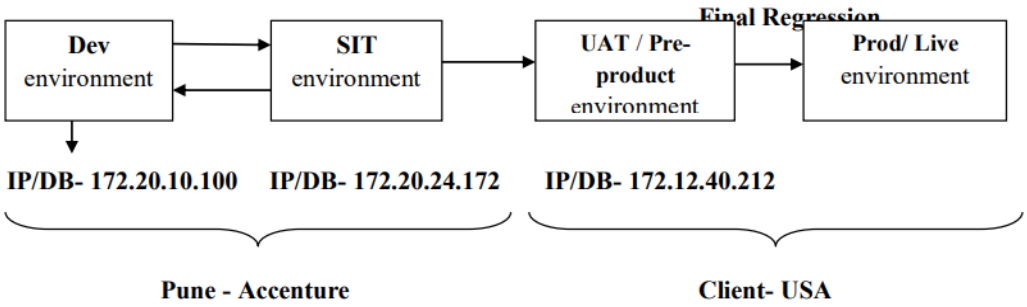
- **Scrum Team**: Scrum team is made by an individual person who works collectively to achieve a particular task. The team works in a bond to deliver committed and requested products.
- **Scrum Master**: Scrum Master is responsible for the proper execution or working of the scrum team. Being a servant – leader and a coach, he ensures the proper productivity of a team towards scrum sprint goal.
- **Product Owner**: The product owner has the responsibility to deliver a complete picture of what to build and to convey that idea to the team.

- **What are the different artifacts in scrum?**
  - **There are three scrum artifacts:**
    - **Product Backlog:** The product backlog includes all the items that will be delivered for the product. This list is constantly evolving. The commitment of the product backlog is the product goal.
    - **Sprint Backlog:** Sprint backlog is a list of all items committed to being delivered within a sprint. Once decided, the sprint backlog cannot be changed. The commitment of the sprint backlog is the sprint goal.
    - **Product Increment:** Product increment is usually delivered at the end of the sprint, which is a workable increment of the overall product.

- **How do you calculate a story point?**
  - A story point is calculated by taking into consideration the development effort+ testing effort + resolving dependencies and other factors that would require to complete a story.

❖ Testing Environments:

- In my project 4 types environment
  - **Dev environment -** developer work +coding + whit box testing
  - **SIT environment (System integration testing)** – tester test case design, test case execution, defect and defect report
  - **UAT environment** - Client location (Developer + tester)
  - **Prod/ Live environment** -End user application



- **Dev environment:**
  - Developer will do coding & white box testing.
  - Developer will sent the build deploy into SIT environment.
  - Dev environment (feature 250 line code) then do build delivery (250 line) to SIT environment (feature-250 line code).
  - Developer will sent Mail to Tester (throw JIRA/ HAPLM)
  - In Mail Attachment – Unit testing documents which contains screenshot for testing white box testing, table's name, URI/URL, etc.
- **SIT environment**:
  - Tester will work.
  - Tester do test case design documents and test case execution accordingly.
  - In TCE if we found defects then we will raised to developer.
  - Inform throw Mail (JIRA/ HAPLM).
  - Defect will fix by developer in dev environment then they sent Modified Build.

- Tester will check defects
- In SIT environment while TCE, tester will prepared the test proof then send email to UAT team throw JIRA/ HPAML tool.
    - **Dev environment URL = 172.20.10.100:8080 /https://www.flipkart.com/**
    - **SIT environment URL = 172.20.24.172:8080/ https://www.flipkart.com/**

- **Dev environment Testing:**
    - Unit Testing
    - Integration Testing
- **SIT environment Testing:**
    - Sanity Testing/ Smoke Testing
    - System & functional Testing (BBT)
    - Re-testing
    - Regression Testing, etc.
- **UAT environment Testing:**
    - Alpha Testing
    - Beta Testing
- **Production/live environment testing:**
    - Production Issue
- **Dev environment-**
    - In Dev environment, developer will work.
    - Developer will do coding as per user story.
    - After completion of coding developer will do testing (WBT)
        - Unit Testing
        - Integration Testing

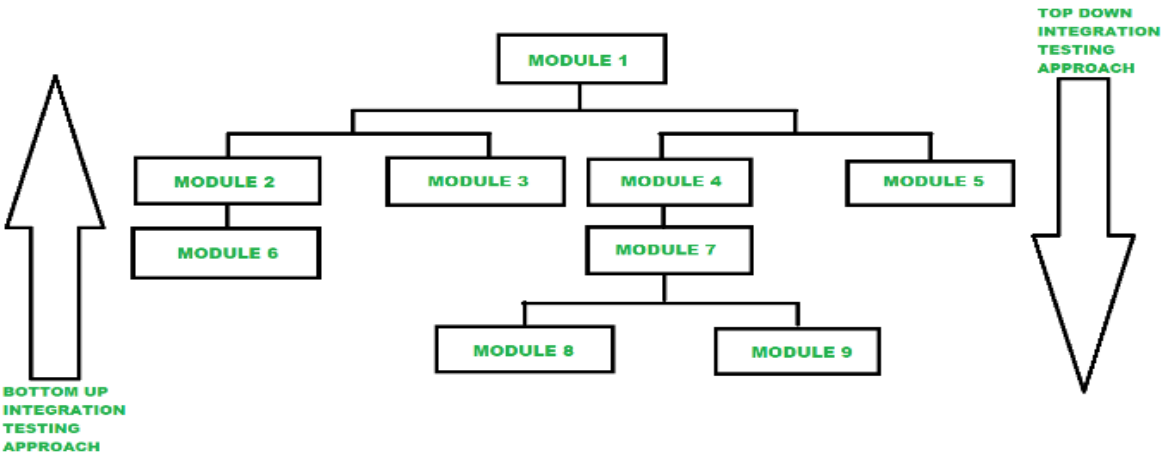
- **Unit Testing:**
  - A unit is a single component or module of a software.
  - Unit testing conducts on a single program or single module.
  - Unit testing is white box testing technique.
  - Unit testing is conducted by the developers.
  - Unit testing documents will contains screenshot for testing WBT, Tables Name, URI/URL, etc.
  - **Unit testing techniques:**
    - ✓ Basis path testing (Every path is tested).
    - ✓ Control structure testing
    - ✓ Conditional coverage (will check all conditions in program will be verified by true and false)
    - ✓ Loops Coverage
    - ✓ Mutation Testing

- **Integration Testing**:
  - Integration testing performed between 2 or more modules.
  - Integration testing focuses on checking data communication between multiple modules.
  - Integrated Testing is white box testing technique.
  - **Types of integration testing:**
    - Incremental integration testing
    - Non- Incremental integration testing

  - *Incremental integration testing:*
  - Incrementally adding the modules and testing the data flow between the modules.
  - Incremental Integration Testing is having 3 approaches:

    - **Top Down -Incremental integration Testing:**

      - Incrementally adding the modules and testing the data flow between the modules and ensure module added is the child of previous module.
      - If we don't have sub-module then developer will prepared stub.
      - Stub- It is temporary program prepared by developer throw XML languages



- **Bottom up -Incremental integration Testing:**
  - Incrementally adding the modules and testing the data flow between the modules.
  - And ensure module added is the parent of the previous module.
  - If we don't have main-module then developer will prepared driver.
  - Driver - It is Temporary program prepared by developer throw XML languages.

- **Sandwich or hybrid Approach:**

  - Combination of top-down & bottom-up approach is called as sandwich approach.

  - Here main-module and sub-module is not implemented so in that case developer will have to create driver and stubs to test the functionality of the browser.

- **Non-Incremental integration Testing:**

- Adding all the modules in a single shot and test the data flow between modules.
- *Drawbacks of non-incremental integration Testing:*
  - We might miss data flow between some of the modules.
  - If you find any defect, we can't understand the root cause of defect.

SIT environment:

- Tester will work.
- Tester do test case design documents and test case execution accordingly.
- **SIT environment Testing:**
  - Sanity Testing/ Smoke Testing
  - System & functional Testing (BBT)
  - Re-testing
  - Regression Testing, etc.

Sanity Testing:

- When developer will sent us a new build and inform throw e-mail (JIRA/ HPALM)
- In SIT environments, Sanity testing it is a first testing which is performed, sanity testing is called **zero level testing.**
- When we got the build, tester will check build stability i.e. checking either these build is stablefor testing or not
- Sanity testing also called as ***tester acceptance testing, build acceptance testing.***
- In Sanity testing we will check:
- ***Validation of the core functionality of application/build:***
  - ✓ Here we mainly check the core functionality with respect to the requirement present in the user story.
  - ✓ Here we check icon, button from which user can proceed to next step.
  - ✓ For Ex. Login Button, Test and agree button, skip, Ok buttons etc.
- ***Validation of the GUI of application/build:***
  - ✓ From the graphical user interface, users first interaction takes place
  - ✓ Hence here we check logo size, dimension of pages is correct or not, also check the images&clear or blurry and its quality.
- ***Validation of the links of application/build:***
  - ✓ As per the requirements there are hyperlink are present in the particular module.
  - ✓ While performing the sanity testing, those are active and redirected to expected functionality

- ***Validation of the tabs of application/build:***
  - ✓ In tab validation here we check the behavior of web elements which are working or not.
  - ✓ Web elements is nothing but the text box or search bar etc.
  - ✓ In Tab Validation here we provide input to these field.
  - ✓ Ex. enter user name /password, search bar, enter any text.
- ***Validation of the Page Navigation of application/build:***
  - ✓ In page navigation validation we check the page flow, whether the functionality is behave as expected or not.
  - ✓ In Sanity Testing if build is not stable for testing then tester will reject the build.
  - ✓ In sanity testing, tester will not create log or defect.
  - ✓ In sanity testing, if build is not stable for testing then we will reject build and inform to developer throwMail (JIRA/ HPALM).
  - ✓ In sanity testing, tester will not write test cases.
  - ✓ In Sanity testing build is not stable or Sanity testing issue due to SIT environment problem,system hung out, run time problem, core functionality/ basic functionality, etc.
  - ✓ For Sanity testing, we required 2 to 3 hr. for every new build.
  - ✓ In sanity testing, only tester is involved.

Smoke Testing:

- Smoke testing it is advance version of sanity testing.
- Smoke testing performed when we got the new build, tester will check build stability i.e. checking either these build is stable for testing or not .
- In Smoke testing we will check:
  - Validation the core functionality of application/ build
  - Validation the GUI/ UI of application/ build
  - Validation the link of application/ build
  - Validation the tab/ pages
  - Validation the navigation

- In smoke testing if build is not stable for testing then tester will reject the build and tester will give/ find the root cause of the defect/ issue.
- In Smoke testing = sanity testing + troubleshooting >> >done by tester.
    - In Smoke testing = sanity testing + package validation >>>done by developer.
- **How to do Troubleshooting**
    - Login into application or open application under test.
    - Pass data or verify functionality of application.
    - If we found error or defect then we will go error logs (172.20.24.172:C:\application\Windows NT\Action\Logs.txt)
    - Logs.txt file will contains all details about defect/ issue.
    - In smoke testing, tester will not write test cases.
    - In smoke testing build is not stable or smoke testing issue due to SIT environment problem,system hung out, run time problem, core functionality/ basic functionality, etc.
    - In smoke testing, if we found issue then we found root clause of the defect/ issue and inform to developer throw email (JIRA/ HPALM).
    - In smoke testing, 2 to 3hr for every new build.
    - In smoke testing, tester & developer both involved.
    - In my project, we are following smoke testing.

System & functional Testing (BBT):

- Testing over all functionality of the application with respective client requirements.
- It is a black box testing technique.
- This testing is conducted by testing team.
- After completion of component and integration level testing's we start system testing.
- Before conducting system testing, we should know the customer requirements.
- *System testing focusses on below aspects.*
    - **Usability testing**
    - **Functional testing**
    - **Non-Functional testing**
    - **Performance Testing**
    - **Security testing**

- **User interface testing (GUI):**
  - 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, fonts, sizes, icons, content, and images.
  - **GUI Testing Checklist**
    - Testing the size, position, width, height of the elements.
    - Testing of the error messages that are getting displayed.
    - Testing of the font whether itis 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.
    - Testing UI Elements like button, textbox, text area, check box, radio buttons, drop downs links etc.

- **Usability testing:**
  - Usability testing, tester we will validate user friendlessness of the screen/ application.
  - **Usability testing 2 type:**
  - **GUI(graphical user interface) / UI(user interface) Testing:**
  - **In GUI/ UI testing, as tester validation**.
    - Validation look & feel of the screen / application.
    - Validation ease to use (End user ease) of screen / application.
    - Speed of interface.

- **Manual support Testing-**
  - In manual support testing, as tester we will validate, how manual text will be support in screen / sensitiveness on screen.

Functional testing:

- Functionality is nothing but behavior of application.
- Functional testing talks about how your features should work.
- In Functionality, tester will validation internal feature of the application/ build.
- For validating internal feature we will performed different coverage.
- **Functional testing having testing converges:**

Behavioral coverage testing-

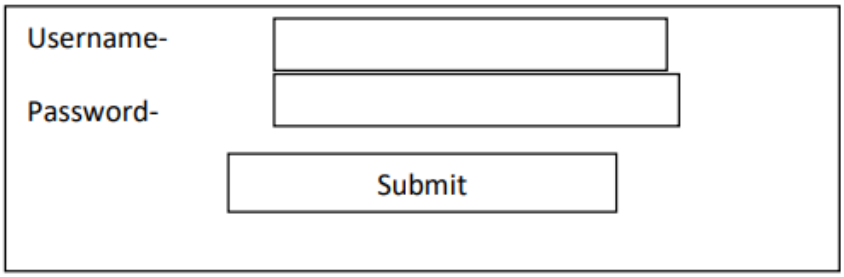
- Tester will test/validate, behavioral of the object/ Web elements present in application.
- Tester will check behavioral of the object/ Web elements & property of the object/ Web elements.

Object/ Web elements	**Behavioral & Property**
Text box	Focus & Un- focus
Check box	Check & un-check OR Tick & Un-tick
Button	Enables & Disables
Radio button	On & off

Input domain converge testing:

- Tester in input domain coverage testing, tester validation different input size / length & datatypes.
- For input domain coverage testing maintaining these size/ length & data types
- **Input domain coverage has 3 techniques:**
- ***Boundary value analysis, equivalent class partitioning, decision table technique.***
- **Boundary value analysis:**
  - BVA we will check input size / length.
  - Ex. Login page- Username- only mobile no, password- 4 to 6

character combination (capital, smallletters & No.)



BVA (Size/Length)	Pass	Fail
Username Text Box	**10 digits**	**8, 11 digits.**
Password Text Box	**4 digits**	**3 digits**
	5 digits	**7 digits**
	6 digits	**8 digits**

- **equivalent class partitioning:**
  - Equivalent class partitioning validate type of data that passes through object.
  - Ex. Login page- username- only mobile no, password- 4 to 6 character combination(capital letters, small letters & 2 No.).

ECP (data type)	Pass	Fail
Username Text Box	**0-9 (int)**	**A-Z, a-z, Special character**
Password Text Box	**A-Z, a-z, 0-9**	**Special character(4 to 5 length)**

- **Decision table technique:**

- Validation different input combination on the object/ web elements
- Ex. Ex. Login page- Username- only Mobile no, Password- 4 to 6 charter combination (Capital,Small letters & 2 No.), Submit Objects Rule 1/ input combination 1.

Objects	Rule 1/ input combination 1	Rule 2/ input combination 2	Rule 3/ input combination
Username	Valid	Valid	In-Valid
Password	Valid	In- Valid	Valid
Submit	Press	Press	Press
Result	Home	Error	Error

- **Error handling converge testing:**
  - Validating different types error will generated in object of application.
  - If we will pass invalid/ wrong data into object/web elements.
  - Tester verifies the error messages while performing incorrect actions on the application.
  - Error messages should be readable.
  - User understandable/simple language.

  - e.g.: **Incorrect data** - - (not specific what is actual wrong)
  - Invalid Username / Password - - (user can understand what was wrong)

- **Backend coverage testing/ database converge testing-**
  - Validating frond end operation that data stored into backend.
  - Backend coverage testing also called database testing
  - DML operations (data manipulation language) such as insert, update, delete, select etc.
  - Table & column level validations (column type, column length, number of columns...) Relation between the tables (Normalization)
  - Functions, stored procedures, triggers, indexes, views etc.

- **Calculation/Manipulation Testing:**
  - *Verify arithmetic calculation with respect to the user perform actions.*
  - Tester should verify the calculations
  - e.g. – Banking application – amount credited and debited should reflect in account.

- **Service based converge testing:**
  - Validating sequential operation/ end to end flow of the application.

- **Links Existence & Links Execution:**
  - Where exactly the links are placed Links existence.
  - Links are navigating to proper page or not Links execution
    - ✓ **Internal Links**
    - ✓ **External Links**
    - ✓ **Broken links**

- **Cookies and sessions:**
  - 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 (If you are idle for some time)

- **Non-Functional testing:**
  - In Non-Functionality, tester will validation External feature of the application/ build.
  - Non- Functionality testing, different testing types:
  - **Recovery coverage testing:**
    - In recovery coverage we ensure that whether the system is able to recover from abnormalsituation to normal situation.
    - *Ex. Filpkart- Payment page/ tab-while doing payment close or refresh the page -close application-add to card page/ tab.*
  - **Compatibility coverage testing (Browser Compatibility testing):**
    - In this we check whether the build/application is compatible with the user expected platform or not.
    - Here we deal with the User Expected Platforms such as operating system and browser.
    - ***Forward Compatibility:***
      - ✓ Here check the application compatibility for **newer version of browser or operating system.**
      - ✓ If we have problem in operating system of SIT environments then Network team handle that issue not tester.
    - ***Backward Compatibility:***
      - ✓ Here we check the application compatibility **for older versions of browser or operating system.**

- ✓ *Backward Compatibility testing having two types:*
- ✓ In my project I have done "Browser Compatibility testing".
  - ***Cross browsing Compatibility testing:***
    - Validating application/ build it support different browser.
    - Ex. Different browser – Chrome, Firefox, Opera, Edge, safari, etc.
  - ***Version control Compatibility testing:***
    - Validating application it support for different version on same browser.
    - Ex. Chrome browser – V90.10, V89.00, V85.00, V70.00, etc.

♦ **Configuration coverage testing/ hardware coverage testing:**

- Validating the application I supporting the hardware (Print, Bluetooth, etc.).
- Configuration coverage testing also called hardware coverage testing.

♦ **Intersystem coverage testing-:**

- In the Intersystem coverage we check how our application is application sharing data with another.

♦ **Sanitation/Garbage testing:**

- In Garbage/sanitization coverage we eliminate extra features/ fields/buttons/text added by developer and which is not part requirement.
- If those extra features are discussed with the client and if he agree to continue with the same in that case we will continue with the extra feature or else we have to remove them.
- If any application provides extra features/functionality then we consider them as bug.

- **Globalizations coverage testing-**
  - Validating application is supporting to all languages.
  - **Local:** In this we test the application in the local language which is specific to region.
  - For example: Marathi, Guajarati, Punjabi, and Bengali. Etc.
  - **International:** In this we test the application in the international language which is specific to the country.
  - **Global:** In this we test the application in the global Language which is followed by whole world.
  - **For regional testing we will use Google translator.**

Performance testing:

- This testing check the overall performance of the build/application.
- In Performance testing we check the Load, Stress, Volume etc.
- **Load Testing:**
  - Increasing the load gradually on application then checking the speed of application. It is done by the tools like Load Runner, Jmeter etc.
- **Stress Testing:**
  - Suddenly increase/decrease the load on application and check the speed of application(increase by 10 members – decrease by 20 – again increase the load all suddenly)
  - e.g., E-commerce site while on sale.
- **Volume Testing:**
  - Check how much data is able to handle by the application.

Security Testing:

- How secure our application (done by skilled tester not by normal testers)
- **Authentication:** Users are valid or not
- **Authorization / Access control:** Permissions of the valid user.
- **What Is Authentication?**
  - Authentication is the act of validating that users are whom they claim to be
  - This is the first step in any security process.
- **Complete an authentication process with:**
- ***Passwords:*** Usernames and passwords factors. If a user enters the correct data, the system assumes the identity is valid and grants access.

- ***One-time pins:*** Grant access for only one session or transaction.
- ***Authentication apps:*** Generate security codes via an outside party that grants access.
- ***Biometrics:*** A user presents a fingerprint or eye scan to gain access to the system.

- **What Is Authorization?**
  - Authorization in system security is the process of giving **the access a specific resource or function.**
  - Giving **someone permission** to download a particular file on a server or providing individual user with **administrative access to an application** are good examples of **authorization.**
  - In secure environments, **authorization must always follow authentication.** Users should first prove that their identities are genuine before an administrators grant them access to the requested resources.

Re-testing:

- In Re-testing tester validating the functionality of the application by passing multiple test data.
- Testing which is performed by passing multiple test data, these testing called retesting.
- Test data we will get from Databases (SQL server).
- If we have old project then we have existing data in databases by firing the SQL queries we get the test data.
- If we have new project then for testing we required test data then tester in SIT environment create the test data.
- If we have project which test data depends on other application.
- Ex. Phone pay – feature UPIID created, mobile no., bank No., debit card, etc. these data related to bank then these test data will provided by BA.
- Whenever the developer fixed a bug, tester will test the bug fix is called re-testing.
- Tester closes the bug if it worked otherwise re-open and send to developer.
- To ensure that the defects which were found and posted in the earlier build were fixed or not inthe current build.

- **Example**
- ~ Build 1.0 was released, Test team found some defects (Defect Id 1.0.1, 1.0.2) and posted.
- Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

Regression testing:

- Testing conducts on modified build to make sure there will not be impact on functionality because of changes like *adding/deleting/modifying* features.
- In re-testing or in BBT, when tester found a defects then tester will raised to developer.
- Then developer fix or resolved the defect and developer will sent Modified build and tester willdo the testing (regression testing).
- Regression testing – Re-execution of test on modified build, to check defect has been fixed/work properly & there is no side impact on interconnected modules.
- *Regression testing = Regret + Action.*
- Regression testing is having 3 types:
  - **Unit regression testing**
    - Testing only the changes/modifications done by the developer.
  - **Regional regression testing**
    - Testing the modified module along with the impacted modules
    - Impact analysis meeting conducts to identify impacted modules with QA & Dev.
  - **Full Regression**
    - Testing the main feature & remaining part of the application.
    - Ex: Dev has done changes in many modules, instead of identifying impacted modules.
    - We perform one round of full regression.

- **Regression testing will performed in 2 times**
  - **In SIT environments** - if we found defects
  - **Final Regression testing-** When application is moving from one environment to anotherenvironment.
  - *In Regression testing we will execute these test cases and it is termed as*

Regression suite.

- *Regression Suite test case are considered as:*
  - Failed test cases.
  - High priority test cases.
  - Extra features test cases or Extra functionality test cases.

User Acceptance Testing/ Pre-product Testing:

- When UAT testing, will start after completion of BBT in SIT environment.
- Tester who is working in SIT environment prepared the test proof and sent to UAT team.
- UAT (user acceptance testing) also called *customer acceptance testing.*
- UAT team works- 2 developer + 1 tester (client location).
- Here we check end to end functional flow of the application with respect to the client requirements.
- Throw remote desktop, we will access UAT sever & do the testing.
- Tester will not work in both environments (SIT & UAT).
- User Acceptance Testing is having 2 types:
- **Alpha testing.**
  - Alpha testing is perform on controlled environments which is known as in-premises testing.
  - The alpha testing with client we check end to end functional flow with development team.
  - As UAT perform before production if any issue/defect/change request added while performingthe UAT, development team will fixed at same time.
- **Beta Testing:**
  - Beta testing is perform on uncontrolled environments which is known as out-premises testing.
  - Beta testing is performed by the end user, perform is in random manner, if found any unexpected behavior of the application then according to the user feedback we will change the functionality of the application.
  - In one module I have performed SIT testing & in another module we have performed UAT testing

- If we found issue/defect in UAT testing, UAT tester will raised these defects to respective developer & SIT Tester.
- SIT tester, he re-produce the defects in SIT environment.
- If defect found in SIT environment then we will inform developer and say that fix the defect ASAP
- If defect not found in SIT environment then SIT tester shows Test proof to UAT tester.

Production testing or Hotfix:

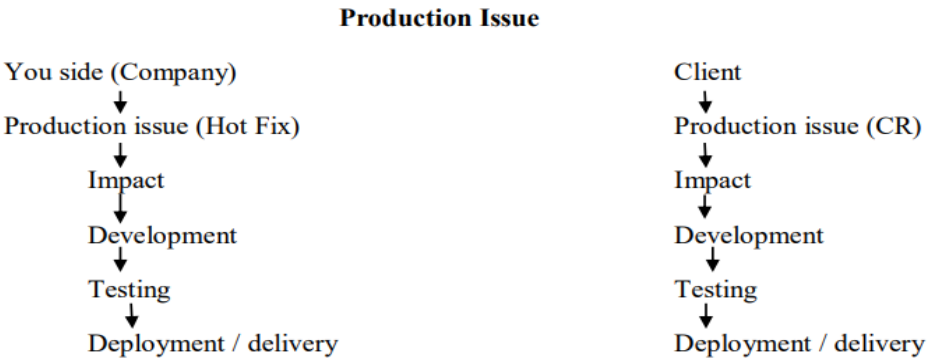
- After completion UAT, then developer will deployed the code/ build from UAT to production.
- If we found a defects in production environment these defects called production defect.
- **Production defect will occurred due to 2 reasons:**
- If Tester have missed functionality while doing the testing, then these production defect is termed as **"Hot Fix".**
- If Client has missed some functionality and these defects found in production issue, then these production defect **"CR (Change request)"**
- If Tester have missed functionality while doing the testing, then these production defect "Hot Fix"

A. If defects/ issue found in production then client will sent an "Escalate Mail" to project team tester will re-produce the defects in SIT environment.

- If defect found in SIT environment then he will inform to developer.
- say fix these issue ASAP and deployment to client (Project Manager ask to Tester sent me apology mail / Reasoned why these has been happed against these production issue)

B. If defects/ issue found in production then client will sent an "Escalate Mail" to project team.Tester will re-produce the defects in SIT environment.

- If defect is not present in SIT environment then tester will email to project manager.
- Developer & Designer, these issue is not present in SIT environment & attached test proof.
- Developer will check the deployment process/ environment problem/ Configuration file then Developer will handle these issue & deployment of client.



- If **client has missed some functionality and** these defects found in production issue, then these production defect **"CR (Change request)".**
- If we will get change request from client then we will accept these change request but if change **request is impact more on development, testing & production.** Then we will inform to the client that Change request we **have considered in next sprint.**
- If we will get change request from client then we will accept these change request and if change request **impact is less then we have to work on** change request then development, testing & production.
- Then change request development & Testing and deployment to client.
- Hotfixes are also known as QFE update quick-fix engineering updates, which can be normally,we will create a hotfix that need to be fixed immediately flow.

- ***Hotfix address very specific issues like:***
  - Adding a new feature.
  - Bug, or security fix.
  - Changing database schema

❖ Testing Terminologies:

- Testing we will scenario/condition:
  - Monkey testing
  - Exploratory testing.
  - AdHoc testing

- **Monkey Testing:**
  - Purpose of monkey testing is break the system code to find out number of defects.
  - When we have less time for testing or test case execution.
  - Then we will follow random approach to perform testing which results random execution of test cases.
  - For performing monkey testing domain knowledge is required.
  - While test case execution we have to consider only positive scenarios whatever the priority of test case like high, medium, and low.
  - If we got issue/defects in monkey testing then we can't deployed user story to client.
  - If client wants build on urgent basis then developer and tester do have to work on Saturday and Sunday there will more chances to complete the work on change request and user story is deploy to the client.
  - Still user story is not completed then project manager will inform the client that user story we have consider in next sprint to work on.
  - Monkey testing is performed in following condition
  - If build is move form SIT environment to UAT environment within 2 hrs.
  - If build is moved to the production environment within 2 hrs.

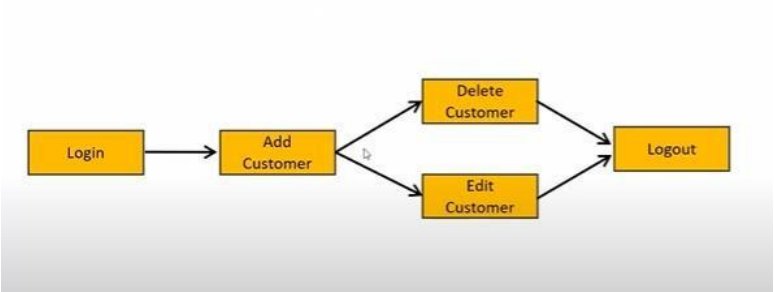
- **Exploratory testing:**
  - Purpose of exploratory testing is to gain the domain or product knowledge.
  - When any new member is working on project who don't have knowledge about the domain.
  - They by referring old test case and execute them to gain knowledge about the domain.
  - In exploratory testing, we consider all test case for execution.
  - Domain knowledge is not required to perform exploratory testing.
  - By perform exploratory testing we learn and analyze the product.

- **Adhoc testing:**
- If we knowledge about the domain but we don't have or less test data then we perform Adhoc testing.
- When we don't have test cases for testing.
- Domain knowledge is required.
- Testing is based on the domain or product knowledge only.
- In Adhoc testing, No any documentation, test case design etc.

❖ End to End Testing :(Integration testing)

- Testing the overall functionalities of the system including the integration among all the modules is called as **end to end testing.**
- End to end testing is also cal**led as Integration testing.**
- Integration testing will performed on complete module.
- Integration testing performed by Tester
- **End to End Testing is as follows:**

  - Login
  - Add new customer
  - Edit customer
  - Delete customer
  - Logout



❖ Globalization Testing:

- Performed to ensure the system or software application can run in any cultural or local environment.
- Different aspects of the software application are tested to ensure that it supports every language and different attributes.
- It tests the different currency formats; mobile number formats and address formats are supported by the application.
- For example, Facebook.com supports many of the languages and it can be accessed by people of different countries. Hence it is a globalized product.

❖ Localization Testing:

- Performed to check system or software application for a specific geographical and cultural environment.
- Localized product only supports the specific kind of language and is usable only in specific region.

- Its testes the specific currency format, mobile number format and address format is working properly or not.
- For example, Baidu.com supports only the Chinese language and can be accessed only by people of few countries. Hence it is a localized product.

❖ Positive Testing:

- Testing the application with valid inputs is called as **positive testing.**
- It checks whether an application behaves as expected with positive inputs.
- **For example -**
- Enter Only Numbers
- 12345
- **Positive Testing.**
- There is a text box in an application which can accept only numbers. Entering values up to 99999 will be acceptable by the system and any other values apart from this should not be acceptable.
- To do positive testing, set the valid input values from 0 to 99999 and check whether the system is accepting the values.

❖ Negative Testing:

- Testing the application with invalid inputs is called as **Negative Testing.**
- It checks whether an application behaves as expected with the negative inputs.
- **For example:**
- Enter only numbers.
- Aced
- Negative Testing.
- Negative testing can be performed by entering characters A to Z or from a to z.
- Either software system should not accept the values or else it should throw an error message for these invalid data inputs.
- ***Positive & Negative Test Cases:***
- ***Requirement:***
- For Example, if a text box is listed as a feature and in FRS it is mentioned as Text box accepts 6 - 20 characters and only alphabets.

- **Positive Test Cases:**
  - Textbox accepts 6 characters
  - Textbox accepts up to 20 chars' length.
  - Textbox accepts any value in between 6-20 chars' length.
  - Textbox accepts all alphabets.

- **Difference between smoke testing and sanity testing?**

Sanity testing	**Smoke testing.**
When we get new build from developer.	If we found defect in sanity testing then we perform smoke testing.
It is also called as zero level testing/tester acceptance testing.	It is combination of sanity testing and package validation. Sanity testing troubleshoot = tester Sanity testing package validation=developer
Here we validate-core functionality, GUI ,link, pages, tab etc.	Validation of troubleshooting and package validation.
If we found defect we reject the build.	If we found defect we are finding the root cause of defect and sent mail to developer.
Performed by tester	Performed by tester as well as developer.

- **What is your approach, when SIT environment is crash?**
  - If client want Deployment with the sprint and SIT environment is crash, then tester will access the UAT environment.
  - UAT environment access throw remote desktop.

- **What unit testing documents will contains?**
  - Unit testing documents which contains screenshot for testing white box testing, table's name, URI/URL, etc.

- **Difference between alpha testing and beta testing?**

Alpha testing	**Beta testing**
Alpha testing is performed on service or web based application. Example: whatappweb, paytm, swiggy etc.	Beta testing is perform on product or desktop based application. Example: adobe reader , VLC media player, MSOffice etc.
In UAT, for alpha testing, **developer and tester are involved.**	In UAT, for beta testing developer and tester are not involved **End user perform this testing.**
If defect found in build then immediate fixed.	If defect found in build it will fixed in next version of application.
Client interaction is more is in alpha testing.	End user interaction is more In beta testing.
Alpha testing we check system and functionality of build.	Beta testing we check system and functionality of build.

❖ Priority and Severity:

- Priority & Severity terms it will be defined against defect.
- Priority term defines defect **impact on client business**
- Severity term defines defect **impact on functionality application.**
- **Severity:**
  - **Defect status on Severity critical, high, and medium & low.**
  - **Critical: This defect indicates nothing can proceed further.**
    - Ex: Application crashed, Login Not worked.
  - **High: The main/basic functionality is not working.**
    - Customer business workflow is broken. They can't proceed further.
    - Ex1: Fund transfer is not working in net banking.
    - Ex2: Ordering product in ecommerce application is not working.
  - **Medium: It cause some undesirable behavior, but the feature/application is still functional.**
    - Ex1: After sending email there is no confirm message
    - Ex2: After booking cab there is no confirmation.

- **Low: It won't cause any major break-down of the system.**
    - Ex: Look and feel issues, spellings, alignments, logo mistakes.

Priority:

- Defect status on priority high, medium & low.
- Defect Priority states the order in which a defect should be fixed.
- Priority describes the importance of defect.
- **High:**
    - **The defect must be resolved immediately** as it affects the system severely and cannot beused until it is fixed.
- **Medium:**
    - *It can wait until a new versions/build is created.*
- **Low:**
    - *Developer can fix it in later releases.*

- **Examples:**

- **Low priority-Low severity:**

- A spelling mistake in a page not frequently navigated by users.
- Ex. Spelling mistakes in application (Ex. Submit text changes – Ok text)
- Ex. Button color is not proper (Ex. Button Blue color into red color)
- Any cosmetic or spelling issues which is within a paragraph / page.

- **High Severity- Low Priority:**

- Application crashing in some very corner case.
- Application rarely used functionality is not working (Ex. Paytm –PromoCode).
- Invoice download is not working in application.
- Web page not found when user clicks on a link (user does not visit that page generally).

- **High priority-Low severity:**
  - Slight change in logo color or spelling mistake in company name.
  - Application text are overlapped with each other.

- **High priority-High severity:**
  - Login page is not working- defects**.**
  - Core functionality application in project – defects.

❖ Basic Terms:

- **Error:**
  - If some wrong coding happens application, if we found mistakes in the application error
  - It can be done by developer during coding.

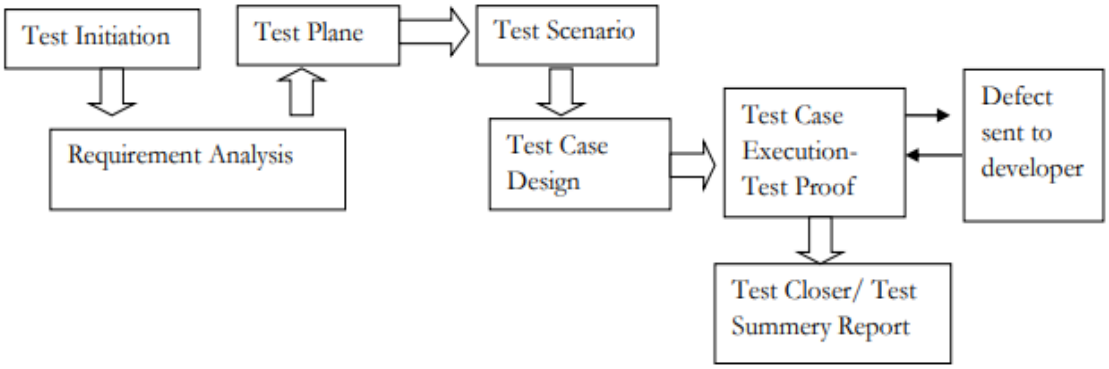
- **Defects:**
  - It can be define as variation between actual and expected results.
  - It is deviation of customer requirements.

- **Bug:**
  - If developer is accepting these defects then it is called Bug.

- **Issue:**
  - If we found major bug in the application these bug, it is called issue.

❖ Software Testing Life Cycle (STLC) –

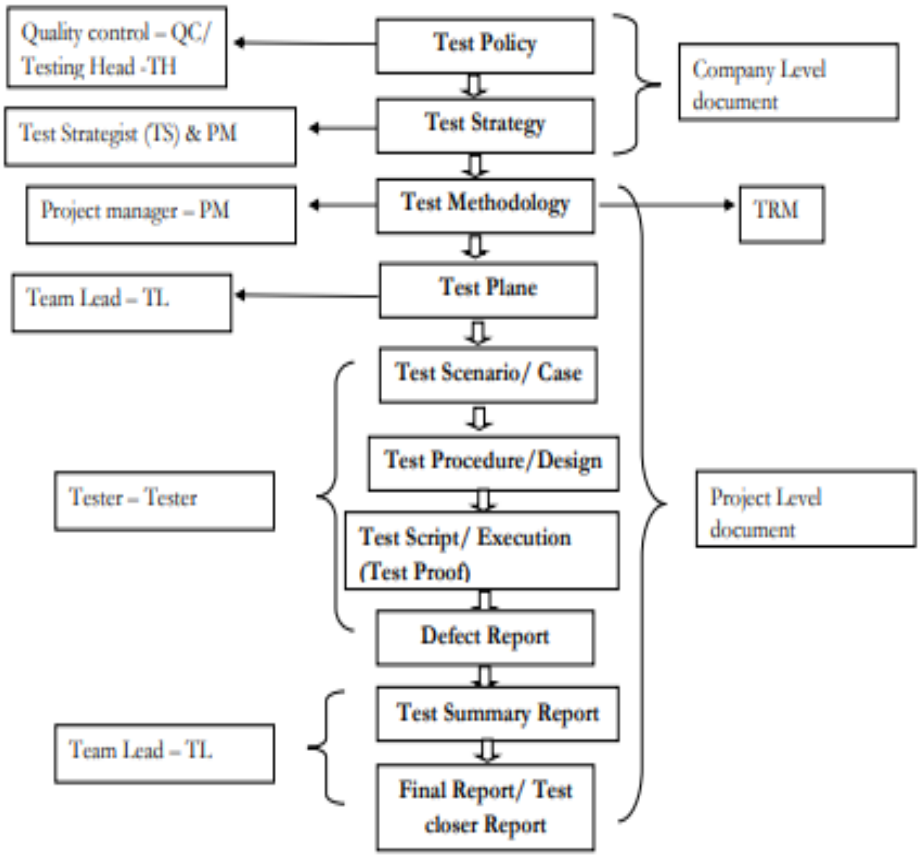
What is your organization Test process?



- **Test Initiation:**
  - In test initiation stage project manager will work.
  - Project manager will prepared test responsibility matrix.
  - After completion of TRM, PM will sent to Test lead.
- **Requirement Analysis:**
  - In this step project manager and test lead involved.
  - They analyze the requirements based on this they select strategy and methodology for Testing.
  - Strategy is nothing but languages such as java, ruby, python, c# etc.
  - Methodology is nothing but testing like manual, automation, database, api etc
- **Test Plan:**
  - Test plan will prepared by test lead.
  - Test plan will contains – Job allocation, resource allocation & estimations.
  - Test plane will contains – who will test, when to test, how to test etc.
- **Test Scenario & Test Case Design:**
  - From a user story tester will identify test scenario.
  - Test cases design from test scenario.
  - Test cases design we will write in excel sheet.
- **Test Case Execution:**
  - In TCE, when we got build from developer then we will start the TCE.
  - In TCE, if we found defect then we will create defects in JIRA/TFS tool & inform to developer throw e-mail.
  - After fixing defects we will perfumed re-testing & regression testing.
- **Test Closure/ Test Summary Report:**
  - Test closure / test summary report will be prepared by test lead/ team lead.
  - Test closure / test Summary report will contains Sprint the US – TCD, TCE, TCS, defect, etc.

- **What is your organization Test documentation?**

  - **<u>Test document hierarchy</u>**



  - **Test Policy:**

    - In test policy stage, testing head & quantity control peoples will work.
    - In test policy documents defines objective of the project.
    - Simple term how much revenue we will generated from the project.
    - Test policy documents, it is company level documents.

  - **Test Strategy:**

    - Tests strategist person will work in test strategist.
    - Test strategist documents defines which strategy we will consider for the project.
    - It is company level documents.

- **Test Methodology-**
  - Tests methodology documents will be prepped by project manager.
  - Test methodology defines mapping between development stages & testing factor.
  - Test methodology stage project manager will papered a TRM (Test reasonability matrix) documents.
  - TRM documents also called as Test matrix.
  - It is project level documents.
  - While preparing TRM document, focus on parameters
  - **Functional / requirements of application**
  - **Risk in the Project.**

Development (Stage) → **Testing factor ↓**	BRS, SRS	Design	Coding	Testing	Support
Requirement phase Testing	Yes	Yes	Yes		
UI Testing	No	Yes	Yes	Yes	Yes
Functional Testing	No	No	Yes	Yes	Yes

- **Test Plan:**
  - Test plan will prepared by test lead.
  - Test plan will contains – Job allocation, resource allocation & estimations.
  - Test plane will contains – who will test, when to test, how to test etc.

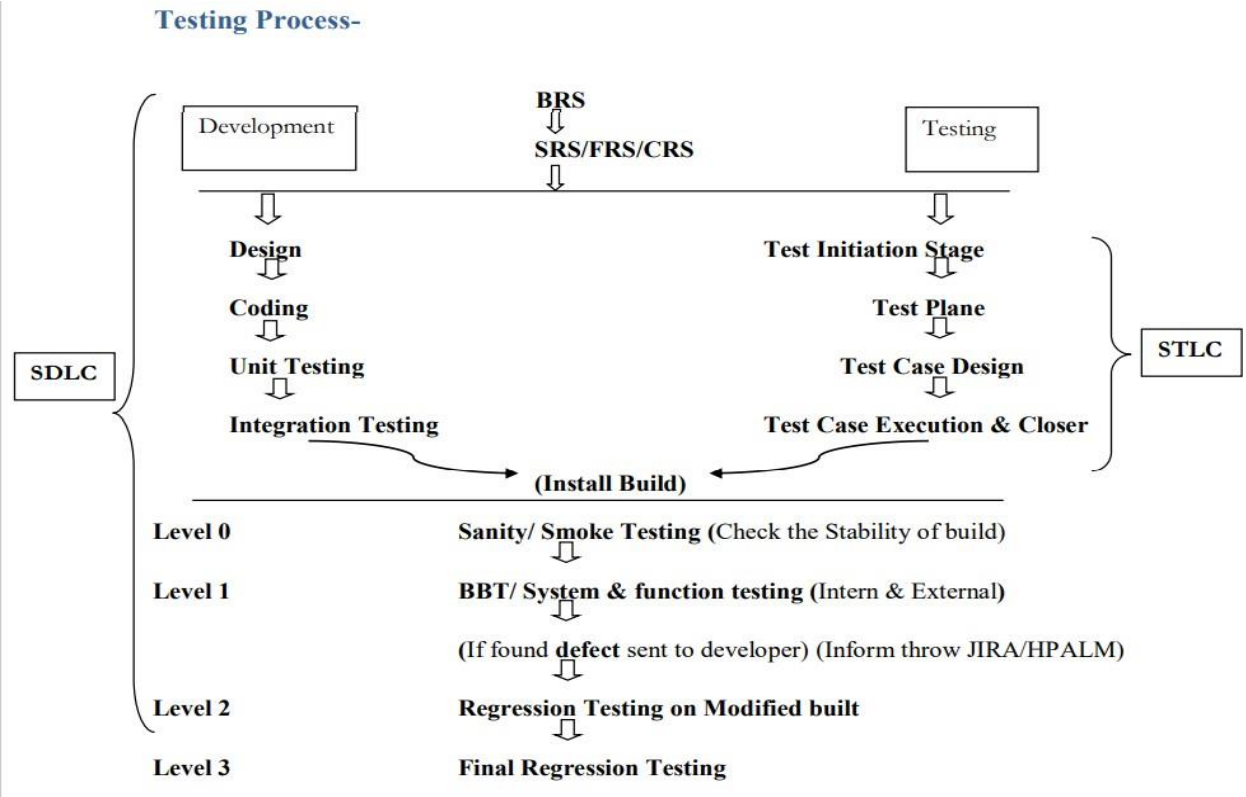
- **Test Scenario & Test Case Design:**
  - From a user story tester will identify test scenario.
  - Test cases design from test scenario.
  - Test cases design we will write in excel sheet.

- **Test Case Execution:**
  - In TCE, when we got build from developer then we will start the TCE.
  - In TCE, if we found defect then we will create defects in JIRA/TFS tool & inform to developerthrow e-mail.
  - After fixing defects we will perfumed re-testing & regression testing.

- **Defect Report:**
  - In TCE, if we found defects, we will created defects in JIRA/ TFS.
  - Inform to developer through email.
  - After defects fix then we will performed e-testing & regression testing.
  - It is project level documents.

- **Test Closure/ Test Summary Report:**
  - Test closure / test summary report will be prepared by test lead/ team lead.
  - Test closure / test Summary report will contains Sprint the US – TCD, TCE, TCS, defect, etc.
  - It is prepared by test lead.

- **Explain the development and testing process that followed in organization?**



- **Explain test plan that followed in your organization?**

- **Test Plan:**
  - Test plane will be prepared by Test lead.
  - **While preparing Test planes will consider following point.**
    - **Job allocation:**
      - Team lead assign work to individual tester like who work in SIT, Regression and UAT environment.
      - Assign work to individual tester in their respective environment.
    - **Resource allocation:**
      - Here test lead check how much resources are available and check environment availability to perform testing to achieve the object of product.
    - **Estimation:**
      - They provide estimation or time slot required to perform testing.
    - **Main aim of test plan :**
      - To decide start and end of the testing.
    - **Test plan contains:**
    - **Test plane ID:**
      - Test lead create an Id for the modules they have tested.
    - **Test items:**
      - Test items defines details about functionality that we want to check.
    - **Iteration:**
      - It is nothing but the availability of web elements or objects in module known as iteration.
    - **Feature to be tested:**
      - The list of all the features within the scope are to be tested will be listed out here in this.
    - **Feature not to be tested:**
      - The list of all the features that are not for planned for the testing will be listed out here in this.

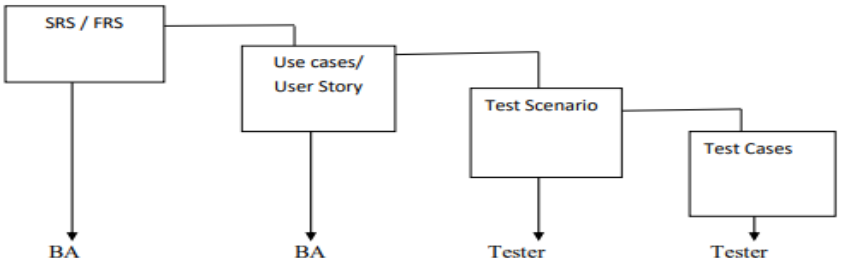
- **Test pass/ fail:**
    - Decide the criteria for test case execution.
- **Test deliverable**
    - The lists of all the documents that are to be prepared during testing process will be maintained here in this section.
    - Ex: Test case document, Defect profile document etc.
- **Test Environment:**

- The clear details of the environment that is about to be used for testing the application will beclearly maintained in this section.
- Ex: Depending upon the Project the Environment will be selected.

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

- **Testing Task:**
    - Who has to do what will be clearly planned and maintained in this.
- **Staff training (New module):**
    - To accomplish this project successfully any staff or training is required then that informationwill be clearly maintained in this section.
- **Sign Off & Approval:**
    - Who has approved this document, when it is approved will be maintained in this section.
    - **Ex: Test Manager.**

- **Explain the terms test scenario and test case in detail?**



- **Software Requirements specification :(SRS)**
  - SRS defines functional requirement to develop & system requirement that will be used.
  - SRS will be prepared by business analyst.
  - SRS derived from business requirement specification document.
  - SRS will contains:
    - Functional requirement
    - Functional flow diagram
    - Use case
    - Screenshot.

- **Use Case:**
  - Use cases defines a single/ specific requirement.
  - Use cases will be prepared by business analyst
  - Use cases derived from SRS.
  - Use cases will contains
    - Description
    - Acceptance criteria.

- **Test Scenario:**
  - Test scenario will be prepared by tester against the user story.
  - Test scenario will be derived from US.
  - Test scenario defines “What to Test”.
  - Test scenario defines ways to test the functionality.
  - Form one Test scenario, multiples test cases will prepared.
  - Test scenario defines the High level test cases.
  - Test scenario always defines in positive ways.

- **Test case:**
  - Test cases will be prepared by tester against the test scenario.
  - Test cases will be derived from test scenario.
  - Test cases defines “How to Test”.
  - Test cases defines input, process, and output.
  - Form one test cases, test data, result, etc.
  - Test cases defines the low level test cases.
  - Test cases always defines in positive way & negative way.

- **Explain the traceability matrix?**
  - In the traceability matrix the Test cases & defects are link/mapped with user story.
  - 2 types Traceability matrix:
    - **Forward Traceability matrix-.**
    - In forward traceability matrix, test cases should be link/mapped with the user story.
    - Test cases written in excel sheet, these excel sheet attached with user story.
    - **Backward Traceability matrix:**
    - In Backward Traceability matrix, defects should be link/mapped with the user story.
    - Defects will be created in JIRA/ TFS tool.
    - In JIRA/ TFS tool defects link with user story.

- **Who is responsible for writing test cases?**
  - Test cases will be written by tester
  - While preparing test cases, we will consider
  - Business logic related test cases
  - Input domains related test cases
  - UI/GUI related test cases.

- **What are the different defect reporting tool do you know?**
  - There are number of defect reporting tools available in market like *Clear Quest, Dev Track, Jira, **Quality Center, Bug Zilla etc.***
  - *In my project we are using JIRA for defect management as well as project management.*

- **What are different field we have in excel sheet while writing the test case?**

  - **Test Case:**
    - Step by step actions to be performed to validate functionality of AUT (How to test).
    - Test case contains test steps, expected result & actual result.
  - **Test Case Contents:**
    - Sr. No.
    - Test case id:
    - Priority
    - Reference
    - Test scenario.
    - Pre-requisite.
    - Test data.
    - Test steps.
    - Expected result
    - Actual result.
    - Pass/fail.

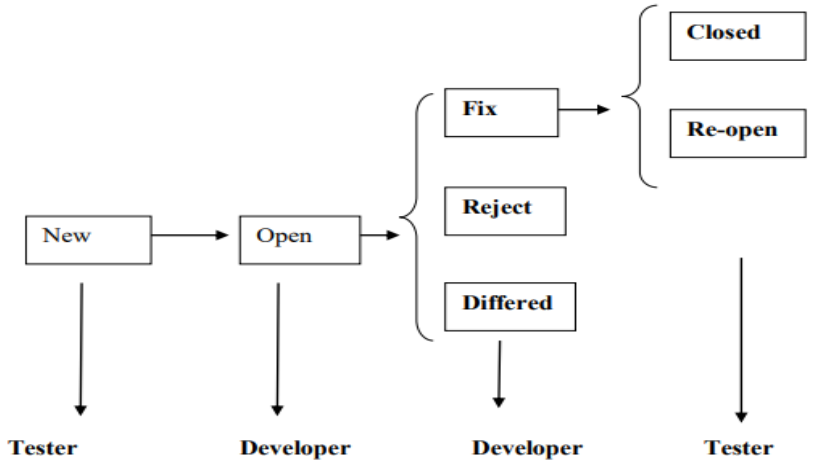
- **What is defect resolution?**

  - After receiving the defect report from the testing team, development team conduct a reviewmeeting to fix defects. Then they send a resolution type to the testing team for further communication is known as **defect resolution.**
  - Resolution Types: -
    - Accept
    - Reject
    - Duplicate
    - Enhancement
    - Need more information
    - Not Reproducible
    - Fixed
    - As Designed.

- **What exactly defect report contains?**
  - **Defect Report Contents:**
  - **Defect_I0:** Unique identification number for the defect.
  - **Detect Description:** Detailed description of the defect including information about the modulein which defect was found.
  - **Version**: Version of the application in which defect was found.
  - **Steps:** Detailed steps long with screenshots with which the developer can reproduce that defects
  - **Date Raised:** Date when the defects raised.
  - **Reference**: where you Provide reference to the documents like. Requirements, design, architecture or may be even screenshots of the error to help understand the defect.
  - **Detected By:** Name/ID of the tester who raised the defect.
  - **Status**: Status of the defect, more on this later.
  - **Fixed by:** Name/i0 of the developer who fixed it.
  - **Date closed:** Dat**e** when the defect is closed.
  - **Severity :**Which describes the impact of the defect on the application
  - **Priority:** which is related to defect fixing urgency. Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively.

- **What you have to maintain while executing the test cases?**
  - As per test cases we have to test functionality.
  - While test case execution, tester will prepared the test proof.
  - In TCE, if we found defects then raised/ create in JIRA/ TFS tool.
  - Inform to developer throw email (JIRA/ TFS).
  - Test proof will contains- Test cases against screen-shot.

- **Explain defect or bug life cycle in detail?**
  - **Defect:** when tester found error in application then it is called as defects.
  - **Defect life cycle:** The journey of the defects from their start to end (New to close) it is called as defect life cycle.



  - The journey of defect is goes like from **New** >>**Assign**>>**open**>>**fixed** >>**retest**>>**verified**>>**closed.**
  - **New:**
    - When tester will found defect in the application or build then tester will log defect in a Jira and **mark defects status as New.**
    - We will assign that defect to the developer at that time **defect status as Assign.**
    - Tester will inform to developer through mail by JIRA.
  - **Open:**
    - When developer start analyzing that defect that time developer will mark the **defects status as Open.**

- **Fix :**
  - When developer found it is valid defect then developer will fix and sent to the tester and **mark defects as Fix.**
  - Developer will inform to tester through email.

- **Retest:**
  - On the fixed defects, tester will do testing and regression testing to ensure that functionality is working fine or not. **At the time defect status as Retest.**

- **Verified:**
  - Once we are done with retesting and ensure functionality is working fine or defect is no longer present in the build **at that time defect status as Verified.**

- **Closed:**
  - When verification of the defect is done and once all the functional scenario is working fine at that time we will **change the defect status as closed.**

- **Re-open:**
  - On the fixed defects, tester will do retesting and regression testing.
  - If he found defect have not been fixed by developer.
  - Functionality is not working fine and **mark defect status as Re-open.**
  - Tester will inform to developer through email.

- **Reject:**
  - When developer checking defect and he found defect is invalid defect and **mark defect status as Rejected.**

- **Duplicate:**
  - If the same defect is logged multiple time or it is corresponds to same functionality at that time, developer **mark the defect status as Duplicate and it is closed.**

- **Deferred:**
  - When developer found, these defect can't be fixed in current sprint as it requires more time to fix.
  - At that time developer **mark the defect status as deferred.**
  - Defect status as deferred will be decided by project manager, business analyst and designer.
  - Project manager will prepare a user story for these deferred **defect as "Carried over bug"**

- **Re-producible defects:**
  - Re-producible defects are those defects which is producing again and again.
  - For re-producible defect, In JIRA tool we will **marked as Re-producible defects =Y/N.**

- **Bug Leakage:**
  - Defect those found which is missed from SIT environment and found in UAT environment.
  - Tester in SIT environment, he will reproduce that defects.

- **What are the different report? Have you involved in preparation of report?**
  - In my project, during test case executions tester maintains **test proof.**
  - We will also maintain defects which contains all defects related to user stories.
  - Test summary report is prepared by test lead which contains all records such as test case design, test case executions details, test proof, defect report etc. of current sprint
  - Test closure report is also prepared by test lead. It indicate sign-off from the testing team.

- **What is review? What are the different types of review followed in your organization?**
  - Review defines correctness & completeness of the documents.
  - In my project, we have 4 type of review for the test cases.
    - **Self-review, Peer review, internal review, External review**
    - **Self-review:**
      - Tester will do test cases review, these review is called self-review.

- **Peer review:**
  - Test lead/ senior tester will do review the test cases.
  - Tester will sent the email to test lead/ senior tester.

- **Internal review:**
  - In Internal review, business analyst will review test cases.
  - Tester will sent the email business analyst.
  - Business analyst set up the meeting with Tester for test cases review

- **External review:**
  - In external review- client/ UAT tester will review the test cases.
  - Tester will send the email to test lead.
  - In External review for test cases review for that test lead set up meeting with client/ UAT tester.
  - In these meeting, Tester will shows/ explain their test cases to client/ UAT Tester.
  - In My project, we have done internal Review / External Review.
  - In review, if we got the suggestion from client/ UAT Tester/ BA then Tester willaccept these suggestions.
  - In Excel sheet, we will accept these suggestion and write in suggestion,will change test case as per suggestions.

- **What are the 7 principles of software testing?**
  - **Present of defect in testing**
    - Test the software in order to find the defects.
  - **Early Testing**
    - Start software testing at early stages means from the beginning when you get requirements.
  - **Defect clustering**
    - Highly impossible to give the bug free software to the customer.

- **Exhaustive testing**
  - Should not do Exhaustive testing. Means we should not use same type of data for testing everytime.
- **Pesticide Paradox**
  - Testing is context based means decide what types of testing should be conducted based on type of application.
- **Testing dependency**
  - We should follow the concept of Pesticide Paradox means, if you are executing same cases for longer run, they won't be find any defects.
  - We have to keep update test cases in every cycle/release in order to find more defects.
- **Bug/ Error free application (95% < defects Build reject)**
  - We should follow defect clustering. Means some of the modules contains most of the defects.
  - By experience, we can identify such risky modules. 80% of the problems are found in 20% of the modules.

- **What is roles & responsibility?**
  - Requirement analysis / US analysis/ understand.
  - Identify Test scenario against US.
  - Test cases writing against Test scenario.
  - Test cases review (Internal/ external review).
  - Test cases execution (Test proof).
  - Defect raised/ created.
  - Defect report against current sprint.
  - Test summery report (Helping to Test lead).
  - Client interaction (Sprint review meeting- For respective US we will demo/ review)

- **Which parameter consider in Test cases review?**
  - Test case should be simple.
  - Test cases should be understandable.
  - Test cases should be functionality against the US
  - Test cases should be business logic related test cases
  - Test cases not should be duplicated.
  - Test cases should follow standard format.

- Test cases should grammatically correct.
- Test cases should be with spelling mistake

- **When you will know that testing is completed?**
  - All test cases execution should be completed,
  - All defect should be fixed & tested (retesting & regression testing).
  - Tractability matrix should be completed.

- **How many test cases you will write per day?**
  - **How many test cases you will write per US?**
  - It totally depends on complicity of US.
  - User story >> in an average, I will write 25 to 30 TCD/ US.

- **How many defects you will write or raised per day? How many defects you will write per US?**
  - It totally depends on development against US.
  - User Story In an average, I will write 4 to 5 defects/ US

- **How many automation test script you will write per day?**
  - In an average, I will write 5 to 6 test script.

- **What are testing techniques for test cases writing?**
  - **Test techniques/ coverage for writing the test cases**
    - Behavioral testing techniques / coverage
    - Input domain testing techniques / coverage (BVA, ECP, Decision)
    - Error handling testing techniques / coverage
    - Backend testing techniques / coverage
    - Service level testing techniques / coverage
    - Calculation based testing techniques / coverage

- **What problem you have faced in your carrier? OR What problem you have faced in your last project?**
    - If we have less knowledge about project domain.
    - If we have less time for testing.
    - If we have less resources / less tester.
    - If developer is not communicate properly/ developer making silly mistake.
    - If we have problem in environments.
    - If we have frequent changes in requirements.

- **What is your approach, if developer is not accepting the defect?**
    - If tester will found the defects, then tester will raised the defect in JIRA/ TFS tool.
    - While creating the defects, screenshot will attachment with defects & inform throw the mail.
    - We will take call with developer & share the screen with developer and shows the defects.
    - Tester will inform to Test lead & PM throw mail.
    - Same we will inform in daily stand up.

- **What is your approach, if a defect has not occurred in Dev environment but present in SIT environment?**
    - If tester will found the defects, then tester will raised the defect in JIRA/ TFS tool.
    - While creating the defects, screenshot will attachment with defects & inform throw the mail.
    - We will take call with developer & share the screen with developer and shows the defects.
    - Tester will inform to Test lead & PM throw mail.
    - Same we will inform in daily stand up.
    - Problem of these Defect – Deployment/ Environment.
    - These defects will be occurred due to Configuration file.

- **What are the entry and exit criteria for test execution?**
  - **Entry criteria:**
    - Coding should be completed
    - Test cases should be ready and base lined
    - RTM should be updated
    - Test data should be read and base lined
    - Test environment/set up should be ready.
    - S/w tools should be ready and approved.
  - **Exit criteria:**
    - All test cases must be executed and passed
    - All defects identified must be fixed, retested and closed.
    - Test execution summary report must be prepared.

- **What is test log?**
  - It is a report of what tests have been executed and their status like pass/ fail.
  - It is also known as Test execution report.

- **Do you run all regression tests for every bug fixed?**
  - No, I didn't run regression test cases for every bug fixed. I run regression tests once for every build.

- **When you fill the data in the application form, how do you ensure that the data is stored in the correct tables and columns?**
  - We can write an SQL query to retrieve data from the data base and compare the query result with the data we have filled in the application forms.

- **What is test case?**
  - Test case is a set of inputs, conditions and expected outcomes which a tester will deter- mine whether an application is working correctly or not.

- **How do you know for which functionalities you should write test case?**
  - My lead writes top level requirements in QC and assigns to each team member.
  - We divide test requirements further into sub requirements.
  - Then we identify test conditions for each sub requirement and create test cases. Test cases are reviewed after that. Reviewed and approved test cases will go to ready state.

- **How do you know your test cases are completed?**
  - We follow two step approaches to ensure that test cases are completed.
  - **Reviews--** it ensures that quality of the test cases is good.
  - **Requirement traceability matrix ---**it ensures that all requirements have been covered through test cases.
- **How do you find whether a test case is a good test case or bad test case?**
  - A good test case is one which finds the bug or one which has a high probability of finding the bug.
  - A good test case should be documented clearly, so that it can be executed by anyone without any difficulties and confusion.
- **What is the percentage of positive and negative test cases that you write?**
  - Approx. 30% positive and 70% negative.
- **Do you update the test cases after receiving build based on the application screen?**
  - During execution, if we feel any test case requires an update, we will do it with the approval of the team lead. But this work is very limited.
- **What is an entry criterion for test closure?**
  - Decision to stop testing
- **Should you understand the whole project functionality or only the functionality assigned to you?**
  - I should have a big picture of the whole project.
  - In other words I should have an over- view of the whole project and detailed screen and field level understanding of the assigned functionalities.
- **What parameters do the test manager considers to take the decision to stop testing?**
  - The important parameters a test manager looks into are
  - Whether all requirements have been developed or not.
  - Whether all requirements have been covered through testing.
  - Whether all requirements have been handled through fixed or differed status.

- **What are the exit criteria for test closure?**
  - Checking whether planned deliverables have been delivered.
  - Finalizing and archiving test ware.
  - Handover of test ware for maintenance.
  - Analyzing lessons learned for improvement of test maturity.
  - Testing sign off.

- **What is testware?**
  - Test ware is Artifacts produced during the testing process.
  - Test ware include test cases, test plan, automation scripts, test data, test environment set-up and clear up procedures and any additional software or utilities used in testing.

- **What is lessons learnt document?**
  - No of test cases/scenarios blocked.
  - No of defects verified and their respective status.
  - Weekly status reporting.
  - Test case summary.
  - Issues found.
  - Issues resolved.
  - Critical issues which are still open and which requires immediate attention from the client side.
  - The report should also contain high plan for the next week.

- **What is web server log?**
  - Every time a web page is requested, the webserver automatically logs the following in-formation.
  - The IP address of the visitor.
  - Date and time of the request.
  - The URL of the requested file.
  - The URL, the visitor came from immediately before.
  - The visitors web browser type and operating system.

- **How do you check broken links?**
    - Many tools are available for this. We use tools like menu.

- **How many test cases can you execute per day?**
    - It depends on the size and complexity of the test cases. Approximately i.e. execute around 50 test cases per day which comes to 40 pages approx.

- **Explain different test execution strategies?**
    - There are 3 test execution strategies.
    - They are pass1, pass2, pass 3
    - **Pass1 Test execution strategy:**
        - In this execution model one execution cycle will be there.
        - In this one execution cycle. Itself testers log defects and retest that defect.
        - This is useful in stable, small with.
    - **2nd pass:**
        - Development team releases new build claiming all the defects are fixed.
        - Testing team retest all defects with adhoc regression.
        - If new defects are found development team release new build and the life cycle is repeated until no new defects.
    - **3rd pass:**
        - Testing team runs full regression suite and this phase completes only when full regression is completed.
        - This is good model for large, complex and critical projects.
        - In case of getting large no of defects in pass -2 strategy, one may have to move to pass-3 strategy.

- **What is code review?**
    - Code review is the process of reviewing the code written.
    - Code reviews are conducted for the code developed by the developer and also for the automation scripts developed by the automation engineer.

- **How do you say review was successful?**
  - If every reviewer prepares well before the review and provides good comments for improvement of the work product, we can say that the review was successful.
- **How much information you can review in one day?**
  - Per hour we review around 20 pages if it is documentation and 200 lines if it is code.
- **Explain what do you document during the review process?**
  - We document page and line number of defect, origin of defect, severity of defect.
  - We also document other information like work product ID, reviewers, etc.
- **What are the roles present in the review?**
  - **Manager:**
    - Decides on execution of reviews.
    - Allocates time in project schedules.
    - Determines if the review objectives have met.
  - **Moderator:**
    - Leads the review, including planning and running the meeting.
    - Follows-up after the meeting.
  - **Author:**
    - The author is the person who has created the item to be reviewed.
    - The author may also be asked questions within the review.
  - **Reviewer:**
    - The reviewer are the attendees of the review who attempt to find errors in the item under review.
    - They should come from different perspectives in order to provide a well-balanced review of the item
  - **Scribe:**
    - The scribe or recorder is the person who is responsible for documenting issues raised during the process of the review meeting.

- **Test Cycle Closure**
  - **Activities:**
    - Evaluate cycle completion criteria based on Time, Test coverage, Cost, Software,
    - Critical Business Objectives, Quality.
    - Prepare test metrics based on the above parameters.
    - Document the learning out of the project
    - Prepare Test summary report
    - Qualitative and quantitative reporting of quality of the work product to the
    - Customer.
    - Test result analysis to find out the defect distribution by type and severity.
    - Deliverables
      - Test Closure report
      - Test metrics

- **Explain Test Matrix?**

  - **Defect Density: Number of defects identified per requirement/s**
    No.of defects found / Size(No. of requirements)
  - **Defect Removal Efficiency (DRE):**
    (A / A+B) * 100
    (Fixed Defects / (Fixed Defects + Missed defects)) * 100
    - A- Defects identified during testing/ Fixed Defects
    - B- Defects identified by the customer/Missed defects
  - **Defect Leakage:**
    (No.of defects found in UAT / No. of defects found in Testing) * 100
  - **Defect Rejection Ratio:**
    (No. of defect rejected /Total No. of defects raised) * 100
  - **Defect Age:** Fixed date-Reported date
  - **Customer satisfaction** = No.of complaints per Period of time

- **Explain defect related Matrix?**

  - **% of Test cases Executed:**
    No.of Test cases executed / Total No. of Test cases written) * 100
  - **% of test cases NOT executed:**
    (No.of Test cases NOT executed/Total No. of Test cases written) * 100
  - **% Test cases passed**
    (No.of Test cases Passed /Total Test cases executed) * 100
  - **% Test cases failed**
    (No.of Test cases failed / Total Test cases executed) * 100
  - **%Test cases blocked**
    (No.of test cases blocked / Total Test cases executed) * 100