Testing technology, tool or practice you think is the "next big thing" in Software Testing

Suganya Kannan

December 21, 2020

Data Analytics and Artificial Intelligence(AI) will play a major role in all phases of the software development life cycle. It is not surprising that software testing will also be benefited from this. Internet of Things(IoT) is highly growing in the digital world and testing technologies have to be developed or adapted for the domain. The practice that I think is the "next big thing" in Software Testing is Model-Driven Engineering(MDE). MDE can help to tackle the challenges in new domain. Test models help to identify the defects in the system effectively. The testing environment can also be modeled. This helps in reducing the huge time and effort incurred in the process of software testing. It helps to reduce a lot of efforts by analyzing similar applications and their testing process.

User experience is very critical for a software application to be accepted and to be successful. Most often designers are not already aware of what the system should look like. To enhance the design based on the feedback of the users, the system undergoes user acceptance testing such as A/B split testing. The multiple design alternatives could be tested with the help of A/B split testing. In the split testing, different variants of the same application are given to different groups of people and their interaction with the system is recorded and analyzed. It is not feasible to cover all the variants using A/B testing as it will incur huge manpower and expenses. It becomes difficult to compare the results based on the different variants as each design is independent and no information about the system is stored. Design models can help to solve this problem by integrating both the regular design and additional uncertainty information. This uncertainty model can then be used to generate the software application with all its variants. The variants can then be compared with the user analysis to solve the uncertainties.

Automation tools with the help of MDE and AI could be made more intelligent. The tool should be provided with knowledge about certain bugs and their impacts. The severity level of the bugs could be made based on the intention of the application. For example, for a safety-critical application, more severity should be given to a security model. In the case of an application that heavily depends on the results of mathematical calculations such as the aircraft projecting, more severity should be given to the mathematical model. MDE combined with AI will help the domain experts, developers, testers, and other stakeholders to communicate in a common language. The main drawback of this approach is that it will be a bit overwhelming for the testers as they have to acquire knowledge of the modeling environment.