How Automation Testing Breeds Better Software

(Technical Topic)

The True Societal Impact of Automation Testing

(STS Topic)

A Thesis Project Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Abram Johan
April 2, 2024

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed: Abram Johan

Advisors

Kathryn A. Neeley, Department of Engineering and Society

Harnessing Automation Testing to Help Eliminate Software Failures

Introduction

In general, poor and inadequate software testing leads to poor software quality. It permits code that is riddled with various bugs, glitches, and faults. While this mostly occurs in amateur programming spaces, many of the major software failures of the world were attributed to shortcomings in its software testing. Some notable cases include the notorious Y2K bug, the Nest thermostat malfunctions, and even the Mariner 1 Spacecraft failure. According to Patel (2024), advanced software testing techniques are not formally taught in most computer science curriculums. Software testing courses are mostly optional, and apart from some brief instruction on debugging methods, the only testing many students are required to learn are elementary. Therefore, many computer science students and graduates aren't proficient in any testing other than basic unit testing. In fact, a good portion of software students and developers don't even know what automation testing is.

If this situation remains the same, these problems will continue to persist. Not only will the quality of software be substandard, but companies will be more exposed to potential software issues. Consumers will be more vulnerable to receiving faulty products as well. Moreover, it would only be a matter of time until we encounter a software failure of catastrophic scale. Improving and correcting the software testing knowledge and skills of developers should be one of the larger priorities of both educators and software companies. I propose that we utilize and promote automation testing as a tool to prevent software faults early in the development process.

Technical Topic: How Automation Testing Breeds Better Software

Automation testing is a critical addition to modern software testing, and it has revolutionized the way that software is tested. By automating certain functions of an application and testing them programmatically, developers and testers can significantly reduce the risk of software bugs and failures. With frameworks such as Selenium, developers can write scripts that execute different functionalities, allowing for a convenient and efficient way to test the various scenarios involved in different user interactions. Another benefit of this is that it allows tests to be easily reproduced, so tests can be rerun whenever changes are made to software taking almost no time compared to manual testing. The process is also intuitively more reliable, as it does not depend on any human action.

Manual testing is highly limited as it is more time consuming, and it requires a human to physically perform the tests. Moreover, it is more susceptible to error, it is not reusable, and it can't be run in parallel with other tests. Automation testing mitigates these issues as it is more reliable, efficient, reusable, accurate, and cheap. Despite these clear advantages, according to software expert Shinji Kanai, "globally, an average of only 15% of tests were automated." Though this form of testing has great upside, there is no call for its widespread use. This indicates a lack of education, awareness, and advocacy surrounding automation testing within the software development industry.

Various empirical studies have shown that automation testing has a positive impact on software quality, software projects, and software production. One study, led by Yuqing Wang (2022), outlined the impact of automation testing on software engineering using a study on survey responses from 37 open-source java projects. The study included a statistical analysis of various considerations, broadly assessing the impact of test automation maturity on product quality, release cycle frequency, and the continuous integration (CI) success of open-source projects. The results revealed that test automation has a positive impact on product quality, leads to shorter release times, and correlates with greater CI success.

Another study, steered by Dudekula Rafi (2012), contrasted the benefits and limitations of automated software testing using systematic literature review and practitioner survey. The study, which is shown below in Figure 1, revealed that automation testing required a high initial investment into automation setup and 80% of the practitioners surveyed disagreed that automation testing would fully replace manual testing. However, Rafi stated that “the survey showed that benefits of test automation were related to test reusability, repeatability, test coverage and effort saved in test executions.” Other data suggests that the initial investment for automation setup leads to much greater savings in the long term. According to Chatterjee, a community expert, “it lessens the related costs of maintaining a large testing team” and “in the long run, it is highly cost-efficient.” This cost-efficiency also stems from identifying bugs early, so they don’t infect the software and cost much more later in the development process.

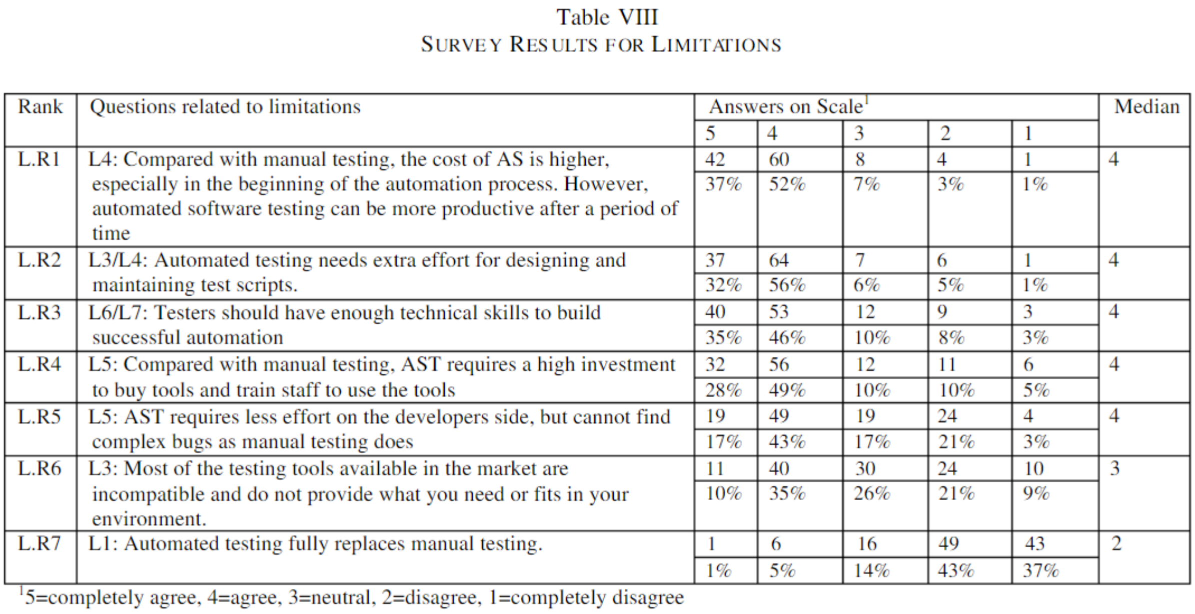

Table VIII
SURVEY RESULTS FOR LIMITATIONS

Rank	Questions related to limitations	Answers on Scale[1]					Median
		5	4	3	2	1	
L.R1	L4: Compared with manual testing, the cost of AS is higher, especially in the beginning of the automation process. However, automated software testing can be more productive after a period of time	42 37%	60 52%	8 7%	4 3%	1 1%	4
L.R2	L3/L4: Automated testing needs extra effort for designing and maintaining test scripts.	37 32%	64 56%	7 6%	6 5%	1 1%	4
L.R3	L6/L7: Testers should have enough technical skills to build successful automation	40 35%	53 46%	12 10%	9 8%	3 3%	4
L.R4	L5: Compared with manual testing, AST requires a high investment to buy tools and train staff to use the tools	32 28%	56 49%	12 10%	11 10%	6 5%	4
L.R5	L5: AST requires less effort on the developers side, but cannot find complex bugs as manual testing does	19 17%	49 43%	19 17%	24 21%	4 3%	4
L.R6	L3: Most of the testing tools available in the market are incompatible and do not provide what you need or fits in your environment.	11 10%	40 35%	30 26%	24 21%	10 9%	3
L.R7	L1: Automated testing fully replaces manual testing.	1 1%	6 5%	16 14%	49 43%	43 37%	2

[1]5=completely agree, 4=agree, 3=neutral, 2=disagree, 1=completely disagree

Figure 1. Practitioner survey results from Rafi's study on the benefits and limitations of automated software testing. Results are scaled from 1-5, legend is presented under the table and questions cover logistical questions about automation testing (Rafi 2012).

The empirical research described above helps demonstrate the technical advantage of companies implementing automation into their testing procedures. However, it doesn't quite depict the impact that a software testing tool like this could have on software quality, software products, and on the world.

STS Topic: The True Societal Impact of Automation Testing

As technology advances, a significant portion of our surroundings depend on software. Everything from our phones to our cars to even our software depends on its functionality. While this helps optimize our everyday lives, it also makes us as consumers reliant on tech/software companies. Inadequacies in their work can be vulnerabilities in our lives. That is the impact that software has on the world, and its grasp on humanity is only growing wider and stronger every day. This is why adequate software testing practices and education is of utmost importance in the current day.

Software testing aims to prevent all software errors from ever reaching clients. According to Rahul Parwal (2024), it is "fundamentally an information-gathering and learning process. As a stakeholder in the product, testing permits you to obtain early feedback about the system." The goal is to catch bugs early in the development process so they cannot impact newer systems when they are created. However, the question is: What happens when testing goes wrong? That is the question Parwal aims to answer in his article, and it is the crux of the topic. For companies, they take a hit on their reputation, lose revenue, receive slow feedback, push out poor products, and they may even receive lawsuits. However, these types of issues are to be expected with creating software as it is within the company's control. Consumers, on the other hand, have no agency on the topic and can also be impacted by software flaws.

Consumers, while not responsible, are not immune to the consequences of software failures. In fact, they are often the ones affected the most. As shown in figure two, test automation maturity positively impacts product quality, so it is the consumers that suffer from poor testing. When a banking company's system fails, the customer is the one who loses their data and transactions. The customer is the one who is susceptible to financial losses because of the error. These types of situations have occurred before and on scales much larger. Deepak Arora's (2021) article depicts some of the larger software failures in the past. According to him, "The biggest software failures in history are IT outages, Ransomware attacks, including data leakages that have affected millions of customers." The article highlights many failures including the T-mobile data breach, the Tesla memory failure that resulted in 135,000 recalls, and even a security issue on Zoom. He preaches that "It is important to understand the uncertainty of the future. Therefore, any software or application made must be built on a solid foundation."

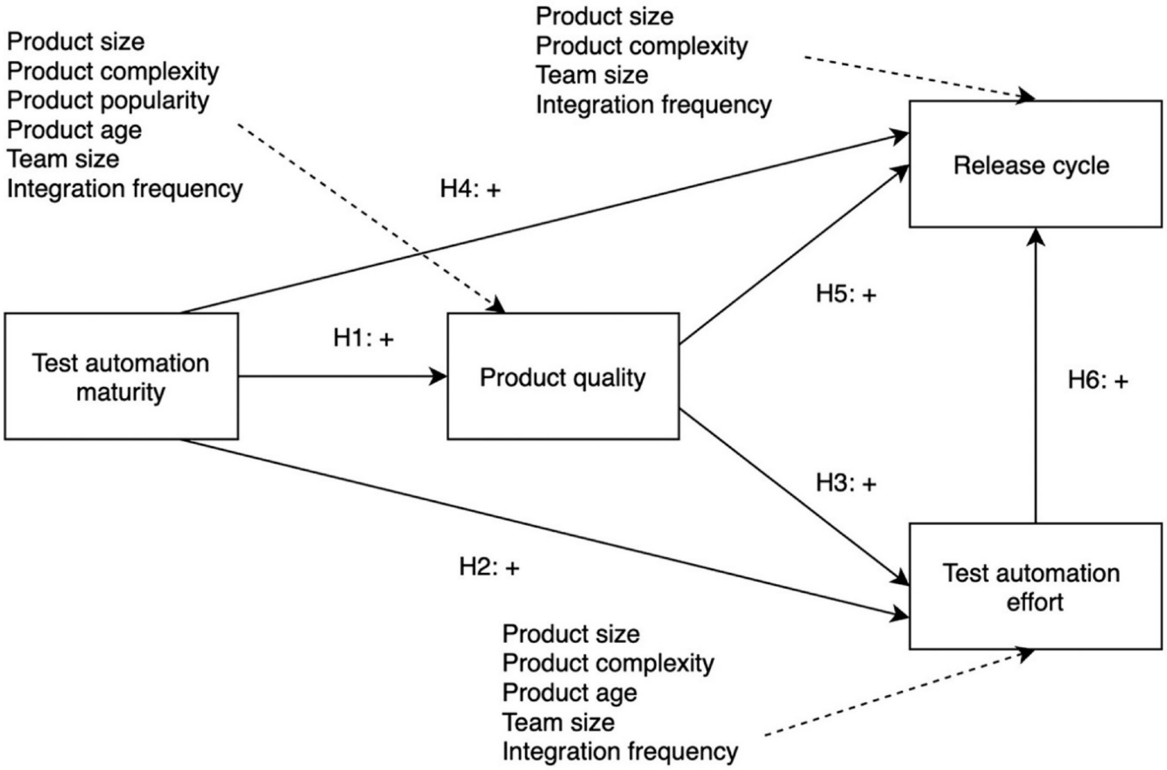

Figure 2. Diagram of the impact of test automation maturity. Test automation has positive impacts on product quality and reduces test automation effort (Wang et al., 2022).

Automation testing is one of the most simple, effective, and efficient ways to mitigate software failures. As proven earlier, it has a profound impact on software quality and its versatility can make it a great option. As testing requirements get more complex and rigorous, automation is the way to maximize testing coverage in the long run. With more companies utilizing it, we can proactively safeguard against potential disruptions, effectively protecting the interests of both companies and their consumers. In essence, automation testing can extend beyond just the software development realm. It can serve as a measure to uphold integrity in software systems but also in protecting the interests and well-being of consumers in an increasingly digitized world.

Conclusion

The anticipated deliverables of implementing automation testing in software engineering are multifaceted. From a technical standpoint, the adoption of automation testing frameworks is expected to promote increased education and awareness among software developers, promoting better programming practices, and ultimately resulting in fewer faults in software products. This transition towards automation testing would show a shift in our current software testing methodologies, offering a systematic approach to identify and resolve issues early in the development process.

If successfully implemented and integrated into software development education and industries, automation testing has the potential to address the systemic problems presented earlier in this document. By minimizing software failures with better testing methods, organizations can mitigate risks associated with poor software quality. They can ultimately enhance product reliability and customer satisfaction. Moreover, automation testing can contribute to the resolution of issues related to inadequate testing protocols and the lack of standardized testing practices, ultimately fostering a culture of quality assurance and continuous improvement in software development processes. (1570 words)

References

Arora, D. (2021). Top Software Failures Due to Lack of Testing. *BugRaptors(website)*. https://www.bugraptors.com/blog/top-software-failures-due-to-lack-of-testing.

Summary: This article simply highlights many of the top software failures that occurred due to a lack of testing. Including cases of companies like Slack, T-Mobile, Yahoo, and many others, the article emphasizes the importance of a solid foundation in software development. This includes early and regular software testing. The company failures resulted in huge breaches of data, millions of dollars lost, and in some cases even casualties of death. Therefore, it really conveys the importance and impact of software testing. (80 words)

Chatterjee, S. (2023). What is Test Automation: Benefits, Limitations, Tools, and Best Practices. *BrowserStack(website)*. https://www.browserstack.com/guide/what-is-test-automation

Summary: This informational article helps outline the meaning, importance, benefits, and many other aspects of automation testing. At its core, the report explains that automation testing is important because it has major implications for continuous testing, project collaboration, scalability, continuous delivery, bug detection, and even security. Moreover, it explains how automation testing can benefit developers by providing speedy tests, increased test coverage, better consistency, higher cost efficiency, and enhanced reusability. The article also illustrates that the automation testing market will skyrocket in the future, citing models that project massive growth between 2022 and 2032. (93 words)

Grater, M. (2005). Benefits of Using Automated Software Testing Tools to Achieve Software Quality Assurance. *CiteSeerX(website)*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=699a4af7d5eb936da0df36d9f080fe3d598d5cd0

Summary: This document is a capstone report about how beneficial automated testing tools are for achieving software quality assurance. The report asserts that the increasing complexity of software development demands that developers use automated software testing tools. Using academic literature, it proceeds to analyze the benefits of automation testing in relation to five factors: quality, reliability, usability, security, and performance. The reports findings indicate that automation is faster, more reliable, accurate, cheaper etc. It states that it helped locate defects faster and earlier, and it also helped remove more bugs before product release. (92 words).

Kanai, S. (2022). 15 Benefits of Automated Testing in App Development. *Headspin(website)*. https://www.headspin.io/blog/15-benefits-of-automated-testing-in-app-development

Summary: This educational article illustrates why automation testing is necessary in today's world, highlighting its edge and benefits over manual testing. The article also points out

that only 15% of the current apps are tested using automation currently. The article mentions how automation testing is more reliable than manual testing as it removes human error, it is less time-consuming, and it has more testing coverage. Automation testing also leads to faster feedback cycles, allows testing on multiple platforms in parallel, reusability of test scripts, and better test insights. (87 words)

Lemos, O., Silveira, F, Ferrari, F., & Garcia, A. (2018, March). The impact of Software Testing education on code reliability: An empirical assessment. *Journal of Systems and Software Volume 137, 497-511*. https://doi.org/10.1016/j.jss.2017.02.042

Summary: This study speaks on software testing, focusing on the impacts of testing education on both enhancing programming skills and developing more reliable software. Moreover, it demonstrates how common inadequate knowledge is, emphasizing its impact. The study contains an experiment on students and a survey of professors. The experiment results showed that even basic testing training rendered developers more than twice as likely to produce correct implementations. The survey with the professors showed that even developers as skilled as them lack the formal testing expertise that is so crucial. (88 words)

Parwal, R. (2024). What Happens When Testing Goes Wrong? The Impact of Poor Software Quality. *Muuk Test(blog)*. https://muuktest.com/blog/impact-of-poor-software-quality

Summary: This article outlines some of the impact that improper software testing can have and the impact of its consequent poor software quality. It emphasizes that poor testing can/will result in bugs in code. If not tested throughout development, the bugs could infect the rest of the software and lead to a multitude of issues. Not only can it impact revenue, but it could also ruin your reputation. Moreover, faulty software could leave you susceptible to lawsuits in certain situations. (79 words)

Patel, V. (2024). Software Testing in Logistics & Transportation: Benefits, and Examples. *QAble(website)*. https://www.qable.io/blog/software-testing-in-logistics-and-transportation

Summary: This educational article outlines the major benefits of software testing, focusing primarily on its impact on logistics and transportation. The article starts by mentioning how software testing enhances customer satisfaction, efficiency, and productivity: the key factors of creating a successful product. However, the article goes on to say that software testing also has an impact on the global economy with its influence in logistics and transformation. Playing a part in supply chain management, warehouse management, transport management etc., software failures could have massive implications on important product shipments. (88 words)

Rafi, D., Katam, Mäntylä, M. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. *ResearchGate(website).*

https://www.researchgate.net/publication/261392355_Benefits_and_limitations_of_automated_software_testing_Systematic_literature_review_and_practitioner_survey

Summary: This scholarly article delves into the topic of automation testing, aiming to bridge the gap between academic and practitioner perspectives. To do so, the paper investigates the benefits and limits of test automation. Using a survey of 115 software professionals, the study concluded that benefits of test automation were related to test reusability, repeatability, test coverage, and effort saved in test executions. While initial setup proved to be difficult, the tangible benefits of the testing were apparent. (77 words)

Wang, Y., Mäntylä, M. V., Liu, Z., & Markkula, J. (2022). Test automation maturity improves product quality—Quantitative study of open source projects using continuous integration. *Journal of Systems and Software Volume 188, 111259).* https://doi.org/10.1016/j.jss.2022.111259

Summary: This scholarly article outlines the impact of automation testing on software engineering using a study on survey responses from 37 open source java projects. The study includes a statistical analysis of various considerations, broadly assessing the impact of test automation maturity on product quality, release cycle frequency, and the continuous integration (CI) success of open source projects. The results reveal that test automation has a positive impact on product quality, leads to shorter release times, and correlates with greater CI success. (81 words)