



# Testing Competence: Scripted Testing Versus Exploratory Testing

<sup>1</sup>Vandna Dahiya, <sup>2</sup>Sushila

<sup>1</sup>Maharshi Dayanand University, Haryana, vandnadahiya2010@gmail.com

<sup>2</sup>Food Corporation of India, sushiladahiya35@gmail.com

## ABSTRACT

Automation of software testing is a complex problem. Some of the developing organizations for software product continue to shrink rapidly. One of the reasons behind is the testing strategy adopted by the testers, which sorts most of the gap. Today, most of the testers adopt scripted testing as a strict testing strategy and emphasis minimum or no stress on exploratory testing. Scripted testing surely has some advantages but suffers from monotonous way of testing where as exploratory testing is a free style of testing and allows discovering of new problems. It influences our domain knowledge and helps in getting deeper insights of software product by employing human creativity. This paper presents the importance of ensuring a fine balance between both the ways of testing to increase the effectiveness of testing function.

ISSN 2454-308X



9 770024 543081

**Keywords: Scripted testing, exploratory testing, testing strategies.**

## I. INTRODUCTION

Testing methodologies have undergone several changes in which the scripted testing has taken preference over all. Quality assurance team focuses on whether the application under development fulfills all the requirements or not rather than trying to evaluate the behavior of the application. Scripted testing is such a technique, which follows a script written by the testers themselves or someone else. No deviation is allowed from the path of script which includes documented test cases and test steps. Tester has to color within in the lines only. He has to follow all the instructions written in script with the highest possible degree and then to report. On the other hand, exploratory testing is a free style testing and relies on the tester to develop the testing path 'on the go'. It allows a tester to go where an end user might go rather than where a script tells them to go. It gives degree of freedom as well as responsibility to the tester to continually optimize the quality of testing by generating the test cases as needed and then further exploring the new test cases as required with the gaining in experience and creativity of the tester. It is a test approach that can be applied to any test technique, at any stage in the development process where the key is engagement of the tester, and the tester's responsibility for managing his time.

## II. RELATED WORK

Cem Kaner introduced the concept of efficiency measurement for scripted as well as exploratory testing. Various other scholars gave their views and explained the roles of both type of testing. Andrew Thompson mentioned the factors to keep in mind while choosing an approach among scripted and exploratory such as domain knowledge of the tester, complexity of system, documentation level, deadlines, verification, risk levels etc. These factors can be described as below-



*Domain knowledge of the tester-* It depends on the understanding of the tester, how well he gets the operation of the software and the processes that would be supported by the system.

*Complexity of system-* Complexity of the system affects the appropriateness of a particular testing strategy. Detailed test planning is required if the dependency level is high among the modules.

*Documentation level-* Scripted testing flows with the level of documents. If document specifications and functional specifications are deficient than scripting testing cannot be applied. If documents do not exist, than exploratory testing can be applied more suitably.

*Time limits and deadlines-* Time limit specifies the suitability of testing strategy. If there is lead-in time, test design can be conducted before execution. Testing can be documented if specifications are available otherwise; there is a need to start with exploratory testing.

*Resource availability-* The availability of resources in person-days also affects the strategy needed.

*Coverage-* Some systems and applications are such that the level of test coverage must be accurately measurable and demonstrable.

*Skills required-* Strategy of testing depends on the skills of the testing team.

*Verification-* Verification requires something to compare against. Formal script for testing requires the documentation whereas exploratory testing can be done with the expectations of engineers.

*Risk Levels-* Acceptance of risk level is dependent on the criticality of business. Exploratory testing has the risk of not covering all the aspects of testing.

In this paper, efforts are done to extend their views towards pros and cons of both the testing strategies.

### **III. CURRENT ROLE OF SCRIPTED TESTING AND EXPLORATORY TESTING**

Scripted testing is the most used strategy today with a very less scope for exploratory testing. There are various reasons responsible for it like the advancement in technology, which causes a huge requirement in testing volumes of modules daily. The huge costs and need to make efficient software emphasizes on need of artificial testing with tools, which is again scripted testing. Scripted testing is a more preferable testing over exploratory and is widely adopted in industry. Scripted testing allows a better control over budget and the outcome of the quality. As the size of applications increased, it became necessary to adopt a more process-oriented approach to test these complex systems, which can be offered by scripted testing. Qualitative documentation is one feature of scripted testing which is often not possible with scripted. It gives better scope to planning and predictability of costs, time and efforts. This feature comes with the disadvantage of inflexibility. Whenever there is any change in the test scenario, there is a need to change the script as well else it will not be of use. Scripted testing is not adaptive to changes.

Exploratory testing on the other hand is a free style testing which shows how the software actually works. Quality here is dependent upon the skills of tester, the more the tester knows about the product and different test methods, the better the testing will be. In scripted testing, test cases are designed in advance, which includes both the steps and the input and expected results. Whereas when performing exploratory testing, results may be predicted and expected;



others may not. The tester configures, observes, and evaluates the product and its behavior, investigating the result, and reporting information that seems likely to be a bug or an issue. The testers try to evaluate the behavior of the application under development through extensive application, business knowledge, and innovative methods simulating possible user behavior while managing time. Here, roles are not divided for testing. Tester thinks of test cases himself, executes them, based on the results, and executes the new tests. But due to lack of documentation, there is least scope of planning; test cases are in mind of tester. We cannot tell how much work is done and how much is left for testing.

#### IV. NEED AND SIGNIFICANCE OF EXPLORATORY TESTING

Exploratory testing gives chances to explore the area, which are not thought by the developers or have been missed in script completely. Test charters are used which are important element that give shape and form to an exploratory testing model. Exploratory testing may uncover many unexpected shortcoming of a program. It thus makes the release as bulletproof as possible. Scripted testing is monotonous and boring at times when a tester is engaged with same kind of test suites, steps and results. It also becomes a weeding paradox where only similar kind of error is found out with the same set of scripts every time. They will found same kind of bugs every time. It is like classroom teaching where a teacher is concerned mostly with the grades of students and not with what they actually learned. This approach thus limits learning and so does in testing. No deviation from the script is possible. Test designer are needed to be highly skilled and are costly person. Testers can be appointed as beginners or fresher who execute the scenarios.

The main advantage of exploratory testing is that less preparation is needed. When the important bugs are found, we can quickly remove them, and further can focus in their direction more. This approach is more intellectual as combing the inference of already executed results. The previous results guide for future testing. Tester can move in target rich environment. All this process accelerates bug removal. Also, this type of testing is very flexible and adaptive to changes. Test suites are in mind of tester. He can change these as soon as needed. Exploring provides non-functional requirements validation and performance profiling. Close collaboration of testing and developing team helps in feedback on effects. Annual cost of production support is reduced.

We can see both the strategies with a simple example of testing of a small module and generating test cases for both the testing styles.

Suppose we have a small program in C to find the roots of a quadratic equation  $ax^2+bx+c$

Input range [5,95]

Then,

Roots are real if  $(b^2-4ac) > 0$

Roots are imaginary if  $(b^2-4ac) < 0$

Roots are equal if  $(b^2-4ac) = 0$

Equation is not quadratic if  $a=0$

Using scripted approach for boundary value analysis, we can generate 13 test cases as shown in table 1.



Table 1 Expected outputs for a quadratic equation

Test Case	m	l	n	Expected Output
1	5	50	50	Not quadratic
2	6	50	50	Real roots
3	50	50	50	Imaginary roots
4	94	50	50	Imaginary roots
5	95	50	50	Imaginary roots
6	50	5	50	Imaginary roots
7	50	6	50	Imaginary roots
8	50	94	50	Imaginary roots
9	50	95	50	Equal roots
10	50	50	5	Real roots
11	50	50	6	Real roots
12	50	50	94	Imaginary roots
13	50	50	95	Imaginary roots

This is the scripted approach we design for the testing of inputs for this module. But while testing, we may come across various other combinations which this testing will miss like what if there are both the input with minimum value of interval or what if both having maximum value of interval. These types of test cases then generated in the mind of tester and execute simultaneously. This give more error free module rather than scripted testing alone.

#### V. IMPLEMENTING A COMBINED STRATEGY

Exploratory testing is not perfect as while it can explore the most uncovered areas, it can miss very simple problem in lack of direction of script. Test cases cannot be reviewed in advance. Most testing professionals focus on scripted test execution; the tester's instinct pushes her/him to explore the system completely. Structured methods and initiatives can improve the efficiency of exploratory testing when carefully combined with mainstream scripted execution. The factors such as struggle of IT sector with lack of subject matter experts and increasing cost of QA suggest a need of combining both the strategies as follows. Testers are encouraged to think beyond their test scripts thanks to the focus of IT organizations on quality of applications. In the perfection, we could see the mixture of both the testing which allows us to design the testing with as many angles as possible and thus making the release as sound as possible. Exploratory testing gives facility of testing the 'wild' and can touch the areas never bothered by developers; but without restraints, it can miss the smallest bugs like if we bother too minutely to something, we can miss the most minor issue like anything else in the life. A good collaboration of both the testing strategies thus needed.

#### VI. CONCLUSION

The role of exploratory testing is although more important yet it is not suggested to use this approach in isolation due to its instability of documentation. The high demand of testing huge



number of volumes of software also controls the behavior of testing. The large and complex business modules cannot withstand a risk of cost and time; thus a preplanning and predictability as well as review is needed of whole test scenario. At the end of the day, economics dictate what is feasible is the concept of introducing quality into the product and pushing for early detection of defects is forcing QA teams to develop test strategies, which adopt exploratory testing and integrate it with scripted testing seamlessly. It is also shifting the focus towards validation at the level of business transactions, grooming of software matter experts who will have ability to adopt exploratory testing methods effectively. Training and awareness is also a key contributor for the success of exploratory testing methods in this ever-growing tester's community. QA managers must implement exploratory testing techniques as an additional strategy, combined with scripted testing, to improve the overall quality of applications.

## REFERENCES

1. Cem Kaner, James Bach, "The Nature of Exploratory Testing", [online] Available: <http://www.testineducation.org/a/nature.pdf>
2. James Bach, [http://www.satisfice.com/articles/what\\_is\\_et.shtml](http://www.satisfice.com/articles/what_is_et.shtml)
3. Gregory Mooney, <http://blog.smartbear.com/testing/exploratory-vs-scripted-testing-one-or-the-other-or-both/>
4. Justin Hunter and Rikard Edgren, "Exploratory Testing vs Scripted Testing: Rich Terminology," [online], <http://thetesteye.com/blog/2009/11/exploratory-testing-vs-scripted-testing-rich-terminology/>
5. Andrew Thompson "How to Choose between Exploratory and Scripted Testing"[article] <https://www.stickyminds.com/article/how-choose-between-exploratory-and-scripted-testing>
6. [http://en.wikipedia.org/wiki/Test\\_script](http://en.wikipedia.org/wiki/Test_script)
7. Cem Kaner and James Bach "The Nature of Exploratory Testing" <http://www.testineducation.org/a/nature.pdf>
8. TtCalc : <http://ttcalc.sourceforge.net/>
9. <http://www.intexsoft.com/blog/item/73-scripted-testing-vs-exploratory-testing.html>
10. Defect Detection Efficiency: Test Case Based vs. Exploratory Testing <http://www.soberit.hut.fi/jitkonen/Publications/Itkonen-M%E4ntyl%E4-Lassenius-2007-ESEM.pdf>
11. Mixing Scripted and Exploratory Testing <http://testertested.blogspot.com/2007/03/mixing-scripted-and-exploratory-testing.html>
12. E. Kowalczyk, A. M. Memon, M. B. Cohen, "Piecing together app behavior from multiple artifacts: A case study", 2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE), pp. 438-449, Nov 2015.
13. Mixing Scripted and Exploratory Testing, [online] Available: <http://testertested.blogspot.com/2007/03/mixing-scripted-and-exploratory-testing.html>.
14. Cem Kaner, "Defining Exploratory Testing", [online] Available: <http://www.satisfice.com/kaner/?p=42>.