SDLC & AGILE

1. **What is Software Testing?**
   - Process of executing a program or application with the intent of find software bugs using functional and automation tools
   - Process of validating/verifying a software program/application
   - Testers should test to break approach, not test to pass.

2. **What is Software Requirements Specification?**
   - A software requirements specification is a document which acts as a contract between the customer and the supplier.
   - This SRS contains all the requirement of the end user regarding that application. SRS can be used as a communication medium between the customer and the supplier.
   - The developer and tester prepare and examine the application based on the requirements written in the SRS.
   - The SRS documented is prepared by the Business Analyst by taking all the requirements for the customer.

3. **Software Development Life Cycle (SDLC) - What is SDLC?**
   - SDLC defines the phases in **building** of software or application.
     - Project Planning
     - Requirement Gathering (Gathering information used to plan project, Identifying risks)
     - Design (How the application will be built)
     - Coding (developing) (Based on requirements, developers will write the application)
     - **Testing**
     - Production (deployment)(Releasing product)
     - Maintenance (Making sure product is stable, looking at customer report on bugs and fixing it)

4. **Software Testing Life Cycle (STLC) - What is STLC?**
   - STLC defines the phases in **testing** of software or application. In STLC process in different activities are carried out to improve the quality of the product.
     - Requirements analysis
     - Test Planning
     - Test Designing
     - Test Environment Setup
     - Test Execution
     - Test Reporting

5. **What is the difference between STLC and SDLC?**
   - STLC is part of SDLC. It can be said that STLC is a subset of the SDLC set.
   - STLC is limited to the testing phase where quality of software or product ensures. SDLC has vast and vital role in complete development of a software or product.
   - However, STLC is a very important phase of SDLC and the final product or the software cannot be released without passing through the STLC process.
   - STLC is also a part of the post-release/ update cycle, the maintenance phase of SDLC where known defects get fixed or a new functionality is added to the software.

6. **What is requirement?**
   - Requirements convey the expectation of users for the software or product.
   - Process to gather requirements from client, analyze and document them is known as requirement engineering.
   - Goal of requirement engineering is to develop and maintain sophisticated and descriptive SRS 'System Requirements Specification' Document

7. **Where is the requirement coming from?**
   - Customers give requirements for the application
   - Talk to the End-users → the person that will be using this application the most
   - Talk to Partners –
   - Talk to Domain Experts – coders and developers that have already build this application similar before or someone that is an expert the type of product being built
   - Industry Analysts and Information about competitors

8. **When the testing starts?**
   - Testing starts from testing the requirements (not after the coding phase which seems like the most likely answer.)
   - We have to make sure the requirement is correct in first place. With the wrong requirement it is impossible to build bug free application.

9. **How to tell if the requirement is good or bad?**
   - Requirement must be (SMART)
     - **S**pecific →User should be able to login. Authorized user with valid username and password should be able to login
     - **M**easurable → User should able to login very fast (in 2 second after clicking login button).
     - **A**ttainable
     - **R**ealistic
     - **T**estable → User should able to download the receipt very fast (in 2 second

10. **Why we test?**
    - To build bug free application.
    - To satisfied end user and client.
    - To build great product to generate more revenue.
    - I love testing and testing is my passion.

11. **What is tester's main responsibility?**
    - To find bug as much as possible as early as possible. Make sure most of the bug gets fixed.
    - To satisfy the end user and client buy delivering bug free and user-friendly application.

12. **What is the job responsibility of a tester or Software Development Engineering in test role?**
    - Write automation of testing and set up the same for varieties platforms like web or mobile.
    - Managing and handling bug report.
    - Maintaining the proper communication channel between the developer and the client.
    - Preparing and delivering test cases.

13. **Is 100% testing possible?**
    - We can't test the application 100% since there are unlimited scenarios that we can't even imagine.
    - Software testing is risk-based activity based on **priority of the functionality** we can test as much as much as possible.
    - Even though 100% testing is not possible, but I believe 100% customer satisfaction is certainly possible.

14. **What is positive testing? Happy Path testing?**
    - Testing the application with valid inputs. Also called "**Happy Path**" Testing.
      Ex. If you log in with valid username and password it is positive testing.

15. **What is testing hierarchy?**
    - Unit testing → Developers test each module or block of code during development.
    - Component Testing → Component is a standalone functionality that can work by itself. Ex. Amazon Buyer Functionality, Seller Functionality, Prime Video Functionality.
    - Integration Testing → Combine all of the Functionalities. When I integrate them, can I still use all of the functions? Make sure they all still work.
    - System Testing → End-to-End testing. Test everything from beginning to end.
    - Acceptance Testing → Hire a UAT (User Acceptance Testing) Team or Business Analyst can also do Acceptance Testing. After testing has been complete you have to get another team to do acceptance testing so they can confirm the QA teams testing was successful and have the product ready for the customer.

16. **What is 508 Compliance testing?**
    - *(If someone in the interview, ask what 508 testing is, Just tell them what it is. Don't say I have 5-10 years of experience in 508 Compliance testing.)*
    - **It's a requirement for government websites.**
    - All the websites that are used by and for the government. **They have to make sure disabled people can use it.**
    - Example: For healthcare.gov they have Compliance manager and he has a dedicated QA team that does 508 Compliance testing to make sure the website is 508 compliant for disability users.

17. **What is risk-based testing?**
    - Risk-Based testing is defined as the functionalities of a product are tested based on the priority of the deliverables.
    - Risk-Based testing includes testing of crucial features of a product which will have a business impact and the probability of the failure of those features is very high.
    - The priority for all functionalities of a product is set based on the business requirement then the high priority functionalities will be tested first then medium and then low priority functionalities.
    - Risk-Bases testing will be performed when there is no sufficient time to test all the functionalities of a product.
    - Since 100% testing is not possible, we have to do risk analysis. Based on the analysis we have to prioritize our testing activity and test high risk area first. For example:
        o The most critical functionalities
        o The most often used functionalities
        o The most complicated functionalities etc…

18. **How long did it take to build this regression suite?**
    - It took 3 years with; 2 testers 1 manual tester + 1 automation tester
    - when we run:
        o before release
        o after major bug fix
        o after major new functionality
    - where we keep test scenarios and where we as a team take decision which will be executed more than once, in one sprint you test some scenarios.

19. **Tell us one challenge while running regression suite?**
    - Failures. Because regression suite was developed so long ago, and you don't know what has changed. The properties of a button may have changed.

20. **How many environments you have?**
- Development Environment
  - Unit testing
  - Less stable than test environment
- Test Environment
  - Manual testing happens here
  - Replicates the production environment exactly
  - Changes are deployed in intervals
  - Automated **smoke tests** are ran here
    - Runs against the test environment to make sure if the application is stable enough to perform other major testing activities.
    - Run every time changes are deployed to Test environment
    - Can be ran in dev environment
  - Automation tests are ran here
  - Automated Integration tests run here
- Pre-production Environment
  - UAT environment
  - Demo happens here
  - load/performance testing happen here
  - Changes are deployed in big intervals
  - Automated major **regression tests** here (before release)
    - Runs against the UAT environment
    - To find out if new changes result in any defects
    - Runs after major bug fixes and every release
    - This test is decided in test plan
  - Very stable
- Production environment

21. **What is Functional testing?**
- Functional testing team can also be called manual testers it can also be done by automation team (automation functional testing). Similar to Black box testing or manual testers. Just testing the specific functionality of the application. Ex. Can user login? Can user logout? Not testing look and feel of application.

22. **What is non-functional testing?**
- Performance testing, Security testing, Ex. Can 2000 user's login to the application at the same time? Can user move to next page in 1 second?

23. **What is unit testing? Have ever done unit testing?**
- It is part of the white box testing. It's done by the developers before they deploy the code from Development environment to QA environment.
- Since it is performed by developers, I have never done unit testing yet. But I think I can learn it and do it if it is needed.

24. **What is component testing?**
- Testing each component of the application separately. In application it could be one component. One component has stand-alone functionality. Ex. in amazon.com Seller functionality can be one component. Buyer can be another component. Also, Amazon prime videos can be another component.

25. **Smoke Test →**
- **TESTING order:** Code → Unit Testing → Integration Testing → Sanity Testing →*Smoke Testing* → Functional Testing
- In our project, there are five modules like; login, view user, user detail page, new user creation and task creation
- In these five modules, the developer will first perform the smoke testing by executing all the major functionality of modules like; user is able to login with valid login credentials or not, after login new user can be created or not, user that is created is viewed or not etc.

26. **Which part of regression test should be automated?**
- Tests which are stable
- Repeated frequently
- Simple and require no tester input are good candidates for automation

27. **How do you ensure that your regression tests are effective?**
- The regression tests should be wide and detailed enough to allow catching defects. You can also eliminate duplicate test cases, merge test cases and automated tests as feasible.

28. **A number of critical bugs are fixed in software. All the bugs are in one module, related to reports. The test manager decides to do regression testing only on the report's module.**
- Regression testing should be done on other modules as well because fixing one module may affect other modules.

29. **Which technique can be used to achieve input and output coverage?**
- It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.

30. **How do you run your regression? How often, how many VMs, how many days, how many tests?**
- Regression is scheduled before every release and we release 4 times a year (2 Spring release and 2 Fall release).
- Regression also happens when there is a major bug fix.
- Around 500 feature files and 1300 scenarios.
- Regression tests are kicked off by jenkins. Tests are executed on the jenkins server (VM). My Linux server is RedHat.
- The latest run look more than 12 hours.
- **Another answer is**; I have built a suite of regression tests. They are feature files with regression tag. And I have a job in jenkins that kicks of the regression tests. It uses the maven command to trigger the test. The maven command includes that tag name: mvn test -D cucumber.options ="--tags @Regression".
- At the end of the execution, jenkins generates HTML report with detailed tests steps and screenshots.

31. **What is black box testing? What are the different black box testing techniques?**
- Black box testing is the software testing method which is used to test the software without knowing the internal structure of code or program.
- This testing is usually done to check the functionality of an application. The different black box testing techniques are;
  - Equivalence Partitioning
  - Boundary value analysis
  - Cause effect graphing

32. **What is Equivalence partitioning testing?**
- Equivalence partitioning testing is a software testing technique which divides the application input test data into each partition at least once of equivalent data from which test cases can be derived. By this testing method it reduces the time required for software testing.
- Example: When testing a grade calculation system, a tester determines that all scores from 90 to 100 will yield a grade of A, but scores below 90 will not.
- Which technique can be used to achieve input and output coverage? It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.

33. **What is Boundary value testing?**
   - Test boundary conditions on, below and above the edges of input and output equivalence classes.
   - For instance, let say a bank application where you can withdraw maximum $1000 and a minimum of $100, so in boundary value testing we test only the exact boundaries, rather than hitting in the middle. That means we test above the maximum limit and below the minimum limit.
   - For example, of my credit card: Activated date is lower boundary. Expiration date 10/2019 is upper boundary. $0 is lower boundary for spending limit. $25.000 is upper boundary for spending limit.

34. **Why does the boundary value analysis provide good test cases?**
   - Because errors are frequently made during programming of the different cases near the 'edges' of the range of values.

35. **Why we use decision tables?**
   - The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface.
   - The other two specification-based techniques, decision tables and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g. inputs).
   - This technique is sometimes also referred to as a **'cause-effect'** table. The reason for this is that there is an associated logic diagramming technique called **'cause-effect graphing'** which was sometimes used to help derive the decision table.

36. **What is white box testing and list the types of white box testing?**
   - White box testing technique involves selection of test cases based on an analysis of the internal structure (Code coverage, branches coverage, paths coverage, condition coverage etc.) of a component or system.
   - It is also known as Code Based testing or Structural testing. Different types of white box testing are
     o Statement Coverage o Decision Coverage

37. **In white box testing what do you verify?**
   - Verify the security holes in the code
   - Verify the incomplete or broken paths in the code
   - Verify the flow of structure according to the document specification
   - Verify the expected outputs
   - Verify all conditional loops in the code to check the complete functionality of the application
   - Verify the line by line coding and cover 100% testing

38. **What is Gray Box Testing?**
   - Grey box testing is the hybrid of black box and white box testing.
   - In gray box testing, test engineer has the knowledge of coding section of the component and designs test cases or test data based on system knowledge.
   - In this tester has knowledge of code, but this is less than the knowledge of white box testing. Based on this knowledge the test cases are designed and the software application under test treats as a black box & tester test the application from outside.

39. **What is the difference between static and dynamic testing?**
   - **Static testing**: During Static testing, the code is not executed, and it is performed using the software documentation.
   - **Dynamic testing**: To perform this testing the code is required to be in an executable form.

40. **What is maintenance testing?**
   - Triggered by modifications, migration or retirement of existing software.

41. **What is Integration Testing?**
   - Integration testing is black box testing. Integration testing focuses on the interfaces between units, to ensure that units work together to complete a specific task.
   - The purpose of integration testing is to confirm that different components of the application interact with each other. Test cases are developed with the purpose of exercising the interfaces between the components.
   - Integration testing is considered complete, when actual results and expected results are same. Integration testing is done after unit testing. There are mainly three approaches to do integration testing:
     o **Top-down Approach →** tests the components by integrating from top to bottom.
     o **Bottom-up approach →** It takes place from the bottom of the control flow to the higher-level components
     o **Big bang approach →** In this, different module are joined together to form a complete system and then testing is performed on it.

42. **What is Scalability Testing?**
   - Scalability testing is testing performed in order to enhanced and improve the functional and performance capabilities of the application. So that, application can meets requirements of the end users.
   - The scalability measurements is done by doing the evaluating the application performance in load and stress conditions. Now depending upon this evaluation, we improve and enhanced the capabilities of the application.

43. **What is Storage Testing?**
   - In Storage Testing we test those functionalities of the application which is responsible for storing the data into DB.
   - The data entered by the end user in GUI or front end, is the same data which is stored in the database.
   - The storage testing determines that the data taken from the front end of the application is stored in correct place and in correct manner in the database.

44. **What is Stress Testing?**
   - Stress testing tests the software with a motive to check that the application do not crashes if we increase the stress on the application by increasing the large number of users working on the application.
   - We can also apply the stress on the application firing the lots of process which cannot be handled by the application.
   - We perform the stress testing on the application evaluate the application capabilities at or beyond the limits of its specified requirements to determine.
   - Generally, this is a type of performance testing performed in a very high level of load and stress condition.

45. **What is Test Harness?**
   - A test harness is a collection of software and test data required to test the application by running it in different testing condition like stress, load, data- driven, and monitoring its behavior and outputs. Test Harness contains two main parts:
     o Test execution engine
     o Test script repository
   - Automation testing is the use of a tool to control the execution of tests and compare the actual results with the expected results. It also involves the setting up of test pre-conditions.

46. **What is test coverage?**
   - Test coverage means is how many test cases that we have and what functional area those test cases are covering.

47. **What is a V-Model?**
   - A software development model that illustrates how testing activities integrate with software development phases.

48. Which of the following is likely to benefit most from the use of test tools providing test capture and replay facilities?
- Regression testing
- Integration testing
- System testing
- User Acceptance Testing

49. What is Acceptance testing?
- The Acceptance testing will be performed after QA testing. In my current project it is done by UAT team. After UAT team performing the acceptance testing the code will go to production.
  - Development environment(where developers write code and perform unit testing)
  - QA environment (where we test the application.)
  - UAT environment (after the code is tested QA environment it will be deployed to the UAT environment. UAT testing team will perform testing to make sure it fits the business requirement. It is also called staging environment.
  - Production environment(is when the end user can see the real application)

50. What is the difference between UAT (User Acceptance Testing) and System testing?
- **System Testing**: System testing is finding defects when the system undergoes testing as a whole, it is also known as **end to end testing**. In such type of testing, the application undergoes from beginning till the end.
- **UAT**: User Acceptance Testing (UAT) involves running a product through a series of specific tests which determines whether the product will meet the needs of its users.

51. What is continuous integration ?
- Developers can check–in and check–out codes into the system when making new code changes to the application.
- Whenever a developer Check-in a new code into the system, there is a server called **Continuous integration (CI) server**.
- The **CI server** is continuously looking for new code. Once the new code is added to the application the CI server will immediately recognize that code has been checked in. *(there is a tool integrated with this maybe Jenkins or something).*
- That tool will kick out the automated smoke test to check the basic functionally of the application. Then it will say weather that code affected the app negatively or not.

52. How is code deployed to production environment?
- From Local
  - check in code to Git using pull and push (in my company it is SVN)
  - Run unit tests
  - Deploy changes to server
    - Done by jenkins automatically every time developer checks in code.
    - After it passes deploy changes to Dev environment
- From Dev environment
  - Deploy changes to Test environment done by Jenkins.
  - Can be scheduled or manually triggered
- From Test → Deploy changes done by Jenkins
- From Pre-production

53. Agile Framework?
- **Role** : PO, SM, Team
- **Ceremonies** : -Sprint Planning, Daily Scrum, Sprint Review, Sprint Retro, Grooming Session
- **Artifacts** : Product backlog, - Sprint backlog, -Burnout chart

54. **What is Agile?**
   - Agile is **iterative** product **development** methodology that is alternative to the waterfall methodology.
   - Scrum : Team plans for amount of work for the next sprint
   - Kanban : No sprint planning, stories are picked up as is, but you still have everything else

55. **Why do we need Agile? Waterfall and Agile?**
   - Because waterfall methodologies have following disadvantage;
     o Requirement cannot be change or hard to change once document is signed.
     o In waterfall before completing the one phase you can't move to the next phase. For example, before coding phase is completed testing cannot be started.
     o Customer can't see what they are going to get until very late stage in development life cycle.
     o It takes longer time to go to the production. By the time product goes to the market it might be outdated already.
   - Agile has following advantages :
     o The change is welcomed. For example, after the sprint demo if client does not like something, we can take their feedback and improve the product. Requirement change is OK.
     o Since it is iterative development process, the development team can developed piece of functionality, get feedback and improve next iteration. So, the product will be continuously improve.
     o Waste is eliminated in agile with the help of scrum master. For example, if I am blocked, I don't have to wait and waste my time. Since team members communicates with each other efficiently we can be more productive by preventing duplicated effort.
     o Waterfall emphasizes tools and platform, like C# .NET, but agile emphasizes people. You can have best tool but at the end people are using those tools. I believed inspired people can make amazing products even they have less money or less resources.

56. **What kind of Agile methodology did you use in your previous projects?**
   - I have heard Extreme programming(XP) , Kanban and Scrum. But I have only worked with scrum only.

57. **Scrum is an Agile framework, right? Name few other Agile frameworks.**
   - Yes, Scrum is an Agile framework. Few other Agile frameworks are –Feature Driven Development Test Driven Development, Kanban

58. **What are the different roles in Scrum? Scrum roles?**
   - Product owner is actually the stakeholder of the project.
     o He represents the project requirements before the team.
     o He is responsible to have a vision of what to build and convey his detailed vision to the team.
     o He is the starting point of an agile scrum software development project.
   - Scrum team is formed by the collective contribution of individuals who perform for the accomplishment of a particular project.
     o The team is bound to work for the timely delivery of the requested product.
   - Scrum master – Scrum master is the leader and the coach for the scrum team who checks whether the scrum team is executing committed tasks properly.
     o He is also responsible to increase the efficiency and productivity of the team so that they can achieve the sprint goal effectively.

59. **How do you describe a scrum team?**
   - If you put 5 rock stars together it doesn't mean they are a team, or it doesn't mean they can build great product.
   - For me the team is a group of people who are sharing the same goal , moving to the same direction , who trust each other and who will effectively communicate and collaborate with each other to build great product. There should be no star individual but a star team.

60. What are the responsibilities of a Scrum Master?

- Tracking and monitoring
- Understanding requirements properly
- Work to reach the project goal
- Process checking master and quality master
- Protect the team from detachments
- Improving the performance of the team
- Lead the meetings and resolve issues
- Resolution of conflicts and impediments
- Communication and reporting

61. What is a negative test case?

- Negative test cases are created based on the idea of testing in a destructive manner. For example, testing what will happen if inappropriate inputs are entered into the application. Wrong login information

62. What do you understand by the term "Scrum of Scrums"?

- Let us assume an active project on which seven teams are currently working. Each team is responsible to lead its own scrum meeting. But, in order to coordinate and communicate with different teams, it is required to organize a separate scrum meeting. There is one team leader from every team, known as ambassador, who is responsible to represent his team in the scrum of scrums.
- **The scrum meeting organized to hold a coordination between scrum teams is known as the scrum of scrums.**

63. Shippable product/increment?

- The piece of the product is made, and it keeps getting added functionality form each sprint
- The increment must align to the development team's *Definition of Done*
  - When the product increment is delivered, it needs to meet "Definition of Done"
  - Acceptance criteria is fulfilled
  - Product owner accepts the user stories
- The increment must be acceptable by the P.O

64. What is BurnDown Chart?

- Graphic representation of the rate at which work is completed and how much work remains to be done

65. What is Verification and Validation?

- Verification happens during developing by testers and developers; it is a process of evaluating software at development phase and to decide whether the product of a given application satisfies the specified requirements.
- Validation by testers; is the process of evaluating software at the end of the development process and to check whether it meets the customer requirements.

66. What is Definition of Ready?

- Acceptance Criteria is cleared/reviewed & Point/hours are given

67. What is parking lot?

- In Agile it means this: In meeting when you have a problem that is not really relevant to other people we should not keep discussing that item in the meeting because we are wasting other people's time.< Let's make it **parking lot** item > means whoever is interested in that issue can talk after the meeting.

68. What is sprint workflow?

- How a story moves from to do to done and lifecycles - what happens when something is blocked, etc.

69. What is User Story?

- *(Note: basically, a user story is just a requirement)* User story is a short simple description minimum **shippable** product.
- It normally looks like this: As **<end-user>** I want to do **< action>** So that I can **<benefit>**.
  - As amazon user I should able to login, so I can buy stuffs online

70. You said "shippable", what do you mean by that?

- Well, You can't really say As a user I want to put my username in the username field.
- So, I can write my username in there. It has to be complete functionality. Putting user name is not a shippable functionality. But able to login is a complete functionality. That is what I mean by shippable.

71. What is an Epic?

- Epic is a big user story that you cannot complete in one sprint.
- For example, as a user I want to buy online so I don't have to visit the local store. This story is too big, and it cannot be completed in one sprint. So, we can call it Epic instead of user story. It should be divided to multiple user stories like:
  - As a customer I want to be able to login so I can view my account.
  - As a customer I want to be able to search for a product so I can buy them.
  - As a customer I want to be able to proceed to checkout so I can pay for the item that I am going to buy.
  - As a customer I want to be able to logout so I can protect my account.
  - As you can see< As a customer I want to be able to buy...> can be divided to multiple user stories. The team can pick one or more user story in every sprint.

72. Agile experience in your most recent project?

- Our sprint is 4 weeks and we have release every 3 sprints as a release cycle
- We have 7 people in my team. 3 developers (Shwan, Simon, Sinan), 1 automation (Me) and 1 functional testers(Usman), also 1 SM (Yasin) and 1 PO (Simon B.).
- We start a sprint with Sprint Planning Meeting and
  - we discuss about the team's priority features and product backlog items and
  - we learn the part of the application which we are going to developed.
  - Choosing story based on *velocity* and *capacity*
    - **Velocity**: Number of story points delivered/demo in a sprint. For example: if team planned 30 story points (Business value); worth of user stories in a sprint and able to deliver as planned then team's velocity is 30
    - **Capacity**: Total number of available hours for a sprint is Team's capacity. Calculates holiday and PTO hours
  - This meeting is held every week and lasts for almost 1 hour. We get general idea than we do Sprint Grooming meeting for giving some estimates for the tasks.
    - Team, SM, and PO get together to ensure work items are relevant and useful
    - Ask questions to P.O of user stories
    - Re-define acceptance criteria
    - Writing new stories
    - Breaking epics into user stories
    - Understand the story to give proper estimation/to prevent under/over estimate

  How do you estimate?

  Based on my experience and complexity of the story and it is something I worked on before.
- After sprint starts, we do Daily Standup Meeting
  - everyday morning and we discuss what did we do yesterday, what will we do today and is there any blocker.
  - Just we synchronize info about the sprint.

- End of the sprint, usually we do Sprint Demo/Review Meeting .
  - It is just to show customer what we build sprint (PO can put feedback)
  - As an SDET in my team, I have done presentation sometimes and go over through the functionalities in the conference room.
  - Client or stakeholders or business people they ask questions what they don't know.
- After Sprint Demo, we do Sprint Retrospective Meeting .
  - In sprint Retro, we talk about what was good in last sprint, what kind of mistakes we made.
  - We go over them and make sure that we don't make the same mistakes again.
  - If we did something good and improvements, we would continue doing it.
  - This meeting that is held at the sprint review meeting or at the end of the sprint; it lasts for 2-3 hours.

73. **What is Acceptance criteria?**
- Acceptance criteria is the way that we know the user story is successfully developed or not.
- Statements of requirements that are described from the point of view of the user to determine when a story is "done" and working as expected
- 3 parts examples
  - Input → valid email address
  - Process → marking messaging
  - Outcome → marketing message design matches the specs provided by marketing

74. **What is rat hole?**
- Since there is a lot of communication going on in agile team, team has to discuss a lot of stuffs. But sometimes the discussion will last too long for one topic and it is not really productive. We will say it is **<rat hole>** it means we should not keep taking about that issue too long and move forward.

75. **What types of Test cases?**
- I cover different scenarios
  - Positive
  - Negative
  - Boundary Value Analysis

76. **Test Case?**
- Test case is a specific condition to check against the Application Under Test. It has information of test steps, prerequisites, test environment, and outputs.
- Test case describes the functionality and test steps.
  - Test Case ID
  - Step number
  - Description of the functionality
  - Expected result
  - Actual Result

77. **How many Test cases (in your regression suite) do you usually complete in a week?**
- 10 small test cases, 7-8 medium, 2-3 larges
- OR It depends on the project. In COOLSIS we have 2000 test cases. In 4Stay, we have around 700 test cases.

78. <mark>**How long it will take to run your regression suite?**</mark>
- It depends on the project. In my current project out of 2000 test cases in the regression suite around 1500 are already automated. If we use 10 virtual machines to perform parallel execution it takes 2 to 3 days to execute automated test cases. Also, manual testers will execute some manual test cases, but I am not sure how much they are executing. I believe they are only executing some important test cases only after prioritization.

79. **What do you do when you run your automated script or what do you when you run regression?**
   - First, I have to execute my script. Once the script execution is completed, I have to analyze the run result to see if there is any failed test cases. If there is failed test cases I have to determine if it is failed due to legitimate application issue or it is caused by some script issue. (the script can be failed due to automation code issue too) if it is caused by application issue, I will try to manually reproduce it and log a defect if I can reproduce. If it is due to my script, I have to fix it. But this is not the case most of the time.

80. **What are the steps you take to automate?**
   - Learn the functionality
     - Reading requirements
     - Knowledge transfer session with B.A
     - Ask teammates
   - Manually test it
     - Making sure I understand each step properly
     - Understand expected results
   - Automate it
     - Create POM pages
       - Add necessary elements/methods I am going to use and add PageFactory design pattern
       - Create a driver class with Singleton pattern
   - Validate the tests using TestNG Assertions

81. **What percentage of position is automation vs manual?**
   - 80-85% automation 15-20% manual

82. **When do you choose automated testing over manual testing?**
   - If the test cases are high priority test cases.
   - If the functionality is critical functionality.
   - Shakeout or smoke-test test cases.
   - If the test cases are too long and too difficult to execute. The regression test cases based on the priority.
   - We should automate as much as possible.

83. **When do you do automation in your sprint?**
   - When developers are done with their part
   - When code is deployed to QA/test environment
   - When testing framework is set up
   - When all manual tests are done
   - Smoke tests are passing

84. **What is Test Plan?**
   - Test plan is a word document that described the testing scope
     - High level test cycle
     - Defect life cycle
     - Entrance Criteria (defines what all need to start the testing)
     - Exit Criteria (defines what the testing is finished)

85. **What are the tables in test plans?**
- Test design, scope, test strategies , approach are various details that Test plan document consists of.
  - Test case identifier
  - Scope
  - Features to be tested
  - Features not to be tested
  - Test strategy & Test approach
  - Test deliverables
  - Responsibilities
  - Staffing and training
  - Risk and Contingencies

86. **What is the difference between a test plan and a QA plan?**
- A test plan lays out what is to be done to test the product and includes how quality control will work to identify errors and defects.
- A QA plan on the other hand is more concerned with prevention of errors and defects rather than testing and fixing them.

87. **What is a peer review?**
- Peer reviews are reviews conducted among people that work on the same team. For example, a test case that was written by one QA engineer may be reviewed by a developer and/or another QA engineer.

88. **How can you tell when enough test cases have been created to adequately test a system or module?**
- You can tell that enough test cases have been created when there is at least one test case to cover every requirement. This ensures that all designed features of the application are being tested.
- A2-That is the reason we need to have requirement traceability matrix. We can tell how many requirements has been covered by test cases and how many still left from RTM.

89. **Who approves test cases?**
- The approver of test cases varies from one organization to the next. In some organizations, the QA lead may approve the test cases while another approves them as part of peer reviews.

90. **Who writes test plans and test cases?**
- Test plans are typically written by the quality assurance lead while testers usually write test cases.

91. **What is the purpose of test design technique?**
- Identifying test conditions and Identifying test cases.

92. **Difference between Test case and Test script?**
- Test case terminology mostly used for Manual Testing whereas Test Script mostly used for Automation Testing
- **A test case is a documentation which specifies input values, expected output and the preconditions for executing the test.** It's also a layout of the low-level details on how to test the scenario
- A test script in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected.

93. **What should be included in a test strategy?**
- The test strategy includes a plan for how to test the application and exactly what will be tested *(user interface, modules, processes, etc.).* It establishes limits for testing and indicates whether manual or automated testing will be used.

94. What will you do when script fails?

- In my experience, I will identify the failure,
  - if it is this due to application error, sync error, script issue or environment is down, first of all I analyze the result by reproduce it through Jenkins run only the fail one,
  - if it is due to synchronization issue, I will add extra time by using implicit, explicit and some custom expected conditions,
  - If it is script issue I will debugging (identify) my script and fix it, analyze the exceptions,
  - if it is real defect then I will log defect.

95. Test Scenario?

- Make sure that end to end functionality of application under test is working as expected
- The tester needs to put his/her foot in the end users' shoes to check and perform the action as how they are using application under test
- T.S can have many test cases associated with it, Before executing the T.S we need to think of test cases for scenario
- Test Scenario: Validate the login page
  - Test Case 1: Enter a valid username and password
  - Test Case 2: Reset your password
  - Test Case 3: Enter invalid credentials
- In each test case are detailed steps and condition for execution

96. Requirement Traceability Matrix (RTM)

- RTM is used to make sure that all test cases cover the requirement or not. It is like excel sheet.

97. What can be done to develop a test for a system if there are no functional specifications or any system and development documents?

- When there are no functional specifications or system development documents, the tester should familiarize themselves with the product and the code. It may also be helpful to perform research to find similar products on the market.

98. What are the functional testing types?

- Unit Testing
- Smoke testing
- Sanity testing
- Integration Testing
- System Testing
- Regression Testing
- UAT (user acceptance testing)

99. What is the difference between sanity testing and smoke testing?

- When sanity testing is conducted, the product is sent through a preliminary round of testing with the test group in order to check the basic functionality such as button functionality. Smoke testing, on the other hand is conducted by developers based on the requirements of the client.

100.   What steps are involved in sanity testing?

- Sanity testing is very similar to smoke testing. It is the initial testing of a component or application that is done to make sure that it is functioning at the most basic level and it is stable enough to continue more detailed testing.

101. What is the difference between WinRunner and Rational Robot?

- WinRunner is a functional test tool but Rational Robot is capable of both functional and performance testing. Also, WinRunner has 4 verification points and Rational Robot has 13 verification points.

102. **What is the difference between QA and testing?**
- The goals of QA are very different from the goals of testing.
- The purpose of QA is to prevent errors is the application while the purpose of testing is to find errors.

103. **Explain random testing.**
- Random testing involves checking how the application handles input data that is generated at random. Data types are typically ignored, and a random sequence of letter, numbers, and other characters are inputted into the data field.

104. **What is the difference between Quality Control and Quality Assurance?**
- Quality control (QC) and quality assurance (QA) are closely linked but are very different concepts. While QC evaluates a developed product, the purpose of QA is to ensure that the development process is at a level that makes certain that the system or application will meet the requirements.

105. **What is the role of QA in a project development?**
- QA team is responsible for monitoring the process to be carried out for development.
- Responsibilities of QA team are planning testing execution process.
- QA Lead creates the time tables and agrees on a Quality Assurance plan for the product.
- QA team communicated QA process to the team members. QA team ensures traceability of test cases to requirements.

106. **What makes a good QA or Test manager?**
- Knowledge about Software development process
- Improve the teamwork to increase productivity
- Improve cooperation between software, test, and QA engineers
- To improvements the QA processes.
- Communication skills.
- Able to conduct meetings and keep them focused

107. **What is the difference between regression testing and retesting?**
- Regression testing is performing tests to ensure that modifications to a module or system do not have a negative effect on previous releases. Retesting is merely running the same testing again. Regression testing is widely asked manual testing interview questions and hence further research to understand this topic is needed.

108. **Explain the difference between bug severity and bug priority.**
- Bug severity refers to the level of impact that the bug has on the application or system while bug priority refers to the level of urgency in the need for a fix.
- Usually the severity is defined in terms of financial loss, damage to environment, company's reputation and loss of life. Priority of a defect is related to how quickly a bug should be fixed and deployed to live servers.

109. **What is the difference between system testing and integration testing?**
- For system testing, the entire system as a whole is checked,
- whereas for integration testing, the interaction between the individual modules are tested.

110. **Explain the difference between functional and structural testing.**
- Functional testing is considered to be behavioral or black box testing in which the tester verifies that the system or application functions according to specification. Structural testing on the other hand is based on the code or algorithms and is considered to be white box testing.

111. What is difference between Pilot and Beta testing?

- The differences between these two are listed below:
  - o A beta test when the product is about to release to the end user whereas pilot testing take place in the earlier phase of the development cycle.
  - o In beta testing application is given to a few users to make sure that application meet the user requirement and does not contain any showstopper whereas in case of pilot testing team member give their feedback to improve the quality of the application.

112. What is Alpha testing?

- Pre-release testing by end user representatives at the developer's site.

113. What is a failure?

- Failure is a departure from specified behavior.

114. What are Test comparators?

- Is it really a test if you put some inputs into some software, but never look to see whether the software produces the correct result?
- The essence of testing is to check whether the software produces the correct result, and to do that, we must compare what the software produces to what it should produce.
- A test comparator helps to automate aspects of that comparison.

115. Describe how to perform Risk analysis during software testing?

- Risk analysis is the process of identifying risk in the application and prioritizing them to test. Following are some of the risks:

  1.New Hardware.   3. New Automation Tool.   5. Availability of application
  2. New Technology.   4. Sequence of code delivery.   test resources.

- We prioritize them into three categories these are:
  - o High magnitude: Impact of the bug on the other functionality of the application.
  - o Medium: it can be tolerable in the application but not desirable.
  - o Low: it can be tolerable. This type of risk has no impact on the company business.

116. What is Silk Test?

- Silk Test is a tool developed for performing the regression and functionality testing of the application. Silk Test a tool is used when we are testing the applications which are based on Window, Java, web or traditional client/server.
- Silk Test help in preparing the test plan and management of those test plans, to provide the direct accessing of the database and validation of the field.

117. What is difference between Master Test Plan and Test Plan?

- Master Test Plan contains all the testing and risk involved area of the application whereas Test case document contains test cases.
- Master Test plan contain all the details of each and every individual test to be run during the overall development of application whereas test plan describe the scope, approach, resources and schedule of performing test.
- Master Test plan contain the description of every tests that is going to be performed on the application whereas test plan only contain the description of few test cases. during the testing cycle like Unit test, System test, beta test etc
- Master Test Plan is created for all large projects but when it is created for the small project then we called it as test plan.

118. **When is a test considered to be successful?**
   - The purpose of testing is to ensure that the application operates according to the requirements and to discover as many errors and bugs as possible. This means that tests that cover more functionality and expose more errors are considered to be the most successful.

119. **What is defect?**
   - When the expected result does not match the actual result, it is defect.

120. **Define defect density?**
   - Defect density is the total number of defects per lines of code.

121. **What is Defect Life Cycle (DLC)?**
   - New → Assigned → Open → Fixed → Retested →Close

122. **What are the categories of defects?**
   - **Wrong**: The requirements are implemented incorrectly in the application.
   - **Missing**: When requirement given by the customer and application is unable to meet those application.
   - **Extra**: A requirement incorporated into the product that was not given by the end customer. This is always a variance from the specification but may be an attribute desired by the user of the product.

123. **What to do when you find a defect?**
   - If I find a defect, before reporting it I reproduce the bug that I need to make sure that is a valid defect.
   - If it is a small issue, I will go to the developer desk, and he can fix it right away.
   - If it is a big issue, then I open my JIRA and log the defect.
   - If I am not sure it is bug or not, I will talk to SME (subject matter expert it means the person who knows the application better than anyone).

124. **How should testing be conducted?**
   - Testing should be conducted based on the technical requirements of the application.

125. **If developer says not a defect, what to do?**
   - I always make sure that it is a real defect that's why I reproduce it.
   - I take screenshots and give all the steps to reproduce the defect.
   - Actually, one of my biggest challenges that I faced in my current project is that.

126. **Can you test a program and find 100% of the errors?**
   - It is impossible to find all errors in an application mostly because there is no way to calculate how many errors exist. There are many factors involved in such a calculation such as the complexity of the program, the experience of the programmer, and so on. This Manual testing interview questions is the trickiest questions considered by testers.

127. **What is the difference between debugging and testing?**
   - The main difference between debugging and testing is that debugging is typically conducted by a developer who also fixes errors during the debugging phase. Testing on the other hand, finds errors rather than fixes them. When a tester finds a bug, they usually report it so that a developer can fix it.

128. **What is considered to be a good test?**
   - Testing that covers most of the functionality of an object or system is considered to be a good test.

129. When should testing be stopped?

- It depends on the risks for the system being tested. There are some criteria bases on which you can stop testing.
  - Deadlines (Testing, Release)
  - Test budget has been depleted
  - Bug rate fall below certain level
  - Test cases completed with certain percentage passed
  - Alpha or beta periods for testing ends
  - Coverage of code, functionality or requirements are met to a specified point

130. What is the difference between top-down and bottom-up testing?

- **Top-Down** testing begins with the system and works its way down to the unit level.
- **Bottom-up** testing checks in the opposite direction, unit level to interface to overall system. Both have value but bottom-up testing usually aids in discovering defects earlier in the development cycle, when the cost to fix errors is lower.

131. What is the average size of executables that you have created?

- This is a simple interview question about our experience with executables. If you know the size of any that you've created, simply provide this info.

132. Have you performed tests on the front-end and the back-end?

- When I test Front-End, I am actually testing the UI by open up the application and perform testing on UI. If I have done anything on the UI, I have to perform backend testing to see if the change has been made in the database as well. For example, when I update a parent contact information or create new application, I connect to the database and check if the changes are applied to the data or the new application is created or not.

133. What is difference between Front End Testing and Back End testing?

- Front End Testing is performed on the Graphical User Interface, whereas Back End Testing involves databases testing.
- Front end consist of web site look where user can interact whereas in case of back end it is the database which is required to store the data.
- When ender user enters data in GUI of the front-end application, then this entered data is stored in the database. To save this data into the database we write SQL queries.

134. What is the most difficult problem you've found during testing?

- *(This is a simple interview question in which you should provide an example}.* This is one of most tricky manual testing interview questions as your answer will decide your job. You need to answer in such a way that your problem-solving skills and your job. You need to answer in such a way that your problem-solving skills and your eagerness to learn new things, and your dedication towards the job will indicated by your answers.

135. What is your challenge in scrum?

- Since scrum emphasizes cross functional team (it means developer should able to test and testers should able to develop) it is hard to be part of development team as a traditional QA tester. Because generally QAs don't know how to write code. That is why I have to keep myself very competitive. Whenever I have time, I am learning more coding like Java.
- Time change issue → When I save a date the date entered to the database one, they earlier.

136. What is Automation Testing?

- The process of performing testing automatically which reduces the human intervention this is automation testing.
- The automation testing is carried out with the help of some automation tool like QTP, Selenium, WinRunner etc.
- In automation testing we use a tool that runs the test script to test the application; this test script can be generated manually or automatically. When testing is completed then tools automatically generate the test report and report

137. When will you automate?
- If it is taking a lot of manual effort. I run at least once manual and after that I automate it.
- Automation is good for most repetitive functionality

138. What tests can be automated?

• Regression tests	• Functional tests	• Database
• Smoke tests	• API	

139. When will you NOT automate?
- If functionality keeps changing
- If functionality is used only once during the entire project
- **Ad-hoc test** cannot be automated.

140. What is the duration of a scrum sprint? How long is your sprint?
- In my current project our script cycle is 4 weeks. How long is your sprint here? 2 weeks or 4 weeks? (sometimes it is good to ask question. Remember you should not act like an ATM. They generally forget people only answering question. There should be a balance.)
- Our team size is 7 members. 1 SM, 1 PO, 3 developer, 1 MT, 1 AT

141. What is Velocity?
- Velocity is the rate at which team progresses print by sprint.
- I can also say that it cannot be compared to two different scrum teams.

142. What is the "build breaker"?
- The build breaker is a situation that arises when there is a bug in the software.
- Due to this sudden unexpected bug, compilation process stops, or execution fails, or a warning is generated.
- The responsibility of the tester is then to get the software back to the normal working stage removing the bug.

143. What do you know about impediments in Scrum? Give some examples of impediments.
- Impediments are the obstacles or issues faced by scrum team which slow down their speed of work.
- If something is trying to block the scrum team from their getting work "Done" then it is an impediment.
- Impediments can come in any form. Some of the impediments are given as
  o Resource missing or sick team member
  o Technical, operational, organizational problems
  o Lack of management supportive system
  o Business problems
  o External issues such as weather, war etc.
  o Lack of skill or knowledge
- Solution : Teamwork, work hard, communicate well, online connect, mentoring and training

144. What is the difference and similarity between Agile and Scrum?
- Agile is a broad spectrum, it is a methodology used for project management while Scrum is just a form of the Agile that describes the process and its steps more concisely.
- Agile is a practice whereas scrum is a procedure to pursue this practice.
- The similarity that → Agile involves completing projects in steps or incrementally. The Agile methodology is considered to be iterative in nature. Being a form of Agile, Scrum is same as that of the Agile. It is also incremental and iterative.

145. What is increment? Explain.
- An increment is **the total of all the product backlogs items completed during a sprint**.
- Each increment includes all the previous sprint increment values as it is cumulative.
- It must be in the available mode in the subsequent release as it is a step to reach our goal.

146. What do you understand by Daily stand-up?
- The daily stand-up is an everyday meeting (most preferably held in the morning) in which the whole team meets for almost 15 minutes to find answer to the following three questions –
    o What was done yesterday? What is your plan for today?
    o Is there any impediment or block that restricts you from completing your task?
- The daily stand-up is an effective way to motivate the team and make them set a goal for the day.

147. What do you know about Scrumban?
- Scrumban is a Scrum and Kanban based model for the software development.
- This model is specifically used for the projects that need continuous maintenance, have various programming errors or have some sudden changes.
- This model promotes the completion of a project in minimum time for a programming error or user story.

148. State some of the Agile quality strategies?
- Iteration
- Refactoring
- Dynamic code analysis
- Short feedback cycles
- Reviews and inspection
- Standards and guidelines
- Milestone reviews

149. Do you know about Agile Manifesto & its Principles? Explain in brief.
- This is the theory which most of agile/scrum roles aspirant should be on tips.
- Four manifesto values and 12 principles should be explained as much as possible as part of this question.
- Even if it's not explained in 100% accurate manner it should be fine, but intentions of values and principles should come out e.g.
- Manifesto
    o Individuals and interactions over processes and tools
    o Working software over comprehensive documentation
    o Customer collaboration over contract negotiation
    o Responding to change over following a plan
- Guiding Principles
    o Customer Satisfaction
    o Welcome Changing Requirements
    o Working Software is Delivered Frequently (Weeks rather than months)
    o Close, Daily Cooperation between Business People and Developers
    o Project are built around motivated individuals, who should be trusted
    o Face-to-Face Conversation is the best form of communication
    o Working software is the primary measure of progress
    o Sustainable development, able to maintain a constant pace
    o Continuous attention to technical excellence and good design
    o Simplicity - The art of maximizing the amount of work not done - is essential

- o Best architectures, requirements and designs emerge from self-organizing teams
- o Regularly, the team reflects on how to become more effective, and adjusts accordingly

150. What is the use of burn-up and burn-down charts?
- The burn-up chart illustrates the amount of completed work in a project whereas the burn-down chart depicts the amount of work remained to complete a project.
- Thus, the burn-up and burn-down charts are used to trace the progress of a project.

151. Is there any drawback of the Agile model? If yes, explain.
- Yes, there are some drawbacks of the Agile model, some of them are like;
  - o It is not easy to make a prediction about the effort required to complete a task. It becomes more problematic in case of large projects as it becomes difficult to get an idea of the total effort required.
  - o At sometimes, it's not possible to properly focus on the design and documentation of the project
  - o In case the requirements of the client are not understood properly, the final project will not meet the customer requirements. Thus, it will lead to the customer dissatisfaction.
  - o Only the leader who has considerable experience in Agile methodologies is capable to take important decisions. The team members with little or no experience are not involved in decision-making, thus they don't get chance to advance their knowledge.

152. Define Zero Sprint and Spike in Agile.
- Zero Sprint can be defined as the preparation step of the first sprint in Agile.
  - o There are some activities that are required to be done before actually starting the project.
  - o These activities are considered as the Zero sprint; the examples of such activities are – setting the environment for development, preparation of backlogs etc.
- Spike is the type of story that can be taken between the sprints.
  - o Spikes are commonly used for the activities related to the design or technical issues such as research, design, prototyping, and exploration.
  - o There are two types of spikes – functional spikes and technical spikes.

153. What is the role of the Scrum Master?
- The scrum master is the leader as well as coach of the Scrum team.
- The SM is responsible to serve and protect the team from any kind of block that could affect the performance.
- The main role of the SM is to motivate his team to achieve the sprint goal.
- He is focused to build a self-organized and motivated team where each member is familiar with the implementation of Agile and Scrum principles and applications.
- The SM keeps a proper check on the scrum team if they are executing committed tasks properly.
- He is also responsible to increase the efficiency and productivity of the team so that they can achieve the sprint goal effectively.

154. What do you know about a story point in Scrum?
- A story point in Scrum is the unit for the estimation of total efforts that are required to perform or complete a particular task.

155. What is the role of Sashimi in Scrum methodology?
- Sashimi plays an important role in Scrum methodology.
- Sashimi is a technique used by Scrum to check the completion of all the functions created by the developers.
- Using this technique, all the requirements such as analysis, designing, coding, testing and documentation that are used in the constitution of a product are checked and only after that the product is displayed.

156. What do you understand by the term Agile testing?

- Agile testing is a software testing practice that is fully based on the agile principles of software development. It is an iterative methodology where the requirements are the outcome of collaboration between the product owner and team. The agile principles and applications are applied to meet the customer requirements by successful completion of the project.

157. Is it ever suggested to use waterfall over Scrum? If yes, explain when.

- Yes, sometimes it is suggested to use waterfall model over Scrum.
- It is done when the customer requirements are simple, well-defined, fully understood, predictable, and are not subjected to change until the completion of the project.

158. Why does Scrum encourage the use of automated testing for projects?

- Scrum encourages the use of automated (automated performance or automated regression) testing to make the fastest possible delivery of the project . *you may explain some tools that you have used for automating*

159. Explain some common matrices for Agile.

- **Velocity** → Velocity is the average number of points from last 3-4 sprints. It is measured by the summation of the all approved estimates of the stories. It gives an idea of the capacity, progress etc.
- **Cumulative Flow Diagram** → With the help of it, an inspection is done over the uniform workflow. In this diagram/graph, the x-axis represents time whereas the y-axis represents the number of efforts.
- **Work Category Allocation** → it is an important factor that gives a quick information of the time investment i.e. where the time is being invested and which task should be given priority as a factor of time.
- **Time Coverage** → It is the time that is given to a code during testing. It is calculated in percentage as a factor of the number of lines of code called by test suite and the total number of relative lines of code.
- **Business Value Delivered** → It is a term which denotes the working efficiency of the team. The business objectives are assigned numerical values 1,2,3 and so on, as per the level of priority, complexity, and ROI.
- **Defect Removal Awareness** → It is the factor that helps the team to deliver a quality product. The identification of an active number of defects, their awareness, and removal plays an important role in delivering a high-quality product.
- **Defect Resolution Time** → It is a procedure through which the team members detect the defects (bugs) and set a priority for the defect resolution. The procedure of fixing errors/bugs or defect resolution comprises of multiple processes such as clearing the picture of defect, schedule defect fixation, completing defect fixation, generation, and handling of resolution report.
- **Sprint Burndown Matrix** → The sprint burndown chart is a graph to represent the number of non-implemented or implemented sprints during as Scrum cycle. This matric helps to track the work completed with the sprint.

160. Name some methodologies and development where you have used Agile model.

- Some of the methodologies and development where Agile model can be used are –
  - Crystal methodologies
  - Lean software development
  - Dynamic development and Feature driven development

161. Share your experience as Scrum M/Product O/Agile team member and what were your primary responsibilities?

- The trick in this question is whether while explaining you are showing self-organizing and self-motivational team.

162. What was the length of sprints/iterations in your project?

- The idea here is to judge in which kind of environment you have worked. There will be definitely follow up question like was this length fixed in the beginning and never changed? Have you tried with more than this length or less than that?

163. **What do you know about "Planning Poker" technique?**
- Planning poker, also known as Scrum Poker, is a card based agile technique that is used for planning and estimation. To start a session of planning poker technique, the agile user story is read by the product owner.
- The steps performed in the poker planning technique are –
  - Each estimator has a deck of poker cards with the values such as 0, 1, 2, 3, 5, and so on, to denote story points, ideal days or something else that the team uses for estimation.
  - Each estimator has a discussion with the product owner and then privately selects a card on the basis of their independent estimation.
  - If the cards with same value are selected by all estimators, it is considered as an estimate. If not, the estimator discusses the high and low value of their estimates.
  - Then again, each estimator privately selects a card and reveals. This process of poker planning is repeated to reach a general agreement.

164. **How have you done user story mapping & estimation of stories in your projects?**
- Have you used any estimation technique like planning poker, t-shirt, sizing etc.? Whatever technique you used in your project just mention it very clearly.

165. **How is agile testing methodology different from other testing methodologies?**
- The agile testing methodology involves the division of whole testing process into multiple small segments of codes. In every step, these segments of codes undergo testing.
- There are a number of additional processes involved in agile testing methodologies such as team communication, strategic modifications for optimal results and many others.

166. **What is the biggest challenge you faced in your project while handling the Scrum team members?**
- Challenges generally faced in the initial stages of scrum is stabilizing the velocity, team members conflicts, sticking to time-boxing etc.
  - Application should be stable enough to be tested. o Testing always under time constraint
  - Understanding the requirements.
  - Domain knowledge and business user perspective understanding.

167. **Which tests to execute first?**
- Testing the Complete Application. o Regression testing.
- Lack of skilled testers. o Changing requirements.
- Lack of resources, tools and training

168. **Do you have a Scrum Master certification?**
- If you are a certified scrum master, just share the details of your certification like certification exam, score obtained, and the year of passing the certification exam. In case you don't have a certification, mention and highlight your experience in the particular field. Also, let the interviewer know if you are planning to invest in the certification in the near future.

169. **Do you hold any agile certification? Why did you choose this certification?**
- Agile and Scrum methodologies are used to complete a project at earliest.
- Implementing agile principles results in customer satisfaction whereas scrum is known for its flexible feature as per the requirements.

170. **Have you worked with offshore team before?**
- No, I don't. (*Offshore basically means that the team is situated in a different country but is still employed by your company*)

171. **What are the common UI test automation tools?**

- Selenium
  - o Cucumber
  - o TestNG
- Appium
- Protractor
- Winium
- UFT/QTP
- **Katalon Studio**

172. **What is Testware? Test ware?**

- It is the subset of software which helps in performing the testing of application.
- Testware are required to plan, design, and execute tests. It contains documents, scripts, inputs, expected results, set-up and additional software or utilities used in testing.
- Testware is term given to combination of all utilities and application software that required for testing a software package. It is special because it has;
  - o Different purpose
  - o Different metrics for quality and
  - o Different users

173. **How does a client or server environment affect testing?**

- There are lots of environmental factors that affect the testing like speed of data transfer data transfer, hardware, and server etc. while working with client or server technologies, testing will be extensive.
- When we have time limit, we do the integration testing. In most of the cases we prefer the load, stress and performance testing for examine the capabilities of the application for the client or server environment.

174. **Given some example with details regarding some of the typical experience or excessive load working day of a tester or software development engineer in test (SDET) resources?**

Three key tasks are always taken huge time for the tester in any day:

- Understanding the requirements of the project.
- Preparing and executing require test cases based on the client expected functionalities.
- Reporting about the bugs identified on individual functionality developed for the client to the developer and retest the same after redelivery by the developer for ensuring expected functionality properly deliver without any common bug.

175. **Explain about some expert comments of how one tester can decide that provided product are actually ready to move in the live environment?**

This is one of the critical decisions, so it never been taken by the single person or junior guys. Only developer and tester are not involved for taken this decision, higher management is periodically involved in that. Management test mainly ensure by validating below for ensuring product delivery are bugless:

- Validating bug reports provided by the tester. How bug got resolved and retesting done by the tester or not.
- Validating all the test cases written by the tester for that specific functionality, documentation, and confirmation taken from the tester on the same.
- Run automate test cases for ensuring new functionalities does not break any existing functionality.
- Sometimes validating test coverage report, which ensures all the developing component has been covered by test cases written.

176. **Explain about some expert comments of how one tester can decide that provided product are actually ready to move in the live environment?**

This is one of the critical decisions, so it never been taken by the single person or junior guys. Only developer and tester are not involved for taken this decision, higher management is periodically involved in that. Management test mainly ensure by validating below for ensuring product delivery are bugless:

- Validating bug reports provided by the tester. How bug got resolved and retesting done by the tester or not.
- Validating all the test cases written by the tester for that specific functionality, documentation, and confirmation taken from the tester on the same.
- Run automate test cases for ensuring new functionalities does not break any existing functionality.
- Sometimes validating test coverage report, which ensures all the developing component has been covered by test cases written.

177. **What Test Techniques are there and what is their purpose?**

Test Techniques are primarily used for two purposes: a) To help identify defects, b) To reduce the number of test cases.

- **Equivalence partitioning** is mainly used to reduce the number of test cases by identifying different sets of data that are not the same and only executing one test from each set of data
- Boundary Value Analysis is used to check the behavior of the system at the boundaries of allowed data.
- State Transition Testing is used to validate allowed and disallowed states and transitions from one state to another by various input data
- Pair-wise or All-Pairs Testing is a very powerful test technique and is mainly used to reduce the number of test cases while increasing the coverage of feature combinations.

178. **What information should be included in a defect or bug report?**

- A brief summary of the defect
- A full description of the defect including steps to reproduce
- Screenshot attachments if required
- Date the defect was found and raised
- Who reported the defect.
- Severity and/or Priority of the defect
- Which component is the defect assigned.

TYPE OF TESTS

There are many different types of testing that you can use to make sure that changes to your code are working as expected. Not all testing is equal, though, and we will see here how the main testing practices differ from each other.

Manual vs. automated testing

- At a high level, we need to make the distinction between manual and automated tests. Manual testing is done in person, by clicking through the application or interacting with the software and APIs with the appropriate tooling. This is very expensive as it requires someone to set up an environment and execute the tests themselves, and it can be prone to human error as the tester might make typos or omit steps in the test script.
- Automated tests, on the other hand, are performed by a machine that executes a test script that has been written in advance. These tests can vary a lot in complexity, from checking a single method in a class to making sure that performing a sequence of complex actions in the UI leads to the same results. It's much more robust and reliable than automated tests – but the quality of your automated tests depends on how well your test scripts have been written.
- Automated testing is a key component of continuous integration and continuous delivery and it's a great way to scale your QA process as you add new features to your application. But there's still value in doing some manual testing with what is called exploratory testing as we will see below.

The different types of tests

Smoke testing

- Smoke tests are basic tests that check basic functionality of the application. They are meant to be quick to execute, and their goal is to give you the assurance that the major features of your system are working as expected.
- Smoke tests can be useful right after a new build is made to decide whether or not you can run more expensive tests, or right after a deployment to make sure that they application is running properly in the newly deployed environment.

Integration tests

- Integration tests verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected. These types of tests are more expensive to run as they require multiple parts of the application to be up and running.

Regression Testing

- Regression means retesting the unchanged parts of the application. Regression testing is a testing that is done to verify that a code change in the software does not impact the existing functionality of the product.
- This testing makes sure that the product works fine as previously with the newly added functionality or any change in the existing feature or once the bug fix is done. Previously executed test cases are re-executed in order to verify the impact of change.
- Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine, and the new changes have not introduced any new bugs. This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.
- Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the system in Regression Testing, so typically automation testing tools are used for these types of testing.

Functional tests

- Functional tests focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.
- There is sometimes a confusion between integration tests and functional tests as they both require multiple components to interact with each other. The difference is that an integration test may simply verify that you can query the database while a functional test would expect to get a specific value from the database as defined by the product requirements.

FUNCTIONAL TESTING
- Unit Testing
- Smoke testing
- Regression Testing
- Sanity Testing
- Integration Testing
- Acceptance Testing
- GUI Testing
- Usability Testing
- System Testing
- Alpha Testing
- Beta Testing
- User Acceptance Testing

NON-FUNCTIONAL TESTING
- Performance Testing
- Load Testing
- Volume Testing
- Stress Testing
- Security Testing
- Installation Testing
- Penetration Testing
- Compatibility Testing
- Migration Testing
- Recovery testing
- Reliability Testing
- Usability Testing
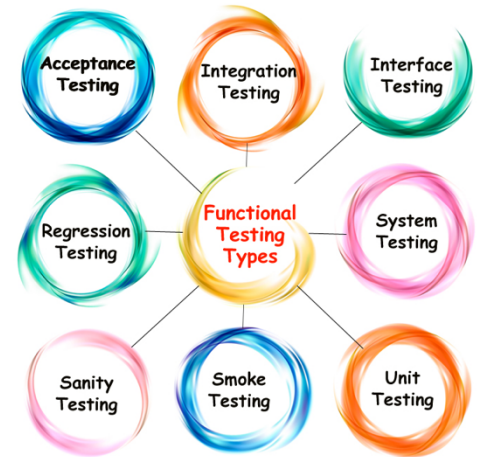
Unit tests

- Unit tests are very low level, close to the source of your application. They consist in testing individual methods and functions of the classes, components or modules used by your software. Unit tests are in general quite cheap to automate and can be run very quickly by a continuous integration server.

End-to-end tests

- End-to-end testing replicates a user behavior with the software in a complete application environment. It verifies that various user flows work as expected and can be as simple as loading a web page or logging in or much more complex scenarios verifying email notifications, online payments, etc...
- End-to-end tests are very useful, but they're expensive to perform and can be hard to maintain when they're automated. It is recommended to have a few key end-to-end tests and rely more on lower level types of testing (unit and integration tests) to be able to quickly identify breaking changes.

Black Box Testing

- Black box testing, which is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing, is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value. The main focus in black box testing is on the functionality of the system as a whole.
- The term 'behavioral testing' is also used for black box testing. Behavioral test design is slightly different from the black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged.

Types of Blackbox testing:
- Functional Testing,
- Non-functional,
- Equivalence Partitioning,
- Boundary Value Analysis,
- Decision Table Testing,
- Error Guessing,
- Graph-based Methods,
- Comparison,
- State Transition tests

White Box Testing

- White Box testing is based on the knowledge about the internal logic of an application's code.
- It is also known as Glass box Testing. Internal software and code working should be known for performing this type of testing. Under these tests are based on the coverage of code statements, branches, paths, conditions etc.
- Conclusion
- The above-mentioned Software Testing Types are just a part of testing. However, there is still a list of more than 100+ types of testing, but all testing types are not used in all types of projects. So, I have covered some common Types of Software Testing which are mostly used in the testing life cycle.
- Also, there are alternative definitions or processes used in different organizations, but the basic concept is the same everywhere. These testing types, processes, and their implementation methods keep changing as and when the project, requirements, and scope changes.

Types of Whitebox testing:
- Unit Testing,
- Execution Testing,
- Mutation Testing,
- Operations Testing
- Statement Coverage,
- Branch Coverage,
- Path Coverage,
- Security Testing,
- Graph-based Methods,
- Comparison,
- State Transition tests,

Acceptance testing

- Acceptance tests are formal tests executed to verify if a system satisfies its business requirements. They require the entire application to be up and running and focus on replicating user behaviors. But they can also go further and measure the performance of the system and reject changes if certain goals are not met.

Performance testing

- Performance tests check the behaviors of the system when it is under significant load. These tests are non- functional and can have the various form to understand the reliability, stability, and availability of the platform. For instance, it can be observing response times when executing a high number of requests or seeing how the system behaves with a significant of data.
- Performance tests are by their nature quite costly to implement and run, but they can help you understand if new changes are going to degrade your system.

Ad-hoc Testing

- The name itself suggests that this testing is performed on an ad-hoc basis i.e. with no reference to the test case and also without any plan or documentation in place for such type of testing. The objective of this testing is to find the defects and break the application by executing any flow of the application or any random functionality.
- Ad-hoc testing is an informal way of finding defects and can be performed by anyone in the project. It is difficult to identify defects without a test case but sometimes it is possible that defects found during ad-hoc testing might not have been identified using existing test cases.

Boundary Value Testing

- This type of testing checks the behavior of the application at the boundary level.
- Boundary value Testing is performed for checking if defects exist at boundary values. Boundary value testing is used for testing a different range of numbers. There is an upper and lower boundary for each range and testing is performed on these boundary values.
- If testing requires a test range of numbers from 1 to 500 then Boundary Value Testing is performed on values at 0, 1, 2, 499, 500 and 501.

- Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and looking for defects that exist in the application. Sometimes it may happen that during this testing major defect discovered can even cause system failure.
- During exploratory testing, it is advisable to keep a track of what flow you have tested and what activity you did before the start of the specific flow.

How to automate your tests

- An individual can execute all the tests mentioned above, but it will be very expensive and counter-productive to do so. As humans, we have limited capacity to perform a large number of actions in a repeatable and reliable way. But a machine can easily do that rapidly and will test that login/password combination works for the 100th time without complaining.
- To automate your tests, you will first need to write them programmatically using a testing framework that suits your application. PHPUnit, Mocha, RSpec are examples of testing frameworks that you can use for PHP, JavaScript, and Ruby respectively. There are many options out there for each language so you might have to do some research and ask developer communities to find out what would be the best framework for you.
- When your tests can be executed via script from your terminal, you can have them be automatically executed by a continuous integration server like Bamboo or use a cloud service like Bitbucket Pipelines. These tools
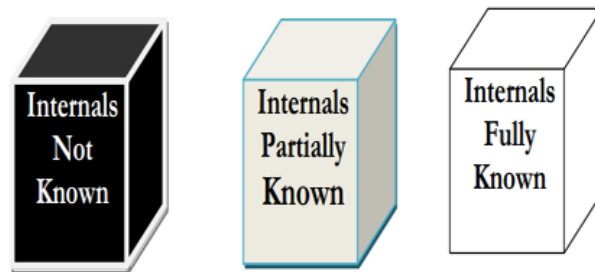
Comparison between the Three Testing Types

	Black Box Testing	Grey Box Testing	White Box Testing
1.	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2.	Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
3.	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4.	-Testing is based on external expectations -Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5.	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6.	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
7.	This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested