The Taxonomy of Factors influencing Effectiveness of Exploratory Testing

Rashmi N
Dayananda Sagar College of Engineering
S.M. Hills, K.S. Layout
Bangalore, India.

Suma V
Dayananda Sagar College of Engineering
S.M. Hills, K.S. Layout
Bangalore, India.

ABSTRACT
In today's world as our lives and livelihood depend largely on the software systems, malfunctioning of these software systems is highly intolerable. Hence, the delivery of high quality software becomes a big challenge for IT industries. However, various strategies are followed by IT industries to deliver a quality product within the estimated time and budget. Software Testing is one of these strategies that contribute towards the quality of the product. The two main approaches of Software Testing are Test Case Based Testing and Exploratory Testing. It has been empirically proved that the defect detection efficiency of exploratory testing is as good as the traditional test case based testing. Hence, this paper aims at finding the factors that influence the defect detection effectiveness of Exploratory Testing.

General Terms
Software Quality, Software Testing, Defect Detection Effectiveness, Test Case Based Testing, Exploratory Testing et. al.

1. INTRODUCTION
Since software plays a vital role in every day-to-day activities, development of quality software is one of the primary focus of any organization. Defect free software is yet again one of the modes through which quality can be accomplished in software products [1]. Effective defect management is therefore achieved through efficient testing strategies [2].

Software Testing is mainly classified into Manual and Automation Testing [3]. Though automation testing has gained popularity, it is observed that more serious and interesting defects can be found through Manual Testing [4]. Manual testing is further be classified into Test Case Based Testing and Exploratory Testing. Test Case Based Testing is an approach of testing where a set of predefined test cases are designed during the test planning stage and executed in the later stages of software development [5]. The test cases are designed based on the requirement specification using several test case design techniques such as equivalence partitioning, boundary value analysis technique etc [6][7]. This approach of testing emphasizes decidability, predictability and accountability [8]. Some of the benefits of the Test Case approach are as follows.

1. Each test case designed and optimized for important quality attributes such as power, credibility etc.
2. Test cases are reviewed by the stakeholders.
3. Test cases are reused.
4. Comprehensiveness of the test cases are known to testers.

In spite of the above benefits the major drawback of Test Case Based Testing approach is the amount of cost required for this approach is big. In addition, specifying full set of test cases before their execution may lead to project failures. This is because the risk profile such as change in the requirement specification, change in the platform, change in the competition and a change in the user expectation etc. associated with a project evolves over a period of time. As a result, the test cases should also change. But the test cases remain the same. Executing the same set of tests again and again may not provide much information about the quality of the product under test [8].

However, exploratory testing is based on simultaneous learning, design and execution of the test cases by the testers. Exploratory Testing emphasizes learning and adaptability of the test cases [8].

Exploratory Testing was introduced by Cem Kaner [8]. The benefits of Exploratory testing include

1. Increased effectiveness in terms of the defect count and the impact of detected defects.
2. Simultaneous learning ensures that the tester is not following any pre-specified test scripts and is actively learning about the system and gaining knowledge about the behavior and failures of the system under test. This helps the testers come up with better and powerful tests.
3. Reduced documentation preparation before executing test cases.
4. No need of comprehensive requirements document since exploratory testing is based on the experience and knowledge of the product gained from various other sources.
5. Enables rapid feedback from testing to both the developers and testers[8].

Exploratory testing is found to be significant in various situations. First, when a rapid feedback or learning of the product is required. Second, when there is either no or hardly any time left for systematic testing approaches. Third, when there is a need to investigate the status of a particular risk. Fourth, when scripted tests needs to be more diversified. Fifth, where the regression testing can be done by exploring. Sixth is when the testing is being done from end user's point of view or based on user's manual.

Because of the advantages mentioned above, it is beneficial to practice exploratory testing as a complimentary approach to the traditional Test Case Based Testing [9]. However, an in-depth knowledge of various factors that influence the success nature of this approach of testing is being further investigated and enhanced. This study therefore aims to comprehend the

taxonomy of factors that influence the defect detection effectiveness of Exploratory Testing.

2. LITERATURE SURVEY

Increased dependence on software and the severe consequences of software failures have made IT firms to deliver high quality products. Donald E Harter and Mayuram S. Krishnan et. al in paper[10] investigate the relationship between process maturity, quality, cycle time and effort. They find that higher levels of process maturity are associated with higher product quality, but also with increased development effort [10].

Hence, software quality has emerged as one of the pivotal aspects of any software development process. Quality of software depends on the process followed during its development. Aman Kumar Sharma, Dr. Arvind Kalia et. al in paper[11] discuss the factors and sub factors of quality and provide a categorized, logical and understandable hierarchy of sub factors for the benefit of software developers and academicians towards research progress in the very domain [11].

Additionally, high quality software is one which is defect free, produces predictable results with less manageable results and so on. However, there exist several defect management strategies such as inspections, prototypes, testing and correctness proof to ensure the development of high quality software. Suma V and Gopalakrishnan Nair T R in paper[12] discuss various aspects of inspection such as defect prevention, the significance of inspection, analysis of the inspection process through a case study and hence conclude that inspection is the most efficient and effective way of defect prevention [12].

Further, software metrics play an important role in Software Engineering discipline. Any process that can be measured is better controlled. T R Gopalakrishnan Nair and Suma V in paper[13] introduce new inspection metrics such as Depth of Inspection, people metric and inspection performance metric and hence contribute towards getting valuable information regarding inspection process [13].

Natalia Juristo et. al in paper[14] presented a detailed situation of testing techniques and also analyze the maturity level of knowledge in software testing. They finally conclude that the current testing knowledge is limited [14].

Rashmi N and Suma V in paper [15] present the basics of exploratory testing such as the characteristics, benefits characteristics of exploratory testers are presented.

Juha Itkonen et. al in paper [16] performed a controlled experiment to compare the defect detection efficiency of test case based testing and exploratory testing and found no significance difference between the two approaches with regards to defect detection efficiency, technical type, detection difficulty or severity [16].

Rashmi N and Suma V in paper [17] present that the integration of exploratory testing with the traditional test case based testing increases the defect detection efficiency [17].

Above research study indicates the impact of testing the software before release and the rigorous ongoing research to accomplish the same. Further, the survey also indicates the current trend of advancement in exploratory testing.

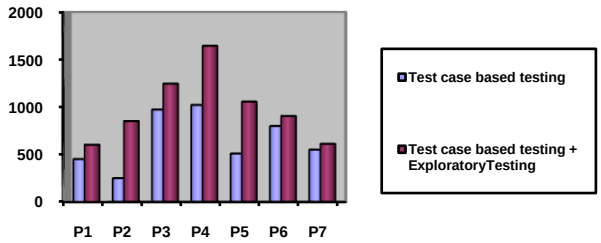
3. RESEARCH WORK

Practicing Exploratory testing as a complimentary approach is found to be beneficial. This is evident from the following table Table 1 which represents the data collected in terms of the defect counts from an organization. The data is collected from two companies performing scripted and exploratory testing. The companies are CMMI level 5 companies specializing in developing software products for high end printers and solutions for pharmaceutical companies. The projects here being compared are of embedded and life sciences type. The projects here are implemented using C, C++ on WINDOWS platform. All projects are non-critical in nature. All projects are of standalone type and of maintenance type. Data collection is through the data centers and quality assurance departments of the above mentioned company.

Table 1. Effectiveness of the combined approach

Projects	**Test Case Based Testing Approach**	**Integrated Approach (TCBT + ET)**	**Additional defects detected using Exploratory Testing**
P1	451	602	151
P2	251	851	600
P3	972	1244	272
P4	1022	1644	622
P5	510	1056	446
P6	800	905	105
P7	550	612	62

The table can also be represented graphically as shown below.



Literature survey has thrown light that Test Case Based Testing emphasizes decidability and accountability whereas Exploratory Testing emphasizes learning and adaptability. Hence, this study, aims to comprehend the factors that influence learning and adaptability as emphasized by exploratory testing[8].

Small and large organizations are always faced by two important issues such as lifelong learning and learning in work[18]. An increasing emphasis is being laid on individual and collective learning to enhance competitive advantage [19]. Hence, researchers have explored learning within the organizational context and identified the key factors at three levels namely organizational, functional and personal level as shown in Table 2.

Adaptability refers to the ease with which software allows differing system constraints and user needs to be satisfied [20]. Four important factors through which adaptability is analyzed are user, the platform, the environment and activity as shown in table 4[20]. A user model refers to various user characteristics. The user model consists of characteristics of a particular user. An interaction platform is about the physical characteristics of the devices. The third factor is the environment, which covers the physical, social and organizational elements that are outside of the interactive system (platform & user). The fourth factor is the activity. The activity factor includes task characteristics and the general activity and the user's goal [20].

Table 2. Factors influencing learning

	Organizational	**Learning**
Structure		
Culture		
Senior Management support		
Organization of work		
Work Pressures		
Managerial skills		
	Functional	
Time		
Money		
Staff		
Expertise		
New initiatives		
Role clarity		
	Personal	
Responsibility for learning		
Motivation to learn		
skills		

Table 3. Factors influencing adaptability

	User	**Adaptability**
Knowledge & level of experience		
Socio-demographic characteristics & user role		
Cognitive abilities & emotional state		
Perceptual & motor abilities		
	Interaction platform	
Computing Power & Autonomy		
Input/Output device		
Software Environment		
Connectivity		
	Environment	
Human Environment		
Machine Environment		
Ambient Characteristics		
Spacio-temporal Characteristics		
	Activity	
Task Characteristics		
User's Goal		
General Activity		

The main and the sub factors listed in table 4 are explained below.

3.1 Human Component

Almost all the practices, processes, methods and techniques of Software Engineering discipline are human centric. They all depend on the coordinated activities and decisions taken by the human beings. Hence, Exploratory Testing is also major affected by the Human Component. The factors that belong to this component are as follows.

3.1.1 Experience

Beer et al.[21] define the term experience as "practical knowledge that is developed in direct observation or participation in activities or as the amount of professional experience". Exploratory Testing is mainly dependent on the experience and intuition of the testers [9]. Experience may be classified as domain experience and testing experience [22]. Domain experience is obtained by working in a particular domain for a long time and testing experience is the experience obtained by working in various testing projects from multiple domains [21]. Experience helps in the design and the execution of the test cases and in the identification of defects[9].

Table 4. Taxonomy of Factors influencing Exploratory Testing

Factors affecting the effectiveness of Exploratory Testing	**Human Component**	H1	Experience
		H2	Knowledge
		H3	Learning styles
		H4	Tester's skill set
		H5	Tester's Creativity
	Technical Component	T1	Process Models
		T2	Documentation
		T3	Tools & Environment
		T4	Software Type
		T5	Fault Type
		T6	Testing Technique
		T7	Training Certification
	Organizational Component	O1	Organizational Structure
		O2	Organizational Culture
		O3	Organizational Maturity
		O4	Management Commitment
		O5	Organizational Stability
	Resource Component	R1	Time
		R2	Effort
		R3	Budget

3.1.2 Knowledge

Knowledge is applied to different exploratory testing tasks. First, knowledge is used as information to guide exploratory test design. Second, knowledge can be used to identify failures, i.e., as an oracle to distinguish between a correct, expected outcome and an incorrect, defective, outcome [22]. Third, knowledge, along with observed actual behavior of the tested system, can be used to create new, better tests during exploratory testing. Three important types of knowledge namely domain knowledge, system knowledge and generic software engineering knowledge are utilized by testers to create good exploratory tests and hence detect defects while performing exploratory testing. [22].

3.1.3 Learning styles

A learning style is a person's "characteristic strengths and preferences in the ways they take in and process information[23]. The learning styles can be classified as Sensory/Intuitive, Visual/Verbal, Inductive/Deductive, Active/Reflective, and Sequential/Global[23]. These learning styles are applied to create good test cases while performing exploratory testing[23].

3.1.4 Tester's Skill set

An exploratory tester possesses analytical and logical skills to analyze a product, to evaluate risks tools, and to think critically, among others. When a tester has the skills to listen, read, think and report rigorously and effectively without the use of pre-defined test cases, exploratory approach to testing can be many times as productive(in terms of revealing vital information) as the scripted one[24].

3.1.5 Tester's Creativity

An exploratory tester is first and foremost a test designer. He is able to craft tests that systematically explore the product[25]. Also exploratory testers produce more and better ideas than novices by making creative use of the heuristics such as guidelines, generic checklists, mnemonics or rules of thumb[9].

3.2 Technical Component

3.2.1 Process Models

Exploratory testing is best suited for agile projects where there is lack of comprehensive documentation[26].

3.2.2 Documentation

Existence of good documentation of the process helps the exploratory tester to produce some good tests else the tests may be based purely on the experience and knowledge of the tester[9].

3.2.3 Tools & Environment

Usage of automation tools and debugging tools during exploratory testing can increase the speed of testing. The use of some of the tools like MindJet help in getting a good knowledge of the heuristics which further helps in producing good exploratory tests[22]. Environment aspects such as the platform, competition, user expectations, physical environment in which the product operates such as light, sound etc. also affect exploratory testing[8].

3.2.4 Software Type

Exploratory testing is suitable for GUI rich commercial applications. However, it is also applicable to real-time embedded systems for avionics applications, including the Space Shuttle Inertial Measurement Unit, satellite systems, and other navigation systems [27].

3.2.5 Testing Technique

A testing technique can be selected based on the best set of quality attributes[8]. Exploratory testing is an approach and hence any testing technique such as scenario-based testing, stress testing, model-based testing can be prepared in an exploratory way.

3.2.6 Fault Type

According to a study [16] exploratory testing is able to detect more number of defects related to user interface and usability

issues.

3.2.7 Training & Certification

Exploratory testers depend mainly on their creativity, memory of past events, and intuition. Exercises that encourage participants to develop their creativity, sharpen their memory, or focus their intuition will play an important role[25]. Classes, courses, seminars and workshops are conducted by various experts on exploratory testing. These courses help the novice and experienced testers to enhance their test execution skills [25]. These technical skill set enhancement forums act as assurance schemes for exploratory testers.

3.3 Organizational Component

3.3.1 Organizational Culture

According to a study [29] there are four sub factors identified in this category. Attitude to process improvement, attitude to quality, work time policy, and personal effect on working practices. These sub factors are the specifics that affect Exploratory Testing. For example, a tester's attitude towards the quality of the software where the tester does not sign-off an artifact from testing unless he has the confidence that it follows the quality expected in the company[29].

3.3.2 Organizational Structure

Organizational structure deals mostly with the organization of the company. Two factors are identified: number of roles, and consistent understanding of software engineering practices. In some companies there are multiple roles for an individual personnel within one project and also he participates in multiple projects. This helps in the learning ability of an exploratory tester and hence results in good exploratory tests. Consistent understanding of the conceptions about the software engineering practices followed in the company between the developers and the managers helps in the improvement of the practices of the company [30].

3.3.3 Organizational Maturity

The sub factors of organizational maturity are effective measurement, use of modern programming practices and availability of technical support. Measurement can be applied to organizations, projects, processes, and work products [30].

Session based test management technique of exploratory testing provides few metrics which helps in measuring the coverage of the product[30]. For projects practicing modern programming practices such as agile programming exploratory testing is most suited [28].

3.3.4 Management Commitment

The sub factors of management commitment are the senior management commitment to project, lack or loss of organizational commitment to project[30]. Practicing exploratory testing as a complimentary approach to the traditional testing helps the management to achieve its goal of estimated quality [28].

3.3.5 Organization Stability

The sub factors of organization stability are resources shifting from the project due to changes in the organizational priorities, unstable organizational environment, effect of corporate politics on projects, organization undergoing restructuring during the project, rate of organizational change (growth or decline)[30]. For example, doubling the headcount of software engineers in a company affects the potential economy of scale of the software development process[30].

3.4 Resource Component

The resources used during software testing are time, people, and budget. The main aim of the manager is to allocate the resources in a constrained manner so that the effort involved in testing is reduced and hence the overall budget[31].

3.4.1 Time

According to James Bach, exploratory testing can be practiced when defects of critical nature and high priority need to be investigated in a short span of time while there is hardly ample of time available for implementation of systematic testing approaches. In addition, one of the characteristic of exploratory testing is rapid feedback from the testers regarding the behavior of the product to the developers[9].

3.4.2 Effort

Another important characteristic of exploratory testing is reduced documentation. The tests are not defined before. Rather, the focus is on finding the defects by exploration. Hence, the effort involved in exploratory testing is less compared to traditional test case based testing [9].

3.4.3 Budget

Exploratory testing when performed in combination with the Test Case Based Testing can reduce the technical debt thereby reducing the budget associated with the project[32].

4. ACKNOWLEDGMENTS

Authors would like to thank all the industry people who extended their valuable support and help in compliance within the framework of non-disclosure agreement.

5. CONCLUSION AND FUTURE WORK

Software Testing is an effective means of delivering a quality software. The success of software testing depends on the choice of the testing approach being used. There exists mainly two approaches of software testing to test different quality attributes of a software. The approaches are Test Case Based Testing and Exploratory Testing. Though Test Case Based Testing is very popular and being practiced in many software industries, this approach has certain disadvantages. However, as a solution practicing exploratory testing as a complimentary approach to the Test Case Based Testing is found to be beneficial in several ways.

From literature survey we understand that Test Case Based Testing emphasizes decidability, predictability and accountability while Exploratory Testing emphasizes learning and adaptability. There exists a number of test case design strategies that are categorized under traditional Test Case Based Testing. These are based on theories for effectively revealing defects in software and have been studied using only the Test Case Based Testing approach. Studying the factors that affect the defect detection effectiveness and efficiency of Exploratory Testing is an important part of research. Hence, this paper aims at studying the factors influencing the defect detection effectiveness of Exploratory Testing. In this regard, as a first step of our research, we present the defect detection efficiency of the integrated approach of the traditional Test Case Based Testing and Exploratory Testing through a case study involving seven projects. From this we understand that the defect detection efficiency of the integrated approach is better compared to the traditional Test Case Based Testing alone. This encourages us to study the factors influencing learning and adaptability. Finally, we group the factors into four components as human, technical, organizational and resource components. Further, more research is required to study the effect of predefined test

cases in comparison to other approaches to manual testing.

6. REFERENCES

[1] Humphrey, Watts S. "The Software Quality Challenge." CROSSTALK, The journal of Defense Software Engineering (June 2008).

[2] Meyer B, ETH Zurich & Eiffel Software, Santa Barbara, CA, "Seven Basic Principles of Software Testing", Computer ,Volume:41 Issue: 8 ,IEEE Computer Society, Aug. 2008, 99 - 101.

[3] Jussi Kasurinen, Ossi Taipale, and Kari Smolander, "Software Test Automation in Practice: Empirical Observations", Hindawi Publishing Corporation, Advances in Software Engineering, Volume 2010, Article ID 620836, 18 pages, oi:10.1155/2010/620836

[4] Andersson, C. and P. Runeson, "Verification and validation in industry - a qualitative survey on the state of practice," Proceedings of International Symposium on Empirical Software Engineering, 2002, pp. 37-47.

[5] Cem Kaner, "The Nature of Exploratory Testing", 2004.

[6] Myers, G.J., The Art of Software Testing, New York: John Wiley & Sons, 1979.

[7] Beizer, B., Software Testing Techniques, New York: Van Nostrand Reinhold, 1990.

[8] Cem Kaner J.D. "A Tutorial in Exploratory Testing", April 2008.

[9] Itkonen, J. and K. Rautiainen, "Exploratory testing: a multiple case study," Proceedings of International Symposium on Empirical Software Engineering, 2005, pp. 84-93.

[10] Donald E Harter, Mayuram S Krishnan, Sandra A Slaughter, "Effects of Process Maturity on Quality, Cycle time and effort in Software Product Development", Management Science, Vol,46, No. 4, April 2000.

[11] Aman Kumar Sharma, Dr, Arvind Kalia and Dr. Hardeep Singh, "An analysis of Optimum Software Quality Factors", IOSR Journal of Engineering Apr. 2012, Vol. 2(4) pp: 663-669.

[12] V. Suma, T. R. Gopalakrishnan Nair, "Defect Management in Software Development", Book on Recent Advances in Technologies", ISBN 978-953-307-017-9, pp 379-404, Intec web Publishers, Vienna, Austria, November 2009.

[13] V. Suma, T. R. Gopalakrishnan Nair, "Defect Management Depth of Inspection and Inspection Performance metric", Crosstalk, Nov/Dec 2011.

[14] Juristo N., A.M. Moreno, S. Vegas, "Reviewing 25 years of Testing Technique Experiments ",Empirical Software Engineering, vol. 9(1-2), pp. 7-44, 2004.

[15] Rashmi N, Suma V, "Exploratory Testing for Effective Software Development", RATE-13.

[16] Juha Itkonen, Mika V. Mäntylä, Casper Lassenius, "Defect Detection Efficiency: Test Case Based vs. Exploratory Testing", Proceedings of International Symposium on Empirical Software Engineering and Measurement, pp. 61–70, 2007.

[17] Rashmi ,N , Suma .V, "Defect Detection Efficiency: A Combined Approach", International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-3 Number-3 Issue-11 September-2013.

[18] Sally Sambrook, "Factors Influencing Learning in Work: a comparison of two research projects (European- and United Kingdom-based)", European Educational Research Journal, Volume 1, Number 3, 2002.

[19] Moingeon, B. & Edmondson, A. (Eds) (1996) Organisational Learning and Competitive Advantage. London: Sage.

[20] Tarpin-Bernard F., Marfisi-Schottman I, Habieb-Mammar H., anameter: the first steps to evaluating adaptation ,sixth workshop on user-centred design and evaluation of adaptive systems, umap09 user modeling, adaptation, and personalization, trento, Italy, pp. 11-20, 2009.

[21] Beer, A. and R. Ramler, "The Role of Experience in Software Testing Practice," Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2008, pp. 258-265.

[22] Juha Itkonen, Mika V. Mäntylä, and Casper Lassenius. "The role of knowledge in failure detection during exploratory software testing". IEEE Transactions on Software Engineering, May 2011, 17 pages.

[23] Andy Tinkham and Cem Kaner, "Learning Styles and Exploratory Testing", Pacific Northwest Software Quality Conference (PNSQC), 2003.

[24] James Bach, "Exploratory Testing Explained", 2003.

[25] Andy Tinkham, Cem Kaner, "Exploring Exploratory Testing", 2003.

[26] James Lyndsay, "Testing in an agile environment", Workroom Productions Ltd, 2007.

[27] Phil Laplante, "Exploratory Testing for Mission Critical, Real-Time and Embedded Systems", IEEE Transactions on Reliability, pp. 449-482, Volume 59, Number 3, September 2010.

[28] Jonathan Bach, "Session Based Test Management", 2000.

[29] Bern, A., S. J. A. Pasi, U. Nikula and K. Smolander (2007). Contextual Factors Affecting the Software Development Process – An Initial View. Second AIS SIGSAND European Symposium on Systems Analysis and Design, Gdansk, Poland, University of Gdansk, pp. 72-79.

[30] Clarke, P. and O'Connor R.V. "The situational factors that debugging SRGM", International Journal of Computer Applications (0975 – 8887), Volume 18– No.2, March 2011, Software and Technology, Vol. 54, Issue 5, May 2012. pp.433-447.

[31] SK.Md.Rafi, Shaheda Akthar, " Resource Allocation to Software Modules in Software Testing with Imperfect-debugging SRGM", International Journal of Computer Applications (0975 – 8887)Volume 18– No.2, March 2011.

[32] Syed Muhammad Ali Shah, Marco Torchiano, Antonio VetroMaurizio Morisio, "Exploratory testing as a source of testing technical debt".