Automation Testing: Streamlining Quality Assurance

A Technical Report Submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Abram Francis Johan

Fall 2024

Technical Project Team Members

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

Rosanne Vrugtman, Department of Computer Science

Automation Testing: Streamlining Quality Assurance

CS4991 Capstone Report, 2024

Abram Johan
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
atd2ec@virginia.edu

ABSTRACT
iWorks, an innovative software firm based in Reston, Virginia, decided to streamline the testing process for their web application by employing automation testing and continuous integration/deployment. To achieve this, I utilized Selenium to generate automation scripts, and used GitLab to deploy a CI/CD pipeline. When smoke testing the application, I had to write various scripts, each of which automated and executed a function of the app. Then, using TestNG, I created unit tests to ensure that the automated functions executed properly. I then deployed a Gitlab pipeline that builds, tests, and deploys the code. From this process, I found several small bugs in the app's interface, state, and authentication process, which I reported to the developers for a fix. Moreover, I put in place a CI/CD and test suite for developers to use anytime now or in the future. As with all software maintenance, I will need to adapt and tweak the scripts and tests as the app evolves.

1. INTRODUCTION

Without proper software testing, much of the technology we have today would be either faulty or dysfunctional. Everything from the phones we scroll on, the apps we use, and websites we browse use software to function. This type of work is typically handled by software engineers, who are expected to create tests during and after development to ensure that their code works as expected. During software engineering internship, one of my tasks was to test the company's web application using automation testing.

Automation testing is essentially the use of external software to programmatically execute tests on the target software. One popular tool for this is Selenium, a framework geared towards supporting browser automation. When testing a browser web application, Selenium allows testers to write scripts that can access Google Chrome browser and automatically execute functions of the app. For example, a basic test script could automate the process of logging onto an app by entering valid credentials.

Selenium allows for the creation of automation scripts, but it does not serve well to create formal tests. For this, I used TestNG, a popular testing framework, to develop unit tests for each of the automation scripts. In tandem, the two allow for the creation of sharp automation tests. This allows for a streamlined method to ensure functionality throughout large development changes. It also helps quantify bugs, identify faults, and overall maintain a higher caliber of product for the company.

2. RELATED WORKS

Automation testing has proven to be a worthwile form of testing since its conception. With it, developers can seamlessly execute continuous integration of

their systems. This can help run many tests on parallel threads and it forgoes the time-consuming chore of manual smoke testing. This helps them identify their bugs more readily and ultimately, push their products out more quickly. This is proven in a group research study led by Wang, et al (2022). They examined 37 open-source java projects, conducting a full statistical analysis of the projects' life cycles. According to Wang: "higher levels of test automation maturity (assessed by standard best practices in literature) are significantly associated with shorter release cycles." Moreover, their study concluded that test automation maturity has a positive association with product quality.

Automation testing also helps companies reduce their failures, which obviously helps save time in development. This ultimately saves money for the company because it enables them to identify bugs early at the unit level and skip the tedious task of fixing the bugs on a system level. Moreover, companies have fewer maintenance requests from their customers, enhancing their reputation, which is invaluable. There is more accuracy in programmatic testing too, as there is less human error involved. According to Karanth (2024): "At the end of the day, delivering a quality product to the market beats any other type of savings and cutbacks." This statement emphasizes how automation testing can prove to be invaluable in the long run.

3. PROJECT DESIGN

My design pushes for an increase in automation testing in college education for all computer science students. This work would be the first step in making automation testing a mandatory part of computer science/software engineering curricula. With this step, my work will aim to optimize the state of testing in and industry, mimimize the number of total software errors, and improve the speed of software production.

To achieve this, my work will implement an academic experiment in college education that will evaluate the impact of automation testing on computer science students' ability to create good software. The experiment will aim to bridge the cap between the theoretical knowledge college students receive and the technical skills they will need in industry.

The experiment will be structured with two groups of computer science students: Group A and Group B. To keep their knowledge levels as consistent as possible, we will choose two groups of students enrolled in the same course. To ensure that these students are already somewhat proficient in programming, we will choose students in some software development course. We will remove any students from either group that already have experience with automation testing. Group A will serve as the control group and will not receive any additional training on automation testing. Group B, on the other hand, will receive a comprehensive training on automation testing, including rigrorous practice with it. The training must include theoretical lectures, along with practical exercises including use of Selenium and TestNG at least.

Students of both groups will have to build a website. It should be a large-scale project utilizing both front and back end, so it would be a semester-long proeject done throughout the course. All students should be given the same requirements for what their website should look like to ensure that their code can be evaluated in the same way. Once both projects are complete, comprehensive testing will be conducted on the projects. This phase will be conducted to identify and record any bugs, errors or inconsistencies in the website's functionality.

The testing of the sites will be conducted from both groups throughout the experiment to analyze their varying learning experiences. Then, at the end of the project, the websites will be thoroughly examined, using a combination of manual testing by the research leaders and automation testing led by Group B students. The data will be collected from both groups separately. When documenting the errors for comparison, we will only evaluate the manual testing. Testers will follow the same script for a run-through of the website's functions. It will be documented in terms of minor functionality errors, major functionality errors, and visual faults. Once all data is recorded, the research team will gather all the data and determine if there was any significant discrepancy between the average number of errors in both groups' websites. This will be done using some form of statistical analysis to reveal whether any difference in the data is trivial or relevant.

4. ANTICIPATED RESULTS

As this experiment requires access to a sample of students and their course curricula, I will not be able to actually conduct the experiment. Because of their use of automation testing techniques, I anticipate a significant improvement in software quality in Group B. I expect their website to have fewer bugs and errors, thereby validating the benefits of implementing automation testing in computer science education. Ideally, the results of this experiment should yield positive results so that we can make such education mandatory. In a world where the need for testing has become increasingly imperative, the integration of measures like this is critical.

5. CONCLUSION

This project is important as it would serve to provide more exposure to automation testing. Moreover, it would be a strong first step toward making automation testing a staple in software testing procedure. As it is expected that the experiment yields results that reveal the positive impact of automation testing, it is my hope that we are able to make automation testing a mandatory component of computer science education. This way, all software engineers in the industry will be well prepared to use automation testing in their jobs.

This is especially important in this age as software gets increasingly large and complex. Because software failures and defects cost companies millions of dollars a year, it is imperative that developers can find a way to better identify them earlier on. Additionally, the advent of artificial intelligence has produced a whole new realm of programming. With students, software engineers, and programmers all beginning to rely more on artificial intelligence to create large code bases, it is now more important than ever to shift to a method of testing that is efficient on a large scale. This work would represent a crucial step in educating students, preventing software defects, easily identifying these defects, and ultimately creating more robust software.

6. FUTURE WORK

In the future, further work will need to be done to evaluate more long-term impacts of automation testing on the careers and professional success of the students. Another need would be the refinement of curriculum material and instructional methods on the topic. Professors would need to become well-versed on the topic so they can teach their students correctly. Additionally, the exploration of advancements in automation testing capabilities could be beneficial. With the help of quality assurance professionals, advancements in automation testing technologies could help accommodate evolving industry trends.

REFERENCES

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

Karanth, D. (2023). Why Automation Testing is Necessary. *cprime(blog)*. https://www.cprime.com/resources/blog/why-automation-testing-necessary/