



What Are the Elements of Software Applications to Automate for Enterprises?



Software test automation for enterprises can be a challenge given the need to adhere to heterogeneous architectures, quality processes, and compliance requirements. This begs the question: is it possible to achieve the level of test automation that complex systems and modern delivery schedules demand for big enterprises? The answer is “yes”, provided enterprise intelligent automation is implemented for end-to-end business-critical and complex features in phases. So, what is enterprise test automation all about and why is it important for business enterprises? Let us find out in the below-mentioned segment.

What is enterprise test automation?

It refers to the testing of processes or value chains of business enterprises to validate their smooth functioning in an omnichannel environment. Since every enterprise app or website is mission-critical and helps stakeholders derive insights and make critical decisions, they should perform at their optimal level and be devoid of any vulnerabilities or bugs. The apps could be BI tools, ERP solutions, and payment processing solutions, among others. Given the scale and complexity of enterprise applications and websites and their need for achieving cross-

device and cross-platform compatibility, manual testing should be given a go. It is only through enterprise process automation using the right frameworks and tools that the performance of these software solutions can be evaluated for end-to-end business processes and their ability to meet business goals.

What should be automated in software QA automation?

Any automation testing services for enterprises should include the following elements or aspects:

- Website or app that is business-critical for the enterprise and needs continuous monitoring.
- Web or app features that are too complex for manual testing and are prone to human error.
- Time-consuming features that need to be tested repeatedly to ensure optimal software performance.
- Unit tests to check whether the class or function can operate in isolation.
- Integration of APIs to check if various components of the app or website and any connected third-party software work seamlessly.
- Performance tests to evaluate the software application's scalability, accessibility, stability, and performance when subjected to different thresholds of traffic load.
- End-to-end testing of a user's journey across software components, both at the API or UI level.

However, it is not easy for any automation testing company to implement automation for software applications that form part of any enterprise's software suite. Let us discuss the strategies that [test automation services](#) can implement to remove the barriers to automation.

Apply automation across the technology stack: Any test automation process is preceded by the development of a framework. It is only when the framework is ready for implementation after testing that test scripts are used. With changes in the software application to be automated, the test scripts and framework need to be updated and tested on a continuous basis. This requires suitable resources to be tapped into so that they can take a specific approach to script-based automation. Any QA tester finds it time-consuming to build, scale, and maintain the scripts and framework.

This type of challenge gets amplified several times for an enterprise. For instance, to test a heterogeneous application stack comprising packaged applications such as Salesforce, SAP, Oracle EBS, databases, mainframes, mobile and web front ends, and others, multiple frameworks need to be built. Further, more tools and frameworks need to be configured, learned, and integrated. So, ensure every technology, tool, and approach is connected to provide centralized visibility and foster collaboration.

Address the test maintenance issues: Test scripts and frameworks should be maintained for the success of any test automation activity. Or else, the test results can be riddled with false positives or negatives, thereby losing their credibility. So, test stability should be considered from the beginning of the test automation initiative. Check how the tool responds to expected variations and how to keep it in sync with the application's changing requirements. Make sure the tests are run with appropriate data and within a completely stable test environment. Also, test suites should be built from modules that are easily updatable and used all across the board.

Conclusion

Enterprise test automation is the pressing need of the hour, given the quality and security dimensions of applications taking center stage. Any [software test automation](#) service should plan the processes or applications to be tested by streamlining the bottlenecks and achieving, in the process, a well-performing enterprise ecosystem catering to the needs of all stakeholders.

Resource

James Daniel is a software Tech enthusiastic & works at Cigniti Technologies. I'm having a great understanding of today's software testing quality that yields strong results and always happy to create valuable content & share thoughts.

Article Source: dev.to