

## **WHY MANUAL TESTING REMAINS RELEVANT**

Vinnitsia National Technical University

### ***Анотація***

*У даній роботі увагу приділено актуальності та затребуваності ручного тестування в процесі розробки програмного забезпечення.*

**Ключові слова:** програмне забезпечення, автоматизація, автоматизоване тестування, ручне тестування, забезпечення якості.

### ***Abstract***

*In this paper the relevance and demand for manual testing in the software development process are considered.*

**Keywords:** software, automation, automated testing, manual testing, quality assurance.

### **Introduction**

Every day we see hundreds of new applications and products come out into the market. A lot of testing takes place before they are released to the public. Automated testing can speed up the process and is often seen as a replacement for manual testing. However, manual testing still has a critical role in the QA process.

### **Research results**

The increase of test automation adoption has unjustly framed manual testing as an archaic and unnecessary practice. After watching an automation suite swiftly execute an entirely library of test cases, it can be easy to tunnel vision on the great benefits of automation. However, the value of manually executing your tests cannot be understated. Here are a few reasons why manual is still relevant as ever.

Initial automation requires an increased investment in both, time and resources. Attempting to initialize automation while close to the end of your testing cycle would be a moot effort; the time you take to set up and the sudden resource shift means you'll be nearing your release date before you can start running reliable and core automated testing. During that same timeframe, you could be focusing your testing resources towards manual execution [1].

Like any code, your automation will contain errors. An error filled automation script may be misinterpreted as failed functionality in your tested application, or even worse your automation script will interpret an error as a correct functionality. Manually testing your core, critical-path functionality ensures that your test case is passing from a user perspective, with no room for misinterpretation.

The advent of automated testing platforms for Responsive and UI testing has provided a much appreciated convenience. However, it should be a boost to your UI testing efforts. These programs validate your test cases by checking element distance, image placement, and alignment of elements in relation to each other. Because of this, there are more than a dozen ways that something such as alignment between a menu and logo can be misinterpreted; a manual tester would immediately be able to catch something that looked "off", and fail the test case [2].

Some scenarios are simply not feasible to automate. They are either actually impossible due to technological limitation and the complexity of the scenario, or the resource cost of automating it greatly outweighs the cost of a simple manual test.

While the end result of Automation is ease, the set up of framework and development of scripts are no easy tasks. An effective automator has a foundation of programming skills, as well as an inherent understanding of test design. These skills are learned over years of experience in both QA and Development, and acquiring somebody with these specific skillsets, especially on short notice, is not a simple process. On the other hand, the majority of Manual test cases are simple to execute and can easily be taught. Follow the

steps in the test case, and validate that your actual results are consistent with the expected results.

In the context of Agile testing, automation is of great benefit. Having a library of tests reliably and quickly executable truly helps with test completion and coverage during a tight sprint. By that same token, manual testing is a quick way to execute for any test cases that are not yet automated. There may be no time to build automation for new features introduced in the current build, making manual the best option for test completion [3].

### Conclusions

As a conclusion, the need for increased test coverage across an ever increasing range of software and devices has made test automation more important than ever. As automation continues to grow, it can be easy to forget about the wide spectrum of benefits manual testing still has to offer. Appreciating the value of both approaches will make for a wholesome testing experience.

By dedicating your QA resources to only one of these approaches, you will miss many opportunities to improve quality. Using manual and automated testing together will lead to a higher quality, more stable product.

### REFERENCES

1. Keep calm and carry on – manual testing is not dying. Available at: <https://techbeacon.com/app-dev-testing/keep-calm-carry-manual-testing-not-dying> (accessed 20 February 2019)
2. Mukesh Sharma. Software Testing 2020: Preparing for New Roles. Boca Raton, CRC Press, 2016. 141 p.
3. 13 Reasons Why Manual Testing Can Never Be Replaced. Available at: <https://testlio.com/blog/manual-qa-testing-best-practices/> (accessed 20 February 2019)

**Носковенко Юлія Павлівна** — студентка групи ІАКІТ-18м, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця, e-mail: [angel479870@gmail.com](mailto:angel479870@gmail.com)

**Тулчак Людмила Володимирівна** — старший викладач кафедри іноземних мов, Вінницький національний технічний університет, м. Вінниця

**Noskovenko Yulia P.** — Faculty of computer systems and automation, Vinnytsia national technical University, Vinnytsia, e-mail: [angel479870@gmail.com](mailto:angel479870@gmail.com)

**Tulchak Liudmyla V.** — Senior Lecture, Chair of Foreign Languages, Vinnytsia National Technical University, Vinnytsia