

## 1) What is a software? Different types of software?

- ◆ In short computer software is a set of programs, procedures, functions, associated data and/or its available documentation.
  - i) System Software
  - ii) Application Software
  - iii) Client server Application
  - iv) Web based application
  - v) Mobile application

## 2) Describe the differences between client server and web-based application?

- ◆ For client server application the software must be installed in the client's machine. Client server application built on two tier architecture.
- ◆ Web based application can be accessed through a browser. Web-based application is usually built on multi-tier architecture

## 3) What is Software Quality Assurance (SQA)?

- ◆ SQA involves the entire software development process.
- ◆ Monitor and improve the development process.
- ◆ Making sure that the requirement matched with the expectation.
- ◆ Ensure if there is any bug, we fix it.
- ◆ It is oriented PREVENTATION.



## 4) What is Software Testing?

- ◆ Software testing is a process to evaluate the actual software product matches expected requirement to make sure the software product is bug free.
- ◆ I intentionally do negative and positive testing if there are any fixes required.
- ◆ It is mostly focused on DETECTION.
- ◆ Example: I can give you an example of negative testing and positive testing. Let's say I want to log in to the Facebook account and I put the wrong username and password and my expectation was to get the failed result and it meets my expectation which is called negative testing. And positive testing is exactly the opposite where I put the valid username and password to get successful result and met my expectation which is called positive testing.

## 5) What is main goal of software testing?

- ◆ The goal of Software Testing is to ensure the quality of the software that meets the expectation of production needs which compare the User Requirements to developed product.

## 6) What is SDLC?

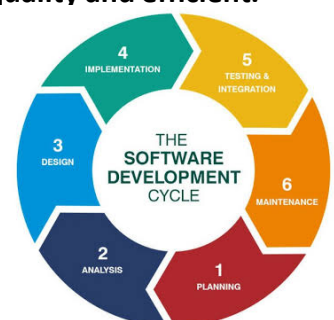
- ◆ SDLC referred to as the Software Development Life Cycle.
- ◆ It is the process of developing software through business needs/requirements.

## 7) What is SDLC used for?

- ◆ SDLC useful in developing the products or customer which ensures the high quality and efficient.

## 8) Describe the phases of Software Development Life Cycle?

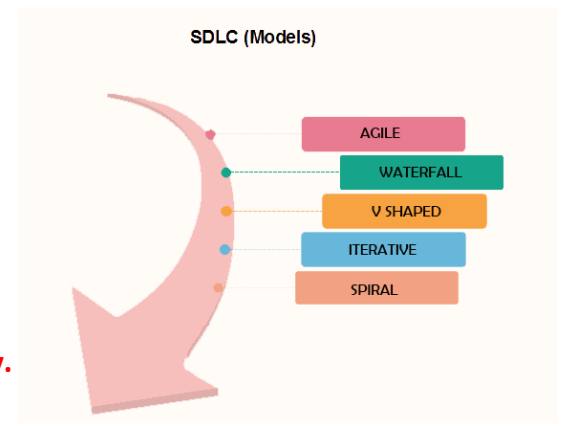
- ◆ Phases of SDLC: The entire SDLC process divided into the following stages:
  - i) **Phase 1:** Planning and Requirement Analysis
  - ii) **Phase 2:** Defining Requirements and designing architecture
  - iii) **Phase 3:** Building or Developing the Product
  - iv) **Phase 4:** Testing the Product



v) Phase 5: Deployment and Maintenance

9) What are the different types of SDLC methodologies?

- ◆ Waterfall methodology.
- ◆ Agile methodology.
- ◆ Incremental Model.
- ◆ V-shaped model.
- ◆ Kanban.



10) Describe the differences between Waterfall and Agile Methodology.

- ◆ Waterfall:
  - i) Waterfall is sequential methodology where each phase of the SDLC is completed chronologically.
  - ii) In waterfall the development cycle is longer.
  - iii) In Waterfall detailed documentation plays an important role.
  - iv) It's hard changing the requirements once the project development starts.
- ◆ Agile:
  - i) In Agile is an incremental methodology where all the phases of SDLC occurs at the same time.
  - ii) In agile the cycle is broken down into smaller sprints each lasting 2 to 3 weeks.
  - iii) In agile preferred discussion and continuous improvement.
  - iv) Agile allows changes in project development requirement.

11) How would you strategize a release in Agile?

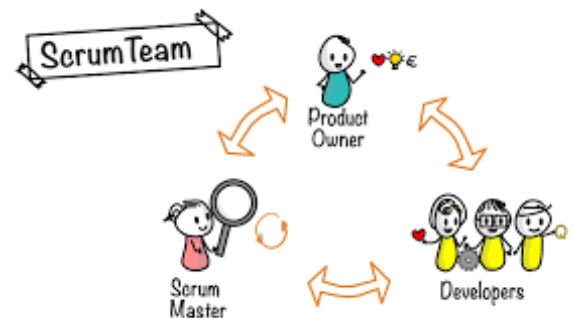
- ◆ I will have the daily scrum meeting to discuss on what each team member is working with.
- ◆ A sprint planning meeting to discuss the backlog and decide which user story will be developed in this sprint and tested.
- ◆ Sprint review or demo meeting where QA team show to the product owner stories that were developed, tested and ready for release.
- ◆ We also have a retrospective meeting to discuss how to improve our agile development this is the way to have a release in an agile project.

12) What is Scrum?

- ◆ Scrum is a software development process framework for project management based on Agile Methodology.
- ◆ Scrum uses self-organizing cross-functional teams to incrementally develop products in short cycles.

13) What are the roles in Scrum?

- ◆ The roles in Scrum:
  - i) Product Owner
  - ii) Development Team
  - iii) Scrum Master
  - iv) End Users



14) What is a Sprint?

- ◆ Sprints are core components of the agile methodology.
- ◆ Sprint is the basic unit of development in Scrum which progress via iterations called Sprint.
- ◆ It lasts from two to three weeks.

15) Describe Scrum Development Process? Or What are the meetings you attend in your Agile methodology?

- ◆ Backlog grooming: As an on-going activity, the product owner and the scrum team should be actively reviewing their backlog and ensuring the work is appropriately prioritized and contains clear information. This allows each piece of work to be more easily planned into a sprint as it can be more accurately estimated. As a tester I actively try to be involved in this process as it is my first opportunity to assess the requirements and the information provided. It also allows me a chance to gather information required for testing, which allows me to provide more reliable estimates.
- ◆ The objective is to be in a position where for any piece of work presented in planning you know exactly what the work requires. You should then have a good idea of what you will test and therefore provide reliable estimates. If this is not the case, then the work cannot be effectively planned into the sprint.
- ◆ **Sprint Planning meeting, Daily Scrum Meeting, Review Meeting, Sprint Retrospective.** CREATE BY - ATUL KUMAR (LINKEDIN)
- ◆ A **SPRINT PLANNING MEETING** is the starting point of Sprint. It is the meeting where the entire scrum team gathers, the SCRUM Master selects a user story based on the priority from the product backlog and the team brainstorms on it. Based on the discussion, the scrum team decides the complexity of the story and sizes it as per the Fibonacci series. The team identifies the tasks along with the efforts (in hours) which would be done to complete the implementation of the user story.
- ◆ On each day of the sprint, all team members attend a **DAILY SCRUM MEETING** for 15 to 20 minutes to provide updates on the workday before, the task they will perform today and highlight any obstacles.
- ◆ At the end of every sprint cycle, the SCRUM team meets again and demonstrates the implemented user stories to the product owner. The product owner may cross verify the stories as per its acceptance criteria. It's again the responsibility of the Scrum master to preside over this **REVIEW MEETING**.
- ◆ **SPRINT RETROSPECTIVE** is conducted at the end of each sprint with the entire team including Scrum Master and to discuss what went well, what did not go well and what steps should be taken to improve the process.

#### 16) What is Defect/Bug?

- ◆ Defect is the unusual behavior or problem in the software program.
- ◆ It's a condition when a product does not meet the requirement or end-user expectation.

#### 17) What are the few types of defect?

- ◆ Arithmetic,
- ◆ Logical,
- ◆ Interface,
- ◆ Syntax,
- ◆ Multithreading.

#### 18) List the most popular bug tracking tools available in the market?

- ◆ Most of the organization uses a defect tracking tool like Quality Center, Jira, Bugzilla, Team Foundation Server, and Clear Quest to report the software problem.

#### 19) What should you do after finding a defect?

- ◆ **RECREATE THE DEFECT:** Once you find a defect, we must try to recreate (meaning that we should be able to reproduce it) at least 3 times so that we are sure that it is a defect.

CREATE BY - ATUL KUMAR (LINKEDIN)

- ◆ **ATTACH THE SCREEN SHOT (SUPPORTING DOCUMENT):** Once we confirm that it is a defect, and then it is a good idea to attach supporting documents when we log (write) a defect. For example, screen shot, requirement document etc.
- ◆ **LOG THE DEFECT:** Now, the next step is, we need to log it. Depending on the company what kind of tools they are using. In some cases, Excel sheet is used log defects.

#### 20) Explain Bug Life Cycle?

- ◆ When the bug is logged it gets NEW status
- ◆ When the bug is accepted it gets OPEN status
- ◆ When the bug is assigned to developer it gets ASSIGNED status
- ◆ When the bug is being fixed it gets "In Progress Status"
- ◆ When the bug is fixed it gets "DEV COMPLETE" or "FIXED" status
- ◆ When the bug is tested and passed it gets "VERIFIED" status
- ◆ If a decision is made to fix the bug in later release it gets "DEFERRED" status

#### 21) What is the biggest bug you find out in your last projects/framework?

- ◆ I cannot go details about the project based on the confidentiality of the company.
- ◆ But I can tell you when I was logging in to recently, I saw a null error and assign to developer.

#### 22) What is STLC?

- ◆ STLC stands for Software Testing Life Cycle which performed by the testing team to ensure the quality of the software.

#### 23) Different stages of STLC?

- ◆ Requirement Analysis – it should be clear, consistent, and testable.
- ◆ Test Planning – developing the test cases and manage them using tools.
- ◆ Designing/Development – developing the test cases based on the requirement.
- ◆ Environment Setup – when integrated environment is ready to validate the product/test case.
- ◆ Execution – we execute the case in real time and try finding bugs.
- ◆ Closure – once testing is completed. Matrix, reports are ready to publish/share with the team.

#### 24) Difference between SDLC vs STLC?

- ◆ **SDLC**
  - i) Concept – Business Analyst gathers requirements. Development team analyzes the requirements.
  - ii) Requirement – the development team starts analyzing from the architecture and the design perspective.
  - iii) Design Stage – The architecture of SDLC helps you develop a high-level and low-level design of the software based on the requirements.
  - iv) Development Stage – Development team starts developing the software. Integrate with different systems. Once all integration is done, a ready to test software or product is provided.
  - v) Testing Stage – The actual testing is carried out in this phase. It includes unit testing, integration testing, system testing, defect retesting, regression testing, etc.
  - vi) Implementation & Maintenance - Once sign-off is received from various testing team, application is deployed in prod environment for real end users.
- ◆ **STLC**
  - i) Concept – no active involvement but can participate on meetings.
  - ii) Requirement – read requirement document try to understand as requirement: clear, consistence and testable.

- iii) Design Stage – read design document, writing test cases based on design prototype of application.
- iv) Development Stage – black box tester not active but white box tester do unit testing.
- v) Testing Stage – all testing needs to do in this stage with manage Defect Life Cycle.
- vi) Implementation & Maintenance – do regression test if any update version or defect fix.

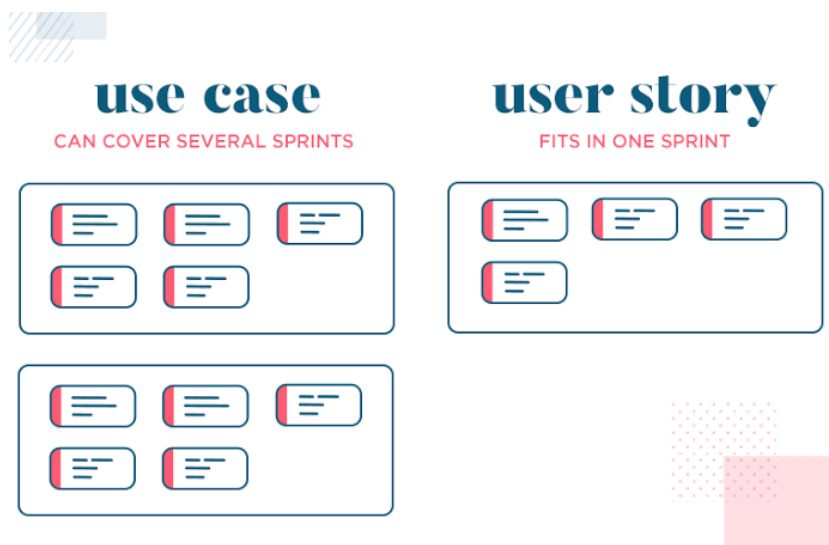
## 25) During SDLC, what is your role?

- i) Discuss with the BS and requirement are clear, consistent, and testable and have the required acceptance criteria.
- ii) Design phase – write the test cases.
- iii) Develop – help the developer with unit testing if you are a white box tester.
- iv) Testing – executing the test case
- v) Deployment – Regression Risk and fix

## 26) Difference between Use case and user stories?

### i) Use Case:

- (1) Used in Waterfall Methodology
- (2) Very detailed
- (3) It is documented in the FRD
- (4) Time consuming to create
- (5) Format:
  - (a) Use case name
  - (b) Description
  - (c) Pre-condition
  - (d) Post-condition
  - (e) Basic path
  - (f) Alternative paths
  - (g) Exception path



### ii) User stories:

- (1) Used in Agile Methodology
- (2) Less detailed
- (3) It is logged onto the JIRA tool
- (4) Relatively faster
- (5) Format:
  - (a) As a WHO  
I want WHAT  
So that WHY
  - (b) Acceptance Criteria
    - (i) Given (a context)
    - (ii) When (an event)
    - (iii) Then (an outcome)
  - (c) Supporting Models

## 27) What is verification and validation?

### ♦ Verification:

- i) **Verification typically involves reviews and meetings to evaluate documents, plans, code, and specifications.**
- ii) **This can be done with checklists, issues list, walkthroughs, and inspection meetings.**
- iii) **It answers the question, Am I building a product, right?**
- ◆ **Validation:**
  - i) **Validation typically involves actual testing and takes place after verifications are completed.**
  - ii) **This can be done by doing testing such as white box, Gray box & Black box testing.**
  - iii) **It answers the question, Am I building the right product?**

#### 28) What is the difference between Black-Box and White Box Testing?

- ◆ **BLACK BOX TESTING:** Black Box Testing is a software testing method where testing is possible without the internal knowledge of code.
  - i) **Functionality Testing is a form of Black box testing.**
- ◆ **WHITE BOX TESTING:** White Box Testing is a software testing method where testing is done with the internal knowledge of code.
  - i) **Unit testing is a form of white box testing.**

#### 29) What is Gray box testing?

- ◆ **It's a combination of white and black box testing.**
- ◆ **The aim is to search for defects and tester knows partially internal structure.**

#### 30) Level Of TESTING?

- ◆ **UNIT TESTING**
  - i) **Unit testing is a type of software testing where individual units of a software are tested.**
  - ii) **The purpose is to validate that each unit of the software code performs as expected.**
  - iii) **Unit Testing is done during the development (coding phase) of an application by the developers.**
  - iv) **Unit testing is typically done by the programmer/developer.**
- ◆ **INTEGRATION TESTING**
  - i) **Integration testing is performed to determine that every single piece of an application such as back-end data, front-end operating system, hardware, software, networking connectivity and all sub systems are interacting with each other as per requirement.**
- ◆ **SYSTEM TESTING**
  - i) **System Testing ensures the entire operation of an application which includes the internal and external functionalities, Operating System, front-end, back-end, database, networking, and integrations are working as per requirements. System Testing is usually performed after all the modules are developed and connected to each other. How much CPU, hardware capacity you have?**
- ◆ **UAT (USER ACCEPTANCE TESTING)**
  - i) **UAT also known as beta or end user testing. UAT is done in the staging environment of testing.**
  - ii) **It defined as testing the software by the user or client to determine whether it can be accepted or not.**
- ◆ **FUNCTIONAL TESTING**
  - i) **Functional testing is a type of software testing and a type of black box testing whereby the system is tested against the functional requirements/specifications.**
  - ii) **It is detail steps and required to verify the each and every module, every box, buttons and their properties are behaving as expected by the specifications.**

- iii) The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.
- iv) Example of functional testing types are unit testing, regression testing, Integration testing.

◆ **NO-FUNCTIONAL TESTING**

- i) Non-functional testing is defined as a type of software testing to check non-functional aspects/features of an application
- ii) Example of non-functional testing type are load/performance testing, usability testing and security testing.
- iii) An example of non-functional testing would be to check how many people can login into a software at the same time.

◆ **508 TESTING**

- i) Section 508 testing allows for an organization to ensure that the software meets 508 Compliance requirements. It is for people with disabilities to help.

◆ **DATABASE TESTING**

- i) Database testing includes the testing of actual data and database reliability. SQL scripting is generally used to test database.

◆ **PERFORMANCE TESTING**

- i) Performance testing is a type of testing where we check the system response time.
- ii) We use JMeter do performance testing. We can also do inside selenium project.

iii) **LOAD TESTING**

- (1) **Testing an application under heavy loads.**
- (2) **How many people can be in the application at same time.**

iv) **STRESS TESTING**

- (1) **Stress Testing is measured to check the extra load for extended period of time.**
- (2) **How many people can be in the application same time for how long.**

◆ **USABILITY TESTING**

- i) This testing is to determine the end user interaction to an application contains the user-friendly functionalities that serve the software operation easy and simple. Programmers and Testers are not usually appropriate for Usability Testing.

◆ **COMPATIBILITY TESTING**

- i) Compatibility testing is done to ensure the application is works properly with different platform and browser.
  - (1) Platform compatibility – for different operating system
  - (2) Browser compatibility – for different web browser

◆ **USER INTERFACE TESTING**

- i) User interface Testing is to check a user's interaction with the software.
- ii) The goal of UI testing is to ensure that the objects within the UI function as expected and conform to industry standards

◆ **END-TO-END TESTING**

- i) End-to-End testing is testing a complete application environment in a situation that mimics real-world use such as:
  - (1) Using Selenium Java to do Front End testing.
  - (2) Using API and RestAssure to do Middle End testing.
  - (3) Using SQL to do Backend testing.



ii) And network communications, or interacting with other hardware, applications, or systems.

◆ **SHOW STOPPER**

i) A showstopper is a defect or bug that stops the user for further action (testing).

ii) It has no work around. In other words, it stops everything, and the user cannot go any further.

◆ **REGRESSION TESTING**

i) Regression testing is a type of testing that is done to verify that a code change in the software doesn't impact the existing functionality of the product.

ii) This is to make sure the product works fine with the new functionality, bug fixes or any change in the existing feature.

iii) We executed the test cases over and over to verify.

31) **What is Regression Fix?**

◆ Fix means, bug or defect fixed.

◆ Regression Fix means if any bug change, any defect fix, then we must run the regression suite again. This is call regression fix.

32) **What is Regression Risk?**

◆ Risk means, new feature added.

◆ Regression Risk means if there is any new feature added, any new thing added, any functionality added then, I have to run those test case again.

33) **What is Regression Strategy?**

◆ Regression testing involves testing the unchanged parts of an app again and again.

◆ It ensures that the previous functionality of an application is working as intended and the new feature that has been added doesn't introduce any new bug or error.

◆ We approach using test automation such as QTP (Quality Test Professional).

34) **What is Regression Testing in details?**

◆ Repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.

◆ In simple words, We do regression testing by re-executing the tests against the modified application to evaluate whether the modified code breaks anything which was working earlier. Anytime we do modify an application, we should do regression testing.

◆ Regression testing gives confidence to the developers that there is no broken functionality after modifying the production code. It makes sure that there are no unexpected side effects.

35) **When do we do Regression Testing?**

◆ We do regression whenever the production code is modified. Usually, we do execute regression tests in the following cases:

◆ When new functionalities are added to the application. Example: A website has a login functionality which allows users to do login only with Email. Now the new features look like "Providing a new feature to do login using Facebook".

◆ When there is a Change Requirement Example: "Remember Password" should be removed from the login page which is available earlier

◆ When there is a Defect Fix. Example: Imagine, 'Login button' is not working in a login page and a tester reports a bug stating that the login button is broken. Once the bug is fixed by the developers, testers test it to make sure whether the Login button is working as per the expected result. Simultaneously testers test other functionalities which are related to login button.



- ◆ When there a Performance Issue Fix. Example: Loading the home page takes 5 seconds. Reducing the load time to 2 seconds
- ◆ When there is an Environment change. Example: Updating the Database from MySQL to Oracle

### 36) How we do regression testing?

- ◆ Regression Testing Manual or Automation?
- ◆ Regression tests are generally extremely tedious and time-consuming. We do regression testing after every deployment, so it would make life easy to automate test cases instead of running manually on each and every time. If we have thousands of test cases, it's better to create automation test scripts for the test cases which we do on every build(i.e., regression testing).
- ◆ Automated regression test is the best practice and it is the choice of organizations to save a lot of time and to run nightly builds.

### 37) Regression Testing example?

- ◆ Assume there is an application which has a functionality to 'Add', 'Save', and 'Delete'. These functions allow users to add data, save data and delete data. Now developers are developing a new feature i.e., 'Update'. This feature allows users to edit and update the data. As a tester, you have to verify the introduction of a new feature (ie., Update) impacting the existing features (such as 'Add', 'Save', and 'Delete' functions) or not.

### 38) How do we perform Regression Testing?

- ◆ Steps to carry out to perform regression tests are
  - i) Step 1: Get clear understanding on what are the changes made to the software
  - ii) Step 2: Analyse which part of the software might be impacted with the modified code
  - iii) Step 3: Make sure whether you have to follow which type of regression such as Unit, Partial, or Full regression.
  - iv) Step 4: Execute the selected test cases.

### 39) What are the types of Regression Testing?

- ◆ There are three types of Regression Testing.
  - i) Unit Regression - Unit Regression Testing is done during the unit testing phase and the code is tested as a single unit without any dependencies. Dependencies if any are temporarily blocked to test the unit independently.
  - ii) Partial Regression - Partial Regression Testing means when regression testing is done on a module where the change in the code is related to that particular module and there wont be any change impact on other modules.
  - iii) Complete Regression - Complete Regression Testing means when regression testing is done on a number of modules where the change in the code is related to the other module is uncertain.

### 40) What are the Regression Testing Techniques?

- ◆ Software maintenance is the process of modifying a software application which includes improvements, bug fixing, deletion of existing features, and optimization. When we do these modifications, the system may work incorrectly. To avoid issues while modifying the code, we do follow regression testing techniques. These techniques help us to eliminate the risks associated with the modifying code.
- ◆ Following are the various techniques.

- i) **Retest All** - All the test cases in the existing test bucket or test suite are re-executed to verify that there are no bugs that occurred due to the modified code. This is a very expensive method as it needs enormous time as well as resources when compared to the other techniques.
- ii) **Regression Test Selection** - By using the regression test selection technique, we select a part of test cases from the test suite to run to verify whether the modified code affects the application or not instead of re-executing the complete test suite.
- iii) **Test Case Prioritization** - By using this technique, we select the test cases with high priority first to run instead of a medium and low priority ones. Test case priority depends on its business impact, frequency, and criticality.

#### 41) Difference Between Re-Testing and Regression Testing

- ◆ Retesting is a testing types to test the functionality or bug again to ensure that the defects which were found and posted in the earlier build were fixed or not in the latest build.
- ◆ Regression testing is a testing types to repeat the testing of an already tested program, after modifications, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software component.

#### 42) What are the Challenges of Regression Testing?

- ◆ There are some challenges in performing regression testing, even though it plays a crucial role in the QA process.

#1. **Right Tool** - Selecting the appropriate tool plays a key role in the success of regression test automation. If we select an inappropriate tool then regression test automation will become time-consuming and won't generate good ROI.

#2. **Resources** - To utilize the selected tool in a proper manner to generate good ROI, we need to have skilled resources who can use the tool effectively.

#3. **Time Consuming** - Regression testing involves running the same test cases again and again. It leads to large test suites and consumes more time to complete the test execution. Sometimes due to time constraints, we may not run the entire regression test suite.

#4. **Test Coverage** - To overcome the time consumption, we have to minimize the test suites. We may not achieve maximum test coverage when minimizing the test suites.

#### 43) What are the Automated Regression Test Tools?

- ◆ In most cases, the software undergoes frequent changes and we spent a lot of time to execute these regression test cases and which leads to high-cost and time consuming.
- ◆ To avoid these problems we can choose automated regression tests.
- ◆ Automation testing is a great way to run regression tests in efficient and cost effective way. It also helps testers not to test the repeated and boring tests.
- ◆ In general, regression test suites have to be updated frequently. So while selecting a tool for your needs make sure the tool has the ability to update a test suite.
- ◆ Here are some of the regression test tools that help you in automating your regression test cases efficiently and quickly.
  - i) **Selenium**: This is a free tool that is used to automate web applications.
  - ii) **Katalon**: Katalon studio comes with Katalon Recorder which acts as a record and playback tool. Katalon Studio IDE supports the creation of test cases in Java and groovy.
  - iii) **Ranorex**: Ranorex Studio is a codeless test automation tool that accelerates regression tests for desktop, web, and mobile.

#### 44) How To Select A Regression Test Suite?

- ◆ Most of the bugs found in the production environment occur because of the changes did or bugs fixed at the eleventh hour i.e. the changes done at a later stage. The bug fix at the last stage might create other issues/bugs in the Product. That's why Regression checking is very important before releasing a Product.
- ◆ Below is a list of test cases that can be used while performing this Test:
  - i) Functionalities which are frequently used.
  - ii) Test cases that cover the module where the changes have been done.
  - iii) Complex test cases.
  - iv) Integration test cases which include all the major components.
  - v) Test cases for the core functionality or feature of the Product.
  - vi) Priority 1 and Priority 2 test cases should be included.
  - vii) Test cases that frequently fail or recent testing defects were found in the same.

45) How many test cases have you automated in a day?

- ◆ Example: "The number usually depended on the complexity of the test cases. With a limited complexity, I could reasonably automate five to six test cases per day. For more complex scenarios, I was able to automate only one or two test cases per day. I would also break down the most complex test cases into elements like take input, calculation and verifying output to organize the project. Some of these test cases took me two days to complete. So it really depends on the complexity of the test case."

46) How would you select a test case for automation?

- ◆ Example: "I would first determine whether the test case is critical to the business or if it just serves an end-to-end scenario. Then, I would plan the execution frequency according to automation needs and begin the test run, while monitoring for defects. As the automation test concludes, I would analyze the results within a time frame that does not exceed the time it would take for manual execution."

47) How do you decide which tools to use for automation testing?

- ◆ Example: "I would start by looking at the requirements to identify which testing scenarios I want to automate. Then I would research the tools that support the project requirements. For these, I determine whether the budget can support necessary resources, either by supplying the necessary tools or hiring skilled personnel as needed. Finally, I would compare the tools needed to ensure they meet key criteria and choose tools that not only fit the criteria but stay within the company budget as well."

48) What types of tests would you not automate?

- ◆ Example: "I would not automate test cases that are seldom executed or that are more quickly executed manually. I also would avoid automation testing on cases that extend to exploratory and usability testing, as these test types could present misuse of time and resources."

49) Regression Testing is needed to what extent?

- ◆ Well, it depends on application nature and scope of changes. If the scope is large then testing needs to perform thoroughly and it will take time. The complete information can be obtained from the developer about the scope, nature of applications, the total amount of change.

50) What is the best time to perform the Regression Testing?

- ◆ Mostly, the regression testing is carried out throughout the development process and this is a continuous process too. Still, the best time to perform regression testing is after smoke testing or at the end of function testing after a short release.

51) What are the best practices you followed to perform the Regression Testing successfully?

- ◆ Here, are some of the best practices listed below based on my experience to perform the regression testing successfully without any failure.
  - i) The test cases should run automatically daily in the evening when changes are made to the application.
  - ii) In this way, bugs can be identified at an early stage and it will be fixed immediately.
  - iii) It will also reduce the risks associated with the release by covering all the defects earliest at the end of the release cycle.

52) Why do we need the Regression testing tools?

- ◆ Regression tools are vital to automate the testing process. When we are loaded with a plenty of test cases whose manual execution is impossible and time – consuming too. Here, we need regression testing tools where test cases can be executed automatically and it saves your precious time too. To which level, you wanted to automate the testing process depends on a number of test cases remaining to execute.

53) Have you used any regression testing tool during any project?

- ◆ Yes, I have worked on selenium testing tool during my past work experiences. This is a popular automation tool where test scripts are automated as the convenience.

54) Do Regression testing of GUI application is possible?

- ◆ Well, testing of GUI applications is tough when changes are made to the interface. Still, this is possible to write the effective test cases that need continuous improvements every time the GUI is changed. The process is a little difficult so the best idea is utilizing a number of testing tools to make the process easier.

55) What are the strategies to follow while conducting regression testing?

- ◆ Well, there are two popular strategies that you should follow –
  - i) Either you may run all the test cases,
  - ii) Or you need to select a set of test cases based on the nature and scope of the changes.

56) When will QA team perform Regression Testing?

- ◆ Regression testing is performed when a new build comes to the QA team and it is prone to bugs. Here, quality testers need to remove the bugs during each release cycle and make sure that application is still working fine as earlier.

57) Can you explain the concept of Regression Testing with the help of Example?

- ◆ Yes, I will explain to you the concept of regression testing with a simple example and easy to understand for everyone. Take an example of a project having multiple modules like Admin module, employee module, personal details modules etc. there is some error in admin module like users are not able to login to their account even with the right credentials. So, this is an error and need to be fixed by the development team. Here, Development team modified existing code or added new code to fix this issue and the module is sent back to the testing team again.
- ◆ Now the testing team will check either modified code has not affected the behavior of the rest of the application or it is giving output as per expectations only or still needs changes. This is called the Regression testing where new code should never affect the functionality of application during bug fixing. I hope you better understand the concept now why Regression testing is necessary and how is it performed.

58) What are the important facts to focus on while writing Regression test cases?

- ◆ Writing a test case is the most important step while performing test execution process and in-depth knowledge can always make the testing process easier and quicker.
- ◆ To write the test cases, you should have a clear understanding of the client requirements.
- ◆ Each requirement should be presented in the form of test cases and make sure nothing important details are left out.
- ◆ Based on the document specifications, remember that functional and non-functional requirements are represented well.
- ◆ Test cases should be reviewed from time to time and make sure that they are not repeated.
- ◆ As soon as changes are made to the application, design new test cases if old ones are not valid.
- ◆ The structure of a test case should be kept simpler so that it can be validated for a wider range and understood by everyone.

#### 59) What is Testing Pyramid?

- ◆ Testing Pyramid is a framework that can help both developers and QAs create high-quality software. It reduces the time required for developers to identify if a change they introduced breaks the code. It can also be helpful in building a more reliable test suite.
- ◆ Testing Pyramid had 3 level of testing:
  - i) Unit Testing (using Junit or TestNG Assert, assertTrue, Assert.assertEquals...)
  - ii) Integration Testing (API testing manually with postman or rest assured with automation testing)
  - iii) End to end Testing (UI level testing using multiple scenarios cucumber feature file, StepDefinition, Pagefactory with Selenium and Java)

#### 60) How many test cases did you do in your regression suite?

- ◆ In my project, I have regression suite running every night with around 300 test cases. Every morning I check to see the HTML reports is passing at least 98%. I identify any failed test cases and if it's a major issue I report to my team member.

#### 61) How to write Regression test cases? What are the criteria?

- ◆ Regression test cases are also based on the requirement documents. They are written more into detail and with every release (build), the testers need to do regression testing.
- ◆ The criteria for regression testing are; there should be no major defects while we do our smoke test and functional testing.

#### 62) How did you do functionality testing?

- ◆ For functionality testing I analyzed requirement, identified test scenario, prepare test data, developed test cases, executed test cases, and circulated test execution report.

#### 63) Difference between Sanity, Smoke and Ad Hoc testing in short amount of time?

- ◆ **SANITY TESTING** – Sanity testing will have details step with specific feature/test is running from test suite to check if system is running or not. It verifies the bugs those fixed in the previous build and new features. We need details documentation to test.
- ◆ **SMOKE TESTING**– Smoke testing is high level testing with minimum steps to verify the application is running or not. It verifies the main functionalities but not in deep. We need minimum documentation to test.
- ◆ **ADHOC TESTING** - Ad hoc testing is informal software testing where we try to find out if there are any errors. No need any documentation.

#### 64) What is Test Case?

- ◆ Test Case is a detail document that describes step by step process to execute a test.

- ◆ Detail steps to test specific function of an application
- ◆ Expected and actual result
- ◆ Pass or fail result
- ◆ Description for each step
- ◆ Information on person executing the test

65) How do you develop Test Cases?

- ◆ We develop Test Case and based on analyzing the Test Plan and SRS.

66) What do you include in a test case?

- ◆ Test cases should include Test case ID, Test case title, Description, Test steps, expected result and Actual result (once tested).

67) When to develop Test Cases?

- ◆ Before the application is ready
- ◆ Developed the detail functional requirement are ready and analyzed.
- ◆ Detail user interface design.
- ◆ A developer writes codes based on the test cases.

68) What is the flow of writing Test Cases?

- ◆ Application's all objects, modules and screens or windows are related to each other by the design based on business rules.
- ◆ Following the business rules to complete a specific scenario that follows the flow of functionalities of an application to write the test cases is called "flow of writing test cases".

69) How do you determine what to test in an application?

- ◆ First, we have the test cases (or test scripts) that are written based on the requirement document.
- ◆ This pretty much covers what functionalities to test.
- ◆ Therefore, looking at the test cases tells us what to test in the application.

70) Difference between test scenarios, test plan, test script?

- ◆ **Test Scenarios:** A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.
- ◆ **Test Cases:** It is a document that contains the steps that has to be executed, it has been planned earlier.
- ◆ **Test Script:** It is written in a programming language and it's a short program used to test part of functionality of the software system. In other words a written set of steps that should be performed manually.
- ◆ **Test Strategy:** Test strategy is set of guidelines which describes how testing needs to be done. It is used at the organizational level. It is normally developed from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as base.

71) What is Test Plan? How do you develop Test Plan?

- ◆ Test plan is the overall strategic (plan) or detailed document that describes entire testing Process to meet the objectives of the testing requirements.
- ◆ What is the deliverable and scope?
- ◆ Test Plan is developed based on the
  - User Requirement Specification (URS),
  - Business Requirement Specification (BRS),
  - System Requirement Specifications (SRS).

72) 1) Test Strategy for all the project in a company

- ◆ It does not change for each project delivered by the company
- ◆ It's a set of guideline for testing
- ◆ It is created by project manager
- ◆ It's a general approach
- ◆ It's a long term plan

2) Test Plan for each project

- ◆ It will include deliverable for the specific project
- ◆ Date for testing
- ◆ Possible Issue or blocker
- ◆ Scope or percentage of application to be tested
- ◆ Created based on the requirement/ specification

73) What includes preparing a standard test plan?

- ◆ A good Test Plan includes Introduction of testing, Scope, Purpose and Objectives, Reference Documents, Acronyms, Type of Testing to be conducted, Role & responsibilities, Test Environment, Requirements, Scenarios, Requirement Traceability Matrix, Pass & Fail Criteria, Entrance & Exit Criteria, Schedule, Resources, and Assumption & Risk Analysis.

74) How to write User Acceptance Test plan & test cases?

- ◆ The way of writing Test Plan and Test Cases is the same in all the test phases. However, specifically for User Acceptance Testing, the testers use data nearly real data (meaning that the data is very much like the production data or real data).

75) What are the types of test cases that you write?

- ◆ We write test cases for smoke testing, integration testing, functional testing, regression testing, load testing, stress testing, system testing and so on.

76) What is RTM?

- ◆ The purpose of the Requirement Traceability Matrix (RTM) is used to ensure that each requirement is linked with its associated test case, so that testing can be done as per the requirements.
- ◆ I can identify any gap using RTM where the test missing.
- ◆ RTM is developed by business analyst and maintained by SDLC team.

77) Why RTM is useful?

- ◆ The main goals for this matrix are:
  - i) Make sure software is developed as per the mentioned requirements.
  - ii) Helps in finding the root cause of any bug.

78) How many requirements you can have in a test case?

- ◆ It depends on the test cases.
- ◆ For Example, one test case can be broad where we can have multiple requirements.
- ◆ Also, multiple test cases can have one or more requirements.

79) How can it be known when to stop testing?

- ◆ This can be difficult to determine. Many modern software applications are so complex and run in such an interdependent environment that complete testing can never be done.
- ◆ Common factors in deciding when to stop are.
  - i) Deadlines (like release deadlines or testing deadlines).



- ii) Test cases completed with certain percentage passed.
- iii) Test budget has been depleted.
- iv) Coverage of code, functionality, or requirements reaches a specified point.
- v) Bug rate falls below a certain level; or
- vi) Alpha or Beta testing period ends.

**80) How can it be determined if a test environment is appropriate?**

- ◆ A Procedure Check out or Dry Run is required to execute for determining whether the test Environment is appropriate for testing an application.

**81) Have you done any Back End Testing and/or if you did, how did you do it in your last project?**

- ◆ Yes, I did. Database testing is done to ensure that the data integrity, access, and retrieval. It is done to ensure that
  - i) The data input in the UI is saved correctly in the database
  - ii) The data output is correct
  - iii) Data structure and schema is correct
  - iv) Provide an example from your project

**82) Can you create a few negative and positive scenario for login test cases? (Boundary Test)**

- ◆ Valid Username & valid Password
- ◆ Valid username & Invalid password
- ◆ Invalid username & valid password
- ◆ 3 Boundary Tests – Password should be 8 digits
  - i) Test by 7 digits
  - ii) Test by 8 digits
  - iii) Test by 9 digits
- ◆ Verify the logo of the login page/page title
- ◆ Testing the Login Button
- ◆ Test the customer service button
- ◆ Forget password button

◆ Phone number two and four digit I need is it didn't say they only said that suppose the password is 8 digit right so you are protected at 7 you have to test it at 8 and we have to test it at night right one before one at the boundary one after the boundaries so this is called a boundary test now do you already have three test cases right since three different scenario you need to think of three different scenarios and not just say that oh I put in 10 different username 10 different wrong username no you have to give a specific scenario I'm testing the specific thing right so I'm doing a boundary test for the password right then another one you can say the password they said had to be one capital at least one capital right so you do it all lowercase and one with all uppercase right and then one with one lowercase N you know one uppercase right so at least the one with all uppercase supposed to sell all workers suppose well the one that has uppercase and lowercase that was supposed to pass so now you have the boundary test three of them and then three of them for the uppercase and lowercase right and then another scenario could be like this passwords had to have a symbol but this is like common factors right now right you know anytime you want to create an account they said that you have to have it the uppercase and lowercase 8 digit and have a symbol this is like a very common thing right but those things you need to test OK so you can create you know also you can you can test out that logo you see like every login page will have a logo right so company logo so we can think of like OK I have to test that company logo and number another thing you can say I have to test out the title I have to make sure the title is right I hope so you know the if the title is wrong then that's that that fail right so that's a positive test OK so think of it whenever you put in you know a negative test.

**83) What are the four environments in testing?**

- ◆ Dev environment -
- ◆ Test/QA environment -
- ◆ Staging environment –UAT testing
- ◆ Production Environment – NO Automation/Manual testing should be done in production unless manager ask you to do in manual testing.

**84) Two production issues came up and how will you write the scenario for this issue?**

- ◆ I will contact my Business Analyst and try to find out the step to reproduce the issue.
- ◆ I will discuss with QA team member and developer.
- ◆ And then I will open a Ticket/bug report in JIRA with description, severity, priority etc
  - i) Assign to dev
  - ii) Dev will discuss the issue.

85) Let's say you find a bug and you assigned to dev and he rejected what will you do?

- ◆ First, I will document the bug when I assigned to the developer. If the dev rejected the bug.
- ◆ Then I will have a conversation with developer for the reason why it was rejected.
- ◆ Then I will go to Business analyst and if they agree that this is not a bug then I will close the bug.
- ◆ If the BA doesn't agree that this is a bug, then I will escalate to the project manager.

86) What is epic?

- ◆ An epic is a large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories. An epic can't be more than 10 points.

87) What did you do with JIRA?

- ◆ JIRA is an issue tracking tool and in my last project I use JIRA to manage all user stories, tasks and defects.
- ◆ As a QA, I access user stories in JIRA and create subtasks to complete the user story. First, I create subtasks such as create testcases, automate test cases, and execute test cases.
- ◆ While test development and test execution in progress I change JIRA status. Also, I write test cases in JIRA using the plugins such as JBehave.
- ◆ I also opened bug using JIRA and communicated to development team. While opening bug, I add a summary, steps to reproduce, attach a screenshot, severity, priority and assign the bug to developer.
- ◆ Every year I have 200 – 250 stories and my last 5 years' experience I did around 1000 – 1250 stories in JIRA.

88) **Velocity chart** - Track the amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

- ◆ How many story point we can deliver in a sprint
- ◆ Dependent of the complexity not on hours
- ◆ Divide total number of story point by the number of sprint
- ◆ It is a rate of delivery
- ◆ This chart is average for all the sprint

**Burndown chart** - Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

- ◆ Daily activity of work that is done
- ◆ Work left to do vs time
- ◆ Completed work in a sprint and if any work remaining and actual line of estimated work
- ◆ It will have a Ideal, estimated
- ◆ This chart is for each sprint

89) What is a Business Requirements Document (BRD)?

- ◆ BRD provides a complete business solution for a project, including the documentation of customer needs and expectations.
- ◆ BRD fulfills the following objectives.
- ◆ Gain agreement with stakeholders.
- ◆ Provide clarity on the business requirements.

- ◆ Describe the solution that meets the customer/business needs.
- ◆ Determine the input for the next phase of the project.

90) How many test cases did you do in your last 5 years?

- ◆ Yearly I did 250 – 300 test cases and in 5 years I did 1200 test cases.
- ◆ Explanation:
  - 48 weeks out of 52 weeks
  - 2 months are buffer zone
  - 40 weeks left
  - 2 weeks sprint – 9 days
  - 20 sprints
  - Each sprint we can do 18 points
  - Every day we can do 2-point stories.

91) How many team members you have in your team?

- ◆ Are you talking about my QA team or Agile team? In my agile team I have 12 members in my team:
  - 1 Business Analyst (BA),
  - 1 Project Manager,
  - 1 Database Analyst (DBA),
  - 1 Manual testers,
  - (2) Automation Engineer,
  - 6 Developers.

92) If you have to automate scripts for 2 or more websites that have similar functionality what would your automation approach be?

- ◆ I have a properties file where I keep URL and site-related data.
- ◆ I can create a method in utility manager so that when the project is run with maven configuration, depending on the dEnv, the project will run for two or more websites in the maven configuration; you need to set up a goal as "clean test -Denv=envName" and you need to set the base directory to the project in question

## JAVA

93) What are the 3 Java components?

- ◆ JDK
- ◆ JRE
- ◆ JVM

94) Difference between JDK, JRE and JVM?

Here are the major differences between JDK vs. JRE vs. JVM:

JDK	JRE	JVM
The full form of JDK is Java Development Kit.	The full form of JRE is Java Runtime Environment.	The full form of JVM is Java Virtual Machine.

JDK is a software development kit to develop applications in Java.	It is a software bundle which provides Java class libraries with necessary components to run Java code.	JVM executes Java byte code and provides an environment for executing it.
JDK is platform dependent.	JRE is also platform dependent.	JVM is platform-independent.
It contains tools for developing, debugging, and monitoring java code.	It contains class libraries and other supporting files that JVM requires to execute the program.	Software development tools are not included in JVM.
It is the superset of JRE	It is the subset of JDK.	JVM is a subset of JRE.
The JDK enables developers to create Java programs that can be executed and run by the JRE and JVM.	The JRE is the part of Java that creates the JVM.	It is the Java platform component that executes source code.
JDK comes with the installer.	JRE only contain environment to execute source code.	JVM bundled in both software JDK and JRE.

#### 95) What is JAVA?

- ◆ Java is a cross-platform object-oriented programming language.

#### 96) Why we use JAVA?

- ◆ Java works on different platforms like Windows, Mac, Linux, iOS and Android.
- ◆ It is easy to learn and simple to use.
- ◆ It is open-source and free.
- ◆ It is secure, fast and powerful.
- ◆ JAVA can be used to create complete applications that can run on a single computer.

#### 97) What is the current JAVA version?

- ◆ The current JAVA version is JAVA 15.

#### 98) Which version of JAVA you use in your current project?

- ◆ Java 8.

#### 99) JAVA Project Structure?

Project – Topmost

Packages – Collections of Classes

Classes – Blueprint of an object

#### 100) Class Structure in JAVA project?

Class

Variable

## Method

### All Selenium Code

### All JAVA Code

#### 101) What is object?

- ◆ **Object is an instance of a class**
- ◆ Object allocates memory when it is created in JAVA
- ◆ Many ways to create object in JAVA such as
  - By New Keyword
  - By Newinstance() method
  - By Clone () method
  - By Factory method
  - By Deserialization

#### 102) How many ways we can use object?

- ◆ By variable
- ◆ By method
- ◆ By constructor

#### 103) What is class?

- ◆ **Class is a blueprint of an object**
- ◆ Class is a group of similar objects
- ◆ Class is declared once
- ◆ Class doesn't allocate memory when it is created in JAVA
- ◆ There is only one way to define class in JAVA using CLASS KEYWORD

#### 104) What is Java Class Loaders?

- ◆ Class loaders are used to find different types of classes in a Java Project, such as system class loader, extension class loader, and the bootstrap class loader. A class loader is an object that is responsible for loading classes. If given a binary name for the class, it will locate the class and keep that class in the memory to be used next.

#### 105) What is class body?

- ◆ The class body contains all the code that provides for the life cycle of the object created from class.

#### 106) What a class can contain in JAVA?

- ◆ Variable, Methods, Constructors, nested class and interface.

#### 107) What is variable?

- ◆ The real data is variable.
- ◆ Variable is a container to hold the data.
- ◆ Variables are typically used to store information which your Java program needs to do its job.
- ◆ Each variable has name and data type.

#### 108) How many types of variable in JAVA?

- ◆ There are 3 types of variable in JAVA which are:
  - Local Variable – a variable that is declared inside the method is called local variable. Static and non-static methods. It only method use it.

- ii) **Class Variable** – a variable that is declared as static is called static/class variable. It can access the static variable. Its interface uses it.
- iii) **Instance variable** – a variable that is declared inside the class but outside the method called instance variable. We can use all the methods. Its constructor uses it.

#### 109) How do we use STATIC word in JAVA?

- ◆ **Static means class level.**
- ◆ **We do not need to create object to call it.**
- ◆ **We can call it directly without object by using**
  - i) **Static variable**
  - ii) **Static method**
  - iii) **Static interface**

#### 110) Difference between path and class-path?

- ◆ **Path** – Path defines the location where system can look up the executable files.
- ◆ **Class-path** – Class-path is the location of the Java class files.

#### 111) How many types of data in JAVA?

- ◆ **In Java, there are two types of data which are:**
  - i) **Primitive –**
    - (1) we do not need to create an object.
    - (2) Char, Boolean, byte, short, int, long, float and double
  - ii) **Non-primitive –**
    - (1) we need to create an object.
    - (2) We can use the method from the class.
    - (3) String, class, interface & Arrays

#### 112) Default values for primitive and non-primitive data types?

Primitive Data Type in Java

Name	Default Value	Size	Type
byte	0	1 byte	Integral Value
short	0	2 byte	Integral Value
int	0	4 byte	Integral Value
long	0	8 byte	Integral Value
float	0.0f	4 byte	Floating Point
double	0.0d	8 byte	Floating Point
char	'\u0000' (means 0 in ASCII)	2 byte	Character
boolean	false	1 bit	Boolean

#### 113) What is a method in JAVA?

- ◆ **A method is a behavior of an object.**
- ◆ **Method is not the same name as class name.**
- ◆ **Method use the void word.**

#### 114) What is method body?

- ◆ **The method body is where all the action of a method takes place.**

#### 115) How many types of method in JAVA?

- ◆ **Two types:**
  - i) **Normal method** – method name and body.
  - ii) **Abstract method** – a method without body is known as abstract method. A class can extend only one abstract class.
  - iii) **A class contain an abstract method than the class convert to abstract class.**

#### 116) What is modifiers in JAVA?