How to get test automation right

It is practically impossible to operate a rapid software release cycle without automated testing. Manual testing is too slow to keep up with the brisk pace of agile development teams. Later on, during system testing and validating integrations with other systems, both the time spent and the overall workload become intolerable.

Test automation is notoriously expensive to perform and even more expensive to maintain, even with the most advanced tools. Therefore, automation should first be applied to tests that are frequently repeated and infrequently changed.

Test automation, if done well, may save quite a bit of human effort. Bigger gains, however, come from time saved. In a best-case scenario, test automation can accelerate your testing by as much as 90% simply by eliminating waiting times from the process.

Let's consider a simple example. A software team is releasing a new system version for testing. It is Thursday. The testers are still busy verifying some corrections from the previous test release, but they will be ready to start by Monday. You've lost more than three days. The testers will complete the test round in three days, i.e. next Wednesday. That is when the development team will be able to start correcting errors found in the tests. Some of the corrections are easy. Some will require so much work that they will be tested in the next test batch in three weeks. After the three weeks a similar test cycle will begin again. Now the testers will be testing new features as well as corrections to errors they found three weeks ago. And so on.

A testing job that took more than six calendar days could have been completed in a few hours by automated tests. The corrections in the code could have been verified any time by just re-running the tests. The time saved in the feedback cycle is immense.

But test automation is not a silver bullet. While you can reduce the number of testers, you still need people to execute tests manually as well as to design and maintain automated tests. A creative human being is still superior to a machine when it comes to testing new features and figuring out what could go wrong. The machine is superior when it comes to doing the same routine repeatedly. In a world of fast cycles and multiple integrations, "repeatedly" represents a large number.

Test automation is usually perceived as the automation of the execution of tests. This is a somewhat narrow view. Broadly speaking, test automation means:

1. Automating execution of the tests
2. Automating preparation of test data
3. Automating creation of test cases
4. Automating setup and configuration of test tools and environments
5. Automating installation and configuration of the system being tested
6. Automating reporting of test results
7. Automating analysis of test results and
8. Automating reporting of test statistics and quality metrics

Test automation will deliver huge benefits if done right. Again, it all starts from understanding the quality risks, understanding the dependencies, and understanding the software process. Automating a poor process still leads to automatic mediocrity.

[Previous: This is why you need to test from coding to production]

Qentinel Robotic Software Testing

[Next: This is how you find metrics that matter]