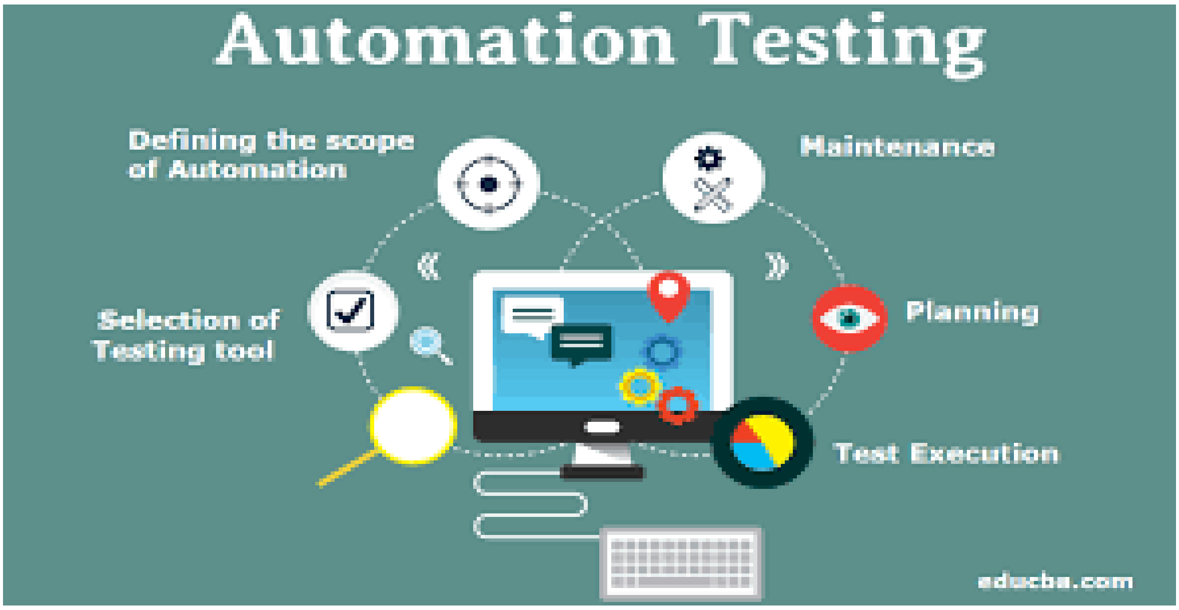
AUTOMATED SOFTWARE TESTING:-

The scope of Automation testing is especially useful in running repetitive tasks that take up valuable time of QAs. For the uninitiated, Automation testing undertakes a lot of different testing types where you can simply run the test cases over and over again with several inputs.



Automated software testing is the method of automatically reviewing and validating software products, such as web and mobile applications. This process ensures that they meet all predefined quality standards for code style, functionality, and user experience. Test automation replaces manual human activity with systems. Even though tests, like regression or functional testing, can be done manually, automating the process will reduce the time taken to perform the tests. Moreover, it takes less time to perform exploratory tests and more time to maintain test scripts, thus, increasing the overall test coverage.

Automated software testing is widely used in application development to help accelerate the launch of apps/software or new features for existing software tools. It is mainly used to assist the testing process done for software tools/apps developed by organizations, including functional and load/stress testing and provides consistent results and data points. Testing experts use test automation tools to create scripts and test cases, and these scripts/test cases can be reused. Therefore, the primary features of test automation can help organizations eliminate human errors, reduce the total time taken for software development, and automate repetitive tasks.

PURPOSE OF AUTOMATION TESTING:

Automation testing serves several important purposes in the software development lifecycle. Let's explore some key reasons why organizations embrace automation testing:

1. Increased Test Coverage: Automation testing enables a broader scope of test coverage. Organizations can leverage a test automation platform and use it to design test scripts to cover various scenarios and test cases, ensuring thorough validation of software functionality. With automated tests, organizations can achieve higher levels of test coverage, resulting in improved software reliability.

2. Consistency and Reusability: Automation testing ensures consistent test execution by removing the element of human error. Using test automation platform to automate testing, you can reuse test scripts across multiple test cycles and different software versions. This reusability not only saves time but also promotes consistency in testing, enabling accurate comparison of results over time.

3. Early Detection of Defects: Automation testing enables early detection issues. By running automated tests at different stages, such as during integration or regression testing, potential bugs can be identified and addressed promptly. Early defect detection helps in reducing the costs.

WHAT KINDS OF TESTS SHOULD BE AUTOMATED?

While testing an application/software, testers cannot automate all processes involved in the testing cycle. Some tests need human supervision and involvement to get better results. Using test automation platforms to automate testing is not an alternative to manual testing but helps and supports the entire testing team by reducing the workload.

In order to determine whether a test is suitable for automation, testers can check if it fits the following criteria:

- The tests should be highly repetitive and take a long period of time to perform if it is done manually
- The testing path must be predictable, as it has been verified earlier through manual testing

- The tests that involve the testing of frequently used features that introduce high-risk conditions
- The tests that require multiple datasets and run on several different hardware or software platforms and configurations
- Tests that are not possible for human manual testing, e.g., thousands of concurrent users trying to log in at the same time

If a test meets all these criteria mentioned above, you can consider leveraging test automation platforms for automation.

AUTOMATED TESTING OF WEB APPLICATIONS:-

Automated web application testing is a process in which testing is performed using a set of automated testing tools. Automating web applications saves time and money as the same set of automated test cases can be reused multiple times in various agile sprints.

Web testing, or web application testing, is a software practice that ensures quality by testing that the functionality of a given web application is working as intended or as per the requirements.

Web application testing allows you to find bugs at any given time, prior to a release, or on a day-to-day basis.

Testing is an essential part of software development.

HOW TO TEST WEB APPLICATIONS?

Given the complexity and variety of modern web applications, a multifaceted testing approach is essential. This section outlines the key web test case scenarios, including functionality, usability, regression, end-to-end, cross-browser, and performance testing.

FUNCTIONAL TESTING

A single end-user can make the whole system crash in minutes, even after unit, integration, and performance tests have passed. This usually happens because the user does something the developers did not expect. The purpose of functional testing is therefore to ensure that the functionality of the software works as intended for an end-user. It tests this through the UI of the application. As opposed to non-functional testing, it doesn't deal with the underlying source code, and doesn't evaluate things like speed, scalability, and reliability. There are many types of functional tests. Examples of functional tests in a web application UI include testing that:

1. The login to your web application is successful across browsers and devices
2. The web application is interacting as intended with external databases and syncing successfully
3. Invoices are being sent and received with the correct information and securely
4. Buttons, text fields, menus, etc., are working as per the requirements

USABILITY TESTING

Usability testing focuses on design aspects rather than functional aspects, assessing the user experience, and how intuitive and user-friendly the web application is.

Key aspects include:

- User Interface Evaluation: Analyzing the layout, design, and navigability.
- User Experience Testing: Assessing the ease of learning and using the application.
- Accessibility Testing: Ensuring the application is accessible to all users, including those with disabilities.

REGRESSION TESTING

Regression testing is critical whenever updates or changes are made to the application. It ensures that new code doesn't negatively impact existing functionality, and typically includes functional tests as part of its scope.

This type of testing:

- Verifies Existing Functionality: Ensures that previous functions still operate as intended after modifications.
- Identifies Unintended Consequences: Catches any new bugs introduced by recent changes.

END-TO-END TESTING

End-to-end testing examines the complete functionality of the web application from start to finish, emulating real user scenarios. It aims to ensure that all components of the application work together seamlessly. End-to-end testing is typically applied to a business' most important or common business processes, to reduce the risk that these key flows break.

This involves:

- Workflow Testing: Ensuring all the integrated parts of the application interact correctly.
- Data Integrity Testing: Confirming that data maintains its integrity throughout all transactions.

CROSS-BROWSER TESTING

With the variety of browsers available, browser-based testing ensures that the web application performs consistently across different browsers and their versions.

This testing type:

- Ensures Compatibility: Verifies that the application functions correctly on various browsers.
- Identifies Browser-Specific Issues: Highlights any layout or functional issues unique to certain browsers.

PERFORMANCE TESTING

Performance testing evaluates the web application's stability and responsiveness under various conditions.

This includes:

- **Load Testing**: Assessing the application's ability to handle high volumes of users.
- **Stress Testing**: Determining the application's breaking point and how it recovers from failure.
- **Speed Testing**: Measuring response times and the speed of page loading under normal conditions.

WEB APP TEST AUTOMATION BEST PRACTICES

Before you start automating your web application tests, make sure you draft a test automation strategy to keep you on track. Things to keep in mind before you start automation are:

1. What are the specific requirements of your web application?
2. What types of tests do you need to automate?
3. Which test automation tool best suits your requirements and goals, as well as the resources on your team?
4. How much maintenance will automation require?

As a first rule of thumb, start small, and once you're comfortable, start scaling your automation efforts. No one wants to end up with hundreds of automated test cases that are impossible to maintain. Rather, think of automation like a bell curve - automate too little, and the ROI on your potential tool costs and onboarding will be too high. Automate too much, and the time you spend on changing or maintaining your tests starts to exceed the time saved. Ideally, find the sweet spot in the middle where return is the highest.

Successful web application testing requires effective test automation processes, clear communication within the team, an efficient strategy, and an automation tool that doesn't impair testers, but enables them.