

Introduction To Adhoc Testing

Mr. Neeraj Chhabra

Abstract—This paper gives an introduction to Adhoc Testing. Adhoc testing is very useful technique in manual testing as well as automated testing. Some approaches are covered for Adhoc testing, which are very helpful in this type of testing. There are some advantages and drawbacks, also covered in this paper.

Index Terms— software testing, adhoc testing, adhoc testing, monkey testing, adhoc testing approach, test techniques, test methods

1 INTRODUCTION

Adhoc Testing is also known as Expert Testing/Monkey Testing. Adhoc Testing is an effective technique to test a system without any defined scenario or test cases. It is done without any planning or documentation. In general, while testing any application or system, we have to follow a certain scenario for executing test cases. But in Adhoc Testing, There are not any specific test cases for execution. So, a tester is free to test the system/application to execute any scenarios, what he thinks. Tester is not bound or has no limitation on execution. He can do anything, what he wants. It is a powerful mechanism, to found as many defect as possible during the testing of system or application. It is up to tester that how can he looks the system to explore that, while adhoc testing.

2 APPROACHES FOR ADHOC TESTING

There is no defined approach for Adhoc testing. It depends on Experience and thinking of the tester that how he thinks and looks the SUT. Experienced tester can find many defects through Adhoc testing because he knows, the common scenarios, which causes defect in a SUT. One can follow some approaches, which can be helpful to a tester in adhoc testing.

A. Find Defect Cluster

Defect Cluster is an area of system, where the possibility of occurring defect, is maximum. While Adhoc testing, a tester should try the scenarios, which can find the defects in particular area. One can find defect cluster by analyzing the test history of test cases, executed before. After that, One can get idea of, in which area, defect may occur more.

B. Interaction with other components

Interaction of one application to another is a common scenario, where the fault may occur. While Adhoc testing, one can interact one component or module to other component or module, so that he can find the unexpected behavior in the SUT.

C. Think like a developer

While Adhoc testing, think like a developer and look the SUT, what you would do, while developing the current module/component. Then you can get answer that where the robustness or correctness of SUT is lacking. You can break the code and may get a fault.

D. Good knowledge of SUT

A tester should have good knowledge of SUT. If a tester has good knowledge of SUT, he can think and execute more scenarios. Adhoc testing requires more and more scenarios to execute. If you execute more scenarios, you may get more faults in SUT.

E. Use your Experience

While Adhoc testing, an experience tester can find more faults than a fresher. A tester get more and more knowledge, when he do adhoc testing, because he gets the knowledge of similar system and can use this experience in other project.

F. Test in sessions

Test the application or software part by part i.e., one functionality at a time. This provides better focus and understanding of the problems, if any.

G. Learn to use different tools

At times certain defects may be brought to light by debuggers, profilers, and task monitors. So a thorough knowledge about these tools can make matters easy for the testers.

H. Record your findings

Make a record of what bugs you found, where in the application they appeared, and the function. These records helps the developer and also testers of future applications. Also record what bugs you didn't find. That may seem odd, but there needs to be a record of what worked.

-
- Neeraj Chhabra is currently working as a SQA Engineer in Nvidia Corporation, India, PH- 08097212586. E-mail: nchhabra@nvidia.com

3 WHEN TO START

Adhoc testing is a free form technique. It should start after execution of formal test suits. It is a right time to approach for Adhoc testing. After cycle execution, system is ready for facing the random scenarios, which can find more faults than formal test case execution. We can do Adhoc testing, while system testing or feature testing. Adhoc testing should be used more and more in Testing Life Cycle. It gives the great support to testing team to execute more and try different scenarios on SUT. Adhoc testing finds a place during the entire testing cycle. Early in the project, Adhoc testing provides breadth to testers understanding of your program, thus aiding in discovery. In the middle of a project, the data obtained helps set priorities and schedules. As a project nears the ship date, Adhoc testing can be used to examine defect fixes more rigorously.

4 BENEFITS

Adhoc Testing is performed without any planning or documentation, so tester has full authority to perform any scenario. He is not bound with any test case. There are many benefits, which increase the effectiveness of Testing.

- ✓ Tester has no test case, so his only aim to find faults by any means
- ✓ It is effective, when there is a time limitation for testing the SUT
- ✓ Tester can find more defects than a formal cycle execution
- ✓ Adhoc testing often used as a complement to other types of testing
- ✓ Increase code coverage Benefits

5 DRAWBACK

Adhoc testing is not structured. That's why, one has to remember all the scenarios, which he performs. Tester has to think a lot for interacting components and work hard for finding different scenarios. The main drawback of Ad-Hoc testing is unstructured and inconsistent level of performance. Also the testing doesn't go by rules. It is similarly to an error guessing methodology. It doesn't have test cases. There are some drawback mentioned below -

- Unorganized testing
- possibility of not covering of major functionality
- can't assume the time to work on the application
- cant show the proof that so and so % of work has been covered
- sometimes cant reproduce the bugs which I found.

6 CONCLUSION

Adhoc Testing is done without any test plan or documentation. So, one doesn't need to wait for any cycle execution or phase. He can easily test the SUT by his knowledge and experience. Ad-hoc testing helps in deciding the scope and duration of the various other testing and it also helps testers in learning the application prior starting with any other testing. It is the least formal method of testing. One of the best uses of ad hoc testing is for discovery. Reading the requirements or

specifications (if they exist) rarely gives you a good sense of how a program actually behaves. Even the user documentation may not capture the "look and feel" of a program. Ad hoc testing can find holes in your test strategy, and can expose relationships between subsystems that would otherwise not be apparent. In this way, it serves as a tool for checking the completeness of your testing. This methodology can be very effective in

- Finding Missing cases
- Determining Priorities of Other testing activities
- Increase Code Coverage

Adhoc Testing can be very effective in any project. Main thing is to execute this at the right time and right place in testing cycle.

REFERENCES

- [1] <http://www.testingcraft.com>
- [2] http://www.en.wikipedia.org/wiki/Ad_hoc_testing
- [3] http://www.testingbrain.com/BLACKBOX/BLACK_BOX_Ad-hoc_Testing.html