Abstract

System testing is an integral, costly, and time-consuming activity in the software development life cycle. As is true for software development in general, reuse of common artifacts can provide a significant gain in productivity. In addition, because testing involves running the system being tested under a variety of configurations and circumstances, automation of execution-related activities offers another potential source of savings in the testing process.

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

System testing is actually done to the entire system against the Functional Requirement Specification(s) (FRS) and/or the System Requirement Specification (SRS). Moreover, the system testing is an investigatory testing phase, where the focus is to have almost a destructive attitude and test not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

The following examples are different types of testing that should be considered during System Testing:

- Functional testing
- Usability testing
- Performance testing (Load, Volume, Stress)
- Compatibility testing
- Security testing
- Smoke testing
- Exploratory/Adhoc testing
- Regression testing
- Reliability testing
- Recovery testing
- Installation testing
- Accessibility

The challenge for developers, QA teams, and management alike is how to speed up their testing processes and increase accuracy and completeness—without breaking their already tight budgets. In the age of accelerating product lifecycles and pressures on reducing the cost, the impact of traditional approach to testing has had a serious impact on IT organizations. The business climate of today is such that IT organizations are asked to do more with fewer resources and without any significant reduction in the quality of the product that is being delivered. When IT organizations make attempts to cut on the cost, Software Testing is often the first item that would be cut.

By implementing automated testing, companies can dramatically increase both the speed and accuracy of their testing processes, providing a higher return on investment (ROI) from software projects while dramatically cutting risk.

RelQ follows FURPSSI (Functionality, Usability, Reliability, Performance, Security, Scalability and Installation & Compatibility) model for System Testing. There is no question that rigorous functional testing is critical to successful application development. By automating key elements of functional testing, companies can meet aggressive release schedules, test more thoroughly and reliably, verify that business processes function correctly, and ultimately generate higher revenue and customer satisfaction from their online operations.
A strategic approach in developing a test automation framework using tools and methodologies will improve the test coverage in regression cycles and reduce the test effort in subsequent release cycles.

RelQ has over 2000 person-years of experience in Software Testing with 400 person-years of extensive experience in Test Automation using various tools and methodologies, on varied domains & technologies. The presentation by Sudheer Mamadapur aims at highlighting the Importance of Test Automation using tools, frameworks and methodologies in System (Functional) Testing for effective software testing and improved testing productivity.