JUnit

# What is JUnit?

* JUnit is a unit testing open-source framework for the Java programming language. Java Developers use this framework to write and execute **automated tests**. In Java, there are test cases that have to be re-executed every time a new code is added.
* JUnit plays a huge role when it comes to regression testing. **Regression Testing** is a type of software testing that checks if the recent changes made to the code do not adversely affect the previously written code.

## What is Unit Testing?

* A **unit** indicates the smallest bit of code that can be fetched out of the system. This small bit can be a line of the code, a method, or a class. The smaller the chunk of code, the better it is, as smaller chunks will tend to run faster.

## What is the need for JUnit Testing?

* Find bugs early in the development phase, which increases the code’s reliability
* Enables the developer to invest more time in reading the code than writing it
* Makes the code more readable, reliable, and bug-free

## Features of JUnit

* **Open Source Network:** Enables developers to write codes fast and with better quality.
* **Provides Annotations:** Provides several annotations to identify test methods.
* **Provides Assertions:** Provides assertions to test expected results.
* **Provides Test Runners:** Has test runners to run tests.
* **Improves Code Quality:** Allows faster code writing, which results in an increase in the code’s quality.
* **Automated Test Running:** All the tests run automatically on JUnit, the results obtained again automatically checked, and it provides feedback.
* **Easily Interpretable Results:** Test results are represented interactively by showing test progress in a bar, thus making them easily interpretable.

## JUnit Annotations

* **JUnit Annotations** refer to the syntactic meta-data added to the Java source code for better structure and readability. Here, syntactic meta-data refers to the type of data representing the structure of a file with references to bytes, data types, and data structures.

## A screenshot of a test Description automatically generated

## What is Automation Testing?

* **Automation testing** is a procedure wherein automated tools are used to write test cases and run them, including testing characteristics such as loading, stress, and performance. The automated process is designed to provide higher efficiency, effectiveness, and accuracy.

## Tools Use

* **HP Quick Test Professional:** Enables testers to execute automated tests to identify, defects or gaps, as opposed to the expected results of the application test.
* **IBM Rational Functional Tester:** Automates functional, regression, GUI, and data-driven testing.
* **LoadRunner:** Simulates a large number of users to measure system performance and behavior under load.
* **Selenium:** An open-source tool used for regression testing, offering playback and recording facilities.
* **SilkTest:** Used for functional and regression testing, it’s the leading functional testing tool for e-business applications.
* **TestComplete:** Lets testers create automated tests for Android, iOS, Microsoft Windows, and Web applications.
* **Testing Anywhere:** Tests applications, controls, GUI front-ends, objects, and websites.
* **Visual Studio Test Professional:** An integrated toolset designed for teams as well as individual developers.
* **WATIR:** Open-source software that lets you write easily read and maintained tests.
* **WinRunner:** An automated functional GUI testing tool that permits users to record and playback user interface (UI) interactions as test scripts.

## Why Use Automation Testing?

* **Automation testing** is an ideal tool for running many tests simultaneously and handling a lot of the work that human testers would consider boring and repetitive. Automation testing also comes in handy for testing multi-lingual sites.

## Advantages of Automation Testing

* **Load Testing:** Automation testing can run thousands of simultaneous tests, acting like millions of users, something that manual testing can’t replicate. By using realistic load tests, developers can put their software through the paces and get a good idea of how it will handle real-world situations.
* **Reliability and Accuracy:** Automated testing can perform the same tests over and over, with no deviations, thereby guaranteeing a consistently accurate performance.
* **Timesaver:** Automation testing brings “set it and forget it” functionality to the testing process. Activate the tests, leave, and see the results when you return.
* **A Fundamental Part of DevOps:** The DevOps design philosophy has been gaining considerable traction in the world of app development, and automation testing is a necessary element.
* **Reusable:** Once it’s in place, developers have a set of tools that can be used (and re-used) to test different versions of apps and software in the future.
* **Flexible Programming:** Automation testing can change the automation testing’s programming, creating increasingly complex tests that can spot obscure weaknesses in the app.

## Disadvantages of Automation Testing

* **Setup Time:** Automation testing requires a lot of beforehand preparation. In the long run, it’s worth it, but for the short-term, it’s a lot of work and can be time-consuming.
* **Particular Skills:** Need skilled people to conduct automation testing. Automation test script-writing requires a proficient programmer.
* **Debugging:** Developers need to make sure that the script is entirely bug-free.
* **Maintenance Issues:** The more screens that a test script contains, the more difficult it is to maintain test files.

## Manual vs. Automation Testing

* **Time Factor:** Automated testing takes less processing time than manual testing.
* Random Testing: Also called “exploratory testing,” it’s allowed in manual testing, not in automation.
* **Start-up Cost:** Automated testing demands a higher investment, but the return on the investment (ROI) is higher in the long run. Manual testing is cheaper, but the value isn’t as high.
* **UI Changes:** Changes to the User Interface (UI) require modifications to an automation testing’s scripts, whereas manual testing works just fine even with small UI changes in effect.
* **Engagement:** The accuracy doesn’t suffer due to monotony and boredom on the part of the test. Manual testing’s repetition risks boring the tester, thereby increasing the likelihood of errors.
* **Frameworks:** Automation testing uses Selenium and other test automation frameworks such as Datadrive, Hybrid, and Keyword. Manual testing doesn’t use frameworks.
* **The Human Factor:** Automation testing has no way of ascertaining user-friendliness or whether or not customers are satisfied. Manual testing includes the human observation element, so testers get a better idea of how user-friendly the system is.
* **Programming Skills/Knowledge:** Automation testing requires programming knowledge, whereas manual testing doesn’t.
* **Best Time to Use:** Automation testing works best when there are a large number of repeatable functional tests, as well for situations calling for load testing, performance testing, and regression testing. Manual testing is better suited for ad hoc, exploratory, and usability testing.

# What is Regression Testing?

* **Regression testing** is a type of software testing conducted after a code update to ensure that the update introduced no new bugs. This is because new code may bring in new logic that conflicts with the existing code, leading to defects. Usually, QA teams have a series of regression test cases for important features that they will re-execute each time these code changes occur to save time and maximize test efficiency.