

ISSN 2063-5346



## A MIXED APPROACH FOR AN EFFECTIVE SOFTWARE TESTING PROCESS

Dr.S.Shakila Anjum<sup>1</sup>, Dr.A.K.Ashfauk Ahamed<sup>2</sup>

---

Article History: Received: 01.02.2023

Revised: 07.03.2023

Accepted: 10.04.2023

---

### Abstract

This research paper attempts to explore the factors that make the relevance of manual testing justified in the context of the widespread use of automation testing in software industry. The major factors which are considered here in this research paper are **Cost, Time, Efforts and Quality**. Through a meticulous study carried out by a researcher by comparing the data and information from various sources, it is identified that automation testing and manual testing cannot be effectively used in isolation but in consolidation both the techniques can complement each other in overcoming their individual demerits to bring about the seamless results.

**Keywords:** Software testing, Quality, Cost, Time, Automation testing

---

<sup>1</sup>Assistant Professor, Department of Computer Applications, B. S. AbdurRahman Crescent Institute of Science and Technology, Chennai, India.

<sup>2</sup>Assistant Professor, Department of Computer Applications, B. S. AbdurRahman Crescent Institute of Science and Technology, Chennai, India.

ashfauk@gmail.com

DOI:10.31838/ecb/2023.12.s1-B.378

---

## 1. Introduction

In software development life cycle, testing is one of the important phases that every IT company follows. Software testing plays a vital role in delivering a quality deliverable. Whatever might be the software development methodology, testing phase is covered for sure to complete the development life cycle. Functional testing or Black box testing can be performed to meet the functional requirements given by the customer. Before automation came into existence, manual testing was performed for testing the software. But now most frequently automation testing is used to perform the regression testing of any software which being developed.

### 1.1 Manual testing

Functional testing was performed by writing the test cases manually in an excel sheet 2 decades ago. After which test management tools came into existence and all the manual test cases were stored or uploaded in the tool for tracking. Defects were mapped to the testcases and in turn to the requirements in the tool itself for better traceability. Then for automation of this manual test cases, automation script was written based on the manual test cases.

### 1.2 Automation testing

Automation testing has been the order of the day all across the IT companies globally. Most of the IT companies have already started adopting automation testing widely. With the aim of reducing the time and cost and increasing the quality of testing of the deliverable, but on the contrary automation testing has been proving to be ineffective in certain cases. When these problems concerning the automation testing were probed through various sources it became evident that automation testing has certain inherent characteristics because of which they fail to meet the expected deliverables.

### 1.2.1 Prerequisite for Automation Testing

Automation can be effective only for stable applications or applications that are partially stable [1]. Due to the fact that test cases should also be developed using the automation tools which requires development knowledge. So when the application code changes, the testing script should also be changed, this is inevitable. Hence the automation testing cannot be started immediately once the application code is ready but it requires some time to develop the automation script. Due to this reason, automation can be effectively utilized for regression testing which is the next iteration of testing.

### 1.3 Comparing the Key factors of Automation and Manual Testing

Automation test engineers require coding skills along with manual testing skills. Hiring an automation engineer is expensive and as well as difficult. In manual testing, all types of testing is possible with an ease. Manual tester can be trained on domain/functional knowledge related to the application under test. Manual tester has extensive functional knowledge and they prepare test cases based on the requirement document and the functional knowledge they have. This test case document will be the input of the automation testing based on this input automation testers write their script.

Documents like letters and reports in the pdf formats, data output in excel sheets, complex functionalities and few complex images are more efficient when tested manually. When the above mentioned features are automated it takes lot of time when compared to manual testing. When compared with manual testing the time difference is very drastic. Also, the first iteration can be only tested manually because once development get completed the available time for developing the automation script will not be sufficient hence with the available time manual testing can be completed easily. Hence

automation testing is highly effective for regression testing and manual testing is highly workable for changing requirements and unstable application.

## 2. Need for a mixed approach

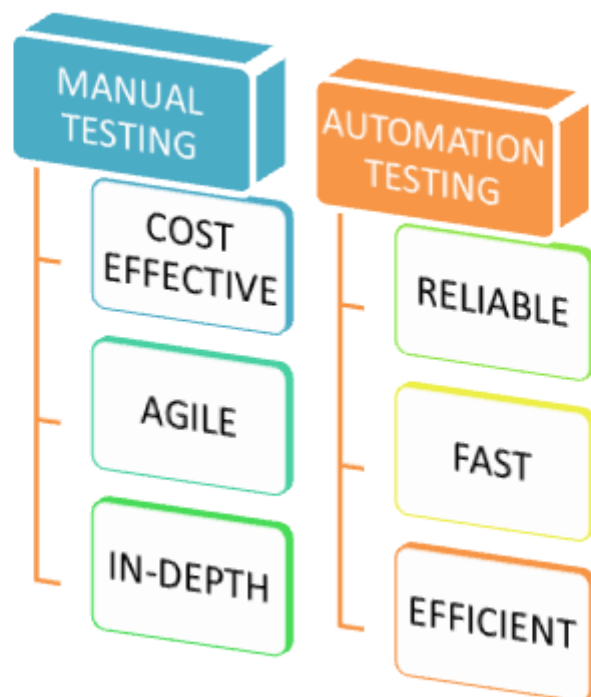
### 2.1 Reasons substantiating mixed approach towards software [2] testing

1. Efforts required for automation is more when compared with manual.
2. Cost of the automation resources are more when compared with manual.

3. Less Time is taken in manual testing for testing complex functionalities/reports/Letters/Excel verification/Images/pdf formats etc., when compared to automation testing [3] [4].

4. In terms of quality of testing [5], test cases are reviewed by peers and additional test cases are added based on the review comments. But there is no such review for automation test scripts [6].

Hence to balance both automation and manual testing we can utilise a mixed approach towards software testing.



## 3. Analysis

Advantages of the factors of using a mixed approach are highlighted.

S.No.	Factors	Manual Testing	Automation Testing	Mixed Approach
1	<b>Cost</b>	Less cost	More Cost	*Medium Cost
2	<b>Time</b>	More Time	Less time	*Average
3	<b>Quality</b>	Good Quality	Good Quality	Best Quality
4	<b>Efforts</b>	More efforts	Medium efforts	Less efforts
5	<b>Coverage</b>	Full	Partial	Full

\* **Note:**

- Reusability and Cost-effectiveness will be achieved when automation is completed.
- Cost is more for automation initially.
- Time taken will be reduced after automation is completed.

Other factors that lead to mixed approach are as follows:

- Automation testing [7] is only feasible for bigger size projects or applications
- Automation can only be performed for stable applications or applications which are partially stable.
- Automation testing is not required [8] for smaller projects
- Change is constant hence stability of the app cannot remain the same for indefinite period of time.
- Cost of the automation is more when compared to manual testing.
- Manual testing is the base for automation testing. Manual tester [9] has the functional knowledge but automation tester has programming knowledge.
- Automation cannot be done [10] for verification of Reports, Letters, Images and pictures.
- For Complex calculations – automation doesn't play a crucial role.
- Excel sheet data verification – becomes complex when automated.
- Any changes to the application under testing require change to the automation testing [11] code which consumes lot of time when compared to manual testing [12].

#### 4. Conclusion

Even for the development of automation testing algorithms, knowledge and expertise acquired through manual testing process will serve as a base. Further to this, it is from only manual testing that the domain and functional requirements can be transferred and transformed into automation testing algorithms and cases. .

Hence it can be inferred from the above facts that manual testing will go hand in hand with automation testing, and will be instrumental in evolving automation testing.

#### References:

1. Kumar, D., & Mishra, K. K. (2016). The impacts of test automation on software's cost, quality and time to market. *Procedia Computer Science*, 79, 8-15.
2. Black, R. (2016). *Pragmatic software testing: Becoming an effective and efficient test professional*. John Wiley & Sons.
3. Kochhar, P. S., Xia, X., Lo, D., & Li, S. (2016, July). Practitioners' expectations on automated fault localization. In *Proceedings of the 25th International Symposium on Software Testing and Analysis* (pp. 165-176).
4. Shamshiri, S., Just, R., Rojas, J. M., Fraser, G., McMinn, P., & Arcuri, A. (2015, November). Do automatically generated unit tests find real faults? an empirical study of effectiveness and challenges (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 201-211). IEEE.
5. Dudekala, R., Katam, M., Kai, P. and Mika, M. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In: *International Workshop on Automation of Software Test Co-located with the International Conference on Software Engineering*.
6. Naujoks, F., Hergeth, S., Wiedemann, K., Schömig, N., & Keinath, A. (2018, September). Use cases for assessing, testing, and validating the human machine interface of automated driving systems. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 62, No. 1, pp.

- 1873-1877). Sage CA: Los Angeles, CA: SAGE Publications.
7. Riley, V. (2018). Operator reliance on automation: Theory and data. In *Automation and human performance: Theory and applications* (pp. 19-35). CRC Press.
  8. Wiklund, K., Eldh, S., Sundmark, D., & Lundqvist, K. (2017). Impediments for software test automation: A systematic literature review. *Software Testing, Verification and Reliability*, 27(8), e1639.
  9. Rodrigues, A., & Dias-Neto, A. (2016, September). Relevance and impact of critical factors of success in software test automation lifecycle: A survey. In *Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing* (pp. 1-10).
  10. Garousi, V., & Pfahl, D. (2016). When to automate software testing? A decision-support approach based on process simulation. *Journal of Software: Evolution and Process*, 28(4), 272-285.
  11. Rösch, S., Ulewicz, S., Provost, J., & Vogel-Heuser, B. (2015). Review of model-based testing approaches in production automation and adjacent domains—Current challenges and research gaps. *Journal of Software Engineering and Applications*, 8(09), 499.
  12. Campos, J. C., Fayollas, C., Martinie, C., Navarre, D., Palanque, P., & Pinto, M. (2016, June). Systematic automation of scenario-based testing of user interfaces. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 138-148).