THE HUMAN SIDE OF TEST AUTOMATION

Evolving your manual processes with automation
INTRODUCTION: HUMAN INTERACTION

We’ve dreamed of the day when robots can do anything that humans can do, alleviating us all from the mundane tasks in our lives. That vision of freedom has been the premise of many a sci-fi film, however the reality has not arrived, either in our daily lives or in the software development environment. According to recent research, organizations have automated only 28%* of all their test cases, leaving the rest to manual processes.

The goal of full automation is unrealistic since human interaction will always be essential. We humans have irreplaceable skills which automation cannot match: such as our ability to test the non-objective aspects of an application like usability, look and feel (is it modern or old-fashioned?), ease of use and visual behavior. Nor is it always possible for automated procedures to properly ‘anticipate’ any and all actions that a user might take. For example, mistaken actions such as clicking a ‘purchase’ button too early in an on-line store application – which a manual tester would make sure performs ‘correctly’ under those conditions.

In research, 39%* of respondents cite device availability and time constraints as the main challenges in the area of mobile testing.

Despite this, automation has been much-hyped, with an inference that it has the ability to replace manual testing. This ‘rip and replace’ strategy for manual testing has universally failed; however the importance and interest in automation remains – particularly in light of the rapid adoption of practices such as Agile and Continuous Integration.

PACE OF CHANGE – THE PRESSURE IS ON

While manual testing can bring great quality to finished software, organizations are now facing the reality that manual tests can’t stay completely manual forever. The pace of development and rapid Agile release cycles dictate that delivering on time and to budget means tightening not lengthening the testing process.

Manual testing is already time-consuming, and becoming more so thanks to the ‘mobile multiplier’ effect of different platforms, devices and operating systems that must be tested. In research, 39%* of respondents cite device availability and time constraints as the main challenges in the area of mobile testing. Meanwhile, think how often browsers get updated: Firefox alone has a fixed release cycle of six weeks. It is unrealistic to expect a test department to re-test an application every six weeks using purely manual processes. Yet even if that application hasn’t changed during that time, testing must be performed to guarantee the application works properly.

There are also situations where manual testing is extremely difficult or impossible. A case in point is load testing, where testers must simulate hundreds or thousands of concurrent users – and will rarely, if ever, have those employee numbers within their four walls to accurately perform ‘real’ testing.

User case 1: Telefónica O2 Business Solutions

Improves on its load testing

Telefónica O2 Business Solutions deals with major customers around the world and manages the IT infrastructure for many of them. The company is frequently asked to check an application because ‘it is running slow’ and the root cause is often the number of users, so load and stress testing is used to identify capacity issues associated with a high volume of users.

By simulating real-world user demand with Silk Performer, Telefónica O2 are able to increase test user volumes from hundreds to thousands compared to their previous open source tooling. As a result they have resolve capacity problems before they have an operational impact, and in turn have increased customer satisfaction.
Virtualization of load through automation is the only practical solution, and can be extremely effective. In one major insurance company, the ability to run load testing more regularly and with less effort allowed a four-fold increase in the concurrent members that could be supported by a key system, while response times of under two seconds were maintained.

Finally, there’s the ‘human cost’ of manually testing to consider. With every change to the software being developed, the manual test process must begin again by hand, instead of allowing automation to take over and quickly re-run all of the required tests. Performing laborious, repetitive tests under pressure can lead to test fatigue and stress, as well as introducing human errors. These can range from misinterpreting the requirements and so consequently writing incorrect code, typing invalid inputs, sorting an input sequence incorrectly, or any number of other issues.

THE HYBRID TESTING MODEL

The ideal testing strategy is a hybrid approach that combines automated and manual testing elements. It automates what can be automated, while recognizing that some processes require manual intervention. The focus is on extending the reach of human testing, rather than replacing it.

More specifically, the hybrid model should integrate requirements together with manual and automated tests, to ensure that the original business needs which have been defined are both tested and implemented.

Regression testing is a typical first choice for the introduction of automation due to the many frequent code changes and the ability to run the regressions in a timely manner. Wherever automation is applied, the goal is to perform repetitive tests with enough variations (data, user actions, gestures, etc) to create different test ‘cases’, even though these will be built up from the same base functionality.

When automation is introduced as part of a hybrid testing solution, the tedium of manual testing is reduced significantly, and tests can be run both during the day and overnight (something that’s impractical or impossible in most organizations). Valuable test personnel can also be freed up for more strategic testing roles, and it is not untypical to see manual test efforts reduced by 25% or more. The same applies to load testing, since tests can be scheduled and run in the background using automation, to save time and improve efficiency.

User case 3: ikeGPS

ikeGPS hardware and software is used across the world by utility companies, defense forces, intelligence groups, energy companies, and other organizations.

To test its integrated hardware and software package, which runs on Windows Mobile, ikeGPS needed to automate since short sprints and complex test requirements made it increasingly difficult to meet demands using a heavily manual testing process.

Developers turned to Silk Mobile’s simple user interface, and immediately appreciated its intuitive, human-like approach which mimics manual testing processes on real mobile devices. This led to rapid adoption, and the solution is now deployed across multiple hardware platforms. Automated mobile tests are run overnight and following Silk Mobile’s implementation, testing effort has been reduced by 25%.
A HUMAN TOUCH TO AUTOMATION

Creators of automation tools rarely discuss the human aspects of their tools, preferring to focus on the technical aspects they deliver. Yet test automation processes call for strong adoption at a grass roots level, and that usually springs from human needs – like working in a familiar way, being able to share different views and knowledge, and replicating human actions effectively. Some 42%* of research respondents cite a lack of availability of the right test tools to build reusable test sets as an issue.

It follows that test automation solutions which capture human input in a highly intuitive way – often without the need to write any code whatsoever – are more likely to mirror the way that manual testers currently work, and will be more easily adopted without the need for software developers to write automation code themselves, or to hire others to do it.

Certain test tools now provide capabilities to perform visual comparisons in a human-like way, or to grab details of an application that even manual testers would not be capable of capturing. Other test automation interfaces closely mimic human (manual) testing processes – for example, mirroring the screen of a mobile device on your desktop so that a test recording can capture all of the human input through a simple interface.

Automation should also reflect that one size does not fit all. Testing software must adapt to the needs of the user, whether they have a technical or a business focus. With role-based test automation tools, the human touch is accommodated through different interfaces to match needs and skills, including non-code based automation. This ensures that all the different views and knowledge within the team are properly captured, resulting in software being delivered at high quality and at higher speed.

SUMMARY

The holy grail of automation has been to eliminate manual effort and let machines do the work. The reality is that the most effective implementation of automation has been to supplement manual effort to eliminate the parts of a procedure that it can do more effectively, while removing the mundane and repetitive tasks that are stressful and frequently boring for humans. That enables a human to do what a human does best and for a machine to do what it does best. It is the ultimate humanity of automation – making the work less onerous for the user and enabling them to feel better about what they ‘have’ to do.

The challenge is to incrementally bring useful automation to bear on redundant and repetitive tasks in order to alleviate the pressure on manual testers. These pressures are increasing due to the mobile multiplier effect and the need to cross browser test across an explosion of devices, platforms and operating systems.

In order to cope, automation must be easy to adopt and have parallels with the manual testing it will automate, such as mimicking end user behavior. The tools should also adapt to the needs of different team members, recognizing that there’s a human side to successful automation as well as a technical one.

Discover more on how Borland brings the human touch to automation through the Silk Portfolio of testing tools here.


42%* of research respondents cite a lack of availability of the right test tools to build reusable test sets as an issue

User case 4: Mitel

Performs tests that are unfeasible manually

Mitel provides business communications solutions and services including unified communications and collaboration software applications, IP telephony platforms and mobile applications.

Product design and verification within Mitel involves a rapidly expanding matrix of tests to be run across multiple software and hardware integrations. Silk Test automation has been used for more than eight years to test literally millions of combinations of end-user interactions, product versions and different types of products. Automation allows Mitel to perform tests it otherwise could not do. For example, regression tests for a new product are set up across multiple platforms, left to run, with the results analyzed the after a couple weeks – without automation, Mitel believes this would take far longer.

“Silk Test is an important part of our delivery environment and a key reason we are able to get to market very quickly with a variety of quality products. In many cases, the cost and effort required to run the tests manually would be prohibitive. We simply wouldn’t be able to get products into people’s hands quickly enough.”

Karl Wagner,
Manager, Software Design Environment, Mitel