

## **ASSIGNMENT 2 (TESTING)**

**Name: Omkar Shrikant Gawade**

**ID : 1001967237**

### **Machine learning in software engineering**

#### **INTRODUCTION**

Software testing is an expensive process and takes anywhere around 25-40% of an overall project's budget but it's necessary for ensuring software works correctly. It's feasible for larger companies to hire dedicated teams for testing but not for small companies. Meanwhile, market pressures favor quick development cycles, leaving little time for thorough testing.

Large and small businesses alike are seeking methods to cut costs and increase the scope and reliability of testing while still meeting customer expectations and timelines. As a result, corporations are looking for automated ways to test software more efficiently, ushering in the era of automated testing.

Over the last decade, there has been a constant effort to deploy software more quickly. Companies are spending a lot of time and effort on end-to-end software delivery pipelines, and containers and their ecosystem are delivering on their early promises. Software testing is an important aspect of the software development process (SDLC). Initially, testing was done by hand, which was a time-consuming and labor-intensive operation. Then there was test automation, which used software to perform tests and find faults. Automation changed testing, bringing with it a slew of advantages such as faster feedback and more test coverage. For DevOps organizations, scaling test automation and maintaining it over time remains a problem..

Today, artificial intelligence (AI) and machine learning (ML) have invaded the software testing sector, ushering in a new age in the software development business. The goal of artificial intelligence in software testing is to make testing smarter and more dependable. AI and machine learning have had a significant influence on software testing, with their adoption making the process easier, quicker, and more accurate.

Software that is well-written must fulfill both functional and non-functional criteria, as well as adhere to appropriate coding conventions. Programmers that follow coding rules regularly produce code that is simple to comprehend and avoids unneeded complications.

"Importance of code quality and coding standards in web application, mobile app, and software development" has further information on the importance of coding rules. How do you make sure your team adheres to code standards? Code review is the only option, but it's a costly process! You need tools that can detect typical violations from coding standards in software engineering to guarantee that reviewers focus on what matters. Machine learning can assist in this situation since ML-based solutions can detect such frequent aberrations. This can have a big influence on the software development projects you're working on.

#### **FIRST ARTICLE**

Software testing is an important and essential part of the software development process (SDLC). With competent software testing, the quality of software may be improved. The significance of software testing can be seen in fact that it has been established as just a full method rather than a single step in the software development life cycle (SDLC). This procedure begins concurrently with the SDLC. The testing procedure can begin as soon as the project's requirements are determined. Software testing life

cycle is a testing process that, like SDLC, is made up of a well-defined set of phases (STLC). The main advantage of STLC is that it involves testers early in the software development process.

However, the issue with software testing is that there are so many test cases to test. It indicates that the software testing domain is too vast to test. We may alternatively state that, given the processing resources and time constraints, none of these test cases can be completed. To put it another way, thorough testing is not possible. As a result, rather of focusing on thorough testing, we should focus on effective testing. Testing must be carried out on a limited number of test cases that cover all areas of software. Effective testing is the term for this. Effective testing entails devising ways that allow specified portions of test cases to be run rather than the entire set of test cases.

There is a potential that other modules of the software will be affected if the software configuration changes for any reason (bugs, addition of new functionality, etc.). In this instance, we'll have to re-test the entire software.

As a result, machine learning ideas are applied to prevent headaches. Many software systems include machine learning (ML) as a regular feature. A trained model in your system may present predictions to users directly to assist them in making a human choice, or it may make automated decisions within the software system itself. Whether the machine learning in a system is built in-house or received from a 3rd party's pre-trained model API, production software that relies on a trained model's predictions must be thoroughly tested, just like any other component of the program.

## **SECOND ARTICLE**

With the extensive use of machine learning to develop software applications, it has become essential to follow best software practices for machine learning-based systems as well. The success of any software application is totally dependent on the effectiveness of the testing strategy to test the software applications. The system developed using machine learning required regular training of models and every time the predictions of the trained model are required to be validated and tested. The use of machine learning in software testing facilitates the software testing. For performing the testing of artificial intelligence-based systems, testing is required at each phase of the model development such as testing of data cleansing, feature engineering etc. The generation of the test cases using machine learning is easier and fastest. Machine learning also helps to design the technique to reduce the test maintenance and provide a simpler test analysis.

Various algorithms of machine learning like the support vector machines, semi-supervised K - means, K - Nearest Neighbor classification, Q-learning Support Vector Regression and Multi-layer perception are applied to generate the effective test cases and also prioritize the test cases. The testing strategies based on machine learning are capable of delivering reliable and robust software within the time and allocated budget.

## **CONCLUSION**

As you can see both views truly support using machine learning in software testing. Though implementation of machine learning in any industry is complex and time consuming, It saves a lot of time and avoids human error. But according to my view, machine learning is really a future of not only software testing but in every application area like healthcare, business predictions, complex calculations, etc. along with that we all need to keep one thing in mind is that for machine learning every problem is solvable until it's not.

## **References**

1.Gao, Jerry; Tao, Chuanqi; Jie, Dou; Lu, Shengqiang (2019). [IEEE 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) - San Francisco East Bay, CA, USA

(2019.4.4-2019.4.9)] 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) - Invited Paper: What is AI Software Testing? and Why.

<https://ieeexplore.ieee.org/document/8705808>

2.Dan cagen; 3 Ways Machine Learning Can Improve Test Automation

<https://www.applause.com/blog/how-machine-learning-improves-test-automation>

3.Erik Fogg, Chief Operating Officer at ProdPerfect, introduces How AI and Machine Learning Will Shape SoftwareTesting, <https://insidebigdata.com/2021/01/27/how-ai-and-machine-learning-will-shape-software-testing/>

4. Scott Lindeman, introduces Software Testing and Machine

Learning; <https://towardsdatascience.com/software-testing-and-machine-learning-45f014968c62>

5.Vedpal J C Bose University of Science & Technology, YMCA, Faridabad; Naresh Chauhan

J C Bose University of Science & Technology, YMCA, Faridabad, Published in2021 5th International Conference on Information Systems and Computer Networks (ISCON) - Role of Machine Learning in Software Testing

<https://ieeexplore.ieee.org/document/9702427>