International Journal of Research in Science And Technology http://www.ijrst.com/

(IJRST) 2012, Vol. No. 2, Issue No. II, Apr-Jun **ISSN: 2249-0604**

COMPARISON OF MANUAL AND AUTOMATION TESTING

VIVEK KUMAR

ABSTRACT

Manual testing is a testing technique, where test engineer test the software manually. The test engineer, who carries out all the test cases and executes on the application manually, step by step and indicates whether a particular step was accomplished successfully or whether it failed, performs manual testing,. In this paper we describe importance of manual testing in age of automated testing because I think that manual testing is a base of automated testing

Keyword: *manual testing, automated testing*

INTRODUCTION

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user, and use most of all features of the application to ensure correct behavior.Large scale engineering projects that rely on manual software testing follow a more rigorous methodology in order to maximize the number of defects that can be found. Manual testing plays an important role in applications where functionalities change quite often. Manual testing is essential, as 100 percent automation is not possible in real-time environment. In some cases manual testing holds upper hand over the automation.

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user, and use most of all features of the application to ensure correct behavior. To ensure completeness of testing, the tester often follows a written test plan that leads them through a set of important. It is performed by the tester who carries out all of the actions on the tested application manually, step-by-step and indicates whether a particular step was accomplished successfully or whether it failed. Manual testing is always a part of any testing effort.
It is the oldest and most rigorous type of software testing. It is requires a tester to perform manual test operations on the software without the help of Test automation.

A manual tester would typically perform the following steps for manual testing:

1. Understand the functionality of program
2. Prepare a test environment.
3. Execute test case(s) manually

4. Verify the actual result
    1. Record the result as Pass or Fail
    2. Make a summary report of the Pass and Fail test cases
    3. Publish the report
    4. Record any new defects uncovered during the test case execution

At the time of manual testing tester need only test case and with the information how to execute those test case. Test Complete gives you an opportunity to create and manage manual tests when testing your application. After adding a Manual Testing project item to your project, you can create a collection of steps to be performed when the application is being tested, with a description and detailed instructions for each step. Test case is also written for all type of testing according to test strategy of test plan. Test engineer writes test cases on base of design document of the software. Manual testing start once the testing team receives a build from the development team.

Stages: - We follow several stages of manual testing. They are:
Unit Testing This initial stage in testing normally carried out by the developer who wrote the code and sometimes by a peer using the white box testing technique.

Integration Testing This stage is carried out in two modes, as a complete package or as a increment to the earlier package. Most of the time black box testing technique is used. However, sometimes a combination of Black and White box testing is also used in this stage.

System Testing In this stage the software is tested from all possible dimensions for all intended purposes and platforms. In this stage Black box testing technique is normally used.

User Acceptance Testing This testing stage carried out in order to get customer sign-off of finished product. A 'pass' in this stage also ensures that the customer has accepted the software and is ready for their use.

COMPARISON TO AUTOMATED TESTING

Test automation may be able to reduce or eliminate the cost of actual testing. A computer can follow a rote sequence of steps more quickly than a person, and it can run the tests overnight to present the results in the morning. However, the labor that is saved in actual testing must be spent instead authoring the test program. Depending on the type of application to be tested, and the automation tools that are chosen, this may require more labor than a manual approach. In addition, some testing tools present a very large amount of data, potentially creating a time consuming task of interpreting the results. From a cost-benefit perspective, test automation becomes more cost effective when the same tests can be reused many times over, such as for regression testing , and when the results can be interpreted quickly.Test automation has become a necessity mainly due to shorter deadlines for performing test activities, such as regression testing, performance testing, and load testing Test automation may be able to reduce or eliminate the cost of actual testing. A computer can follow a rote sequence of steps more quickly than a person, and it can run the tests overnight to present the results in the morning.

If you're testing a compiler, automation might be only a little more expensive than manual testing, because most of the effort will go into writing test programs for the compiler to compile. Those programs have to be written whether or not they're saved for reuse. Suppose your environment is very congenial to automation, and an automated test is only 10% more expensive than a manual test. (I would say this is rare.) Creating an automated test is usually more time-consuming (expensive) than running it once manually. The cost differential varies, depending on the product and the automation style. On the other hand, its disadvantage is this: No human insight. During automated testing, the machine only executes what the conditions of the pre-set steps are. It has no capacity to think outside of the pre-set steps and do exploratory or monkey testing

Conversely, graphical user interfaces whose layout changes frequently are very difficult to test automatically. There are test frameworks that can be used for regression testing of user interfaces. They rely on recording of sequences of keystrokes and mouse gestures, then playing them back and observing that the user interface responds in the same way every time. Unfortunately, these recordings may not work properly when a button is moved or relabeled in a subsequent release. An automatic regression test may also be fooled if the program output varies significantly (e.g. the display includes the current system time). In cases such as these, manual testing may be more effective.

The benefit of using Manual Testing are:-

- Manual testing is cost effective as compared with Automation testing.
- It allows the tester to perform more Ad-hoc testing (random testing). More bugs are found via Ad-hoc testing than via automation. And the more time a tester spends playing with the feature, more bugs can be found.
- One major advantage of manual testing is this: Since a person thinks, therefore, the tester will find ways and means on how to best explore the product aside from the pre-set ways presented to him/her. In short, a person can do exploratory or monkey testing.
- Manual testing can be use for both small and big projects.
- We can easily add and remove the test cases according to project movements.
- Fresh tester can understand very easily the process of manual testing.
- Manual testing is more reliable than automation testing (in many cases automated not cover all cases).
- Manual testing is not related with any programming languages.
- It is covered in limited cost.

Drawbacks of manual testing

- Manual testing can be very time consuming as everything has to be done manually.
- More human involvement.
- In manual testing, the concept of repeatability not so accurate.
- GUI object size difference and color combination etc is not easy to find out in manual testing.
- Actual load and performance is not possible to cover in manual testing for large number of users.
- Running test manually is very time consuming job.

CONCLUSION

One of the main goal of this work is to understand the importance of manual testing.. There is no complete substitute for manual testing. Manual testing is extremely important crucial for testing software applications more thoroughly. Manual testing is always a part of any testing effort. It is especially useful in the initial phase of software development, when the software and its user interface are not stable enough, in beginning the automation does not make sense. Test automation is expensive and it is an addition, not a replacement, to manual testing.

REFERENCES

1. ANSI/IEEE 829-1983 IEEE Standard for Software Test Documentation
2. Craig, Rick David; Stefan P. Jaskiel (2002). *Systematic Software Testing*. Artech House. p. 7. ISBN 1580535089.
3. Itkonen, Juha; Mika V. Mäntylä and Casper Lassenius (2007). "Defect Detection Efficiency: Test Case Based vs. Exploratory Testing". *First International Symposium on Empirical Software Engineering and Measurement*.http://www.soberit.hut.fi/jitkonen/Publications/Itkonen_Mäntylä_Lassenius_2007_ESEM.pdf. Retrieved 2009-01-17.
4. http://softwaretestinginterviewfaqs.wordpress.com/category/testing-in-stages/
5. Mosley, Daniel (2002). *Just Enough Software Test Automation*. Prentice Hall. p. 27. ISBN 0130084689.

6. Bach, James (1996). "Test Automation Snake Oil". *Windows Technical Journal* **10/96**: 40–44.http://www.satisfice.com/articles/test_automation_snake_oil.pdf. Retrieved 2009-01-17

7. Manual testing - Wikipedia, the free encyclopedia available at http://en.wikipedia.org/wiki/ manual testing

8. Kolawa, Adam; Huizinga, Dorota (2007). *Manual Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press. p. 74. ISBN 0470042125.http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470042125.html

9. http://tosca-testsuite.com/Newsletter/Apr11/En/Newsletter.html

10. Elfriede Dustin, et al.:*manual software testing*. Addison Wesley, 1999, ISBN 0-20143-287-0

11. Elfriede Dustin, et al.: *Implementing manual Software Testing*. Addison Wesley, ISBN 978-0321580511

12. Mark Fewster & Dorothy Graham (1999). *Software Test case*. ACM Press/Addison-Wesley. ISBN 978-0201331400.

13. Roman Savenkov: *How to Become a Software Tester*. Roman Savenkov Consulting, 2008, ISBN 978-0-615-23372-7

14. Exploratory Testing, Cem Kaner, Florida Institute of Technology, *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL, November 2006

15. Software Testing by Jiantao Pan, Carnegie Mellon University

16. Leitner, A., Ciupa, I., Oriol, M., Meyer, B., Fiva, A., "Contract Driven Development = Test Driven Development - Writing Test Cases", Proceedings of ESEC/FSE'07: European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering 2007, (Dubrovnik, Croatia), September 2007

17. Software errors cost U.S. economy $59.5 billion annually, NIST report

18. ***a b*** Myers, Glenford J. (1979). *The Art of Software Testing*. John Wiley and Sons. ISBN 0-471-04328-1.

19. Company, People's Computer (1987). "Dr. Dobb's journal of software tools for the professional programmer". *Dr. Dobb's journal of software tools for the professional programmer* (M&T Pub) **12** (1–6): 116.http://books.google.com/?id=7RoIAAAAIAAJ.

20. Gelperin, D.; B. Hetzel (1988). "The Growth of Software Testing". *CACM* **31** (6). ISSN 0001-0782.

21. *until 1956 it was the debugging oriented period, when testing was often associated to debugging: there was no clear difference between testing and debugging.* Gelperin, D.; B. Hetzel (1988). "The Growth of Software Testing".*CACM* **31** (6). ISSN 0001-0782.