

A Survey Paper On Different Software Testing Automated Methods With Its Techniques

Dr.P.A.Selvaraj, Dr.M.Jagadeesan

Abstract: To generate software with high-quality, it is important to utilize software testing methods and tools effectively and efficiently. In this paper, we survey the various software testing approaches are compared and contrast that are used in the various fields. This survey will be helpful for improving software testing methods and tools practices.

Keywords: Software, software testing automation, test cases, test criteria, automated test suite methods.

1. INTRODUCTION:

Nowadays, software plays a vital role which is used in various fields such as aeronautics, medical field, industry, etc., To generate software with high-quality, it is important to utilize software testing methods and tools effectively and efficiently. Software testing is an experiment to test whether the obtained results matches with the expected results. Software testing helps to predict the errors, missing requirements with respect to actual requirement. It can be performed via manually or automated tools. It can be also called as blackbox or whitebox testing. Software testing tools are developed to support particular software testing techniques and method automation either fully or partially.

LITERATURE REVIEW

Enyi Tang, et al. [1] discussed the software development process instability and toolchain to resolve the instability issues. Our numerical analysis toolchain comprises stochastic arithmetic with infinite-precision testing. The proposed technique combines the Stochastic and Infinite-precision testing which effectively detects the problem of instabilities arise by others. An advantage of our proposed work is it not only predicts instability but also diagnoses the reason. It is applicable to the simple testing process. Hee E. Kim, et al. [2] described testing for safety-critical applications for failure probability calculation. He suggested a software failure probability quantification method. The model consists of test cases which hold states, it represents the input trajectory. The proposed method is suggested to nuclear power plant reactor to measure its effectiveness behavior. The advantage of this technique is decreased testing time for small input sequences. Justyna Petke, et al.[3] proposed a combinatorial interaction testing which is significant to test the configuration space of software systems. Algorithm used here is simulated annealing and greedy to test the cases for combinatorial interaction testing. Genetic algorithm also calculated in this method. Algorithm is highly efficient for testing in pairwise with less number of constraints. Miroslav, et al.[4] proposed an exploratory testing approach for detection of defects available in the system.

Administrative overhead generated by system under test, test data and their relevant information are automated in proposed framework. Navigational method is designed in order to avoid redundant test data. Drawback of our model is to provide a consistent guidance for testers since it should be operated continuously during the process of testing. Debdeep Banerjee[5] introduced a purpose of robotic arm for testing the software features available in mobile. Computer vision test automation supported the adding and managing tests via robotic arms. The accuracy is richer in recognition of text with image rectification algorithm. It also can be extended to support the different platforms. Carlos Oliveira [6] described a test suite automated generation method. Test automation tool identifies the software features which are significant to predict the desired automated test generation technique. Accuracy is higher in identifying the domains of CUT space for the different ATSGTs. BaekGyu, et al. [7] suggested self-governing vehicle software testing in the environment which is virtually prototyped. It allows testing the vehicle software in road environment which is virtual helps to decrease the amount of efforts spends on road testing which is physical. The drawback of our system is more time consumption for creating these types of environments. Jes'us Mor'an, et al.[8] proposed a fault prediction testing techniques in mapreduce scenario. It finds out these design faults by generating various configuration on infrastructure. It is automated by utilizing the MRTest engine. The testing techniques can be extended beyond the map reduce framework. Robert M. Hierons, et al.[9] introduced finite state machine testing using parallel algorithms. It constructs the unique input output sequences for finite state machines with millions of states. More scalable and produced accurate results compared to previous existing algorithms.

Automated Methods	Techniques	Merits	Demerits
Prediction of Software Numerical Instability and Diagnosis [1]	Combining Stochastic and Infinite-precision Testing	It not only predicts instability but also diagnoses the reason	Does not supports complicated process
Input-domain Software Testing for Failure Probability Estimation of Safety-critical Applications in Consideration of Past Input	A software failure probability quantification method	Testing time is decreased	Long time is needed for full completion of more test cases

Sequence [2]			
Early Fault detection using Combinatorial Interaction Testing [3]	Simulated annealing and greedy algorithm	Highly competitive for testing in pairwise	More constraints leads to decrease the efficiency of greedy approach
Tapir framework [4]	Team-based navigational Method	Eradicates the administrative overhead	For recording the parts of system under test, a large amount of time is exhausted on administrative tasks
Software test automation for rectification of pictures [5]	Robotic arm	Accuracy is higher	Test automation need to be enhanced to support various platforms
Automated Test Suite Generation methods [6]	Random Testing (RT), whole suite with archive (WSA) and Many-objective sorting algorithm (MOSA)	Selection of ATSG method derived from META framework has 88% accuracy	More datasets can be included
Self-governing vehicle software testing in a virtual circumstances [7]	Virtual prototyping	Decrease the amount of efforts spends on road testing which is physical	More time consumption for creating virtual environment
Fault prediction testing in mapreduce scenario [8]	MRTTest-Random, MRTTest-Wise and MRUnit.	Detects the faults automatically	Need to be expanded beyond the MapReduce framework
Fine State Machine Testing [9]	Parallel Algorithms	More scalable and provides better results	Performance degrades for larger FSMs

- [2] Kim, H. E., Son, H. S., Kim, B. G., Cho, J., Shin, S. M., & Kang, H. G. (2018). Input-Domain Software Testing for Failure Probability Estimation of Safety-Critical Applications in Consideration of Past Input Sequence. *IEEE Access*, 6, 8440–8451. doi:10.1109/access.2017.2765698
- [3] Petke, J., Cohen, M. B., Harman, M., & Yoo, S. (2015). Practical Combinatorial Interaction Testing: Empirical Findings on Efficiency and Early Fault Detection. *IEEE Transactions on Software Engineering*, 41(9), 901–924. doi:10.1109/tse.2015.2421279
- [4] Bures, M., Frajtak, K., & Ahmed, B. S. (2018). Tapir: Automation Support of Exploratory Testing Using Model Reconstruction of the System Under Test. *IEEE Transactions on Reliability*, 67(2), 557–580. doi:10.1109/tr.2018.2799957
- [5] Banerjee, D., Yu, K., & Aggarwal, G. (2018). Image Rectification Software Test Automation Using a Robotic ARM. *IEEE Access*, 6, 34075–34085. doi:10.1109/access.2018.2846761
- [6] Oliveira, C., Aleti, A., Grunske, L., & Smith-Miles, K. (2018). Mapping the Effectiveness of Automated Test Suite Generation Techniques. *IEEE Transactions on Reliability*, 1–15. doi:10.1109/tr.2018.2832072.
- [7] Kim, B., Kashiba, Y., Dai, S., & Shiraishi, S. (2017). Testing Autonomous Vehicle Software in the Virtual Prototyping Environment. *IEEE Embedded Systems Letters*, 9(1), 5–8. doi:10.1109/les.2016.2644619
- [8] Moran, J., Bertolino, A., de la Riva, C., & Tuya, J. (2018). Automatic Testing of Design Faults in MapReduce Applications. *IEEE Transactions on Reliability*, 1–16. doi:10.1109/tr.2018.2802047
- [9] Hierons, R. M., & Turker, U. C. (2016). Parallel Algorithms for Testing Finite State Machines: Generating UIO Sequences. *IEEE Transactions on Software Engineering*, 42(11), 1077–1091. doi:10.1109/tse.2016.2539964

CONCLUSION:

In this paper we have surveyed the trending existing work which focuses on software testing automation with the consideration of multiple environments and architecture. Some of the works proposed for software testing is to reduce the testing time by automated tools and increase the efficiency of software features but limited to fewer test cases. We hope our survey work will help to build a new effective system which supports huge test cases and to work across various platforms.

REFERENCES

- [1] Tang, E., Zhang, X., Muller, N. T., Chen, Z., & Li, X. (2017). Software Numerical Instability Detection and Diagnosis by Combining Stochastic and Infinite-Precision Testing. *IEEE Transactions on Software Engineering*, 43(10), 975–994. doi:10.1109/tse.2016.2642956