**SDLC & Agile**

**TESTING TYPES**

In SDLC, Software testing is a most often used technique for verifying and validating the quality of software. Black-box testing is often used for validation (i.e. are we building the right software?) and white box testing is often used for verification (i.e. are we building the software right?).

* **The black box** is also called Functional testing, a technique that designs test cases based on the information from the specification. Black box testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester.
* **White box testing** is also called structural testing design test cases based on the information derived from source code. White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.
* **Unit testing** is code-based testing which is performed by developers, this testing is mainly done to test each and individual units separately. This unit testing can be done for small units of code or generally no larger than a class.
* **Integration testing** is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing
* **System testing** reveals that the system works end-to-end in a production-like location to provide the business functions specified in the high-level design. It may include usability, regression, and functional if it applies.
* **User Acceptance testing** is conducted by business owners, the purpose of acceptance testing is to test whether the system does in fact, meet their business requirements.
* **Regression Testing** is the testing of the software after changes have been made; this testing is done to make sure that the reliability of each software release, testing after changes have been made to ensure that changes did not introduce any new errors into the system. It is usually scheduled before every Sprint, and often it is performed through Regression Suit – a Set of files in order, hierarchically performed, regression task cases. Therefore regression is automatized to test all and each test case has to be run.
* **Functional Testing** is done for a finished application; this testing is to verify that it provides all of the behaviors required of it. We can say that a functional test verifies specific action or function of the code, in other words, it tests functionalities of the application: Log-in, Log-out, Browse, Search, Check out.
* **Performance testing** is the process of determining the speed, responsiveness, and stability of a software program under a workload. Performance testing can involve quantitative tests done in a lab, or occur in the production environment in limited scenarios.
* **Smoke testing**, or Sanity testing, is an initial testing process to check whether the software under test is ready to be tested, so it is one of the best ways to know the application.
* **Positive and negative testing**, Positive testing is a testing process where the system validated against the valid input data. In this testing tester always check for an only valid set of values and check if an application behaves as expected with its expected inputs. Negative testing is a testing process where the system validated against the invalid input data. A negative test checks if an application behaves as expected with its negative inputs.
* **Ad-Hoc Testing** is done without any test plan or documentation. So, one doesn’t need to wait for any cycle execution or phase. Ad-hoc testing helps in deciding the scope and duration of the various other testing and it also helps testers in learning the application prior to starting with any other testing. This methodology can be very effective in finding missing cases, determining priorities of other testing activities, and increasing code coverage.

**Java**

* **What is an Exception?**

Like an error that happens during runtime/execution.

* **How many types of Exceptions?**
* Checked exceptions

Ex: Thread.sleep(ms); → Java will not compile if you do not handle these exceptions

* Unchecked/Runtime exceptions

Ex: ArithmeticException (int i=100/0;), NoSuchElementException

These exceptions happen during runtime and it is up to a programmer to handle them. Java will not force you to handle them.

* **What do you use for Exception handling in Java?**

try...catch block.

* **What is finally block?**

Finally block is used in Java exception handling. We can use finally block with the only try, or with try-catch. Code in finally block will always execute.

* **When does code in finally block not run?**
* When System.exit(0); st runs, it will make java program to stop and finally will not run.
* JVM Error → OutOfMemory error(when heap memory is full).
* StackOverFlow error(when stack memory is full).

**Selenium**

1. **What is Selenium?**

Selenium is a suite of tools for automated web testing. It is composed of Selenium WebDriver. Selenium WebDriver is used to automate web applications using a browser's native methods.

1. **How many languages does selenium support?**

8: Java, C#, Groovy, Perl, PHP, Python, Ruby and Scala.

1. **What are some advantages of selenium?**

* Selenium is open source and free to use without any licensing cost.
* It supports multiple languages like Java, ruby, python etc.
* It supports multi-browser testing.
* It has a good amount of resources and helping community over the internet.

1. **What are the disadvantages of selenium?**

* Selenium supports only web-based applications
* For any reporting related capabilities have to depend on third-party tools
* No vendor support for tool compared to commercial tools like HP UFT
* We cannot test desktop applications using selenium.
* We have to rely on external libraries and tools for performing tasks like - logging (log4J), testing framework(TestNG, JUnit), reading from external files (POI for excels) etc.

1. **List some scenarios which we cannot automate using Selenium WebDriver?**

* Automating Captcha is not possible using Selenium WebDriver
* We cannot read bar code using Selenium WebDriver

1. **What are the prerequisites to run Selenium Webdriver?**

JDK, Eclipse, WebDriver (selenium standalone jar file), browser, application to be tested.

1. **What do you use to inspect the HTML code?**

* The Chrome Developer Tools (Dev Tools for short) -> Right-click on any page element and select Inspect Element.
* Page Inspector - Firefox Developer Tools -> Right-click on any page element and select Inspect Element.
* Developer Tools (Internet Explorer) -> press F12

# Real Interview Questions

1. Tell me about yourself?
2. Tell me about your testing experience?
3. Are you still working? Why did you switch your career to Test Automation?
4. Tell me about your current framework.
5. Where did you start to automate? How did you start?
6. Can you walk me through over an example how do you automate your one user story?
7. Tell me about your team?
8. Tell me about your project?
9. How would you rate yourself you as a Java?
10. There is a deadline and you have a lot of work to do and it looks impossible to finish on time? How do you handle?