Exploratory testing (ET) is an approach to test software with a strong focus on personal skills and freedom of the tester. ET emphasises the simultaneous design and execution of tests with minimal test documentation. Test practitioners often claim that their choice to use ET as an important alternative to scripted testing is based on several benefits ET exhibits over the scripted testing. However, these claims lack empirical evidence as there is little research done in this area. Moreover, ET is usually considered an ad-hoc way of doing testing as everyone does it differently. There have been some attempts in past to provide structure to ET. Session based test management (SBTM) is an approach that attempts to provide some structure to ET and gives some basic guidelines to structuring the test sessions. However, these guidelines are still very abstract and are very open to individuals’ interpretation.

The main objective of this doctoral thesis is to support practitioners in their decisions about choosing exploratory versus scripted testing. Furthermore, it is also aimed to investigate the empirical evidence in support of ET and find ways to structure ET and classify different levels of exploration that drive the choices made by exploratory testers. Another objective of this thesis is to provide a decision support system to select levels of exploration in overall test process.

The findings presented in this thesis are obtained through a controlled experiment with participants from industry and academia, exploratory surveys, literature review, interviews and focus groups conducted at different companies including Ericsson AB, Sony Mobile Communications, Axis Communications AB and Softhouse Consulting Baltic AB.

Using the exploratory survey, we found three test techniques to be most relevant in the context of testing large-scale software systems. The most frequently used technique mentioned by the practitioners is ET which is not a much researched topic. We also found many interesting claims about ET in grey literature produced by practitioners in the form of informal presentations and blogs but these claims lacked any empirical evidence. Therefore, a controlled experiment was conducted with students and industry practitioners to compare ET with scripted testing. The experiment results show that ET detects significantly more critical defects compared to scripted testing and is more time efficient. However, ET has its own limitations and there is not a single way to use it for testing. In order to provide structure to ET, we conducted a study where we proposed checklists to support test charter design in ET. Furthermore, two more industrial focus group studies at four companies were conducted that resulted in a taxonomy of exploration levels in ET and a decision support method for selecting exploration levels in ET. Lastly, we investigated different problems that researchers face when conducting surveys in software engineering and have presented mitigation strategies for these problems.

The taxonomy for levels of exploration in ET, proposed in this thesis, provided test practitioners at the companies a better understanding of the underlying concepts of ET and a way to structure their test charters. A number of influence factors elicited as part of this thesis also help them prioritise which level of exploration suits testing more in their product’s context. Furthermore, the decision support method provided the practitioners to reconsider their current test focus to test their products in a more effective way.
Structuring Exploratory Testing Through Test Charter Design and Decision Support

Ahmad Nauman Ghazi
Structuring Exploratory Testing Through Test Charter Design and Decision Support

Ahmad Nauman Ghazi

Doctoral Dissertation in Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN
Structuring Exploratory Testing Through Test Charter Design and Decision Support

Ahmad Nauman Ghazi
Structuring Exploratory Testing Through Test Charter Design and Decision Support

Ahmad Nauman Ghazi

Doctoral Dissertation in Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN
To Allah (swt), for blessing me with the abilities and opportunities;
To my late father;
To my mother and sisters, for their constant support, love and prayers;
To my wife Sarah and sons Ajlaan and Ghazwan, for being a continuous source of peace and joy.
“The greatest obstacle to discovery is not ignorance – it is the illusion of knowledge.”

–Daniel J. Boorstin
Abstract

Context: Exploratory testing (ET) is an approach to test software with a strong focus on personal skills and freedom of the tester. ET emphasises the simultaneous design and execution of tests with minimal test documentation. Test practitioners often claim that their choice to use ET as an important alternative to scripted testing is based on several benefits ET exhibits over the scripted testing. However, these claims lack empirical evidence as there is little research done in this area. Moreover, ET is usually considered an ad-hoc way of doing testing as everyone does it differently. There have been some attempts in past to provide structure to ET. Session based test management (SBTM) is an approach that attempts to provide some structure to ET and gives some basic guidelines to structuring the test sessions. However, these guidelines are still very abstract and are very open to individuals’ interpretation.

Objective: The main objective of this doctoral thesis is to support practitioners in their decisions about choosing exploratory versus scripted testing. Furthermore, it is also aimed to investigate the empirical evidence in support of ET and find ways to structure ET and classify different levels of exploration that drive the choices made by exploratory testers. Another objective of this thesis is to provide a decision support system to select levels of exploration in overall test process.

Method: The findings presented in this thesis are obtained through a controlled experiment with participants from industry and academia, exploratory surveys, literature review, interviews and focus groups conducted at different companies including Ericsson AB, Sony Mobile Communications, Axis Communications AB and Softhouse Consulting Baltic AB.

Results: Using the exploratory survey, we found three test techniques to be most relevant in the context of testing large-scale software systems. The most frequently used technique mentioned by the practitioners is ET which is not a much researched topic. We also found many interesting claims about ET in grey literature produced by practitioners in the form of informal presentations and blogs but these claims lacked any empirical evidence. Therefore, a controlled experiment was conducted with students and industry practitioners to compare ET with scripted testing. The experiment results show that ET detects significantly more critical defects compared to scripted testing and is more time efficient. However, ET has its own limitations and there is not a single way to use it for testing. In order to provide structure to ET, we conducted a study where we proposed checklists to support test charter design in ET. Furthermore, two more industrial focus group studies at four companies were conducted that resulted in a taxonomy of exploration levels in ET and a decision support method for selecting exploration levels in ET. Lastly, we investigated different problems that researchers face when conducting surveys in software engineering and have presented mitigation
strategies for these problems.

**Conclusion:** The taxonomy for levels of exploration in ET, proposed in this thesis, provided test practitioners at the companies a better understanding of the underlying concepts of ET and a way to structure their test charters. A number of influence factors elicited as part of this thesis also help them prioritise which level of exploration suits testing more in their product’s context. Furthermore, the decision support method provided the practitioners to reconsider their current test focus to test their products in a more effective way.
Acknowledgements

First and foremost, I would like to express my gratitude to my main supervisor and collaborator Prof. Kai Petersen for his continuous support, guidance and feedback on my work, for the fruitful collaboration on papers, and for always responding to my questions. I am lucky to have him supervising me; his continuous feedback on my ideas and discussions have been a major driving force to complete this thesis. I am also highly indebted to my secondary supervisor Prof. Jürgen Börschinger for his support and feedback on my papers despite his busy schedule. I would also take the opportunity to thank Prof. Niklas Lavesson who served as an external reviewer for my individual study plan throughout these years. His continuous support and timely advice, especially in some of the most crucial times during studies, were instrumental to complete my post graduate studies.

I would like to thank my collaborators Dr. Elizabeth Bjarnason (Lund University), Prof. Per Runeson (Lund University) and Prof. Claes Wohlin (Blekinge Institute of Technology) for their time to discuss my work and provide their precious feedback on papers. In particular, I am thankful to Prof. Claes Wohlin for giving me the opportunity to work in the EASE project during the last year which substantially helped me to do applied research with industrial partners. I am also grateful to Dr. Emelie Engström who was the opponent for my Licentiate defense and provided constructive feedback.

I am grateful to all the practitioners who participated in this research. For sharing their knowledge, insights and time that enabled the studies that are part of this thesis. In particular, I would like to thank Kristoffer Ankarberg (Ericsson AB, Karlskrona) for helping me to conduct research studies with his team members. I would also thank Imran Haider (Ericsson AB, Karlskrona) for his continuous support, encouragement and friendship.

I am also thankful to my colleagues at the SERL group, for providing a nice work environment. In particular, I would like to thank Dr. Nauman bin Ali, Muhammad Usman, and Ricardo Britto for their constant support throughout my post graduate studies.

This thesis would not have been possible without the love, support and patience from my wife Sarah. I am extremely grateful to her and my sons; Ajlaan and Ghazwan for being patient with me working late hours and not being able to give them proper time during my studies. I am deeply grateful to my mother, sisters, nieces and nephews for always being my support, I could have never achieved anything in my life without their support and unconditional love. I would also like to thank both my father-in-law and mother-in-law for their continuous love and support.

This work was partly supported by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se)
Overview of Papers

Papers included in this thesis.

Chapter 2. Ahmad Nauman Ghazi, Kai Petersen and Jürgen Börstler.


Chapter 4. Ahmad Nauman Ghazi, Ratna Pranathi Garigapati and Kai Petersen.

Chapter 5. Ahmad Nauman Ghazi, Kai Petersen, Elizabeth Bjarnason and Per Runeson.

Chapter 6. Ahmad Nauman Ghazi, Kai Petersen, Claes Wohlin and Elizabeth Bjarnason.

# Table of Contents

1 Introduction ........................................... 1  
   1.1 Preamble ........................................... 1  
   1.2 Background ........................................ 4  
      1.2.1 Exploratory Testing .......................... 4  
      1.2.2 Session-Based Test Management .............. 4  
      1.2.3 Test Charters .................................. 5  
   1.3 Research Gaps and Contributions .................. 5  
      1.3.1 Research Questions ............................ 7  
   1.4 Methods .......................................... 9  
   1.5 Overview and Results of Studies ................. 11  
      1.5.1 Chapter 2: Testing heterogeneous systems: An exploratory survey .......... 13  
      1.5.2 Chapter 3: An experiment on the effectiveness and efficiency of exploratory testing .................. 13  
      1.5.3 Chapter 4: Checklists to support test charter design in exploratory testing .......... 14  
      1.5.4 Chapter 5: Exploratory testing: One size doesn’t fit all ........ 14  
      1.5.5 Chapter 6: A decision support method for recommending degrees of exploration in exploratory testing .......... 15  
      1.5.6 Chapter 7: Survey research in software engineering: Problems and strategies .......... 17  
   1.6 Conclusions ..................................... 17  
   1.7 Future Work .................................... 21  
   1.8 Contribution Statement ........................... 22  
   1.9 References ..................................... 24
# Table of Contents

## 2 Testing of Heterogeneous Systems: An Exploratory Survey  27

2.1 Introduction ................................................................. 28
2.2 Related work ................................................................. 29
   2.2.1 Testing in Heterogeneous Systems ............................ 29
   2.2.2 Testing Techniques .................................................. 30
2.3 Research method ............................................................. 32
   2.3.1 Study purpose ......................................................... 32
   2.3.2 Survey Distribution and Sample ................................ 33
   2.3.3 Instrument Design ................................................... 33
   2.3.4 Analysis ................................................................. 34
   2.3.5 Validity Threats ....................................................... 35
2.4 Results ........................................................................... 36
   2.4.1 Context ................................................................. 36
   2.4.2 RQ1: Usage of Testing Techniques ............................ 38
   2.4.3 RQ2: Perceived Usefulness ........................................ 39
2.5 Discussion ...................................................................... 41
2.6 Conclusion ...................................................................... 43
2.7 References ..................................................................... 45

## 3 An Experiment on the Effectiveness and Efficiency of Exploratory Testing  49

3.1 Introduction ................................................................. 50
3.2 Related work ................................................................. 53
3.3 Methodology ................................................................. 54
   3.3.1 Goal definition ......................................................... 54
   3.3.2 Research questions and hypotheses formulation .......... 55
   3.3.3 Selection of subjects .................................................. 56
   3.3.4 Experiment design ..................................................... 58
   3.3.5 Instrumentation ......................................................... 60
   3.3.6 Operation ................................................................. 61
3.4 Results and analysis ....................................................... 62
   3.4.1 Defect count ............................................................. 62
   3.4.2 Detection difficulty, types and severity ....................... 63
   3.4.3 False defect reports ................................................... 77
3.5 Discussion ...................................................................... 78
   3.5.1 RQ 1: How do the ET and TCT testing approaches compare with respect to the number of defects detected in a given time? 78
   3.5.2 RQ 2: How do the ET and TCT testing approaches compare with respect to defect detection difficulty, types of identified defects and defect severity levels? 79
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.3</td>
<td>RQ 3: How do the ET and TCT testing approaches compare in terms of number of false defect reports?</td>
<td>81</td>
</tr>
<tr>
<td>3.6</td>
<td>Validity threats</td>
<td>82</td>
</tr>
<tr>
<td>3.7</td>
<td>Conclusions and future work</td>
<td>86</td>
</tr>
<tr>
<td>3.8</td>
<td>References</td>
<td>88</td>
</tr>
<tr>
<td>4</td>
<td>Checklists to Support Test Charter Design in Exploratory Testing</td>
<td>95</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>96</td>
</tr>
<tr>
<td>4.2</td>
<td>Related work</td>
<td>97</td>
</tr>
<tr>
<td>4.3</td>
<td>Research method</td>
<td>97</td>
</tr>
<tr>
<td>4.4</td>
<td>Results</td>
<td>98</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>102</td>
</tr>
<tr>
<td>4.6</td>
<td>References</td>
<td>104</td>
</tr>
<tr>
<td>5</td>
<td>Exploratory Testing: One Size Doesn’t Fit All</td>
<td>105</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>106</td>
</tr>
<tr>
<td>5.2</td>
<td>Exploratory testing</td>
<td>107</td>
</tr>
<tr>
<td>5.3</td>
<td>Classification of levels of exploratory testing</td>
<td>107</td>
</tr>
<tr>
<td>5.4</td>
<td>Methodology and case details</td>
<td>110</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Threats to validity</td>
<td>111</td>
</tr>
<tr>
<td>5.5</td>
<td>Results from the focus groups</td>
<td>111</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusions and Future Work</td>
<td>114</td>
</tr>
<tr>
<td>5.7</td>
<td>References</td>
<td>117</td>
</tr>
<tr>
<td>6</td>
<td>A Decision Support Method for Recommending Degrees of Exploration in Exploratory Testing</td>
<td>119</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>120</td>
</tr>
<tr>
<td>6.2</td>
<td>Related work</td>
<td>121</td>
</tr>
<tr>
<td>6.3</td>
<td>Solution proposal</td>
<td>122</td>
</tr>
<tr>
<td>6.3.1</td>
<td>General method</td>
<td>122</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Tailoring for ET</td>
<td>123</td>
</tr>
<tr>
<td>6.4</td>
<td>Application</td>
<td>126</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Evaluation method</td>
<td>127</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Results</td>
<td>129</td>
</tr>
<tr>
<td>6.5</td>
<td>Discussion</td>
<td>132</td>
</tr>
<tr>
<td>6.6</td>
<td>Conclusion</td>
<td>135</td>
</tr>
<tr>
<td>6.7</td>
<td>References</td>
<td>137</td>
</tr>
</tbody>
</table>
# Table of Contents

7 Survey Research in Software Engineering: Problems and Strategies 139

7.1 Introduction .......................................................... 140
7.2 Background on the survey research method .......................... 141
  7.2.1 Research objectives are defined .............................. 141
  7.2.2 Target audience and sampling frame are identified ...... 143
  7.2.3 Sample plan is designed ..................................... 143
  7.2.4 Survey instrument is designed ............................... 144
  7.2.5 Survey Instrument is Evaluated .............................. 144
  7.2.6 Survey data is analyzed ...................................... 144
  7.2.7 Conclusions extracted from survey data ................. 145
  7.2.8 Survey documented and reported ......................... 145
7.3 Related Work ......................................................... 146
  7.3.1 Guidelines for survey research in software engineering .... 146
  7.3.2 Problems and strategies ..................................... 147
7.4 Research Method ..................................................... 154
  7.4.1 Research questions ........................................... 154
  7.4.2 Selection of subjects ......................................... 154
  7.4.3 Data collection ................................................ 155
  7.4.4 Data analysis ................................................. 156
  7.4.5 Threats to validity .......................................... 157
7.5 Interview results ..................................................... 158
  7.5.1 Target audience and sampling frame definition and sampling plan 158
  7.5.2 Survey instrument design and evaluation .................. 159
  7.5.3 Data analysis and conclusions .............................. 162
  7.5.4 Reporting ...................................................... 162
7.6 Discussion ............................................................ 162
  7.6.1 Comparison with related work .............................. 162
  7.6.2 Conflicting recommendations and trade-offs .............. 165
7.7 Conclusions .......................................................... 166
7.8 References ........................................................... 168

A Appendix A: Test case template for TCT 175

B Appendix B: Defect report 177

C Appendix C: ET – Test session charter 179
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D Appendix D: Interview guide for the study on survey research in software engineering</td>
<td>181</td>
</tr>
<tr>
<td>D.0.1 Researcher perspective</td>
<td>181</td>
</tr>
<tr>
<td>D.0.2 Respondent's perspective</td>
<td>183</td>
</tr>
<tr>
<td>D.0.3 Concluding questions about survey guidelines</td>
<td>183</td>
</tr>
<tr>
<td>List of Figures</td>
<td>185</td>
</tr>
<tr>
<td>List of Tables</td>
<td>186</td>
</tr>
</tbody>
</table>
Table of Contents
Chapter 1

Introduction

1.1 Preamble

Software testing is an important part of the overall software development lifecycle to improve the software quality. A number of test techniques have been developed in the past to effectively test software while saving cost and time. However, there is a lack of decision support for the test practitioners to choose what technique or approach to use that fits best in their context [8].

When it comes to different test approaches, Itkonen [14] distinguish between scripted testing and exploratory testing. The traditional approach for software testing is scripted testing [1] [24] where test cases are defined, planned and designed prior to their execution.

James Bach defines exploratory testing (ET) as simultaneous learning, test design and test execution [3]. Existing literature shows that ET is a very flexible approach for software testing and can be adapted to different test levels, activities and phases [10][22]. Furthermore, ET is also widely used for testing complex systems of systems [10]. ET as an approach to test software that allows personal freedom and leverages the skills of the tester [17].

Advocates of ET stress the benefits of providing freedom for the tester for test execution where skills, previous experience and testers’ intuition are the main driver; instead of pre-defined test cases. This leads to less overhead in terms of documentation resulting in reduced effort for test script design and maintenance [1]. ET supports testers in learning about the system while testing [1] [17]. The ET approach also enables a tester to explore areas of the software that were overlooked, while designing
test cases based on system requirements [16]. In contrast to the classical approach to software testing, the focus of ET is on providing freedom to testers to explore the software and execute tests without pre-defined scripted test cases. Due to this reason, some argue that ET is an ad-hoc way to test software. However, over the years, ET has evolved to a more structured approach without compromising the basic notions of personal freedom and individual responsibility of the testers [5].

In the past, session-based test management (SBTM) [5] was introduced as an enhancement to exploratory testing which attempts to provide some basic guidelines and structure for practicing ET. SBTM incorporates planning, structuring, guiding and tracking the test effort with tool support for ET.

Some advocates of ET also claim that ET is more effective in defect detection but there is little empirical evidence available to support this claim. There exists only one controlled experiment study where scripted and exploratory approaches were compared [14]. This study was conducted with students and lacked an industrial context. Furthermore, in this study, there existed some design flaws. For example, both controlled groups in this study used both exploratory testing and scripted testing which induced the risk of learning effect. Moreover, for scripted testing, students were provided extra time to design test cases prior to execution. Given these issues, the results from this study show that there is no significant difference in defect detection efficiency between the two test approaches. Therefore, we replicated this study with an updated experiment design, but using the same test object and present this study in Chapter 3. To our knowledge, there existed no further controlled experiments on exploratory testing involving industry practitioners.

In this thesis, we investigate the usage of testing techniques in industry to identify different testing techniques used in practice. The applicability and perceived usefulness of different testing techniques is investigated using an exploratory survey. This survey was carried out with a set of industry practitioners involved in different roles in development of large-scale systems. Three main testing techniques were identified that are used in the context of large-scale systems. Later, we conducted a controlled experiment with the subjects from both industry and academia to investigate effectiveness and efficiency of exploratory testing in comparison with traditional scripted testing. The results from this study show that ET performs significantly better in finding critical defects and is more time efficient as compared to scripted testing. However, during this study and with further discussions with the practitioners, it was found that ET is practiced in an individualized manner and there is no one way of doing it. Therefore, there is a strong need to find ways to structure ET as an approach. Although, SBTM provides some basic guidelines and mentions the basic elements needed to practice ET, there are some shortcomings of this enhancement of ET. For example, SBTM provides a strong focus on having a test charter for each test session. But, practically test
charters are designed differently based on the understanding of the tester. Hence, we identified the need of guidelines for test charter design. To this end, another study was conducted where experienced exploratory testers were interviewed to investigate the most important elements that shall be included in a test charter. Based on the findings in this study, we proposed two checklists for test charter design that would help practitioners design their test charters and practice ET in a more standardized way. We also found out that using different types of test charters can drive the degree of exploration in software testing. Moreover, between the two extremes in software testing, namely: freestyle exploratory testing and scripted testing, there exist a continuum. Therefore, we proposed a taxonomy for the levels of exploration in software testing and exemplified these levels by presenting different test charters that represent these levels of exploration. We also listed distinct elements of a test charter design that help differentiate between different levels of exploration. Based on focus groups conducted at four different companies, we also identified different influence factors that affect test charter design and enable testers to scope the exploration based on the context of their products. Lastly, a literature review and interview study was conducted to investigate the problems encountered by the researcher doing survey research in software engineering. Multiple problems were identified and their mitigation strategies are presented in this thesis. The last study was motivated by the challenges we faced during conducting the survey-based study in Chapter 2.

The main contribution made in this thesis is “to support the practitioners in how to conduct exploratory testing”. This thesis investigated exploratory testing from different perspectives. The following sub-contributions are made:

C1: We explored the usage of different test techniques and approaches (e.g., combinatorial testing, exploratory testing and search-based software testing) in practice through an exploratory survey.

C2: We compared the effectiveness and efficiency of exploratory testing and scripted testing in both academic and industrial contexts.

C3: We investigated the contents of test charters and what factors influence the design of test charters.

C4: We proposed a taxonomy for different levels/degrees of exploration in software testing is presented, and how they relate to test charters. Test charters are a means to influence the degree of exploration.

C5: We developed a decision support method to help practitioners make informed decisions about what exploration levels suit their context best and how to divide time between test activities representing these exploration levels.
Chapter 1. Introduction

C6: We identified different problems in conducting survey research through literature and interviews with researchers and strategies are provided to address these problems. These problems were initially observed when conducting the exploratory survey discussed above in C1.

1.2 Background

1.2.1 Exploratory Testing

Exploratory testing (ET) is an approach that does not rely on the documentation of test cases prior to test execution in contrast with traditional test case based testing. It has been acknowledged in the literature that ET has lacked scientific research [13]. In the past, exploratory testing was seen as an ad-hoc approach to test software. However, over the years, ET has evolved into a more manageable and structured approach without compromising the freedom of testers to explore, learn and execute the tests in parallel. An empirical study comparing the effectiveness of exploratory testing with test-case based testing was conducted by Bhatti and Ghazi [2] and further extended (cf. [1]). This empirical work concludes that ET detects more defects as compared to test case based testing where time to test is a constraint.

ET is defined as simultaneous learning, test design and test execution [3]. ET is perceived to be flexible and applicable to different types of activities, test levels and phases [22]. Existing literature showcases a good amount of evidence regarding the merits of ET, such as its defect detection effectiveness, cost effectiveness and high performance for detecting critical defects [1], [13], [15] and [22]. During the exploratory testing process, the testers may interact with the application and take the information it provides to react, change course or explore the application’s functionality without any constraint [27]. ET is usually done in an iterative fashion [23] to facilitate continuous learning. The factors on which the effectiveness of ET depends are software maturity, the skills of the tester, the product being tested and the time required to test the product [3].

1.2.2 Session-Based Test Management

Session-based test management (SBTM) is an enhancement of ET that helps in tracking the individual tester’s ET progress. In SBTM, the test results are reported in a consistent and accountable way [23]. Session-based test management is a technique that helps in managing and controlling tests that are unscripted. SBTM framework focuses on testing without scripted tests and builds on its strengths such as the speed,
flexibility and range. However, SBTM provides more control and structure to these unscripted tests by explicitly stating the test mission, designing a test charter and through time-boxing the test sessions. Thus, they form a powerful part of the overall test strategy [19], which is a set of ideas that guide the choice of test that in turn guide the test design. Also, the test strategy includes a set of ideas related to project environment, product elements, quality criteria and test techniques [5].

1.2.3 Test Charters

Test charters, which are an SBTM element, play a major role in guiding the testers. The charter is a test plan which is usually generated from a test strategy. The charters include ideas that guide the testers as they test. These ideas are partially documented and are subject to change as the project evolves [5]. SBTM echoes the actions of testers who are well experienced in testing and charters play a key role in guiding the inexperienced testers by providing them with details regarding the aspects and actions involved in the particular test session [4].

The context of the test session plays a great role in determining the design of test plan or the charter [5]. Key steps to achieve context awareness are, for example, understanding the project members and the way they are affected by the charter, and understanding work constraints and resources. When designing charters Bach [5] formulated specific goals, in particular finding significant tests quicker, improving quality, and increasing testing efficiency.

The sources that inspire the design of test charters are manifold (cf. [5][12][17]), such as risks, product analysis, requirements, and questions raised by stakeholders. Mission statements, test priorities, risk areas, test logistics, and how to test are example elements of a test charter design identified from the literature review and their description [1] [5] [9]. Our study will further complement the contents of test charters as they are used in practice.

1.3 Research Gaps and Contributions

The following research gaps have been identified in the related work, and through the exploratory survey and controlled experiment done in Chapters 2 and 3:

Gap-1 Lack of empirical evidence about the claims of exploratory testing advocated that ET is more effective in defect detection and performs better as compared to scripted testing [24].
Chapter 1. Introduction

Gap-2 There is no single way to perform ET and there is a lack of structure in this approach for testing

Gap-3 There exists limited decision support for the practitioners to decide when or when not to use ET during their overall test process

Gap-4 During the course of this thesis, we identified that there exists multiple problems while doing survey research in software engineering, while a synthesis of experiences of survey research is missing.

Gap-1 was identified through the literature and grey material published by the practitioners on their personal blogs. These practitioners and other advocates of exploratory testing have on occasions mentioned that exploratory testing performs much better as compared to the scripted testing. However, we could not find much empirical evidence to support these claims. This gap was further confirmed during execution of the survey presented in Chapter 2.

Contributions: The study presented in Chapter 3 was done to address the above mentioned research gap investigating empirical evidence for the effectiveness and efficiency of exploratory testing in comparison to scripted testing. In the controlled experiment, presented in Chapter 3, both practitioners and students participated as subjects. In the follow-up discussions with the subjects, they affirmed that there is a need to structure the ET approach to gain the benefit of the freedom to explore while mitigating the risk of distracting the tester from the main goal of a test session. This follow-up helped to identify the research gap stated below.

Gap-2 was identified through the literature and grey material published by the practitioners on their personal blogs. This gap was confirmed during the execution of controlled experiment to compare the effectiveness and efficiency of exploratory testing and scripted testing (Chapter 3).

Contributions: During this thesis, we identified the test charters as one of the main elements of session-based test management which is an update of exploratory testing as an initial attempt to provide structure to ET. However, there existed no guidelines for practitioners about how to design test charters. Due to a lack of these guidelines, we observed that practitioners follow an individual approach for designing test charters. In the study presented in Chapter 4, we investigated what are the factors considered by ET practitioners when designing their test charters. This study resulted in two checklists where 30 factors and 25 content elements were elicited. These checklists provide the basis for designing test charters. Later, in Chapter 5, we present a taxonomy for the levels of exploration in exploratory testing and how the testers can induce a specific level of exploration in their testing using a test charter design especially catered for that
level of exploration. In this taxonomy we present 5 different levels of exploration. Furthermore, a number of influence factors were identified in this study and a discussion of how these factors influence the test charter design is also presented in Chapter 5.

*Gap-3* was identified during the synthesis of the interviews conducted with the exploratory testing practitioners (Chapter 4) and also during the focus groups done with four different companies to evaluate the test charter taxonomy (Chapter 5). Both these studies were done to investigate how test charter design can help structuring the exploratory testing.

*Contributions:* Based on the focus groups conducted during the study presented in Chapter 5, we designed a decision support method taking the findings from Chapter 5 in consideration. The levels of exploration and the influence factors identified in that study served as the basis for the decision support method, presented in Chapter 6.

*Gap-4* was identified while designing and executing the exploratory survey presented in Chapter 2. We faced various challenges during the design and execution of the survey presented in Chapter 2, which motivated us review and collect problems and mitigation strategies for survey research in software engineering. Researchers have discussed some of these problems and mitigation strategies as part of their research work implicitly but there existed no literature where these problems are strategies to address these problems were discussed in detail.

*Contributions:* Lastly, to address the above gap, a literature review and interview study, presented in Chapter 7, was done to identify the common problems researchers face when designing and executing survey research in software engineering. The contributions made in this study are:

1. The process for design, execution and analysis of the surveys in software engineering is reported

2. Most common problems encountered by the software engineering researchers when conducting surveys are identified through literature and by interviewing researchers with considerable research experience

3. Multiple mitigation strategies are identified and reported in this study

Figure 1.1 provides an overview of the chapters and maps the contributions.

### 1.3.1 Research Questions

The main aim of this thesis is to support decision making in relation to exploration levels in software testing.

To achieve the above stated aim, the studies are designed considering the following objectives:
Chapter 1. Introduction

Obj-1: To search for empirical evidence for applicability of exploratory testing, its usage and perceived usefulness

Obj-2: To provide empirical evidence of effectiveness and efficiency of exploratory testing in comparison to traditional scripted testing

Obj-3: To investigate ways to structure exploratory testing

Obj-4: To provide a classification for levels of exploration in software testing to scope exploratory testing

Obj-5: To provide a decision support method to aid practitioners choose between different levels of exploration

Obj-6 To find mitigation strategies for the problems in conducting surveys in software engineering

The research questions answered in this thesis are:

RQ-1: What is the practitioner perspective for the usefulness of different testing techniques in context of large-scale software?

RQ-2: How to support practitioners in their decisions about choosing exploratory testing versus scripted testing?

RQ-2.1: Is there a difference between exploratory testing and scripted testing in terms effectiveness and efficiency?

RQ-2.2: How to structure exploratory testing to change its image as an ad-hoc approach?

RQ-2.3: What are the key contents that practitioners include in their test charters and what influences the test charter design?

RQ-2.4: How to scope exploration in exploratory testing?

RQ-2.5: How practitioners can decide and choose between different levels of exploration to devise their overall test strategy in context of system under test?

RQ-3: What are the problems and their mitigation strategies in conducting survey research in software engineering?
1.4 Methods

This thesis takes a mixed-method research approach towards the main objective of the thesis. Therefore, each chapter of this thesis corresponds to an individual research study. An overview of the different research methods along with the contributions of individual studies used to answer the main research questions of the thesis is depicted in Figure 1.1.

Brief introduction of the methods used in this thesis is provided below.

Exploratory survey

A survey is used to collect information from multiple individuals to understand different behaviors and trends [26]. An exploratory survey is used as a pre-study to a more in-depth investigation with the objective to not overlook important issues in that area of research [26]. A structured questionnaire is used to gather and analyze information that serves as the basis for further studies.

In exploratory surveys, the goal is not to draw general conclusions about a population through statistical inference based on a representative sample. A representative sample (even for a local survey) has been considered challenging, Thorn [25] points out that: “This [remark by the author: a representative sample] would have been practically impossible, since it is not feasible to characterize all of the variables and properties of all the organizations in order to make a representative sample.” Similar observations and limitations of statistical inference have been discussed by Miller [20].

Chapter 2 reports a research study based on an exploratory survey. The aim of the survey was to gather data from various companies that differ in characteristics.

Controlled experiment

An experiment provides a formal and controlled investigation to compare different treatments in a precise and systematic manner. A number of treatments can be involved in experiments to compare the outcomes [26]. In software engineering, experiments are often conducted involving human subjects that make the design and execution of the experiment challenging. However, experiments can both be used to test existing theories as well as to investigate the validity of different measures.

In this thesis, we conducted an experiment with 70 human subjects from academia and industry to compare the effectiveness and efficiency of two testing techniques. A detailed discussion and experiment design is provided in Chapter 3.
Chapter 1. Introduction

Figure 1.1: Overview of the thesis
Focus groups

We used focus groups as a data collection method for two of the studies conducted in this thesis. During focus group a group of individuals is selected by the researchers to discuss and comment on specific areas of expertise where these individuals have considerable experience. Focus groups help the researchers to understand the topic under research in a better and concise way with a strong involvement of industrial practitioners. The data collected from focus groups is mainly qualitative and traditional approaches for qualitative data analysis are used to analyze the data. However, we used a slightly different method for data analysis known as repertory grid method [7] [18] [21] which emanates from the personal constructs theory in psychological research. The repertory grid method was adapted, in this thesis, to analyze data collected from the focus groups in a quantitative fashion.

Research studies presented in Chapters 5 and 6 make use of focus groups as the data collection method. In both of these studies focus groups from four different industrial partners comprising of software testers with varying experience in software testing were conducted.

Interviews

Interviews were also used as methods for data collection. During the study presented in Chapter 4, interviews were conducted with industry practitioners whereas for study presented in Chapter 7, a number of researchers were interviewed. All interviews were recorded by the consent of the interviewees and later transcribed and coded for analysis purposes. In both cases, we made use of the thematic analysis [6] technique to analyze the data collected during the interviews.

Literature review

Literature review was done as part of all the studies included in this thesis to identify related works done in the areas of software testing, exploratory testing and surveys in software engineering. Literature review was preferred over systematic literature review due to lack of research literature in exploratory testing. This decision was further motivated due to clear gaps in the research area identified in the start of the thesis.

1.5 Overview and Results of Studies

Each chapter in this thesis corresponds to an individual research study as depicted in Figure 1.1. Table 1.1 provides a mapping of methods used in this thesis, objectives,
Table 1.1: Mapping of methods, objectives, research gaps and research questions

<table>
<thead>
<tr>
<th>Method</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Controlled Experiment</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Focus group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Literature review</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objective</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj-1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obj-2</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obj-3</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Obj-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Obj-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Obj-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research Gap</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gap-1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gap-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gap-3</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gap-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research Questions</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2.1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2.3</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research Context</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Academia</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

research gaps and the research questions. The following sections provide an overview of these studies, including research methodology, results and conclusions.
1.5.1 Chapter 2: Testing heterogeneous systems: An exploratory survey

Chapter 2 explores (1) which techniques are frequently discussed in literature in context of large-scale system testing that practitioners use to test their large-scale systems; (2) the perception of the practitioners on the usefulness of the techniques with respect to a defined set of outcome variables.

Survey is used as the research method in this study. A total of 59 answers were received out of which 27 responses were complete survey answers that were eventually used in this study. The most frequently used technique is exploratory manual testing, followed by combinatorial testing. With respect to the perceived performance of the testing techniques, the practitioners were undecided regarding many of the studied variables. Manual exploratory testing received very positive ratings across outcome variables.

Given that the data indicates that practitioners are often undecided with respect to the performance of the techniques, researchers need to support them with comparative studies and sound evidence. In particular, it needs to be investigated whether the perceptions and experiences of the practitioners can be substantiated in more controlled studies.

1.5.2 Chapter 3: An experiment on the effectiveness and efficiency of exploratory testing

As identified in the study presented in Chapter 2, manual exploratory testing is the most used technique used by practitioners in the context of large-scale systems. We conducted a controlled experiment to compare the effectiveness and efficiency of exploratory testing in Chapter 3.

The exploratory testing (ET) approach though widely used by practitioners lacks scientific research. The scientific community needs quantitative results on the performance of ET taken from realistic experimental settings. The objective of this paper is to quantify the effectiveness and efficiency of ET vs. testing with documented test cases (test case based testing, TCT).

We performed four iterations of the controlled experiment where a total of 24 practitioners and 46 students performed manual functional testing using ET and TCT. We measured the number of identified defects in the 90-minute testing sessions, the detection difficulty, severity and types of the detected defects, and the number of false defect reports.

The results show that ET found a significantly greater number of defects. ET also found significantly more defects of varying levels of difficulty, types and severity levels.
Chapter 1. Introduction

However, the two testing approaches did not differ significantly in terms of the number of false defect reports submitted. We conclude that ET was more efficient than TCT in our experiment. ET was also more effective than TCT when detection difficulty, type of defects and severity levels are considered. The two approaches are comparable when it comes to the number of false defect reports submitted.

In summary, the results of Chapter 3 show that ET found a significantly greater number of defects in comparison with TCT. ET also found significantly more defects of varying levels of detection difficulty, types and severity levels. On the other hand, the two testing approaches did not differ significantly in terms of number of false defect reports submitted.

1.5.3 Chapter 4: Checklists to support test charter design in exploratory testing

In earlier studies, and with further discussion with the industry partners, it was identified that there is a strong need to provide structure to ET which in turn would help the practitioners to conduct ET in an effective manner. Test charters can be helpful to scope the exploration in a test session. Therefore, in Chapter 4, we investigated what to include in test charters and what are the main factors that influence the test charter design. Furthermore, we also investigated the factors that influence the test charter design when planning a test session.

Overall, we found 30 different factors that affect the test charter design and 35 different content elements that test practitioners like to include in their test charters. However, it is not feasible for every test charter to include the identified content elements as it would become an overhead and can limit the exploration to a great extent. Therefore, we present two checklists to help practitioners decide what to include in the test charter while designing and planning a test session. This decision is however heavily influenced by the context of the system under test and the test mission.

The results of this study also show that test charters can be an important driver for scoping the exploration in a test session. We identified that ET cannot be seen as just an opposite pole in comparison with the traditional scripted testing approach. Between freestyle exploratory testing and scripted testing, there exists a whole continuum and there is a need to classify different levels of exploration in software testing.

1.5.4 Chapter 5: Exploratory testing: One size doesn’t fit all

The study presented in Chapter 5, is a focus group study where the main objective was to address the gaps identified in Chapter 4 that there is a need to classify different levels of exploration across the exploratory testing continuum. In this study, we conducted
focus groups with four different companies to identify influence factors that affect the test charter design in the context of these four companies. We also classified different levels of exploration in software testing and presented a classification of exploration levels. These five distinct exploration levels are exemplified using different test charter designs and we depicted how exploration can be scoped between completely freestyle ET and scripted testing by adding or removing specific elements of a test charter.

The taxonomy presented in this study was further evaluated with the industrial partners and detailed discussion was done with the practitioners about each of the identified influence factors in which manner the factors influence each of the exploration level presented in the taxonomy. Furthermore, different advantages and disadvantages for using each level of exploration were elicited as shown in Figure 1.2.

The results of this study show that the presented taxonomy helps the test practitioners understand their test objectives in a much better way and can help them decide in an abstract manner when to use a specific exploration level in the context of their systems under test. Different degrees of exploration have specific advantages and disadvantages as shown in Figure 1.2. Practitioners stated that there is a need of a more concrete decision support method that can help them decide about the time distribution between different levels of exploration when designing the overall test strategy.

1.5.5 Chapter 6: A decision support method for recommending degrees of exploration in exploratory testing

Based on the need, for a decision support method to recommend time distribution for degrees of exploration, expressed by the industry partners in the study presented in Chapter 5, we designed a decision support method to provide recommendations for choosing different degrees of exploration presented in the classification in Chapter 5.

In this study (Chapter 6), based on the theory of personal constructs and the resulting repertory grid technique [18], we designed a decision support method. Personal construct theory presents the repertory grid technique to analyze the personal constructs of people representing their own worldviews in a quantitative way. This technique has emerged from the area of psychological research to analyze qualitative data in a quantitative fashion.

The influence factors and the levels of exploration identified in Chapter 5 serve as inputs for the decision support method. Each of the influence factor was divided into a negative and positive pole and practitioners were asked to provide consensus ratings for each of the factors against each level/degree of exploration. For the analysis, instead of the traditional methods for analyzing repertory grid, we asked the practitioners to prioritize the influence factors most important to their testing in the context of their
Chapter 1. Introduction

Figure 1.2: Advantages and disadvantages of exploration levels

<table>
<thead>
<tr>
<th>Exploration Level</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Style ET</td>
<td>(+) Positive experience with outsourcing test only providing the test object. The focus was on a specific system property (security) – otherwise instruction was &quot;see what you find&quot;</td>
</tr>
</tbody>
</table>
| High Degree of Exploration| (+) Less administration and barriers to testing as it is (a) easier to write test cases and (b) the feeling of the need to follow a script (do not dare to deviate)  
|                            | (+) Facilitates better learning                                                                   |
|                            | (+) More resilient to changes as more changes happen on the detailed level and the charter is abstract. |
|                            | (+) Reduces time to inform and discuss changes that would be required on a more detailed level     |
| Medium Degree of Exploration| (+) Medium + High: time savings due to rapid feedback within a testing session, addresses issue of having so many test cases and time needed to run, as well as the limited time available in release cycles (bi-weekly)  
|                            | (+) Medium + High: when new requirements arrive one cannot think about everything and complement that gap of knowledge through exploration and learning |
|                            | (-) Medium + High: hard to trace coverage (cannot tick of test cases run like in scripted tests) |
|                            | (-) Medium + High: challenging to fulfil conformance requirements with standards and legal requirements (easy to miss requirements in exploration) |
|                            | (+) All ET: Scripted test knowledge biases in conducting ET.                                       |
| Low Degree of Exploration  | (+) Easier to trace coverage                                                                          |
|                            | (-) requires a high amount of detailed information which takes time for thinking and gathering that information |
| Fully Scripted             | (+) Easier to trace coverage                                                                          |
|                            | (-) Too exhaustive with large test suite and frequent release cycles                                 |
|                            | (-) Time-intensive to run with a high number of scripted test cases                                  |
products and the associated test goals. Further, we calculated percentages based on the priority of the influence factors and the consensus ratings for each of these factors applied to the exploration levels. Based on these percentages, we provide recommendations to the practitioners what levels of exploration shall be used in their context and how to distribute time between different levels of exploration.

The decision support method presented in this study was further evaluated with the industry partners and a comparison of their existing test strategy and the recommended test strategy was done.

1.5.6 Chapter 7: Survey research in software engineering: Problems and strategies

During the execution of exploratory survey done in Chapter 2, we identified that there exists various challenges in designing and executing a survey study in software engineering. Due to the increasing need for empirical investigations in software engineering, many researchers nowadays make use of survey research to conduct their research and validate different solutions they propose. One of the most common problems faced in software engineering surveys is insufficient sample size that in turns hinders in drawing generalizable conclusions. Apart from this challenge, various problems are mentioned implicitly in the software engineering research but there existed no research study that discussed most common problems and mitigation strategies; faced by the researchers during the design and execution of software engineering surveys.

To bridge this gap, in the study presented in Chapter 7, a literature review of existing surveys in software engineering was done to identify potential problems and corresponding mitigation strategies. Furthermore, interviews were done with researchers having experience of designing and executing survey studies.

In this study, we identified 24 problems and 65 strategies covering the overall survey process. The elicited problems and strategies will help software engineering researchers to design and execute their survey studies while being aware of existing problems and the associated mitigation strategies.

1.6 Conclusions

Based on the findings in this thesis, the following conclusions are drawn:

RQ-1: What is the practitioner perspective for the usefulness of different testing techniques in context of large-scale software?
Chapter 1. Introduction

- Manual exploratory testing is most used technique, but it is least investigated technique in academia compared to other two techniques identified in this thesis. This provides an opportunity to study the technique and compared with scripted testing.

- Given that, there are positive indications of the use of search-based testing by the industry practitioners, the focus should also be on understanding how and with what success search-based testing can be adopted for testing large-scale in industry.

**RQ-2: How to support practitioners in their decisions about choosing exploratory testing versus scripted testing?**

**RQ-2.1: Is there a difference between exploratory testing and scripted testing in terms effectiveness and efficiency?**

In this study, we executed a total of four experiment iterations (one in academia and three in industry) to compare the efficiency and effectiveness of exploratory testing (ET) in comparison with scripted testing. Efficiency was measured in terms of total number of defects identified using the two approaches (duration of test sessions was 90 minutes each), while effectiveness was measured in terms of defect detection difficulty, technical type of defects, severity levels and number of false defect reports.

Our experimental data shows that ET was more efficient than scripted testing in finding more defects in a given time. ET was also found to be more effective than scripted testing in terms of defect detection difficulty, technical types of defects identified and their severity levels; however, there were no statistically significant differences between the two approaches in terms of the number of false defect reports. The experimental data also showed that in terms of type of subject groups, there are no differences with respect to efficiency and effectiveness for both ET and scripted testing.

We acknowledge that documenting detailed test cases in scripted testing is not a waste but, as the results of this study show, more test cases is not always directly proportional to total defects detected. Hence, one could claim that it is more productive to spend time testing and finding defects rather than documenting the tests in detail.

**RQ-2.2: How to structure exploratory testing to change its image as an ad-hoc approach?**

- Session-based test management (SBTM) is an enhancement to exploratory testing. SBTM provides basic guidelines to conduct exploratory testing and lists
down some key elements including time-boxing of a test session, clear identification of test goal and test charter.

- In order to structure exploratory testing, the most important element is test charter. However, as we found out, test charters are designed through an individualized approach and different individuals design test charters based on their personal opinions. There exist different types of test charters varying from very abstract test mission statements to detailed test plans.

**RQ-2.3: What are the key contents that practitioners include in their test charters and what influences the test charter design?**

To answer this research question, two checklists for test charter design were developed. The checklists were based on nine interviews. The interviews were utilized to gather a checklist for factors influencing test charter design and one to describe the possible contents of test charters. Overall, 30 factors and 35 content types have been identified and categorized.

The factors may be used in a similar manner and should be used to question the design choices of the test charter. For example:

- Should the test focus of the charter be influenced by previous bugs? How/why?
- Are the product’s goals reflected in the charter?
- Is it possible to achieve the test charter mission in the given time for the test session?

With regard to the content a wide range of possible contents to be included have been presented. For example, only stating the testing goals provides much room for exploration, while adding the techniques to be used may constrain the tester. Thus, the more information is included in the test charter the exploration space is reduced. Thus, when deciding what to include from the checklist (Table 4.3) the possibility to explore should be taken into consideration.

**RQ-2.4: How to scope exploration in exploratory testing?**

We identified that scripted testing and freestyle exploratory testing are two opposite extremes of a testing continuum and there are multiple levels in between these two. Therefore, a taxonomy was developed that presents five different levels of exploration in software testing. These levels of exploration are represented by different test charter types with variable characteristics and elements. The more elements included in the test charter design, the less exploration space is provided to the tester. This taxonomy was
evaluated with the help of industry partners and it was identified that the best suited levels of exploration in their context are medium and high degree of exploration.

**RQ-2.5: How practitioners can decide and choose between different levels of exploration to devise their overall test strategy in context of system under test?**

A need for the decision support method was identified earlier in this thesis. In the study presented in Chapter 6, a decision support method was developed to help practitioners choose the levels of exploration for software testing. This method was developed by making use of the levels of exploration presented in the taxonomy developed in Chapter 5 and eliciting the personal constructs of the testers in the focus groups.

The decision support method provides recommendation to the testers about how to distribute their time for different test activities across the software testing continuum ranging from freestyle exploratory testing and scripted testing. These recommendation are based on contextual factors prioritized by the testers that best fit the context of their product.

**RQ-3: What are the problems and their mitigation strategies in conducting survey research in software engineering?**

To answer this research question, the study presented in Chapter 7 was done. We identified problems and related strategies to overcome the problems with the aim of supporting researchers conducting software engineering surveys. The focus was on questionnaire-based research.

We collected data from multiple sources, namely existing guidelines for survey research, primary studies conducting surveys and reporting on the problems and strategies of how to address them, as well as expert researchers. Nine expert researchers were interviewed.

In total we identified 24 problems and 65 strategies. The problems and strategies are grouped based on the phases of the survey research process.

- **Target audience and sampling frame definition and sampling plan:** It was evident that the problem of insufficient sample sizes was the most discussed problem with the highest number of strategies associated with it (26 strategies). Example strategies are brevity (limiting the length of the survey), highlighting the social benefit, using third party advertising, and the use of the personal network to recruit responses. Different sampling strategies have been discussed (e.g. random and convenience sampling). In addition more specific problems leading to losses of in responses were highlighted, such as confidentiality issues, gate-keeper reliability, and the lack of explicit motivations of the practical usefulness of the
survey results.

- **Survey instrument design, evaluation, and execution:** The main problem observed was poor wording of questions, as well as different issues related to biases (such as question-order effect, evaluation apprehension, and mono-operation, over-estimation, and social desirability biases). The strategies were mainly concerned with recommendations for the attributes of questions and what type of questions to avoid (e.g. loaded and sensitive questions), as well as the need for pre-testing the surveys. It was also highlighted that expert discussions are helpful in improving the survey instrument.

- **Data analysis and conclusions:** For data analysis the main problems were the elimination of invalid and duplicate responses as well as inaccuracy of data extraction and analysis. Technical solutions were suggested for the detection of detecting duplications. Invalid responses are avoided through consistency checking and voluntary participation. Finally, the importance of involving multiple researchers in the data analysis has been highlighted.

- **Reporting:** Missing information was highlighted as problematic, including the lack of motivation for the selection of samples. It was also highlighted to report inconsistencies and biases that may have occurred in the survey.

A high number of problems as well as strategies has been elicited. In future work a consensus building activity is needed where the community discusses which strategies are most important and suitable for software engineering research. In addition, in combination with existing guidelines the information provided in chapter 7 may serve for the design of checklists to support the planning, conduct, and assessment of surveys.

### 1.7 Future Work

For future work, we propose to focus on strategic decision support for software testing. A strategic decision is to select the most suitable type of testing given a range of contextual factors. To this end, a classification of exploration degrees and corresponding test charters have been proposed. In addition, a decision making approach for distributing the time between different exploration degrees has been proposed and evaluated in industry practice. Based on these previous contributions, we plan to investigate two types of decisions, namely: automated versus manual testing and independent (external) vs. dependent testing.
1.8 Contribution Statement

Author’s Contribution: Ahmad Nauman Ghazi is the first author of all the papers except one paper that constitutes Chapter 3.

The role of first author was mainly the idea creation, designing the research studies, designing the instruments for data collection, execution, analysis of results and reporting.

Role of supervisors: The supervisors; Prof. Kai Petersen and Prof. Jürgen Böstler, were mainly involved in discussion and provided continuous feedback to help improve the ideas. They also reviewed all the work and provided constructive feedback before submitting the papers.

Role of co-authors: The main role of co-authors has been to discuss and further refine the ideas. Providing support in data collection and analysis as well as providing feedback on the manuscripts to refine the final paper.

A detail of authors’ contributions in individual studies is presented below:

Study presented in chapter 2, was designed, executed and reported by Ahmad Nauman Ghazi. Prof. Kai Petersen and Prof. Jürgen Böstler supported in designing the survey instrument and later reviewed the paper and provided their feedback for refining the paper.

For the paper titled “An Experiment of the Effectiveness and Efficiency of Exploratory Testing”, the main idea was initiated by Ahmad Nauman Ghazi followed by design, execution, analysis and reporting of results. Khurram Bhatti provided support in the data collection during 3 out of 4 iterations of the controlled experiment. The test object was provided by Dr. Juha Itkonen. He also provided his feedback to address the reviewers’ comments before the final publication of the paper. Prof. Richard Torkar and Prof. Anneliese Andrews were mainly involved in discussion about experiment settings. Dr. Wasif Afzal participated in the analysis phase, and in the preparation and refinement of the manuscript.

Paper titled “Checklists to Support Test Charter Design in Exploratory Testing” is presented in chapter 4. The idea for this study was initiated by Ahmad Nauman Ghazi. Ratna P. Garigapati supported the instrument design and data collection for the interviews. Results from this study were later analyzed and subsequently reported by Ahmad Nauman Ghazi and Prof. Kai Petersen.

In the study presented in chapter 5, the idea was initiated through discussions between Ahmad Nauman Ghazi and Prof. Kai Petersen, while reflecting over the interview data of previous study. Ahmad Nauman Ghazi with the support of Prof. Kai Petersen and Dr. Elizabeth Bjarnason conducted the focus groups. The results were later analyzed and reported by Nauman which were further refined through feedback provided by Prof. Kai Petersen, Dr. Elizabeth Bjarnason and Prof. Per Runseson.
Based on the findings from chapter 5, the idea for study presented in chapter 6 was initiated by Prof. Claes Wohlin. Prof. Kai Petersen enhanced the idea based on the repertory grid technique. Later, Ahmad Nauman Ghazi, Prof. Kai Petersen and Dr. Elizabeth Bjarnason conducted focus groups to present and evaluate the developed method to each industrial partner. Prof. Claes Wohlin supported in review, reporting and further refining the study.

The idea for study presented in chapter 7 was initiated by Ahmad Nauman Ghazi. Sri Sai Vijay Raj Reddy and Harini Nekkanti helped in data collection through interviews and literature review. The results were later analyzed and reported by Ahmad Nauman Ghazi and Prof. Kai Petersen.
1.9 References


REFERENCES


Chapter 2

Testing of Heterogeneous Systems: An Exploratory Survey

Ahmad Nauman Ghazi, Kai Petersen and Jürgen Börstler
Submitted to a conference

Abstract: Heterogeneous systems comprising sets of inherent subsystems are challenging to integrate. In particular, testing for interoperability and conformance is a challenge. Furthermore, the complexities of such systems amplify traditional testing challenges. We explore (1) which techniques are frequently discussed in literature in context of heterogeneous system testing that practitioners use to test their heterogeneous systems; (2) the perception of the practitioners on the usefulness of the techniques with respect to a defined set of outcome variables. For that we conducted an exploratory survey. A total of 27 complete survey answers have been received. Search-based testing has been used by 14 out of 27 respondents, indicating practical relevance of the approach for testing heterogeneous systems, which itself is relatively new and has only recently been studied extensively. The most frequently used technique is exploratory manual testing, followed by combinatorial testing. With respect to the perceived performance of the testing techniques, the practitioners were undecided regarding many of the studied variables. Manual exploratory testing received very positive ratings across outcome variables.
Chapter 2. Testing of Heterogeneous Systems: An Exploratory Survey

2.1 Introduction

Over the years, software has evolved from simple applications to large and complex systems of systems [8]. A system of systems consists of a set of individual systems that together form a new system. The system of systems could contain hardware as well as software systems. Recently, systems of systems has emerged as a highly relevant topic of interest in the software engineering research community investigating its implications for the whole development life cycle. For instance, in the context of systems of systems, Lane [17] studied the impact on development effort, Ali et al. [2] investigated testing, and Lewis et al. [19] proposed a process of how to conduct requirements engineering.

Systems of systems often exhibit heterogeneity [18], for instance in implementation, hardware, process and verification. For the purpose of this study we define a heterogeneous system as a system comprised of multiple systems (system of systems) where at least one subsystem exhibits heterogeneity with respect to the other systems [12]. The system of systems approach taken in development of heterogeneous systems give rise to various challenges due to continuous change in configurations and multiple interactions between the functionally independent subsystems. The challenges posed to testing of heterogeneous systems are mainly related to interoperability [39, 25], conformance [25] and large regression test suites [6, 2]. Furthermore, the inherent complexities of heterogeneous systems also pose challenges to the specification, selection and execution of tests.

In recent years, together with the emergence of system of systems research testing of heterogeneous systems received an increased attention from the research community. However, solutions proposed have been primarily evaluated from the academic perspective, and not the viewpoint of the practitioner.

In this study, we explored the viewpoint of practitioners with respect to testing heterogeneous systems. Two main contributions are made:

- **Explore which testing techniques investigated in research are used by practitioners.** Thereby, we learn which techniques practitioners are aware of, and which ones are most accepted.

- **Explore the perception of the practitioners of how well the used techniques perform with respect to a specified and frequently studied set of outcome variables.** Understanding the practitioners’ perception of the techniques relative to each other allows to identify preferences from the practitioners’ viewpoint. The findings will provide interesting pointers for future work to understand the reasons for the findings, and improve the techniques accordingly.
The contributions are made by using an exploratory survey to capture the opinion of practitioners.

The remainder of the paper is structured as follows: Section 2.2 presents the related work. Section 2.3 outlines the research method, followed by the results in Section 2.4. Section 2.6 presents a discussion of observations from the results. Finally, in Section 2.6, we conclude this study.

2.2 Related work

The related work focuses on testing of heterogeneous systems, first discussing testing of heterogeneous systems as such, followed by reviewing solutions of how to test them. However, no surveys could be found that discuss any aspect of testing of heterogeneous systems.

2.2.1 Testing in Heterogeneous Systems

Testing heterogeneous systems is primarily considered to be a challenge emanating from the problem of integration and system-level testing [9] [36]. Therefore, the current research in the area of heterogeneous systems considers it as a subsystem interaction issue [36]. It is also observed that solving the inherent complexities underlying the testing heterogeneous systems is not a priority, most of the related research is focused on addressing the accidental complexities in testing of heterogeneous systems by tuning and optimizing different testing techniques and methods.

A number of research studies discuss system-level testing in general terms without addressing specific test objectives. For automated functional testing, Donini et al. [9] propose a test framework where functional testing is conducted in an external simulated environment based on service-oriented architectures. This demonstrated that functional system testing through simulated environments can be an approach to overcome the challenge of minimizing test sets and obtained test cases are representative of real operation of the system. Wang et al. [36] study heterogeneous systems that exhibit heterogeneity at the platform level and discussed different factors considered in system-level testing of heterogeneous systems. Other than the studies focusing on system and integration testing, a relatively small set of studies attempt to discuss the problem of testing in heterogeneous systems in other test phases. Mao et al. [20] study this problem in the unit test phase whereas Diaz [7] addresses the problem of testing heterogeneous systems in the acceptance testing phase.

Research literature related to testing of heterogeneous systems frequently discusses the interoperability as a common issue. Interoperability testing is also a key test objec-
ative in different applications and technology domains. Xia et al. [39] address the interoperability problem in the web service domain and propose a test method to automate conformance and interoperability testing for e-business specification languages. Narita et al. [25] propose a method supported by a testing framework for interoperability testing for web service domain focusing on communication in robotics domain. However, interoperability remains a challenge in other domains as well. In context of large scale component based systems, Piel et al. [31] present a virtual component testing technique and demonstrated how virtual components can be formed using three different algorithms. This technique was further implemented and evaluated in industrial settings. Furthermore, Kindrick et al. [16] propose a technique combining interoperability testing with conformance testing and conclude that combining the two techniques will reduce the cost of setting up and executing the test management processes improving the effectiveness.

2.2.2 Testing Techniques

In an ongoing systematic literature review three main groups of techniques have been identified that were used to test heterogeneous systems, namely manual exploratory testing, combinatorial testing, and search-based testing. There are more refinements of these categorized techniques.

**Manual Exploratory testing:** Manual exploratory testing (ET) is an approach to test software without pre-defined test cases in contrast with traditional test case based testing. The main characteristics of exploratory testing are simultaneous learning, test design and execution [35, 14]. The tester has the freedom to dynamically design, modify and execute the tests.

In past, exploratory testing was seen as an ad-hoc approach to test software. However, over the years, ET has evolved into a more manageable and structured approach without compromising the freedom of testers to explore, learn and execute the tests in parallel. An empirical study comparing the effectiveness of exploratory testing with test-case based testing was conducted by Bhatti and Ghazi [4] and further extended (cf. [1]). This empirical work concludes that ET produces more defects as compared to test case based testing where time to test is a constraint.

**Combinatorial Testing:** Combinatorial testing is used to test applications for different test objectives at multiple levels. A comprehensive survey and discussion is provided by Nie and Leung [26]. It has been used for both unit and system-level testing in various domains. Combinatorial testing tends to reduce the effort and cost for effective test generation [5]. There exist a number of variants of combinatorial testing, which are used in different domains to test heterogeneous systems.
The problem of testing web services is the most common area in heterogeneous systems that is addressed in literature using different test techniques as discussed in section 2.2.1. Some researchers addressed this problem as a combinatorial testing problem instead of an interoperability issue. Mao et al. [20] and Apilli [3] proposed different frameworks for combinatorial testing to test component based software systems in a web services domain.

Wang et al. [37] study the problem of how interaction faults can be located based on combinatorial testing rather than manual detection and propose a technique for interactive adaptive fault location. Results from this study show that the proposed technique performs better than the existing adaptive fault location techniques.

Changing configurations pose challenges to combinatorial testing techniques. To that end Cohen et al. [6] conducted an empirical study to quantify the effectiveness of test suites. The study shows that there is an exponential growth of test cases when configurations change and subsets of test suites are used, similar to what is common in regression testing.

Mirarab et al. [24] conducted an industrial case study and propose a set of techniques for requirement-based testing. The SUT was software for a range of wireless, mobile devices. They propose a technique to model requirements, a technique for automated generation of tests using combination strategies, and a technique for prioritization of existing test cases for regression testing.

Search-Based Software Testing: Marin et al. [21] present an integrated approach where search-based techniques are applied on top of more classical techniques to derive optimal test configurations for web applications. The authors describe state of art and future web applications as complex and distributed, exhibiting several dimensions of heterogeneity, which all together require new and integrated approaches to test the systems with a criteria to be optimal with respect to coverage vs. effort. The study describes an approach that integrates combinatorial testing, concurrency testing, oracle learning, coverage analysis, and regression testing with search-based testing to generate test cases.

Shiba et al. [33], proposed two artificial life algorithms to generate minimal test sets for r-way combinatorial testing based on a genetic algorithm (GA) and an ant colony algorithm (ACA). Experimental results show that when compared to existing algorithms including AETG (Automatic Efficient Test Generator) [5], simulated annealing-based algorithm (SA) and in-parameter order algorithm (IPO), this technique works effectively in terms of size of test set as well as time to execute.

Another study by Pan et al. [28] explores search-based techniques and defines a novel algorithm, i.e., OEPST (organizational evolutionary particle swarm technique), to generate test cases for combinatorial testing. This algorithm combines the characteristics of organizational evolutionary idea and particle swarm optimization algorithm.
Chapter 2. Testing of Heterogeneous Systems: An Exploratory Survey

The experimental results of this study show that using this new algorithm can reduce the number of test cases significantly.

2.3 Research method

The survey method used in this study is an exploratory survey. Thörn [34] distinguishes statistical and exploratory surveys.

In statistical surveys the goal is not to draw general conclusion about a population through statistical inference based on a representative sample. A representative sample (even for a local survey) has been considered challenging, the author [34] points out that: “This [remark by the authors: a representative sample] would have been practically impossible, since it is not feasible to characterize all of the variables and properties of all the organizations in order to make a representative sample.” Similar observations and limitations of statistical inference have been discussed by Miller [23].

Given that the focus of this research is specific to heterogeneous systems, the population is limited. We were aware of specific companies and practitioners that work with such systems, but the characteristics of companies and their products were not available to us. Hence, an exploratory survey was conducted to answer our research questions. Though, aim was to gather data from companies with different characteristics; different domains, sizes, etc. represented; for the obtained answers, external validity is discussed in Section 2.3.5.

2.3.1 Study purpose

The goal of the survey is formulated based on the template suggested in [38] to define the goals of empirical studies. The goal for this survey is to explore the testing of heterogeneous systems with respect to the usage and perceived usefulness of testing techniques used for heterogeneous systems from the point of view of industry practitioners in the context of practitioners involved in heterogeneous system development reporting their experience on heterogeneous system testing.

In relation to the research goal two main research questions were asked:

1. Which testing techniques are used to evaluate heterogeneous systems?

2. How do practitioners perceive the identified techniques with respect to a set of outcome variables?
2.3.2 Survey Distribution and Sample

We used convenience sampling to obtain the answers. Of interest were practitioners that were involved in the testing of heterogeneous systems before, thus not every software tester would be a suitable candidate for answering the survey. The sample was obtained through personal contacts as well as postings in software engineering web communities (e.g. LinkedIn and Yahoo Groups). 100 personal contacts were asked to respond, and to distribute the survey later. Furthermore, we posted the survey on 32 communities.

Overall, we obtained 42 answers, of which 27 were complete and valid. One answer was invalid as each response was given as “others”, without any further specification. The remaining respondents did not complete the survey. We provide further details on the respondents and their organizations in Section 2.4.1.

2.3.3 Instrument Design

The survey instrument is structured along the following themes.

- **Respondents**: In this theme information about the respondent is collected. This information is comprised of: current position; duration of working in the current position in years; duration of working with software development; duration of working with testing heterogeneous systems.

- **Company, processes, and systems**: This theme focuses on the respondents’ organizations and the characteristics of the products.

- **Test coverage**: Here the practitioners rate the importance of different coverage criteria on a 5-point Likert scale from “Very Important” to “Unimportant”. The coverage criteria rated were specification-based, code-based, fault-based, and usage-based.

- **Usage of testing techniques**: We identified three categories of testing techniques through our ongoing systematic literature review that have been attributed and used in testing heterogeneous systems, namely search-based, combinatorial, and manual exploratory testing (see also Section 2.2). The concepts of the testing techniques were defined in the survey to avoid any confusion. Two aspects have been captured, usage and evaluation. With respect to usage we asked for the frequency of using the different techniques on a 7-point Likert scale ranking from “Always” to “Never”. We also provided the option “Do not know the technique”.

---

1 The survey can be found at https://www.surveymonkey.com/s/RP6DQKF


Chapter 2. Testing of Heterogeneous Systems: An Exploratory Survey

- *Usefulness of testing techniques:* Each technique has been rated according to its usefulness with respect to a set of outcome variables that are frequently studied in literature on quality assurance techniques. The usefulness for each technique for each variable was rated on a 5-point Likert scale from “Strongly Agree” to “Strongly Disagree”. Table 2.1 provides an overview of the studied variables and their definitions.

- *Contact details:* We asked the respondents for their company name and e-mail address. The answer to this question was optional in case the respondents wished to stay anonymous towards the researchers.

<table>
<thead>
<tr>
<th>Variable</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of use</td>
<td>[15] [30]</td>
</tr>
<tr>
<td>Effectiveness in detecting critical defects</td>
<td>[1]</td>
</tr>
<tr>
<td>Number of false positives</td>
<td>[1]</td>
</tr>
<tr>
<td>Effectiveness in detecting various types of defects</td>
<td>[1]</td>
</tr>
<tr>
<td>Time and cost efficiency</td>
<td>[1] [30]</td>
</tr>
<tr>
<td>Effectiveness in detecting interoperability issues</td>
<td>[29]</td>
</tr>
<tr>
<td>Effectiveness for very large regression test sets</td>
<td>[13]</td>
</tr>
<tr>
<td>External product quality</td>
<td>[27]</td>
</tr>
</tbody>
</table>

The design of the survey has been pretested by three external practitioners and one researcher. The feedback led to minor reformulation and changes in the terminology used to become clear for practitioners. Furthermore, the number of response variables has been reduced to make the survey manageable in time and avoid maturation. Furthermore, the definition of heterogeneous system was revised to be more understandable. We further measured the time the respondents needed in the pretest to complete the survey. The time was between 10 and 15 minutes.

2.3.4 Analysis

For reflection on the data (not for inference) we utilized statistical tests to highlight differences for the techniques surveyed across the outcome variables. The Friedman test [11] (non-parametric test) has been chosen given multiple variables (treatments) were studied, the data being on ordinal scale.
2.3.5 Validity Threats

**Internal Validity** One threat to capturing truthfully is if the questions asked in the survey are misunderstood. To reduce this threat we pretested the survey and made updates based on the feedback received. Another threat is maturation where the behavior changes over time. This threat has been reduced by designing the survey so that no more than 15 minutes were necessary to answer the survey.

**Construct Validity** Theoretical validity is concerned with not being able to capture what we intend to capture (in this case the usefulness of different techniques across different outcome variables). To reduce this threat we defined variables based on literature, in particular focusing on variables that are frequently studied when evaluating quality assurance approaches. Given that the study is based on the subjects’ experience, the lack of experience in search-based testing limits the comparability, given that eight respondents did not know the technique, and five have never used it. However, the remaining respondents had experience using it. For the other techniques (manual exploratory testing and combinatorial testing) only few respondents did not know them, or lacked experience. Given that the aim of the study is not to generalize the findings through inference, but rather identify interesting patterns and observations in an exploratory way, threats related to statistical inference were not emphasized.

**External Validity** The exploratory nature of the survey does not allow to statistically generalize to a population. However, as suggested by [34], interesting qualitative arguments can be made such studies. The context captured in the demographics of the survey limits the external generalizability. In particular, the majority of respondents were related to the consulting industry (35.7%), followed by computer industry (28.6%), and communications (25.0%), other industries only have very few responses and are not represented in this study (e.g. accounting, advertising, etc.). With regard to company size, all four size categories are equally well represented. With regard to development models agile and hybrid processes have the highest representation. Only limited conclusions can be drawn about other models.

**Conclusion Validity** Interpretive validity is primarily concerned with conclusions based on statistical analysis, and researcher bias when drawing conclusions. Given that the involved researchers have no particular preference on any of the solutions surveyed based on previous research, this threat can be considered as being under control.
Chapter 2. Testing of Heterogeneous Systems: An Exploratory Survey

2.4 Results

2.4.1 Context

Subjects: Table 2.2 provides an overview of the primary roles of the subjects participating in the survey. The roles most frequently presented are directly related with either quality assurance, or the construction and design of the system. Overall, the experience in years in the current role indicates a fair to strong experience level of the respondents in their current positions.

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Percent</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software developer (implementation, coding etc.)</td>
<td>22.2%</td>
<td>6</td>
</tr>
<tr>
<td>Software architect (software structure, architecture, and design)</td>
<td>18.5%</td>
<td>5</td>
</tr>
<tr>
<td>Software verification &amp; validation (testing, inspection, reviews etc.)</td>
<td>18.5%</td>
<td>5</td>
</tr>
<tr>
<td>Software quality assurance (quality control, quality management etc.)</td>
<td>14.8%</td>
<td>4</td>
</tr>
<tr>
<td>Other</td>
<td>11.1%</td>
<td>3</td>
</tr>
<tr>
<td>System analyst (requirements elicitation, analysis, specification and validation etc.)</td>
<td>7.4%</td>
<td>2</td>
</tr>
<tr>
<td>Project manager (project planning, project measurement etc.)</td>
<td>3.7%</td>
<td>1</td>
</tr>
<tr>
<td>Product manager (planning, forecasting, and marketing software products etc.)</td>
<td>0.0%</td>
<td>0</td>
</tr>
<tr>
<td>Software process engineer (process implementation and change, process and product measurement etc.)</td>
<td>0.0%</td>
<td>0</td>
</tr>
</tbody>
</table>

Looking at the overall experience related to software engineering in years, the average experience is 10.55 years with a standard deviation of 7.04. This indicates that the overall experience in software development is very high.

We also asked for the experience of the practitioners in testing heterogeneous systems themselves. The average experience in testing heterogeneous systems is 4.63 years with a standard deviation of 5.22, while 8 respondents did not have experience as testers on heterogeneous systems themselves. The survey focused on practitioners involved in developing heterogeneous systems though, as those also often gain insights on the quality assurance processes (e.g. people in quality management). Hence, those responses were not excluded.

Company, processes, and systems: The number of responses in relation to company size are shown in Table 2.3. All sizes are represented well by the respondents, hence the results are not biased towards a specific company size.

The companies surveyed worked in 24 different industry sectors (one company can work in several sectors, hence multiple answers were possible). The industries that were represented by the highest number of respondents were consulting (9 respondents), computer industry (hardware and desktop software) (7 respondents), communications (6 respondents), and business/professional services (5 respondents).
The systems developed are characterized by different types as specified in [10]. As shown in Table 2.4 the clear majority of respondents was involved in data-dominant software development, though all types were represented through the surveyed practitioners.

The development models used in the surveyed companies are illustrated in Table 2.5. The clear majority of respondents is working with agile development and hybrid processes that are dominated by agile practices.

**Test coverage**: A key aspect of testing is the test objectives that drive the selection of test cases (cf. [22]). We captured the objectives of the participating industry practitioners in their test case selection as shown in Figure 2.1. Specification-based coverage is clearly the most important criterion for the studied companies, followed by
fault-based coverage. Overall, all coverage objectives are considered important by at least half of the participants.

![Figure 2.1: Importance of test objectives](image)

### 2.4.2 RQ1: Usage of Testing Techniques

We captured the frequency of usage for the three different techniques introduced earlier (search-based, manual exploratory, and combinatorial testing). The frequencies are illustrated in Figure 2.2.

Looking at the overall distribution of usage, it is clearly visible that manual exploratory testing is the most frequently used technique, followed by combinatorial testing and search-based testing. There was not a single respondent indicating of never having used manual exploratory testing.

Search-based testing is the least-used technique, as well as the technique that is least-known. However, 3 respondents who mentioned that they always use search-based testing are all test consultants. Another consultant mentioned frequent usage of the technique along with 2 more respondents who are in education and professional services industries respectively. Only very few respondents are not aware of manual exploratory and combinatorial testing.
Figure 2.3 provides the rating of the variables for the three different techniques studied. To highlight patterns in the data, we also used statistical testing as discussed in Section 2.3.4. The results of the test are shown in Table 2.6.

The highest undecided rates are observed for search-based testing. This can be explained by the observation that people were not aware of the technique, or never used it (see Figure 2.2). Also, a relatively high undecided rate can be seen for combinatorial testing, however, this cannot be attributed to the lack of knowledge about the technique, or that practitioners never used it, as the numbers on both items were relatively low. The opposite is true for manual exploratory testing, where only very few practitioners were undecided.

Variables that are more unique and emphasized for heterogeneous systems (effectiveness in detecting interoperability issues and effectiveness for very large regression test sets) have higher undecided rates for all the techniques. That is, there is a high level of uncertainty across techniques. In the case of regression tests manual exploratory testing was perceived as the most ineffective. For interoperability testing no major difference between the ratings can be observed, which is also indicated by the statistical tests shown in Table 2.6.
Table 2.3: Practitioners' Perceptions of Testing Techniques for Heterogeneous Systems (1 = Strongly Disagree, 5 = Strongly Agree)

<table>
<thead>
<tr>
<th>Perception</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing to improve product quality</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>10%</td>
<td>15%</td>
</tr>
<tr>
<td>Effectiveness of using existing tools</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Time and cost efficiency</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
<td>20%</td>
<td>10%</td>
</tr>
<tr>
<td>Effectiveness in debugging software issues</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
<td>20%</td>
<td>10%</td>
</tr>
<tr>
<td>High number of legacy problems</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Changes in documenting test cases</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Easy to use</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
</tbody>
</table>
Of all techniques, manual exploratory testing is rated exceptionally high in comparison to other techniques for ease of use, effectiveness in detecting critical defects, detecting various types of defects, and in improving product quality. The high rating is also highlighted through the statistical tests, which detected this as a difference in the data sets (see Table 2.6). At the same time, it also received the strongest negative ratings, which was the case for false positives and effectiveness for very large regression test suites.

Table 2.6: Friedman test statistics

<table>
<thead>
<tr>
<th>Item</th>
<th>N</th>
<th>$\chi^2$</th>
<th>df</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use</td>
<td>27</td>
<td>22.522</td>
<td>2</td>
<td>0.000</td>
</tr>
<tr>
<td>Effective in detecting critical defects</td>
<td>27</td>
<td>19.500</td>
<td>2</td>
<td>0.000</td>
</tr>
<tr>
<td>High number of false positives</td>
<td>27</td>
<td>0.090</td>
<td>2</td>
<td>0.956</td>
</tr>
<tr>
<td>Effective in detecting various types of defects</td>
<td>27</td>
<td>17.848</td>
<td>2</td>
<td>0.000</td>
</tr>
<tr>
<td>Time and cost efficiency</td>
<td>27</td>
<td>3.797</td>
<td>2</td>
<td>0.150</td>
</tr>
<tr>
<td>Effective in detecting interoperability issues</td>
<td>27</td>
<td>7.000</td>
<td>2</td>
<td>0.030</td>
</tr>
<tr>
<td>Effective for very large regression test sets</td>
<td>27</td>
<td>1.509</td>
<td>2</td>
<td>0.470</td>
</tr>
<tr>
<td>Helping to improve product quality</td>
<td>27</td>
<td>25.400</td>
<td>2</td>
<td>0.000</td>
</tr>
</tbody>
</table>

2.5 Discussion

Based on the data collected we highlight interesting observations, and present their implications.

**Observation 1:** Interestingly, search-based testing was applied by several practitioners in the scoped application of testing heterogeneous systems (in total 14 of 27 used it at least very rarely), even though in comparison it was the least frequently applied technique. Literature surveying research on search-based testing reported acknowledges that testing is primarily a manual process [22]. Also, in heterogeneous systems we only identified few studies in our search for literature that used search-based testing. Hence, it is an interesting observation that companies are using search-based testing. At the same time, many practitioners were not aware of it at all. This leads to the following lessons learned:

**Lessons learned:** First, given the presence of search-based testing in industry, there exist opportunities for researchers to study it in real industrial environments and to collect experiences made by practitioners; Second, practical relevance of
search-based testing in heterogeneous testing is indicated by the adoption of the technique, which is encouraging for this relatively new field.

**Observation 2:** Although, the survey was targeted towards a specific group of practitioners that have experience with developing and testing heterogeneous systems, the practitioners were largely undecided on whether the techniques used are suitable for detecting interoperability issues. Figure 2.3 shows that search-based testing has comparatively high undecided rates for all the variables.

**Lessons learned:** Practitioners require further decision support and comparisons to be able to make informed decisions about the techniques given the high level of uncertainty. In particular, further comparative studies (which were lacking) are needed in general, and for heterogeneous systems in particular. If people are undecided, adoption is also hindered; hence one should aim to reduce the uncertainty on outcomes for the variables studied.

**Observation 3:** Manual exploratory testing is perceived as very positive by practitioners for the variables “Ease of use”, “Effective in detecting critical defects”, “Effective in detecting various types of defects”, “Time and cost effective” and “Helping to improve product quality”. On the other hand, it has been perceived poorly in comparison to other techniques for the variables “High number of false positives” and “Effective for very large regression-test suites”. Given the context of testing heterogeneous systems, these observations are interesting to compare with findings of studies investigating exploratory testing. Shah et al. [32] investigated exploratory testing and contrasted the benefits and advantages of exploratory and scripted testing through the application of a systematic review combined with expert interviews. Their review is hence used as a basis for the comparison with literature.

The finding with respect to ease of use was understandable, but could also be seen as a paradox. On the one hand there are no perceived barriers as one does not have to learn testing techniques; however, the quality of tests is not known because there is such a high dependency on the skills of the testers (cf. Shah et al. [32]), which could potentially lead to a wrong perception. Shah et al. identified multiple studies indicating time and cost efficiency, and also confirmed that the exploratory testing is good at identifying the most critical defects. Overall, this appears to be well in-line with the findings for heterogeneous systems. With respect to false positives, the practitioners were in disagreement on whether manual exploratory testing leads to a high number
of false positives. Literature on the other hand suggests that fewer false positives are found. With respect to regression testing, the findings indicate the potential for better regression testing in case that sessions are properly recorded, but it was also recognized that it is difficult to prioritize and reevaluate the tests.

**Lessons learned:** Even though not representative, the data indicates a gap between industry focus and research focus. Therefore, research should focus on investigating exploratory testing, how it should be applied, and how efficient it is in capturing interoperability issues to support companies in improving their exploratory testing practices.

### 2.6 Conclusion

In this study we explored the testing of heterogeneous systems. In particular, we studied the usage and perceived usefulness of testing techniques for heterogeneous systems. The techniques were identified based on an ongoing systematic literature review. The practitioners surveyed were involved in the development of heterogeneous systems. Two main research questions were answered:

**RQ1:** Which testing techniques are used to assess heterogeneous systems? The most frequently used technique is exploratory manual testing, followed by combinatorial and search-based testing. As discussed earlier, it is encouraging for the field of search-based testing that a high number of practitioners have made experiences with search-based testing. This may provide opportunities to study the technique from the practitioners’ perspective more in the future. Looking at the awareness, the practitioners were well aware of manual exploratory and combinatorial testing, however, a relatively high number was not aware of what search-based testing is.

**RQ2:** How do practitioners perceive the identified techniques with respect to a set of outcome variables? The most positively perceived technique for testing heterogeneous systems was manual exploratory testing, which was the highest rated in five (ease of use, effectiveness in detecting critical defects, effective in detecting various types of defects, time and cost efficiency, helping to improve product quality) out of eight studied variables. While manual exploratory testing was the most used technique in the studied companies, it is the least investigated technique in the literature on testing heterogeneous systems.

In future work, based on the results of the study, several important directions of research were made explicit:
In order to reduce the uncertainty with respect to the performance of the techniques comparative studies are needed. In particular, in the context of heterogeneous systems variables more relevant to that context should be studied (interoperability, large regression test suits). However, in general more comparative studies may be needed, for instance by comparing their performance on heterogeneous open source systems (e.g. Linux).

Given the positive indications of the adoption of search-based in the industry, the focus should also be on understanding how and with what success search-based is used in the industry for heterogeneous and other systems.

Interesting patterns identified and highlighted in the discussion should be investigated in further depth, two examples should be highlighted: First, does (and if so how) heterogeneity affect the performance of exploratory testing in terms of false positives reported? Second, how could it be explained that manual exploratory testing is so positively perceived? Possible propositions are there is a low perceived entry level of using the technique, while it is at the same time very hard to master given its dependence on the testers’ skills. Furthermore, interestingly it was perceived as being time- and cost efficient, which should be understood further. Overall, large and complex systems have many interactions that could require automation to be able to achieve a satisfactory level of coverage.
2.7 References


REFERENCES


REFERENCES


Chapter 3

An Experiment on the Effectiveness and Efficiency of Exploratory Testing

Wasif Afzal, Ahmad Nauman Ghazi, Juha Itkonen, Richard Torkar, Anneliese Andrews and Khurram Bhatti

Abstract: The exploratory testing (ET) approach is commonly applied in industry, but lacks scientific research. The scientific community needs quantitative results on the performance of ET taken from realistic experimental settings. The objective of this paper is to quantify the effectiveness and efficiency of ET vs. testing with documented test cases (test case based testing, TCT). We performed four controlled experiments where a total of 24 practitioners and 46 students performed manual functional testing using ET and TCT. We measured the number of identified defects in the 90-minute testing sessions, the detection difficulty, severity and types of the detected defects, and the number of false defect reports. The results show that ET found a significantly greater number of defects. ET also found significantly more defects of varying levels of difficulty, types and severity levels. However, the two testing approaches did not differ significantly in terms of the number of false defect reports submitted. We conclude that ET was more efficient than TCT in our experiment. ET was also more effective than
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

TCT when detection difficulty, type of defects and severity levels are considered. The two approaches are comparable when it comes to the number of false defect reports submitted.

3.1 Introduction

Software testing is an important activity to improve software quality. Its cost is well known [63, 13]. Thus, there has always been a need to increase the efficiency of testing while, in parallel, making it more effective in terms of finding defects. A number of testing techniques have been developed to enhance the effectiveness and efficiency of software testing. Juristo et al. [38] present a review and classification of different testing techniques. According to SWEBOK [1], the many proposed testing techniques differ essentially in how they select the test set for achieving the test adequacy criterion.

Due to the high cost of testing, a lot of research has focussed on automated software testing. Automated software testing should ideally automate multiple activities in the test process, such as the generation of test requirements, test cases and, test oracles, test case selection or test case prioritization [4]. The main reason for automation is to have improved test efficiency, especially in regression testing where test cases are to be executed iteratively after making changes to the software [22]. But, as Bertolino [12] argues, 100% automatic testing is still a dream for software testing research and practice. The software industry today still relies heavily on manual software testing [11, 5, 28] where the skills of professional testers and application domain experts are used to identify software defects. Our focus in this paper is on manual software testing as opposed to automated software testing.

The traditional and common approach to software testing is to define and plan test cases prior to execution and then compare their outputs to the documented expected results. Such a document-driven, pre-planned approach to testing is called test case based testing (TCT). The test cases are documented with test inputs, expected outputs and the steps to test a function [35, 2, 5]. The major emphasis of TCT remains on detailed documentation of test cases to verify correct implementation of a functional specification [1]. The test adequacy criterion is thus the coverage of requirements. There are undoubtedly certain strengths with the TCT approach. It provides explicit expected outputs for the testers and handles complex relationships in the functionality systematically [33, 3, 32, 51, 27, 62, 54]. The test case documentation can also provide benefits later during regression testing. In this paper we focus on the actual testing activity and defect detection only.

As opposed to TCT, exploratory testing (ET) is an approach to test software without pre-designed test cases. ET is typically defined as simultaneous learning, test design
and test execution [8, 56, 41]. The tests are, thus, dynamically designed, executed and modified [1]. It is believed that ET is largely dependent on the skills, experience and intuition of the tester. Central to the concept of ET is simultaneous/continuous learning where the tester uses the information gained while testing to design new and better tests. ET does not assume any prior application domain knowledge but expects a tester to know testing techniques (e.g., boundary value analysis) and to be able to use the accumulated knowledge about where to look for defects. This is further clarified by Whittaker [59]: “Strategy-based exploratory testing takes all those written techniques (like boundary value analysis or combinatorial testing) and unwritten instinct (like the fact that exception handlers tend to be buggy) and uses this information to guide the hand of the tester. […] The strategies are based on accumulated knowledge about where bugs hide, how to combine inputs and data, and which code paths commonly break.”

In one sense, ET reflects a complete shift in the testing approach, where test execution is based on a tester’s current and improving understanding of the system. This understanding of the system is derived from various sources: observed product behavior during testing, familiarity with the application, the platform, the failure process, the type of possible faults and failures, the risk associated with a particular product, and so on [41]. Although the term exploratory testing was first coined by Kaner and Bach in 1983, Myers acknowledged experience-based approaches to testing in 1979 [47]. However, the actual process to perform ET is not described by Myers. Instead, it was treated as an ad-hoc or error guessing technique.

Over the years, ET has evolved into a thoughtful approach to manual testing. ET is now seen in industry as an approach whereby different testing techniques can be applied. In addition, some approaches, such as session-based test management (SBTM), have been developed to manage the ET process [7]. Finally, ET has also been proposed to provide certain advantages for the industry [56, 48, 7, 36, 41, 46, 55] such as defect detection effectiveness as well as better utilization of testers’ skills, knowledge and creativity. The applicability of the ET approach has not been studied in research literature. The ET approach, despite its claimed benefits, has potential limitations in certain contexts: when precise repeatability for regression testing is required or when experienced or knowledgeable testers are not available.

There have only been a few empirical studies on the performance of ET or similar approaches, see [23, 21, 33, 14]. In these studies, ET has been reported as being more efficient than traditional TCT. However, as the empirical research on ET is still rare, there is a need to do more controlled empirical studies on the effectiveness and efficiency of ET to confirm and extend the existing results. This scarcity of research on

1Obviously it will help a tester if such knowledge exists (to find expected risks).
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

ET is surprising considering the common notion that test execution results depend on the skills of testers [38]. Generally there has been little empirical investigation on test execution practices and manual testing. Little is known regarding what factors affect manual testing efficiency or the practices that are considered useful by industrial testers [38, 33].

Itkonen et al. [33] compared ET and TCT approaches using time-boxed test execution sessions in a controlled student experiment, where the test execution time was equal among the approaches. They reported higher numbers of detected defects and lower total effort for the ET approach, even though there was no statistically significant difference in defect detection effectiveness between the ET and TCT approaches. Further, the detected defects did not differ significantly with respect to their types, detection difficulty or severity. In the experiment of Itkonen et al. the TCT approach also produced more false defect reports than ET [33]. This study extends the experiment of Itkonen et al. by including both student and industry professionals as subjects and setting an equal total time among the approaches.

In order to advance our knowledge regarding ET and to further validate the claims regarding its effectiveness and efficiency, we have conducted an experiment to answer the following main research question (RQ):

**RQ:** Do testers, who are performing functional testing using the ET approach, find more or different defects compared to testers using the TCT approach?

Our main RQ is further divided into three research questions that are given in Section 3.3.2.

In functional testing, functions or components are tested by feeding them input data and examining the output against the specification or design documents. The internal program structure is rarely considered during functional testing.

In this paper, we use the term *defect* to refer to an incorrect behavior of a software system that a tester reports, based on an externally observable *failure* that occurs during the testing. Our experiment only focuses on testing activity and, thus, excludes debugging and identifying the location of actual *faults*. We also need to make a distinction from pure failure counts, because our analysis does not include repeated failures occurring during the same testing session caused by a single fault.

In summary, the results of our study show that ET found a significantly greater number of defects in comparison with TCT. ET also found significantly more defects of varying levels of detection difficulty, types and severity levels. On the other hand, the two testing approaches did not differ significantly in terms of number of false defect reports submitted.

The rest of the paper is structured as follows. Section 3.2 presents the existing research on ET and TCT. Section 3.3 presents the research methodology, the experiment
design, data collection and analysis. The results from the experiment are presented in Section 3.4. Answers to the research questions are discussed in Section 3.5. The threats to validity are covered in Section 3.6. Finally, in Section 3.7, conclusions and future research directions are presented.

3.2 Related work

A review of experiments on testing techniques is given by Juristo et al. [38]\(^2\). This review concludes that there is no single testing technique that can be accepted as a fact as they all are pending some sort of corroboration such as laboratory or field replication or knowledge pending formal analysis. Moreover, for functional and control flow testing technique families a practical recommendation is that more defects are detected by combining individual testers than techniques of the two families. This is important because, in one way, it shows that the results of test execution depend on the tester’s skills and knowledge, even in test case based testing. There is some evidence to support this argument. Kamsties and Lott found that the time taken to find a defect was dependent on the subject [40]. Wood et al. [61] found that combined pairs and triplets of individual testers using the same technique found more defects than individuals. There are many possible reasons for the variation in the results. Individual testers might execute the documented tests differently; the testers’ ability to recognize failures might be different; or individual testers might end up with different tests even though they are using the same test case design technique. The important role of personal experience in software testing has been reported in testing research. Beer and Ramler [10] studied the role of experience in testing by industrial case studies. In addition, Kettunen et al. [42] recognized the importance of testers’ experience, Poon et al. [50] studied the effect of experience on test case design and Galletta et al. [24] report that expertise increases error finding performance.

ET, as described in Section 3.1, is an approach that does not rely on the documentation of test cases prior to test execution. It has been acknowledged in the literature that ET has lacked scientific research [36]. Later there have emerged a few studies. Nascimento et al. [21] conducted an experiment to evaluate the use of model-based and ET approaches in the context of feature testing in mobile phone applications. They found that ET is better than model-based testing for functional testing and produced better results. The effort was clearly smaller when applying ET compared to the model-based approach.

Also in the context of verifying executable specifications Houdek et al. [23] per-

\(^2\)For recent reviews on software testing techniques, see [37, 4, 19, 49, 20].
formed a student experiment comparing reviews, systematic testing techniques and the
ad-hoc testing approach. The results indirectly support hypotheses regarding the effi-
ciency of experience-based approaches showing that the ad-hoc approach required less
effort, and that there was no difference between the techniques with respect to defect
detection effectiveness. None of the studied techniques alone revealed a majority of
the defects and only 44% of the defects were such that the same defect was found by
more than one technique.

Research on the industrial practice of software testing is sparse. Some studies
show that test cases are seldom rigorously used and documented. Instead, practitioners
report that they find test cases difficult to design and, in some cases, even quite useless
[3, 5, 36]. In practice, it seems that test case selection and design is often left to
individual testers and the lack of structured test case design techniques is not found as
a problem [5]. Research on the ET approach in industry includes a case study [36] and
observation studies on testing practices [35] and on the role of knowledge [34], but to
our knowledge the effectiveness and efficiency of ET has not been researched in any
industrial context.

Even though the efficiency and applicability of ET lacks reliable research, there are
anecdotal reports listing many benefits of this type of testing. The claimed benefits, as
summarized in [36], include effectiveness, the ability to utilize tester’s creativity and
non-reliance on documentation [56, 7, 41, 46, 55].

### 3.3 Methodology

This section describes the methodology followed in the study. First, the research goals
along with research questions and hypotheses are described. After that a detailed de-
scription of the experimental design is presented.

#### 3.3.1 Goal definition

The experiment was motivated by a need to further validate the claimed benefits of
using ET. There are studies that report ET as being more efficient and effective in
finding critical defects. As described in the previous section, it has been claimed that
ET takes less effort and utilizes the skill, knowledge and experience of the tester in a
better way. However, more empirical research and reliable results are needed in order
to better understand the potential benefits of the ET approach.

In this experiment we focus on the testing activity and the effects in terms of de-
fect detection effectiveness. The high-level research problem is to investigate if the
traditional testing approach with pre-design and documented test cases is beneficial or
not in terms of defect detection effectiveness. This is an important question, despite of the other potential benefits of test documentation, because the rationale behind the traditional detailed test case documentation is to improve the defect detection capability [25, 47].

According to Wohlin et al. [60], a goal-definition template (identifying the object(s), goal(s), quality focus and the perspective of the study) ensures that important aspects of an experiment are defined before the planning and execution:

- **Objects of study:** The two testing approaches, i.e., ET and TCT.
- **Purpose:** To compare the two testing approaches in fixed length testing sessions in terms of number of found defects, defect types, defect severity levels, defect detection difficulty, and the number of false defect reports.
- **Quality focus:** Defect detection efficiency and the effectiveness of the two testing approaches.
- **Perspective:** The experimental results are interpreted from a tester’s and a researcher’s point of view.
- **Context:** The experiment is run with industry practitioners and students as subjects performing functional testing at system level.

In this context it might be worthwhile to clarify the words effectiveness and efficiency and how these words are used in the context of this experiment. By effectiveness we mean the fault finding performance of a technique, i.e., the number of faults a technique finds. If we also add a measure of effort, i.e., the time it takes to find these faults, then we use the word efficiency.

### 3.3.2 Research questions and hypotheses formulation

Our main RQ was given in Section 3.1. In order to answer our main RQ, a number of sub-RQs are proposed, along with their associated hypotheses:

**RQ 1:** How do the ET and TCT testing approaches compare with respect to the number of defects detected in a given time?

*Null Hypothesis* $H_{0,1}$: There is no difference in the number of detected defects between ET and TCT approaches.

*Alternate Hypothesis* $H_{1,1}$: There is a difference in the number of detected defects between ET and TCT approaches.
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

RQ 2: How do the ET and TCT testing approaches compare with respect to defect detection difficulty, types of identified defects and defect severity levels?

Null Hypothesis $H_{0,2.1}$: There is no difference in the defect detection difficulty when using ET and TCT approaches.

Alternate Hypothesis $H_{1,2.1}$: There is a difference in the defect detection difficulty when using ET and TCT approaches.

Null Hypothesis $H_{0,2.2}$: There is no difference in the technical type of defects detected using ET and TCT approaches.

Alternate Hypothesis $H_{1,2.2}$: There is a difference in the technical type of defects detected using ET and TCT approaches.

Null Hypothesis $H_{0,2.3}$: There is no difference in the severity of defects detected using ET and TCT approaches.

Alternate Hypothesis $H_{1,2.3}$: There is a difference in the severity of defects detected using ET and TCT approaches.

RQ 3: How do the ET and TCT testing approaches compare in terms of number of false defect reports?

Null Hypothesis $H_{0,3}$: There is no difference in the number of false defect reports when using ET and TCT testing approaches.

Alternate Hypothesis $H_{1,3}$: There is a difference in the number of false defect reports when using ET and TCT testing approaches.

To answer the research questions and to test our stated hypotheses, we used a controlled experiment. In the experimental design we followed the recommendations for experimental studies by [60, 39, 43].

3.3.3 Selection of subjects

The subjects in our study were industry practitioners and students. There were three industry partners, two located in Europe and one in Asia. The subjects were selected using a convenience sampling based on accessibility. The subjects from the industry had experience in working with software testing. Still, they were provided with material on the test case design techniques. In academia, the students of an MSc course in software verification and validation took part in the experiment. They learnt different test case design techniques in the course. Moreover, the students were selected based on their performance, i.e., only students performing well in their course assignments.
were selected. The assignments in the course were marked according to a pre-designed template where a student got marks based on a variety of learning criteria. The final marks on an assignment reflected the aggregate of each of the individual criterion. Out of a total of 70 students, 46 were ultimately selected for the experiment, i.e., top-65%. This selection of top-65% of the students was done before the execution of the experiment, i.e., we did not gather any data from the bottom 35% of the students as they were excluded from the very start.

The total number of subjects who participated in this experiment was 70. Among them there were a total of 24 participants from industry and 46 from academia. The subjects were divided into two groups. The groups are referred to as the ET group and the TCT group, based on the approach they used to test the feature set (experimental object). The approach to be used by each of the two groups (either ET or TCT) was only disclosed to them once they had started their sessions. There were a total of 35 participants in each of the two groups for the four experimental iterations. (The division of subjects in experimental iterations and groups is shown in Table 3.1.)

Further, the following aspects were considered for people participating as subjects.

- **Obtain consent**: To reduce the risk of invalid data and to enable the subjects to perform the experiment according to the objectives, the intention of the work and the research objectives were explained to all subjects (through a meeting in the industry and a presentation to students). It was made clear how the results would be used and published.

- **Sensitive results**: The subjects were assured that their performance in the experiment would be kept confidential.

- **Inducements**: To increase motivation, extra course points were awarded to the students participating in the experiment, but participation was not made compulsory. The industry practitioners were motivated by the prospects of getting important feedback on the performance of the two testing approaches.

### Table 3.1: The division of subjects in experimental iterations and groups.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Type</th>
<th>Total subjects</th>
<th>ET</th>
<th>TCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Academia</td>
<td>46</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>Industrial</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Industrial</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Industrial</td>
<td>13</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>
Table 3.2: Average experience of subjects in software development and software testing in number of years.

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Experience (years)</th>
<th>$\bar{x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>Software development</td>
<td>0.583</td>
</tr>
<tr>
<td></td>
<td>Software testing</td>
<td>0.291</td>
</tr>
<tr>
<td>Industrial practitioners</td>
<td>Software development</td>
<td>2.954</td>
</tr>
<tr>
<td></td>
<td>Software testing</td>
<td>4.045</td>
</tr>
</tbody>
</table>

To characterize the subjects, demographic data was collected in terms of experience in software development and software testing. The demographic data of the subjects is given in Table 3.2. On average, the industrial practitioners were more experienced in software development and software testing than the students, which was expected. However, the students were, on the other hand, knowledgeable in the use of various testing techniques that were taught during the course *software verification and validation*.

### 3.3.4 Experiment design

The experimental design of this study is based on one factor with two treatments. The factor in our experiment is the testing approach while the treatments are ET and TCT. There are two response variables of interest: defect detection efficiency and defect detection effectiveness.

The experiment was comprised of two separate sessions, one each for the ET and TCT group. In the testing session phase, the TCT group designed and executed the test cases for the feature set. The subjects did not design any test cases before the testing session. The rationale was to measure the efficiency in terms of time to complete all required activities. At the start of the session, the TCT group was provided with a template, both for designing the test cases and for reporting the defects. The ET group was instructed to log their session activity as per their own understanding but in a readable format. Both groups were given the same materials and information regarding the tested application and its features. Also both TCT and ET groups were provided with the same jEdit user’s guide for finding the expected outputs. The subjects in TCT group designed their test cases themselves, no existing test cases were provided for them.

All subjects were instructed to apply the same detailed test design techniques: equivalence partitioning, boundary value analysis and combination testing techniques. The same techniques were applied for test case designing in TCT as well as for testing in ET. Same techniques were applied both with the industry and student subjects. This
information was communicated to them prior to the experiment.

Each session started with a 15-minute ‘session startup’ phase where the subjects were introduced with the objective of the experiment and were given the guidelines on how to conduct the experiment. The actual testing was done in a 90 min time-boxed session\(^3\). The defect reports and the test logs were then handed over for evaluation. The following artifacts were provided in the testing session:

- Session instructions.
- A copy of the relevant chapters of the jEdit user’s guide.
- Defect reporting document (TCT only).
- Test case design document (TCT only).
- A test charter and logging document for ET.
- Test data files that are available in the test sessions:
  - A small text file.
  - GNU general public license text file.
  - jEdit user’s guide as a text file.
  - Ant build.xml file.
  - Java source code files from jEdit.
  - C++ source code files from WinMerge\(^4\).

The following artifacts were required to be submitted by the subjects:

- Defect reports in a text document.
- The test cases and the test log (TCT only).
- The filled ET logging document (ET only).
- Test case design document (TCT only).

---

\(^3\) The 90 minutes session length was decided as suggested by Bach [7] but is not a strict requirement (we were constrained by the limited time available for the experiments from our industrial and academic subjects.)

\(^4\) The C++ source code files were given to the subjects as an example to see code formatting and indentation. The purpose was to guide the subjects in detecting formatting and indentation defects.
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

The concept of tracking the test activity in sessions is taken from Bach’s approach of session-based test management (SBTM) [7]. SBTM was introduced to better organize ET by generating orderly reports and keeping track of tester’s progress supported by a tool. Testing is done in time-limited sessions with each session having a mission or charter. The sessions are debriefed with the test lead accepting a session report and providing feedback. The session report is stored in a repository whereby a tool scans it for getting basic metrics, like time spent on various test activities and testing progress over time in terms of completed sessions.

3.3.5 Instrumentation

The experimental object we used in this study was the same as used by Itkonen et al. [33]. It is an open source text editor\(^5\). Artificial faults were seeded in the application at the source code level to make two variants and then recompiled. The variant that we used is referred to as Feature Set-A in the experiment by Itkonen et al. [33]. This variant contained a total of 25 seeded faults. The actual number of faults exceeds the number of seeded faults. The choice to use a text editor was made because editors are familiar to the subjects without requiring any training [33], and it represents a realistic application. In addition, being open source it was possible to seed faults in the application code. The experimental object was only available to the subjects once the functional testing phase was started.

In addition to the test object feature set, we used the following instrumentation, with required modifications: user guide and instructions; test case design template (Appendix A); defect report template (Appendix B); exploratory charter \(^6\) (Appendix C); and feature set defects details.

The Feature Set-A was composed of first and second priority functions:

- First priority functions
  - Working with files (User’s guide chapter 4)
    * Creating new files.
    * Opening files (excluding CZipped files).
    * Saving files.
    * Closing files and exiting jEdit.
  - Editing text (User’s guide chapter 5)
    * Moving the caret.

\(^5\)jEdit version 4.2

\(^6\)The exploratory charter provided the subjects with high-level test guidelines.
3.3.6 Operation

The user guide and the instructions for testing the application were provided to the subjects one day before the experiment execution. The task of the subjects was to cover all functionality documented in the user’s guide concerning Feature Set-A. One subject participated only in one allocated session, i.e., either ET or TCT.

At the start of the testing session, subjects were provided with instructions. The instructions contained details on session arrangement and the focus of the testing session. The TCT group received the template for test case design and reporting defects. The ET group got a vague charter listing the functionality to be tested and an emphasis on testing from user’s viewpoint. Both ET and TCT groups performed the test execution manually.

We executed a total of four experiment iterations, i.e., four instances of the experiment conducted with different subjects under similar experimental setting. Three of these iterations were done in industry (two in Europe and one in Asia) while one of the iterations were done in academia. For each experiment iteration, the ET and TCT groups performed the sessions at the same time (they were located in identical locations but in different rooms).

To provide an identical experimental environment, i.e., testing tools and operating system (OS), each subject connected to a remote Windows XP image. The OS image was preloaded with the experimental object in complete isolation from the Internet. To
Table 3.3: Defect count data summary.

<table>
<thead>
<tr>
<th>Testing approach</th>
<th>defects found (Mean (\bar{x}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>ET</td>
<td>8.342</td>
</tr>
<tr>
<td>TCT</td>
<td>1.828</td>
</tr>
</tbody>
</table>

collect data from this experiment, the logs and defect report forms were filled out by the subjects during the testing session. After the data was collected, it was checked for correctness and the subjects were consulted when necessary.

The experimental design of this study was similar to the earlier experiment by Itkonen et al. [33] and used the same software under test, including the same seeded and actual faults. There are, however, three important differences in the experimental design between the two experiments. First, this study employed only one test session per subject with the purpose of reducing the learning effect of the subjects. We tried to avoid the learning effect because we believed that we would measure the true effect of a particular treatment more accurately. Each subject carried out the experiment one time only using their assigned testing approach. Second, in this experiment the total time provided to both approaches was the same, whereas in Itkonen et al.’s earlier experiment the test case design effort was not part of the time-boxed testing sessions. Both approaches were allocated 90 minutes to carry out all activities involved in their approach. This way we were, in addition, able to measure the efficiency in terms of number of defects found in a given time. Third, the experimental settings were, of course, different. This experiment was executed both in industry and academia, whereas Itkonen et al.’s study [33] used student subjects only.

3.4 Results and analysis

This section presents the experimental results based on the statistical analysis of the data.

3.4.1 Defect count

The defect count included all those reported defects that the researchers were able to interpret, understand and reproduce (i.e., true defects). A false defect (duplicate, non-reproducible, non-understandable) was not included in the defect count. The details of false defects are described in Section 3.4.3. The defect counts are summarized in Table 3.3. The table lists the defect counts separately for both testing approaches.
The mean defect counts for the ET and TCT approaches are 8.342 and 1.828 respectively; ET detected on average 6.514 more defects than TCT. The actual number of defects found by the two approaches was 292 (ET) vs. 64 (TCT). The number of defects detected by both groups were from a normal distribution (confirmed by using the Shapiro-Wilks test for normality). Thus, the number of defects detected were compared using the t-test. Using the two-tailed t-test, we obtained \( p = 1.159 \times 10^{-10} \), hence, the defects found using the two approaches were statistically different at \( \alpha = 0.05 \). The effect size calculated using Cohen’s \( d \) statistic also suggested practical significance, i.e., \( d = 2.065^7 \).

For the number of defects detected in the given time, students found 172 true defects when using ET with a median of 6\(^8\). Practitioners found 120 true defects when using ET with a median of 9. This shows that for the number of students and practitioners applying ET, the practitioners found on average more defects than students. However, the difference is not statistically significant (\( p = 0.07 \)) when applying the Mann-Whitney U test at \( \alpha = 0.05 \) (the data had a non-normal distribution). (We also used the non-parametric Vargha and Delaney’s \( \hat{A}_{12} \) statistic to assess effect size. The statistic \( \hat{A}_{12} \) turned out to be 0.31 which is a small effect size according to the guidelines of Vargha and Delaney [57]).

Students, when applying TCT, found a total of 33 true defects with a median of 1. Practitioners, on the other hand, found a total of 31 defects while applying TCT with a median of 2.5. This shows that practitioners found, on average, more true defects than students when using TCT. However, the difference is not statistically significant (\( p = 0.15, \hat{A}_{12} = 0.35 \)) when applying the Mann-Whitney U test at \( \alpha = 0.05 \) (the data had a non-normal distribution).

### 3.4.2 Detection difficulty, types and severity

The defect reports were classified into three dimensions [33]:

1. Detection difficulty.
2. Technical type.

---

7Cohen’s \( d \) shows the mean difference between the two groups in standard deviation units. The values for \( d \) are interpreted differently for different research questions. However, we have followed a standard interpretation offered by Cohen [18], where 0.8, 0.5 and 0.2 show large, moderate and small practical significances, respectively.

8Median is a more close indication of true average than mean due to the presence of extreme values.
We used the same measure for defect detection difficulty as Itkonen et al. used in their earlier experiment [33]. The detection difficulty of a defect is defined by using the failure-triggering fault interaction\(^9\) (FTFI) number. This number refers to the number of conditions required to trigger a failure [44]. The FTFI number is determined by observing the failure occurrence and analyzing how many different inputs or actions are required in order to make the failure occur. For example, if triggering a failure requires the tester to set one input in the system to a specific value and executing a specific command, the FTFI number would be 2 (i.e., mode 2 defect). The detection difficulty of a defect in this study is characterized into four levels of increasing difficulty:

- **Mode 0**: A defect is immediately visible to the tester.
- **Mode 1**: A defect requires a single input to cause a failure (single-mode defect).
- **Mode 2**: A defect requires a combination of two inputs to cause a failure.
- **Mode 3**: A defect requires a combination of three or more inputs to cause a failure.

To make testing more effective it is important to know which types of defects could be found in the software under test, and the relative frequency with which these defects have occurred in the past [1]. IEEE standard 1044-2009 [31] exists on classifying software defects but the standard only prescribes example defect types such as interface, logic and syntax while recommending organizations to define their own classifications; “the point is to establish a defect taxonomy that is meaningful to the organization and the software engineers” [1]. For the purpose of this study, we have classified defects based on the externally visible symptoms, instead of the (source code level) technical fault type as is common in existing classifications, see, e.g., ODC [17]. We believe that for comparing approaches for manual testing the defect symptoms is an important factor affecting defect detection. The defects were classified into following types based on the symptoms: performance, documentation, GUI, inconsistency, missing function, technical defect, usability and wrong function. The definition of each type of defect with examples appears in Table 3.4. The defect severity indicates the defect’s estimated impact on the end user, i.e., negligible, minor, normal, critical or severe.

In all four modes of detection difficulty, ET found clearly more defects. The difference between the number of defects found in each difficulty level is, consequently, also statistically significant at \(\alpha = 0.05\) using the \(t\)-test \((p = 0.021, d = 2.91)\).
Table 3.4: Description of the types of defects with examples.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Defects in user manual</td>
<td>Manual has wrong keyboard shortcut for inverting the selection in the selecting text chapter</td>
</tr>
<tr>
<td>GUI</td>
<td>Defects in user interface, such as undesirable behavior in text and file selection, inappropriate error messages and missing menus</td>
<td>Uninformative error message when trying to save in an access restricted folder</td>
</tr>
<tr>
<td>Inconsistency</td>
<td>Functions exhibiting inconsistent behavior</td>
<td>Opening a new empty buffer is not possible when only one unmodified empty buffer exists</td>
</tr>
<tr>
<td>Missing function</td>
<td>Defects due to missing functionality and incompatibility issues</td>
<td>Shortcut problems with Finnish keyboard and Autosave does not automatically find the autosave file; prompting for recovery when jEdit is launched after crash</td>
</tr>
<tr>
<td>Performance</td>
<td>Defects resulting in reduced performance of the system</td>
<td>Character input stops after writing few characters fast</td>
</tr>
<tr>
<td>Technical defect</td>
<td>Defects attributed to application crash, technical error message or runtime exception</td>
<td>While holding right arrow-key down an exception is thrown; Goto line crashes if large line number is provided</td>
</tr>
<tr>
<td>Usability</td>
<td>Defects resulting in undesirable usability issues</td>
<td>Open dialog always opens to C: directory; Select lines accept invalid input without a warning message</td>
</tr>
<tr>
<td>Wrong function</td>
<td>Defects resulting in incorrect functionality</td>
<td>An extra newline character is added at the end of the file while saving; if a file created in another editor is opened the last character is missing</td>
</tr>
</tbody>
</table>
Table 3.5: Distribution of defects concerning detection difficulty.

<table>
<thead>
<tr>
<th>Mode</th>
<th>ET</th>
<th>TCT</th>
<th>ET %</th>
<th>TCT %</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0</td>
<td>73</td>
<td>22</td>
<td>25%</td>
<td>35%</td>
<td>95</td>
</tr>
<tr>
<td>Mode 1</td>
<td>117</td>
<td>27</td>
<td>40%</td>
<td>44%</td>
<td>144</td>
</tr>
<tr>
<td>Mode 2</td>
<td>72</td>
<td>11</td>
<td>25%</td>
<td>18%</td>
<td>83</td>
</tr>
<tr>
<td>Mode 3</td>
<td>30</td>
<td>2</td>
<td>10%</td>
<td>3%</td>
<td>32</td>
</tr>
<tr>
<td>Total</td>
<td>292</td>
<td>62</td>
<td>100%</td>
<td>100%</td>
<td>354</td>
</tr>
</tbody>
</table>

In the percentage distribution presented in Table 3.5 we can see the differences between ET and TCT in terms of the detection difficulty. The data shows that for the defects that ET revealed, the proportion on defects that were difficult to detect was higher. In the defects revealed by TCT the proportion of the obvious and straightforward defects was higher.

For students applying ET, the four modes of defect detection difficulty were significantly different using one-way analysis of variance \((p = 4.5e-7, \alpha = 0.05)\). The effect size calculated using eta-squared \((\eta^2)\) also suggested practical significance, i.e., \(\eta^2=0.31^{10}\). We performed a multiple-comparisons test (Tuckey-Kramer, \(\alpha = 0.05)\)\(^{11}\) to find out which pairs of modes are significantly different. The results showed that mode 0 and 1 defects were significantly different from mode 3 defects (Fig. 3.1(a)). In the percentage distribution presented in Table 3.6 we can see the differences in the modes of defects detected by students using ET. The data indicates that students detected a greater percentage of easier defects (mode 0 and 1) as compared to difficult defects (mode 2 and 3).

For practitioners applying ET, the four modes of defect detection difficulty were also found to be significantly different using one-way analysis of variance \((p = 1.8e-4, \eta^2=0.36)\). The results of a multiple comparisons test (Tuckey-Kramer, \(\alpha = 0.05)\) showed that mode 0 defects were not significantly different from mode 3 defects while mode 1 defects were significantly different from mode 3 (Fig. 3.1(b)). In the percentage distribution presented in Table 3.6 we can see a trend similar to when students applied ET, i.e., practitioners detected a greater percentage of easier defects (mode 0 and 1) as compared to difficult defects (mode 2 and 3).

We further applied the multivariate analysis of variance test for identifying any

---

\(^{10}\)\(\eta^2\) is a commonly used effect size measure in analysis of variance and represents an estimate of the degree of the association for the sample. We have followed the interpretation of Cohen [18] for the significance of \(\eta^2\) where 0.0099 constitutes a small effect, 0.0588 a medium effect and 0.1379 a large effect.

\(^{11}\)The term mean rank is used in Tuckey-Kramer test for multiple comparisons. This test ranks the set of means in ascending order to reduce possible comparisons to be tested, e.g., in the ranking of the means \(W > X > Y > Z\), if there is no difference between the two means that have the largest difference \((W & Z)\), comparing other means having smaller difference will be of no use as we will get the same conclusion.

---

66
(a) Multiple comparisons test for the different modes of defects detected by students applying ET.

(b) Multiple comparisons test for the different modes of defects detected by practitioners applying ET.

Figure 3.1: Results of the multiple comparisons test for modes of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different modes of defects, i.e., in Fig 3.1(a) above, the vertical dotted lines indicate Mode 1 and 3 have mean ranks significantly different from Mode 0.)
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

Table 3.6: Percentages of the modes of defects detected by students and practitioners applying ET.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0</td>
<td>46</td>
<td>27</td>
<td>26.74%</td>
<td>22.69%</td>
</tr>
<tr>
<td>Mode 1</td>
<td>68</td>
<td>48</td>
<td>39.53%</td>
<td>40.34%</td>
</tr>
<tr>
<td>Mode 2</td>
<td>39</td>
<td>31</td>
<td>22.67%</td>
<td>26.05%</td>
</tr>
<tr>
<td>Mode 3</td>
<td>19</td>
<td>13</td>
<td>11.04%</td>
<td>10.92%</td>
</tr>
</tbody>
</table>

Table 3.7: Percentages of the different modes of defects detected by students and practitioners applying TCT.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0</td>
<td>7</td>
<td>11</td>
<td>25.92%</td>
<td>45.83%</td>
</tr>
<tr>
<td>Mode 1</td>
<td>15</td>
<td>7</td>
<td>55.55%</td>
<td>29.17%</td>
</tr>
<tr>
<td>Mode 2</td>
<td>5</td>
<td>6</td>
<td>18.52%</td>
<td>25%</td>
</tr>
<tr>
<td>Mode 3</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

significant differences between students and practitioners for defect detection difficulty modes when using ET. The results given by four different multivariate tests indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different modes of defects identified in total ($p$-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root = 0.31, $\eta^2 = 0.14$, $\alpha = 0.05$).

For students applying TCT, the four modes of defect detection difficulty were significantly different using one-way analysis of variance ($p = 0.01$, $\eta^2 = 0.12$, $\alpha = 0.05$). The results of performing a multiple comparisons test (Tuckey-Kramer, $\alpha = 0.05$) showed that mode 0 and 2 defects were not significantly different from any other mode while mode 1 and 3 were found to be significantly different (Fig 3.2(a)). The percentage distribution of different modes of defects is presented in Table 3.7. It shows that no defect in mode 3 was detected by students applying TCT while majority of the defects found were comparatively easy to find (mode 0 and 1).

For practitioners applying TCT, there were no significant differences found between the different modes of defects detected using one-way analysis of variance ($p = 0.15$, $\eta^2 = 0.11$, $\alpha = 0.05$). The percentage distribution of different modes of defects detected by practitioners using TCT is given in Table 3.7. As was the case with students, practitioners also did not find any defects in mode 3 while majority of the defects found were easy (mode 0 and 1).

We further applied the multivariate analysis of variance test for identifying any significant differences between students and practitioners for defect detection difficulty modes of defects when using TCT. The results given by four different multivariate tests
indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different modes of defects identified in total ($p$-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root = 0.27, $\eta^2 = 0.12$, $\alpha = 0.05$).

Table 3.8 shows the categorization of the defects based on their technical type. The ET approach revealed more defects in each defect type category in comparison to TCT (the exception being the ‘documentation’ type where both approaches found equal number of defects). Nevertheless, the differences are very high for the following types: missing function, performance, technical defect, and wrong function. Using a $t$-test, the difference between the number of defects found per technical type for the two approaches is statistically significant at $\alpha = 0.05$ ($p = 0.012$, $d = 2.26$).

The percentage distribution presented in Table 3.8 indicates quite strongly that concerning defects that ET found, the proportion of missing function, performance, and technical defects were clearly higher. On the other hand, the proportions of GUI and usability defects as well as wrong function types defects were higher in defects revealed by TCT.

The results of one-way analysis of variance ($p = 7.8e-16$, $\eta^2 = 0.38$) also showed that students, when using ET, found significantly different technical types of defects. A multiple comparisons test (Tuckey-Kramer, $\alpha = 0.05$) (Fig. 3.3(a)) showed that the
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

<table>
<thead>
<tr>
<th>Type</th>
<th>ET</th>
<th>TCT</th>
<th>ET %</th>
<th>TCT %</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>5</td>
<td>5</td>
<td>1.71%</td>
<td>8.06%</td>
<td>10</td>
</tr>
<tr>
<td>GUI</td>
<td>19</td>
<td>8</td>
<td>6.51%</td>
<td>12.90%</td>
<td>27</td>
</tr>
<tr>
<td>Inconsistency</td>
<td>8</td>
<td>4</td>
<td>2.74%</td>
<td>6.45%</td>
<td>12</td>
</tr>
<tr>
<td>Missing function</td>
<td>65</td>
<td>5</td>
<td>22.26%</td>
<td>8.06%</td>
<td>70</td>
</tr>
<tr>
<td>Performance</td>
<td>62</td>
<td>5</td>
<td>21.23%</td>
<td>8.06%</td>
<td>67</td>
</tr>
<tr>
<td>Technical defect</td>
<td>44</td>
<td>2</td>
<td>15.07%</td>
<td>3.22%</td>
<td>46</td>
</tr>
<tr>
<td>Usability</td>
<td>17</td>
<td>11</td>
<td>5.82%</td>
<td>17.74%</td>
<td>28</td>
</tr>
<tr>
<td>Wrong function</td>
<td>72</td>
<td>22</td>
<td>24.66%</td>
<td>35.48%</td>
<td>94</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>292</td>
<td>62</td>
<td>100%</td>
<td>100%</td>
<td>354</td>
</tr>
</tbody>
</table>

Table 3.9: Percentages of the type of defects detected by students and practitioners applying ET.

<table>
<thead>
<tr>
<th>Type</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>2</td>
<td>3</td>
<td>1.15%</td>
<td>2.54%</td>
</tr>
<tr>
<td>GUI</td>
<td>12</td>
<td>7</td>
<td>6.90%</td>
<td>5.93%</td>
</tr>
<tr>
<td>Inconsistency</td>
<td>4</td>
<td>4</td>
<td>2.30%</td>
<td>3.39%</td>
</tr>
<tr>
<td>Missing function</td>
<td>39</td>
<td>26</td>
<td>22.41%</td>
<td>22.03%</td>
</tr>
<tr>
<td>Performance</td>
<td>38</td>
<td>24</td>
<td>21.84%</td>
<td>20.34%</td>
</tr>
<tr>
<td>Technical defect</td>
<td>26</td>
<td>18</td>
<td>14.94%</td>
<td>15.25%</td>
</tr>
<tr>
<td>Usability</td>
<td>10</td>
<td>7</td>
<td>5.75%</td>
<td>5.93%</td>
</tr>
<tr>
<td>Wrong function</td>
<td>43</td>
<td>29</td>
<td>24.71%</td>
<td>24.58%</td>
</tr>
</tbody>
</table>

defects of the type: documentation, GUI, inconsistency and usability were significantly different from the defects of the type: missing function, performance and wrong function. The percentage distribution presented in Table 3.9 show clearly that students applying ET found greater proportions of missing function, performance and wrong function defects as compared to remaining types of defects.

The practitioners also found significantly different type of defects when using ET as shown by the results of one-way analysis of variance \( p = 4.1e - 10, \eta^2=0.47 \). A multiple comparisons test (Tuckey-Kramer, \( \alpha = 0.05 \)) (Fig. 3.3(b)) showed similar results to students using ET, i.e., the defects of the type: documentation, GUI, inconsistency and usability were significantly different from the defects of the type: missing function, performance and wrong function. The percentage distribution of type of defects found by practitioners using ET (Table 3.9) show a similar pattern to when students applied ET, i.e., practitioners found greater proportions of missing function, performance and wrong function defects as compared to remaining type of defects.

We further applied the multivariate analysis of variance test for identifying any significant differences between students and practitioners for type of defects when using
Figure 3.3: Results of the multiple comparisons test for types of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.3(a) above, the vertical dotted lines indicate that documentation has mean rank significantly different from missing function, performance, technical defect and wrong function.)
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

Figure 3.4: Results of the multiple comparisons test for different types of defects detected by students using TCT. (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.4(a) above, the vertical dotted lines indicate that defects of type documentation have mean rank significantly different from defects of type wrong function.)

ET. The results given by four different multivariate tests indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different type of defects identified in total \( (p\text{-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root = 0.58, } \eta^2 = 0.20, \alpha = 0.05) \).

The results of one-way analysis of variance \( (p = 2.1e-4, \eta^2 = 0.14, \alpha = 0.05) \) also showed that students, when using TCT, found significantly different types of defects. A multiple comparisons test (Tuckey-Kramer, \( \alpha = 0.05 \)) (Fig 3.4(a)) showed that the defects of type wrong function were significantly different than all other types of defects (which did not differ significantly among each other). The percentage distribution shown in Table 3.10 shows that the defects of the type wrong function were detected more than any types of defects.

The practitioners applying TCT, on the other hand, did not find significantly different types of defects as given by the results of one-way analysis of variance \( (p = 0.05, \eta^2 = 0.14, \alpha = 0.05) \). The percentage distribution of types of defects are shown in Table 3.10. As with students using TCT, practitioners also found more wrong function type defects than other types.

We further applied the multivariate analysis of variance test for identifying any
significant differences between students and practitioners for different types of defects when using TCT. The results given by four different multivariate tests indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different types of defects identified in total ($p$-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root = 0.08, $\eta^2 = 0.35$, $\alpha = 0.05$).

Table 3.11 shows the categorization of the defects based on their severities. We can see that ET found more defects in all severity classes. The difference is also statistically significant using a $t$-test at $\alpha = 0.05$ ($p = 0.048$, $d = 1.84$).

The percentage proportions in Table 3.11 show that the proportion of severe and critical defects is higher when ET was employed and the proportion of ‘negligible’ defects was greater with TCT.

Comparing the severity levels of defects found by students using ET show that they found significantly different severity levels of defects (one-way analysis of variance, $p = 3.2e - 14$, $\eta^2 = 0.46$). The results of a multiple comparisons test (Tuckey-Kramer, $\alpha=0.05$) showed that severe and normal defects were significantly different from negligible, minor and critical defects (Fig. 3.5(a)). This is also evident from the percentage distribution of severity levels of defects found by students using ET (Table 3.12). The students clearly found greater proportions of normal and severe defects in comparison to remaining severity levels of defects.

Table 3.10: Percentages of the different types of defects detected by students and practitioners applying TCT.

<table>
<thead>
<tr>
<th>Type</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>GUI</td>
<td>1</td>
<td>1</td>
<td>3.70%</td>
<td>4.17%</td>
</tr>
<tr>
<td>Inconsistency</td>
<td>4</td>
<td>0</td>
<td>14.81%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Missing function</td>
<td>3</td>
<td>2</td>
<td>11.11%</td>
<td>8.33%</td>
</tr>
<tr>
<td>Performance</td>
<td>4</td>
<td>1</td>
<td>14.81%</td>
<td>4.17%</td>
</tr>
<tr>
<td>Technical defect</td>
<td>0</td>
<td>2</td>
<td>0.00%</td>
<td>8.33%</td>
</tr>
<tr>
<td>Usability</td>
<td>3</td>
<td>8</td>
<td>11.11%</td>
<td>33.33%</td>
</tr>
<tr>
<td>Wrong function</td>
<td>12</td>
<td>10</td>
<td>44.44%</td>
<td>41.67%</td>
</tr>
</tbody>
</table>

Table 3.11: Severity distribution of defects.

<table>
<thead>
<tr>
<th>Severity</th>
<th>ET</th>
<th>TCT</th>
<th>ET %</th>
<th>TCT %</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negligible</td>
<td>13</td>
<td>9</td>
<td>4.45%</td>
<td>14.32%</td>
<td>22</td>
</tr>
<tr>
<td>Minor</td>
<td>49</td>
<td>12</td>
<td>16.78%</td>
<td>19.35%</td>
<td>61</td>
</tr>
<tr>
<td>Normal</td>
<td>99</td>
<td>25</td>
<td>33.90%</td>
<td>40.32%</td>
<td>124</td>
</tr>
<tr>
<td>Severe</td>
<td>108</td>
<td>25</td>
<td>36.99%</td>
<td>22.58%</td>
<td>122</td>
</tr>
<tr>
<td>Critical</td>
<td>23</td>
<td>2</td>
<td>7.88%</td>
<td>3.22%</td>
<td>25</td>
</tr>
<tr>
<td>Total</td>
<td>292</td>
<td>62</td>
<td>100%</td>
<td>100%</td>
<td>354</td>
</tr>
</tbody>
</table>
Figure 3.5: Results of the multiple comparisons test for different severity levels of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different severity levels of defects, i.e., in Fig 3.5(a) above, the vertical dotted lines indicate that negligible defects have mean rank significantly different from normal and severe defects.)
Table 3.12: Percentages of the severity level of defects detected by students and practitioners applying ET.

<table>
<thead>
<tr>
<th>Type</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negligible</td>
<td>7</td>
<td>6</td>
<td>4.07%</td>
<td>5.00%</td>
</tr>
<tr>
<td>Minor</td>
<td>28</td>
<td>21</td>
<td>16.28%</td>
<td>17.50%</td>
</tr>
<tr>
<td>Normal</td>
<td>59</td>
<td>40</td>
<td>34.30%</td>
<td>33.33%</td>
</tr>
<tr>
<td>Critical</td>
<td>8</td>
<td>15</td>
<td>4.65%</td>
<td>12.50%</td>
</tr>
<tr>
<td>Severe</td>
<td>70</td>
<td>38</td>
<td>40.70%</td>
<td>31.67%</td>
</tr>
</tbody>
</table>

Table 3.13: Percentages of the severity levels of defects detected by students and practitioners applying TCT.

<table>
<thead>
<tr>
<th>Severity level</th>
<th>Students</th>
<th>Practitioners</th>
<th>Students %</th>
<th>Practitioners %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negligible</td>
<td>8</td>
<td>1</td>
<td>29.63%</td>
<td>4.17%</td>
</tr>
<tr>
<td>Minor</td>
<td>3</td>
<td>4</td>
<td>11.11%</td>
<td>16.67%</td>
</tr>
<tr>
<td>Normal</td>
<td>11</td>
<td>14</td>
<td>40.74%</td>
<td>58.33%</td>
</tr>
<tr>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Severe</td>
<td>5</td>
<td>5</td>
<td>18.52%</td>
<td>20.83%</td>
</tr>
</tbody>
</table>

The practitioners also found defects of significantly different severity levels using ET (one-way analysis of variance, $p = 7.5e-6$, $\eta^2 = 0.40$). A multiple comparisons test (Tuckey-Kramer, $\alpha = 0.05$) (Fig. 3.5(b)) showed results similar to when students applied ET, i.e., severe and normal defects were significantly different from negligible and critical defects. The percentage distribution of severity levels of defects (Table 3.12) also show that practitioners found more normal and severe defects in comparison to remaining severity levels of defects.

We further applied the multivariate analysis of variance test for identifying any significant differences between students and practitioners for severity levels of defects when using ET. The results given by four different multivariate tests indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different severity levels of defects identified in total ($p$-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root = 0.14, $\eta^2 = 0.24$, $\alpha = 0.05$).

The students when using TCT also found significantly different severity levels of defects as indicated by one-way analysis of variance ($p = 0.007$, $\eta^2 = 0.12$, $\alpha = 0.05$). The multiple comparisons test (Tuckey-Kramer, $\alpha=0.05$) (Fig. 3.6(a)) showed that negligible, minor and severe severity levels of defects were different than none other severity level of defects while normal and critical were significantly different. Looking at the percentage distribution of the severity levels of defects found by students using TCT (Table 3.13) show that most of the defects found were of normal severity level while no defect of severity level critical was found.
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

(a) Multiple comparisons test for the severity levels of defects detected by students applying TCT.

(b) Multiple comparisons test for the severity levels of defects detected by practitioners applying TCT.

Figure 3.6: Results of the multiple comparisons test for severity levels of defects detected by students and practitioners using TCT. (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.6(a) above, the vertical dotted lines indicate that defects of negligible severity level have mean rank significantly different from no other severity level.)
The practitioners when using TCT, also found significantly different severity levels of defects, as given by one-way analysis of variance \((p = 0.01, \eta^2 = 0.21, \alpha = 0.05)\). The results of a multiple comparisons test (Tuckey-Kramer, \(\alpha = 0.05\)) (Fig. 3.6(b)) indicate that normal defects were significantly different from negligible and critical defects. Minor and severe defects did not differ significantly with other severity level of defects. The results are somewhat similar to when students applied TCT. The percentage distribution of severity levels of defects found by practitioners is given in Table 3.13. Similar to when students performed TCT, no critical defects are found by practitioners while normal defects were found more than any other severity level of defects.

We further applied the multivariate analysis of variance test for identifying any significant differences between students and practitioners for different severity levels of defects when using TCT. The results given by four different multivariate tests indicate that there is no significant effect of the type of subjects (either students or practitioners) on the different severity levels of defects identified in total \((p\text{-value for Pillai’s trace, Wilks’ lambda, Hotelling’s trace, Roy’s largest root } = 0.14, \eta^2 = 0.20, \alpha = 0.05)\).

### 3.4.3 False defect reports

We consider a reported defect as false if it is either: a duplicate, non-existing, or the report cannot be understood. A defect report was judged as false by the researchers if it clearly reported the same defect that had been already reported by the same subject in the same test session; it was not an existing defect in the tested software (could not be reproduced); or it was impossible for the researchers to understand the defect report. The actual false defect counts for ET and TCT were 27 and 44, respectively. The averages were \(\bar{x}_{ET} = 0.771\) and \(\bar{x}_{TCT} = 1.257\).

It can be seen that on average TCT produced 0.486 more false defect reports than ET. However, the difference is not statistically significant \((p = 0.522)\) when applying the Mann-Whitney U test at \(\alpha = 0.05\) (the data had a non-normal distribution). We also used the non-parametric Vargha and Delaney’s \(\hat{A}_{12}\) statistic to assess effect size. The statistic \(\hat{A}_{12}\) turned out to be 0.463 which is a small effect size according to the guidelines of Vargha and Delaney\(^\text{12}\).

Students applying ET reported 35 false defects with the median number of false defects being 0. On the other hand, the practitioners applying ET reported 9 false defect reports with the median number of false defects also being 0. The statistics indicate that on average both students and practitioners found similar number of false defects. This

\(^{12}\text{Vargha and Delaney suggest that the } \hat{A}_{12}\text{ statistic of 0.56, 0.64 and 0.71 represent small, medium and large effect sizes respectively [57].}\)
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

is also confirmed by Mann-Whitney U test (the data had a non-normal distribution) at $\alpha = 0.05$ ($p = 0.98, \hat{A}_{12} = 0.50$) which indicates non-significant differences in the median number of false defect reports submitted by students and practitioners when applying ET.

Students applying TCT reported 37 false defects with the median number of false defects being 0. On the other hand, the practitioners applying TCT reported 7 false defect reports with the median number of false defects again being 0. The statistics indicate that on average both students and practitioners found similar number of false defects when applying TCT. This is also confirmed by Mann-Whitney U test (the data had a non-normal distribution) at $\alpha = 0.05$ ($p = 0.55, \hat{A}_{12} = 0.55$) which indicates non-significant differences in the median number of false defect reports submitted by students and practitioners when applying ET.

### 3.5 Discussion

This section answers the stated research questions and discusses the stated hypotheses.

#### 3.5.1 RQ 1: How do the ET and TCT testing approaches compare with respect to the number of defects detected in a given time?

In this experiment subjects found significantly more defects when using ET. Hence, we claim, it allows us to reject the null hypothesis: $H_{0,1}$. This result is different from the study by Itkonen et al. [33] where ET revealed more defects, but the difference was not statistically significant. On the other hand, the total effort of the TCT approach in their experiment was considerably higher. One plausible explanation for this is the difference in the experimental design. In this experiment the test case design effort was included in the testing sessions comparing identical total testing effort, whereas in the earlier experiment by Itkonen et al. the significant test case pre-design effort was not part of the testing sessions, comparing identical test execution effort. Considering this difference, our results, where ET shows a significantly higher defect detection efficiency, is in line with the earlier results by Itkonen et al. [33].

The answer to RQ 1 provides us with an indication that ET should be more efficient in finding defects in a given time. This means that documentation of test cases is not always critical for identifying defects in testing, especially if the available time for testing is short. Thus, our experiment shows that ET is efficient when it comes to time utilization to produce more results with minimum levels of documentation (see Subsections 3.3.4 and 3.3.6 for a more detailed description of type of documentation...
used in this experiment). It is important to note that this comparison focuses on the testing approach, meaning that we got superior effectiveness and efficiency by applying the ET approach to the same basic testing techniques as in the TCT approach. We also analyzed the level of documentation done for ET and TCT by subjects. The subjects performing ET provided on average 40 lines of text and screenshots as compared to on average 50 lines of text and screenshots for TCT. The documentation provided by subjects performing ET included brief test objective, steps to reproduce the identified defect and the screenshot of error message received. The subjects performing TCT documented all their test cases before test execution with steps to perform a test case with expected results. Similar to the ET group, they also provided the screenshot of error message received.

Our data in this study does not allow more detailed analysis of the reasons for the efficiency difference. One hypothesis could be that the achieved coverage explains the difference, meaning that in ET testers are able to cover more functionality by focusing directly on testing without the separate design phase. Other explaining factor could be cognitive effects of following a detailed plan. These aspects are important candidates for future studies.

3.5.2 RQ 2: How do the ET and TCT testing approaches compare with respect to defect detection difficulty, types of identified defects and defect severity levels?

The experimental results showed that ET found more defects in each of the four modes of defect detection difficulty. Moreover the difference in the number of defects found in each of the modes, by the two approaches, was found to be statistically significant. Therefore, we are able to reject the null hypothesis: $H_{0.2.1}$. This result strongly indicates that ET is able to find a greater number of defects regardless of their detection difficulty levels. Even more important is the finding that the distribution of found defects, with respect to ET, showed higher percentages for mode 2 and 3 (more complicated to reveal). Based on this data ET is more effective to reveal defects that are difficult to find and TCT, in addition to revealing fewer defects, also reveals more straightforward ones. This indicates that it is challenging for both students and practitioners to design good test cases that would actually cover anything but the most simple interactions and combinations of features, while, when using ET, the subjects are able to more effectively test also the more complicated situations.

For detection difficulty of defects, although the different modes of detection difficulty differed within students and practitioners when applying ET, between students and practitioners there were no significant differences found. When TCT was used,
there were again no significant differences found between students and practitioners for different modes of defect detection difficulty. There was however a trend observed: both students and practitioners, whether applying ET or TCT, detected greater number of easier defects as compared to difficult defects.

In terms of the technical type of defects, ET found, again, a higher number of defects in each of the technical type categories in comparison to TCT (exception being ‘documentation’ type). The differences were found to be statistically significant, therefore the null hypothesis, $H_{0.2.2}$, is rejected. When the distributions of defects regarding technical type are compared, an interesting finding is that TCT revealed a higher percentage of GUI and usability defects. One would expect that ET reveals more of these often quite visible GUI level defects (and usability defects), since the documented test cases typically focus on functional features rather than on GUI level features.

For the different types of defects, there were significant differences within students and practitioners when applying ET, however no significant differences were found between the two groups. A trend observed was that, when applying ET, both students and practitioners found greater proportions of missing function, performance and wrong function defects as compared to remaining type of defects. When TCT was used, there were again no significant differences found between students and practitioners for different type of defects, however a trend common in both subject groups was that more wrong function type of defects were identified than any other type.

In terms of the severity of defects, the actual numbers found by ET are greater than TCT for each of the severity levels and the differences are also statistically significant. We are therefore able to reject the null hypothesis: $H_{0.2.3}$. Considering the distribution of defects, our results show clear differences between the two approaches. The results indicate that ET seems to reveal more severe and critical defects and the TCT approach more of normal and negligible level defects.

For the severity level of defects, there were significant differences within students and practitioners when applying ET, however no significant differences were found between the two subject groups. A trend observed was that, when applying ET, both students and practitioners found greater proportions of normal and sever defects in comparison to remaining severity level of defects. When TCT was used, there were significant differences found within students and practitioners, however between groups there were no significant differences. A trend observed was that more normal severity level defects were identified by the two groups of subjects when TCT was applied.

The answer to RQ 2 is that in this experiment ET was more effective in finding defects that are difficult to reveal and potentially also effective in finding more critical defects than TCT. The TCT approach led testers to find more straightforward defects as well as, to a certain extent, GUI and usability related defects. In addition, TCT revealed proportionally more intermediate and negligible severity level defects. This
could be explained by the fact that test cases were written and executed in a short time and that the subjects were not able to concentrate enough on some of the potentially critical functionality to test. On the other hand, testers did design and execute the tests in parallel when using ET and, hence, ET might have enabled the testers to use their own creativity, to a higher extent, to detect more defects. Our results support the claimed benefits of ET. For defect detection difficulty, technical types, and severity of defects, the differences are higher than reported in the study by Itkonen et al. [33].

3.5.3 RQ 3: How do the ET and TCT testing approaches compare in terms of number of false defect reports?

Our experimental results show that testers reported more false defect reports using the TCT approach. The difference in comparison to ET is smaller than what was reported by Itkonen et al. [33]. This result, even though not statistically significant, might indicate that there could be some aspects in the use of detailed test cases that affect the testers’ work negatively. One explanation can be that when using test cases testers focus on the single test and do not consider the behavior of the system more comprehensively, which might lead to duplicate and incorrect reports. This, in one way, could indicate that ET allows a better understanding of how a system works and a better knowledge of the expected outcomes. Based on the data, however, we are not able to reject $H_{0,3}$.

Our experimental results also confirm non-significant differences in the number of false defect reports submitted by the two subject groups, when applying either ET or TCT.

In summary the experimental results show that there are no significant differences between students and practitioners in terms of efficiency and effectiveness when performing ET and TCT. The similar numbers for efficiency and effectiveness for students and practitioners might seem a surprising result. However, one explaining factor could be the similar amount of domain knowledge that has been identified as an important factor in software testing [34, 10]. In this experiment the target of testing was a generic text editor application and it can be assumed that both students and professionals possessed similar level of application domain knowledge. Also the students selected for the experiment were highly motivated to perform well in the experiment as they were selected based on their prior performance in the course assignments (representing top 65%). It has also been found in certain software engineering tasks that students have a good understanding and may work well as subjects in empirical studies [53]. There is a possibility that if practitioners were given more time (for both ET and TCT), they might have detected more defects by utilizing their experience, however, the design of
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

this study does not allow us to quantify the impact of variation in testing time on how practitioners are utilizing their experience in performing testing. There can be another argument with respect to ET being a non-repeatable process if exact repeatability is required for regression testing. We mentioned in Section 3.1 that ET is perhaps not an ideal technique if precise repeatability for regression testing is required. We, however, do not have the empirical evidence to confirm or refute it since this experiment answers different research questions. The proponents of ET claim that ET actually adds intelligent variation in the regression testing suite by methodically considering choices in input selection, data usage, and environmental conditions [59]: “Testers must know what testing has already occurred and understand that reusing the same tired techniques will be of little bug-finding value. This calls for intelligent variation of testing goals and concerns”. This claim, as we have discussed, awaits empirical corroboration.

One more argument with respect to ET’s effectiveness is the difficulty in finding the actual outcome of a test case, when there are time-pressures or referring to detailed documentation (e.g., the user guide in this paper) is not practical. This might be a factor in the number of false defect reports submitted by subjects performing ET, but we did not perform any such analysis to confirm/refute this possibility.

3.6 Validity threats

Despite our best efforts to obtain valid and reliable results, we nevertheless, were restricted by experimental limitations. This section discusses the most serious threats to the validity of this experiment.

Internal validity: Internal validity with respect to comparing testing techniques means that the comparison should be unbiased.

The defect detection capability of a testing technique is largely dependent on many factors: the type of software under test (SUT), the profile of defects present in the SUT including types, probability of detection, and severity of the defects, and the training and skills of testers [15]. In our comparison of the two testing approaches the SUT was the same for both approaches, we did take into account the types of defects identified and the corresponding severity levels. However, we did not take into account a defect’s probability of detection due to the expected variation in its values caused by inherent subjectiveness in its calculation. The industry participants in our experiment were experienced professionals. Therefore, it is expected that they were able to use their intuition and understanding of the system, to a higher degree. The student participants were selected based on their scores in assignments (representing top 65%) and, hence, were expected to have performed according to their ability. We did not select the
bottom 35% students as subjects as there was a risk of their lack of knowledge in functional testing to be confounded with the end results. The bottom 35% students were also expected to lack motivation in performing the experiment seriously, an important factor in making the results of an experiment more trustworthy and interesting [29]. The lack of knowledge in functional testing and low motivation would also limit their ability to ‘continuous learning’, a concept that ET advocates. The cut-off choice for selecting student subjects (65%) can obviously change in a different context; it was just a better choice in our case given the performance of students. In addition, the application under test was a text editor that is a sufficiently familiar domain for both software development professionals and students. The TCT group did not design any test cases before the beginning of the testing session. Consequently, they were expected to have less time executing test cases in comparison with the corresponding ET group. We acknowledge that ideally there should have been more time for the experimental testing sessions but it is challenging to commit long hours from industry professionals and, furthermore, in an experimental setting like this, one always risks introducing fatigue.

**Conclusion validity**: Conclusion validity is concerned with the relationship between the treatment and the outcome. It refers to using statistical hypothesis testing with a given significance [60]. In our case, the statistical tests were done at $\alpha = 0.05$. It is difficult to provide arguments for a predetermined significance level for hypothesis testing [6] but $\alpha = 0.5$ is commonly used [39]. We did not do a power analysis that could have given us an indication of a more appropriate $\alpha$-level.

Throughout the statistical hypotheses testing, parametric tests were preferred over non-parametric tests provided that the data satisfied the underlying assumptions of each test (since the power efficiency of non-parametric tests is considered to be lower). If the assumptions of a certain parametric test were violated, we used a non-parametric counterpart of that test.

Experiments with humans usually have more variance, and the individual knowledge and skills of the subjects affects the outcome. It is important that in this experiment the knowledge and skill levels for both ET and TCT were similar. While one can claim that ET has larger variance than other techniques, this is not known until we perform further studies. It is, however, likely that ET outcomes depend on competency and experience of testers. This is also acknowledged in Section 3.1: *It is believed that ET is largely dependent on the skills, experience and intuition of the tester.* That is to say, a tester with little or no testing experience, will need more time to learn the new software and to comply with the test process. This is in contrast with an experienced tester who might outperform the inexperienced tester in terms of ET efficiency and effectiveness. This is an inherent limitation while experimenting with ET. This threat can be minimized by a careful selection of human subjects. We believe that our selection
Chapter 3. An Experiment on the Effectiveness and Efficiency of Exploratory Testing

of *experienced* industry professionals and *high-performing* students helped us reduce this threat. With regards to the replication of our experiment, an exact replication is infeasible, as is the case with every other software engineering experiment having human subjects [16], since variation in subject population (and hence the experience) and in contextual factors cannot be entirely eliminated. Thus what we recommend is a theoretical replication [45] of our experiment with a different target population and testing a variant of the original hypothesis, e.g., to investigate the impact of tester experience and domain knowledge on ET outcome. While worthwhile, our experiment is not designed to answer this question. The authors would like to invite future replications of this experiment and are happy to extend all the support for doing that.

**Construct validity**: Construct validity refers to generalizing the experimental results to the concept behind the experiment [60]. The two testing approaches were compared using two constructs: efficiency and effectiveness. The validity and reliability of these constructs are dependent on two factors. First, an identical test suite size for two testing techniques may not translate into the same testing cost [15]. However, this is more true when experimenting with different testing techniques (for example, random testing vs. condition testing). In this study, both groups were instructed to use equivalence partitioning, boundary value analysis and combination testing techniques. The two groups had the freedom to apply any of these techniques for defect detection during the testing session.

The second influencing factor for construct validity in this study is the use of fault seeding. Fault seeding has been used by a number of empirical studies in software testing [26, 30, 58]. The problem with fault seeding, on the other hand, is the potential bias when seeding faults for the purpose of assessing a technique [15]. However, we claim that this bias is somewhat mitigated in this study by systematically seeding a wide variety of faults. Moreover, the actual number of faults in the SUT were greater than the seeded faults and, hence, we can claim that we had a balance between actual and seeded faults and, thus, reaching our goal to have a high variance in the type of faults.

The third influencing factor for construct validity in this study is the use of a single java application as a SUT, which is related to the mono-operation bias [60] that is a threat to construct validity. Using only a single artifact might cause the studied construct to be under-represented. But the choice of jEdit as a test object has some merits, especially considering the current state of experiments in software engineering. Sjøberg et al. [52] conducted a survey of controlled experiments in software engineering. They report that 75% of the surveyed experiments involved applications that were either constructed for the purpose of the experiment or were parts of student projects. This is not the case with jEdit, which is a realistic application (for mature program-
mers) with its homepage\textsuperscript{13} stating hundreds of persons-years of development behind it. Moreover Sjøberg et al. [52] finds that there is no open-source application used in the surveyed experiments while jEdit is an open-source application. With respect to the size of the materials presented to the subjects, the survey [52] states that the testing tasks reported materials in the range of 25 to 2000 lines of code. Our test object is well above this range, with more than 80,000 LoC.

We hope that later experiments on ET can extend the use of test objects beyond our experiment. We sincerely believe that to build an empirical body of knowledge around test techniques, it is important to conduct a number of experiments since, unfortunately, there is no one single perfect study. Basili et al. [9] puts it perfectly: “[…] experimental constraints in software engineering research make it very difficult, even impossible, to design a perfect single study. In order to rule out the threats to validity, it is more realistic to rely on the \textit{parsimony} concept rather than being frustrated because of trying to completely remove all threats. This appeal to \textit{parsimony} is based on the assumption that the evidence for an experimental effect is more credible if that effect can be observed in numerous and independent experiments each with different threats to validity [15].”

**External validity**: External validity refers to the ability of being able to generalize the experimental results to different contexts, i.e., in industrial practice [60]. We conducted a total of four iterations of the experiment, three of which were done in industrial settings. This gives confidence that the results could be generalizable to professional testers. However, the testing sessions in this experiment were short and available time for testing was strictly limited. The SUT in this experiment was also small compared to industrial software systems. It is possible that the experimental results would not be generalizable to larger SUTs and bigger testing tasks. We believe that the results generalize to contexts where available time for a testing task is strictly limited, as in industrial context. We argue that for large and complex applications ET would probably be more laborious because analyzing the correct behavior would be more difficult. But the test case design would also require more effort in that context (i.e., transferring that knowledge into documented oracles). Our experiment is but a first step in starting to analyze these things.

The removal of bottom 35\% of the students also helped us avoid the external validity threat of interaction of selection and treatment [60], i.e., our subject population in case of bottom 35\% of the students would not have represented the population we want to generalize to.

The application domain was rather simple and easy to grasp for our subjects. This improved the validity of our results in terms of the effects of variations in the domain.

\textsuperscript{13}http://www.jedit.org/
knowledge between the subjects. The relatively simple, but realistic, domain was well suited for applying personal knowledge and experience as a test oracle. Our results from this domain do not allow making any statements of the effects of highly complicated application domain to the relative effectiveness or efficiency of the two studied testing approaches.

### 3.7 Conclusions and future work

In this study, we executed a total of four experiment iterations (one in academia and three in industry) to compare the efficiency and effectiveness of exploratory testing (ET) in comparison with test case based testing (TCT). Efficiency was measured in terms of total number of defects identified using the two approaches (during 90 minutes), while effectiveness was measured in terms of defect detection difficulty, defects’ technical type, severity levels and number of false defect reports. Our experimental data shows that ET was more efficient than TCT in finding more defects in a given time. ET was, also, found to be more effective than TCT in terms of defect detection difficulty, technical types of defects identified and their severity levels; however, there were no statistically significant differences between the two approaches in terms of the number of false defect reports. The experimental data also showed that in terms of type of subject groups, there are no differences with respect to efficiency and effectiveness for both ET and TCT.

We acknowledge that documenting detailed test cases in TCT is not a waste but, as the results of this study show, more test cases is not always directly proportional to total defects detected. Hence, one could claim that it is more productive to spend time testing and finding defects rather than documenting the tests in detail.

Some interesting future research can be undertaken as a result of this study:

- Empirically investigating ET’s performance in terms of feature coverage (including time and effort involved).

- Comparing ET’s performance with automated testing and analysing if they complementary.

- Understanding the customers’ perspective in performing ET and how to encourage ET for practical usage.

- Embedding ET in an existing testing strategy (including at what test level to use ET and how much time is enough to perform ET?).
• Develop a more precise classification of ‘tester experience’ so that we are able to quantify the relationship between experience and performance of ET.

• Understanding the limitations of the ET approach in industrial contexts (including when precise repeatability of regression testing is required).
REFERENCES

3.8 References


REFERENCES


REFERENCES


Chapter 4

Checklists to Support Test Charter Design in Exploratory Testing

Ahmad Nauman Ghazi, Ratna Pranathi Garigapati and Kai Petersen
In proceedings of 18th International Conference on Agile Software Development (XP2017), Cologne, Germany. 2017

Abstract: During exploratory testing sessions the tester simultaneously learns, designs and executes tests. The activity is iterative and utilizes the skills of the tester and provides flexibility and creativity. Test charters are used as a vehicle to support the testers during the testing. The aim of this study is to support practitioners in the design of test charters through checklists. We aimed to identify factors allowing practitioners to critically reflect on their designs and contents of test charters to support practitioners in making informed decisions of what to include in test charters. The factors and contents have been elicited through interviews. Overall, 30 factors and 35 content elements have been elicited.
Chapter 4. Checklists to Support Test Charter Design in Exploratory Testing

4.1 Introduction

James Bach defines exploratory testing as simultaneous learning, test design and test execution [3]. Existing literature reflects that ET is widely used for testing complex systems as well and is perceived to be flexible in all types of test levels, activities and phases [7][13]. In the context of quality, ET has amassed a good amount of evidence on overall defect detection effectiveness, cost effectiveness and high performance for detecting critical defects [1] [9] [10] [11] [13]. Session-based test management (SBTM) is an enhancement to ET. SBTM incorporates planning, structuring, guiding and tracking the test effort with good tool support when conducting ET [4].

A test charter is a clear mission for the test session and a high level plan that determines what should be tested, how it should be tested and the associated limitations. A tester interacts with the product to accomplish a test mission or charter and further reports the results [3]. The charter does not pre-specify the detailed test cases which are executed in each session. But, a total set of charters for an entire project generally include everything that is reasonably testable. The metrics gathered during the session are used to track down the testing process more closely and to make instant reports to management [11]. Specific charters demand more effort in their design whilst providing better focus. A test session often begins with a charter which forms the first part of the scannable session sheet or the reviewable result. Normally, a test charter includes the mission statement and the areas to be tested in its design.

Overall, the empirical evidence of how test charters are designed and how to achieve high quality test charters are designed are scarce. High quality test charters are useful, accurate, efficient, adaptable, clear, usable, compliant, and feasible [4]. In this study we make a first step towards understanding test charter design by exploring the factors influencing the design choices, and the elements that could be included in a test charter. This provides the foundation for further studies investigating which elements actually lead to the quality criteria described by Bach [4]. We make the following contributions:

C1: Identify and categorize the influential factors that practitioners consider when designing test charters.

C2: Identify and categorize the possible elements of a test charter.

The remainder of the paper is structured as follows: Section 4.2 presents the related work. Section 4.3 outlines the research method, followed by the results in Section 4.4. Finally, in Section 4.5, we present the conclusions of this study.
4.2 Related work

Test charters, which are an SBTM element plays a major role in guiding inexperienced testers. The charter is a test plan which is usually generated from a test strategy. The charters include ideas that guide the testers as they test. These ideas are partially documented and are subject to change as the project evolves [4]. SBTM echoes the actions of testers who are well experienced in testing and charters play a key role in guiding the inexperienced testers by providing them with details regarding the aspects and actions involved in the particular test session [2].

The context of the test session plays a great role in determining the design of test plan or the charter [4]. Key steps to achieve context awareness are, for example, understanding the project members and the way they are affected by the charter, and understanding work constraints and resources. When designing charters Bach [4] formulated specific goals, in particular finding significant tests quicker, improving quality, and increasing testing efficiency.

The sources that inspire the design of test charters are manifold (cf. [4][8][12]), such as risks, product analysis, requirements, and questions raised by stakeholders. Mission statements, test priorities, risk areas, test logistics, and how to test are example elements of a test charter design identified from the literature review and their description [1] [4] [6]. Our study will further complement the contents of test charters as they are used in practice.

4.3 Research method

Study Purpose and Research Questions: The goal of this study is to investigate the design of test charters and the factors influencing the design of these charters and their contents.

RQ1: What are the factors influencing the design of test charters? The factors provide the contextual information that is important to consider when designing test charters, and complements the research on context aware testing [4].

RQ2: What do practitioners include in their test charters? The checklist of contents supports practitioners to make informed decisions about which contents to include without overlooking relevant ones.

Interviews: Interviews (three face-to-face and six through Skype) were conducted with a total of nine industry practitioners through convenience sampling combined with choosing experienced subjects who are visible in the communities discussing ET (see Table 4.1).

The interviews were semi-structured, following the structure outlined below:
1. *Introduction to research and researcher:* The researchers provide a brief introduction about themselves, followed by a brief description on the research objectives.

2. *Collection of general information:* In this stage, the information related to the interviewee is collected.

3. *Collection of research related information:* This is the last stage where the factors and contents of test charters have been elicited.

*Data analysis:* All the interviews were recorded by consent of the interviewees and later transcribed manually. The qualitative data collected using literature review and interviews was later analyzed using thematic analysis [5]. After thoroughly studying the coded data, similar codes have been grouped to converge their meaning to form a single definite code.

*Validity:* The potential bias introduced by interviewing thought leaders and experienced people in the area who are favorable towards exploratory testing may bias the results, and hence may not be fully generalizable. Though, we have not put any value on the factors and contents elicited, and they may be utilized differently depending on context. That is, identifying the potential elements to include in test charters is the first step needed. To reduce the threat multiple interviews have been used. Using a systematic approach to data analysis (thematic analysis) also aids in reducing this threat.

### 4.4 Results

*RQ1: What are the factors influencing the design of test charters?* Based on interviews with test practitioners, 30 different factors have been identified (see Table 4.2). The

<table>
<thead>
<tr>
<th>Interview ID</th>
<th>Role</th>
<th>Experience in testing</th>
<th>Organizational size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Senior Systems Test Engineer</td>
<td>4 years</td>
<td>More than 500</td>
</tr>
<tr>
<td>2</td>
<td>Test Quality Architect</td>
<td>10 years</td>
<td>50-500</td>
</tr>
<tr>
<td>3</td>
<td>Test Specialist</td>
<td>10 years</td>
<td>50-500</td>
</tr>
<tr>
<td>4</td>
<td>Test Consultant</td>
<td>12 years</td>
<td>More than 500</td>
</tr>
<tr>
<td>5</td>
<td>Test Strategist</td>
<td>3 years</td>
<td>Less than 50</td>
</tr>
<tr>
<td>6</td>
<td>CEO, Test Consultant</td>
<td>30 years</td>
<td>More than 500</td>
</tr>
<tr>
<td>7</td>
<td>Test Manager</td>
<td>20 years</td>
<td>More than 500</td>
</tr>
<tr>
<td>8</td>
<td>CEO, Test Lead</td>
<td>4 years</td>
<td>50-500</td>
</tr>
<tr>
<td>9</td>
<td>Test Quality Manager</td>
<td>13 years</td>
<td>50-500</td>
</tr>
</tbody>
</table>
table provides the name of the factors as well as a short description of what the factor means.

We categorized the factors and identified the following emerging categories, namely:

- **Customer and requirements factors**: These factors characterize the customer and their requirements. They include: F01: Client Requirements, F10: Business Use case, F15: Quality requirements, F27: Client location, and F30: User Journey Map.

- **Process factors**: Process factors characterize the context of the testing in regard to the development process. They include: F21: Process Maturity Level and F25: SDLC Phase.


- **Project management factors**: These factors concern the planning and leadership aspects of the project in which the testing takes place. They include: F05: Timeframe, F06: Project Purpose, F12: Effort estimation, F17: Test Team Communication, F18: Project Plan, and F29: Project Revenue.


**RQ2: What do practitioners include in their test charters?** The interviews revealed 35 different contents that may be included in a test charter. Table 4.3 states the content types and their descriptions.

Similar to the factors we categorized the contents as well. Seven categories have been identified, namely testing scope, testing goals, test management, infrastructure, historical information, product-related information, and constraints, risks and issues.

- **Testing scope**: The testing scope describes what to focus the testing on, be it the parts of the system or the level of the testing. It may also describe what not to focus on and set the priorities. It includes: C02: Test Focus, C03: Test Level, C04: Test Techniques, C10: Exit Criteria, C14: Specific Areas of Interest, C19: Priorities, C28: Coverage, and C33: Omitted Things.

- **Testing goals**: The testing goals set the mission and purpose of the test session. They include: C07: Purpose, C22: Mission Statement, and C24: Target.
## Table 4.2: Factors influencing test charter design

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F01</td>
<td>Client Requirements</td>
</tr>
<tr>
<td>F02</td>
<td>Test Strategy</td>
</tr>
<tr>
<td>F03</td>
<td>Knowledge of Previous Bugs</td>
</tr>
<tr>
<td>F04</td>
<td>Risk Areas</td>
</tr>
<tr>
<td>F05</td>
<td>Time-frame</td>
</tr>
<tr>
<td>F06</td>
<td>Project Purpose</td>
</tr>
<tr>
<td>F07</td>
<td>Test Function Complexity</td>
</tr>
<tr>
<td>F08</td>
<td>Functional Flows</td>
</tr>
<tr>
<td>F09</td>
<td>Product Purpose</td>
</tr>
<tr>
<td>F10</td>
<td>Business Use-case</td>
</tr>
<tr>
<td>F11</td>
<td>Test Equipment Availability</td>
</tr>
<tr>
<td>F12</td>
<td>Effort Estimation</td>
</tr>
<tr>
<td>F13</td>
<td>Test Planning Checklist</td>
</tr>
<tr>
<td>F14</td>
<td>Product Characteristics</td>
</tr>
<tr>
<td>F15</td>
<td>Quality Requirements</td>
</tr>
<tr>
<td>F16</td>
<td>Test Coverage Areas</td>
</tr>
<tr>
<td>F17</td>
<td>Test Team Communication</td>
</tr>
<tr>
<td>F18</td>
<td>Project Plan</td>
</tr>
<tr>
<td>F19</td>
<td>General Software Design</td>
</tr>
<tr>
<td>F20</td>
<td>System Architecture</td>
</tr>
<tr>
<td>F21</td>
<td>Process Maturity Level</td>
</tr>
<tr>
<td>F22</td>
<td>Product Design Effects</td>
</tr>
<tr>
<td>F23</td>
<td>Feedback and Consolidation</td>
</tr>
<tr>
<td>F24</td>
<td>Session Notes</td>
</tr>
<tr>
<td>F25</td>
<td>SDLC Phase</td>
</tr>
<tr>
<td>F26</td>
<td>Tester</td>
</tr>
<tr>
<td>F27</td>
<td>Client Location</td>
</tr>
<tr>
<td>F28</td>
<td>System heterogeneity</td>
</tr>
<tr>
<td>F29</td>
<td>Project Revenue</td>
</tr>
<tr>
<td>F30</td>
<td>User Journey Map</td>
</tr>
<tr>
<td>Content type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>C01: Test Setup</td>
<td>Description of the test environment.</td>
</tr>
<tr>
<td>C02: Test Focus</td>
<td>Part of the system to be tested.</td>
</tr>
<tr>
<td>C03: Test Level</td>
<td>Unit, Function, System test, etc.</td>
</tr>
<tr>
<td>C04: Test Techniques</td>
<td>Test techniques used to carry out the tests.</td>
</tr>
<tr>
<td>C05: Risks</td>
<td>Product risk analysis.</td>
</tr>
<tr>
<td>C06: Bugs Found</td>
<td>Bugs found previously.</td>
</tr>
<tr>
<td>C07: Purpose</td>
<td>Motivation why the test is being carried out.</td>
</tr>
<tr>
<td>C08: System Definition</td>
<td>Type of system (e.g. simple/ complex).</td>
</tr>
<tr>
<td>C09: Client Requirements</td>
<td>Requirements specification of the client.</td>
</tr>
<tr>
<td>C10: Exit Criteria</td>
<td>Defines the “done” criteria for the test.</td>
</tr>
<tr>
<td>C11: Limitations</td>
<td>It tells of what the product must never do, e.g. data sent as plain text is strictly forbidden.</td>
</tr>
<tr>
<td>C12: Test Logs</td>
<td>Test logs to record the session results.</td>
</tr>
<tr>
<td>C14: Specific Areas of Interest</td>
<td>Where to put extra focus on during the testing.</td>
</tr>
<tr>
<td>C15: Issues</td>
<td>Charter specific issues or concerns to be investigated.</td>
</tr>
<tr>
<td>C16: Compatibility Issues</td>
<td>Hardware and software compatibility and interoperability issues.</td>
</tr>
<tr>
<td>C17: Current Open Questions</td>
<td>Existing questions that refer to the known unknowns.</td>
</tr>
<tr>
<td>C18: Information Sources</td>
<td>Documents and guidelines that hold information regarding the features, functions and systems being tested.</td>
</tr>
<tr>
<td>C19: Priorities</td>
<td>Determines what the tester spends most and least time on.</td>
</tr>
<tr>
<td>C20: Quality Characteristics</td>
<td>Quality objectives for the project.</td>
</tr>
<tr>
<td>C21: Test Results Location</td>
<td>Test results location for developers to verify.</td>
</tr>
<tr>
<td>C22: Mission Statement</td>
<td>One liner describing the mission of the test charter.</td>
</tr>
<tr>
<td>C23: Existing Tools</td>
<td>Existing software testing tools that would aid the tests.</td>
</tr>
<tr>
<td>C24: Target</td>
<td>What is to be achieved by each test.</td>
</tr>
<tr>
<td>C25: Reporting</td>
<td>Test session notes.</td>
</tr>
<tr>
<td>C26: Models and Visualizations</td>
<td>People, mind maps, pictures related to the function to be tested.</td>
</tr>
<tr>
<td>C27: General Fault</td>
<td>Test related failure patterns of the past.</td>
</tr>
<tr>
<td>C28: Coverage</td>
<td>Charters boundary in relation to what it is supposed to cover.</td>
</tr>
<tr>
<td>C29: Engineering Standards</td>
<td>Regulations, rules and standards used, if any.</td>
</tr>
<tr>
<td>C30: Oracles</td>
<td>Expected behavior of the system (either based on requirements or a person)</td>
</tr>
<tr>
<td>C31: Logistics</td>
<td>How and when resources are used to execute the test strategy, e.g. how people in projects are coordinated and assigned to testing tasks.</td>
</tr>
<tr>
<td>C32: Stakeholders</td>
<td>Stakeholders of the project and how their conflicting interests would be handled.</td>
</tr>
<tr>
<td>C33: Omitted Things</td>
<td>Specifies what will not be tested.</td>
</tr>
<tr>
<td>C34: Difficulties</td>
<td>The biggest challenges for the test project.</td>
</tr>
<tr>
<td>C35: System Architecture</td>
<td>Structure, interfaces and platforms concerning the system, and its impact on system integration.</td>
</tr>
</tbody>
</table>
4.4 Checklists to Support Test Charter Design in Exploratory Testing

- **Test management:** Test management is concerned with the planning, resource management, and the definition of how to record the tests. Test management includes: C12: Test Logs, C18: Information Sources, C21: Test Results Location, C25: Reporting, C26: Models and Visualizations, C31: Logistics, C32: Stakeholders, and C34: Difficulties.

- **Infrastructure:** Infrastructure comprises of tools and setups needed to conduct the testing. It includes: C01: Test Setup and C23: Existing Tools.

- **Historical information:** As exploratory testing focuses on learning, past information may be of importance. Thus, the historical information includes: C06: Bugs Found, C16: Compatibility Issues, C17: Current Open Questions, and C27: General Fault.

- **Product-related information:** Here contextual product information is captured, including: C08: System Definition, C13: Data and Functional Flows, and C35: System Architecture.

- **Constraints, risks and issues:** Constraints, risks and issues to testing comprise of the items: C05: Risks, C15: Issues, and C29: Engineering Standards.

### 4.5 Conclusion

In this study two checklists for test charter design were developed. The checklists were based on nine interviews. The interviews were utilized to gather a checklist for factors influencing test charter design and one to describe the possible contents of test charters. Overall, 30 factors and 35 content types have been identified and categorized.

The factors may be used in a similar manner and should be used to question the design choices of the test charter. For example:

- Should the test focus of the charter be influenced by previous bugs (F03)? How/why?
- Are the product’s goals (F09) reflected in the charter?
- Is it possible to achieve the test charter mission in the given time for the test session (F12)?
- etc.

With regard to the content a wide range of possible contents to be included have been presented. For example, only stating the testing goals (C22) provides much room
for exploration, while adding the techniques to be used (C04) may constrain the tester. Thus, the more information is included in the test charter the exploration space is reduced. Thus, when deciding what to include from the checklist (Table 4.3) the possibility to explore should be taken into consideration.

In future work we need to empirically understand (a) which are the most influential factors and how they affect the test charter design, and (b) which of the identified contents should be included to make exploratory testing effective and efficient.
4.6 References


Chapter 5

Exploratory Testing: One Size Doesn’t Fit All

Ahmad Nauman Ghazi, Kai Petersen, Elizabeth Bjarnason and Per Runeson

Abstract: Exploratory testing (ET) is a powerful and efficient way of testing software by integrating design, execution, and analysis of tests during a testing session. ET is often contrasted with scripted testing, and seen as a choice between black and white. We pose that there are different levels of exploratory testing from fully exploratory to fully scripted and propose a scale for the degree of exploration for ET. The degree is defined through levels of ET, which correspond to the way test charters are formulated. We have evaluated the classification through focus groups at four companies and identified factors that influence the level of exploratory testing. The results show that the proposed ET levels have distinguishing characteristics and that the levels can be used as a guide to structure test charters. Our study also indicates that applying a combination of ET levels can be beneficial in achieving effective testing.
5.1 Introduction

Advocates of exploratory testing (ET) stress the benefits of providing freedom for the tester to act based on his/her skills, paired with the reduced effort for test script design and maintenance. ET can be very effective in detecting critical defects [1]. We have found that exploratory testing can be more effective in practice than traditional software testing approaches, such as scripted testing [1]. ET supports testers in learning about the system while testing [1], [7]. The ET approach also enables a tester to explore areas of the software that were overlooked while designing test cases based on system requirements [6]. However, ET does come with some shortcomings and challenges. In particular, ET can be performed in very many different ways, and thus there is no one-way of training someone to be an exploratory tester. Also, exploratory testing tends to be considered an ad-hoc way of testing and some argue that defects detected using ET are difficult to reproduce [11]. We pose that there are different levels of exploratory testing, from fully exploratory to fully scripted and propose a scale for the degree of exploration for ET.

There is very little work on providing structure and guidance for ET to utilize different degrees of exploration, despite discussions on how to benefit from ET in both industry and academia [10]. Bach introduced a technique named Session Based Test Management (SBTM) [3] that provides a basic structure and guidelines for ET using test missions. SBTM provides a strong focus on designing test charters to scope exploration to the test missions assigned to exploratory testers. A test charter provides a clear goal and scopes the test session. It can also be seen as a high level test plan [4]. However, little guidance exists of how to utilize the test missions in order to achieve different degrees of exploration.

Recently, Ghazi et al., [4] provided a checklist of contents to support test charter design in exploratory testing. The focus of that research was to support practitioners in designing test charters depending on the context of the test mission and the system under test. However, it remains to explore different degrees of exploration and how they map to the contents of the test charters, e.g. test goals, test steps, etc.

In this article, we present a classification of different levels of exploratory testing (ET), from free style testing to fully scripted testing. We exemplify the ET levels with test charter types that were defined based on studying existing test charters in industry. We have evaluated the classification and the test charter types through focus groups at four companies, Axis Communications, Ericsson, Softhouse Consulting, and Sony Mobile Communications. The focus groups also provided insight into factors that influence the ET levels.
5.2 Exploratory testing

Exploratory testing (ET) is a way to harness the skills, knowledge and creativity of a software tester. The tester explores the system while testing, and uses the knowledge gained to decide how to continue the exploration. The design, execution, and analysis of tests take place in an integrated fashion [12]. The experience and skills of the tester plays a vital role in ET and influences the outcome of the testing [5], [9]. ET displays multiple benefits, such as testing efficiency and effectiveness [1], [10], a goal-focused approach to testing, a high degree of ease of use, flexibility in test design and execution, and providing an interesting and engaging task for the tester [10].

Shah et al. [11] conducted a systematic review of the literature on ET, and found that the strengths of ET are often the weaknesses of scripted testing and vice versa. They conclude that ET and scripted testing should be used together to address the weaknesses of one process with the strength of the other. While Shah et al. do not consider different types of ET, other research highlights the existence of different degrees of exploration.

5.3 Classification of levels of exploratory testing

We have identified five levels of exploratory testing (ET levels) ranging from free style exploratory testing, to fully scripted testing. Figure 5.1 provides an overview of the proposed classification. Each of the five levels are defined by a test charter type that guides the testing. The test charter for each ET level adds an element that sets the scope for exploration space for the tester, from the level of free style testing to fully scripted testing, which is the least exploratory level. On the freestyle level, the tester is only provided with the test object. The exploration space is reduced for each level, by adding further information to the test charter, e.g. high-level goals. The tester is thus further focused for each decreasing ET level, and the reduced freedom leads to a less exploratory approach compared to the previous level. The test charter type for the lowest ET level, i.e. fully scripted, contains both test steps and test data, and thus leaves no space for exploration during test execution.

We provide examples of test charters produced during one of the focus groups in Figure 5.2 The test charter for the highest ET level contains only the goal for the testing, namely to verify a specific function of the system. The test charter for the medium ET level contains additional information, e.g. suitable starting points for the testing. Finally, the test charter for the low ET level contains detailed test activities/steps in addition to goals and other information. The next level, i.e. fully scripted, which is not shown in Figure 5.2, also contains test data. For example, the test charter for the test activity Copy content to card from PC would also specify the content to be copied.
### Figure 5.1: Classification of Levels of Exploratory Testing: ET Levels

<table>
<thead>
<tr>
<th>High degree of exploration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The tester can freely explore the system.</td>
</tr>
<tr>
<td>The tester is provided with one or more high goals for the test session.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Medium degree of exploration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowing the test object, besides that, the tester can freely explore the system.</td>
</tr>
<tr>
<td>The tester is provided with one or more high goals for the test session, also</td>
</tr>
<tr>
<td>during session. Blazing aspects could be too detailed goals, promote a focus on the same time additional restrictions are required, that may bias and thus limit the tester in his/her testing.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Low degree of exploration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The tester is encouraged to choose the test data to be used in the test steps.</td>
</tr>
<tr>
<td>Besides the information for medium degree of exploration, the tester is also</td>
</tr>
<tr>
<td>required to follow certain test steps, which further may bias the tester and reduce the exploration space.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fully scripted:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The tester is provided with the test steps, but also with the test data, which does not</td>
</tr>
<tr>
<td>provide room for exploration steps.</td>
</tr>
</tbody>
</table>

### Chapter 5. Exploratory Testing: One Size Doesn't Fit All
<table>
<thead>
<tr>
<th></th>
<th>High degree</th>
<th>Medium degree</th>
<th>Low degree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test goal/ purpose</strong></td>
<td>To verify adoptable storage</td>
<td>1. Test different SD cards (speed and size)</td>
<td>1. To verify that adopt.storage behaves (as expected) according to the requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Test SD card and internal/external memory</td>
<td>2. To test that no data loss occurs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Test move SD card to other device (phone/PC etc)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Test extract files from SD card</td>
<td></td>
</tr>
<tr>
<td><strong>Set-up (pre-conditions of what needs to be available to test)</strong></td>
<td></td>
<td>1. PC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. 2 devices supported by adopt.storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. SD cards 1 … n</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. 1 device not supporting adopt.storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Test content</td>
<td></td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td></td>
<td>High. There must not be any data loss</td>
<td></td>
</tr>
<tr>
<td><strong>References</strong></td>
<td></td>
<td>See AlMreg.doc</td>
<td></td>
</tr>
<tr>
<td><strong>Test activities</strong></td>
<td></td>
<td>1. Insert SD card</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Setup as internal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Copy content to card from PC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Read content on card from device</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Save</td>
<td></td>
</tr>
<tr>
<td><strong>Additional information used</strong></td>
<td>1. Use previously identified problems as input</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Use Google requirements if needed</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2: Example of test charters for the high, medium and low ET levels
5.4 Methodology and case details

This article is based on the study of test processes at four companies in Sweden involved in large-scale product development in the area of telecommunications and embedded systems. The companies we studied are Axis Communications, Ericsson, Softhouse Consulting, and Sony Mobile Communications. An overview of the companies is provided below. The companies selected the focus group participants based on their experience of testing and their interest in exploratory testing.

We used focus groups [8] as the method for data collection at all four companies and conducted them in two main iterations. The initial two focus groups were exploratory and were conducted at two companies interested in extending their use of exploratory testing in their current test processes (Axis Communications and Sony Mobile Communications). These first two focus groups contained the following steps:

1. Introduce the basic concepts of exploratory testing
2. Present the classification of exploration levels
3. Share examples of test charters type for each level with the participants
4. The participants re-write an existing test case at the different exploration levels using the provided test charter types
5. Open discussion of how each level and test charter types matches the context for their current test practices
6. Elicit factors that affect the level of exploration in testing

We conducted the third and fourth focus groups at Ericsson and at Softhouse Consulting to validate the results from the initial focus groups. The participants were mainly experienced testers, each with more than 20 years of experience of software testing, with a strong focus on exploratory testing. We first conducted a survey with the focus group participants to gauge their views of the degree of impact of ET levels on the factors elicited from the first two focus groups (such as learning). We discussed the outcome of the survey at the (subsequent) focus group. In particular, we discussed the impact on the ET levels of exploration and reached a consensus for each factor.

We audio recorded and transcribed the focus group sessions, and analyzed these to identify resulting advantages and disadvantages of each ET level in the proposed classification.
5.4.1 Threats to validity

The participating practitioners have not experienced all the ET levels and the corresponding test charter types discussed in the focus group. However, they may relate to them given their experience of testing. We reduced this threat by letting the practitioners gain hands-on experience of the test charter types during the focus group (Step 4 above). A common threat of studies with companies is the generalizability of the findings. We partially reduce this threat, by involving four companies. We mitigated the risk of research bias by involving three different researchers in performing the focus groups and in jointly discussing the outcome of these.

5.5 Results from the focus groups

We conducted focus groups in four companies (Axis Communications, Ericsson, Soft-house Consulting and Sony Mobile Communications) to evaluate the classification of ET levels and explore factors that influence these. We outline our research method in section 5.4. The participants of the focus groups discussed factors that influence the ET levels and the corresponding test charter types. We identified six main areas that influence the levels, namely defect detection, time and effort, people-related factors, evolution and change, traceability and quality requirements. We show these factors in Figure 5.3 by presenting two opposing poles for each factor. For example, better learning (indicated as positive by ☺) versus poor learning (indicated as negative by ☹). If the impact on the ET level is neutral this is indicated by ☻.

Overall, the practitioners had a positive view of the higher ET levels, i.e. freestyle and high exploration, for four of the six main areas. The participants noted a positive impact for these levels within the areas of defect detection, time and effort, people-related factors, and evolution and change. In contrast, they expressed a negative impact for factors related to traceability and verifying quality requirements. The participants believed that the more exploratory ET levels have a negative impact on these two areas.

Defect detection: The members of all focus groups highlighted that the exploratory approach will identify more significant defects. However, one participant stated that this may only be the case if you know what the faults may be, i.e. the tester should have the skills to identify where significant faults are most likely to occur. Thus, the testers skills plays a vital role in exploratory testing, as also confirmed by empirical studies on ET [9]. These skills are also required to judge whether an explored behavior is a critical defect or not. However, the participants also pointed out that people taking a new perspective often find new defects. One practitioner said: every time we get new people in the team, we find new defects in the system. This highlights one of
the benefits of exploration, namely that of not biasing the search for defects, for example, through pre-existing test cases and prior knowledge embedded in scripted tests. This may be the case for lower ET levels, i.e. low exploration and fully scripted. Some participants pointed out that high exploration comes with challenges with reproducibility of detected defect. One participant said that the problem is when you have higher degree of exploration the developers want to have very detailed steps to reproduce it. However, when you focus on the reproducing you lose the exploration which is a

Figure 5.3: Overview of factors influencing the ET levels derived from the focus groups and the survey
drawback with the fully scripted level.

**Time and effort:** Many participants highlighted time efficiency as one of the benefits of the high and medium ET levels. One practitioner explained this by saying that we can get a better overview quickly with higher ET levels. These higher ET levels also require less effort to prepare the test, compared to the lower and fully scripted ET levels. One participant explained that the many details of the low ET levels require an upfront investment to develop test cases before you can execute them. Another participant adds by saying that there is less administration if you have a high degree of exploration because then you have quite openness and it is much easier to write test cases. The focus group participants indicated that the effort to maintain test cases at the higher ET levels is less since changes are more likely to affect the level of detail of test cases at the lower ET levels. For example, the tester would need to update the test steps.

**People factors:** The participants highlighted that the higher ET levels are beneficial for encouraging critical thinking, challenging the system when testing, and that they support learning. One participant said that at the highest ET level (freestyle) learning might take longer time. But that it would probably be a better approach from the beginning to understand what the testing is. This participant also said that you only do fully scripted when you know the system and it is monotonous and you can also get tired of it. However, some participants also expressed positive learning effects from fully scripted testing and that it is definitely easier to start learning testing when it is fully scripted. If we do freestyle then it would be difficult because it requires skills and some form of competence or otherwise you are completely lost. One participant suggested that to make full use of the higher ET levels, i.e. freestyle and high exploration you have a mentor that tells you explore this and then you explore and test. When you have questions, you go back and ask/discuss with the mentor. Several participants pointed out that a tester’s experience plays an important role and that less experienced testers are often able to identify new defects since they bring a new perspective to a project. At the same time, a tester with less experience may find it hard to conduct freestyle or high exploration testing since they do not have the domain knowledge required. Hence, there are additional factors that affect the degree of learning. Participants also pointed that with scripted testing one problem can be that if you just keep following the test steps then there is a chance that you miss the approval criteria. Finally, the participants pointed out that learning from the requirements occurs during the derivation of the tests from the detailed requirements. They also highlighted motivation as an important distinguishing factor, where testers quickly get bored when testing at low degrees of exploration included fully scripted testing. The participants also highlighted that the impact and effect of the ET levels may vary well vary throughout the development cycle. They said that the higher ET levels might be particularly useful during the early
phases of testing to explore and learn about the system. The testers may then design new tests that later become scripted tests, which are used for regression testing in later stages when the project is closer to releasing software.

**Evolution and change:** The participants highlighted that it is easier to design new tests for higher ET level (freestyle and high degree of exploration) since this requires less effort; you have an openness and it is much easier to write test cases. In line with this, the participants also expressed that changes can be more easily implemented given that the higher ET levels are more resistant [to change] since you don't need to change a lot of details. They also expressed that the communication around changes to tests is simplified for these higher ET levels and that when some behavior has changed and you just discuss and notify that this has changed instead of going in details every time. However, the practitioners also said that the higher ET levels are more challenging when requirements are added or changed, since information of the new requirements is needed to guide the testing.

**Traceability:** All focus groups have highlighted that the difficulties of tracing coverage is a major drawback for higher ET levels both regarding coverage of code and of requirements. One participant said: The sense of coverage is much lower as compared to when you ticked off 100 test cases in scripted tests. This issue also applies to requirements coverage since test cases at the higher ET levels per definition do not include any mapping to individual requirements.

**Quality requirements:** Several participants highlighted that the higher ET levels are not suitable for conformance requirements. One participant said that We do have a lot of conformance with different standards and legal requirements and if you don't have this kind of [low] ET level then it is easier to miss. The participants expressed different views on this regarding performance. On the one hand, load on the system may be better generated with scripted automated tests. In one case, a participant highlighted that for performance testing you have to continuously compare it to different firmware and we need to have similar tests again then we can't really explore a lot. However, it is also important to consider the end user perspective and make observations while testing at higher ET levels.

### 5.6 Conclusions and Future Work

Earlier in this article, we discussed that there have been some attempts in past to provide structure to ET and guide the test process by defining clear test missions as well as time-boxing the sessions. Our current work aims to provide practitioners a better understanding of ET practices by introducing levels of exploration. The levels of exploration that we propose have distinct elements that help the practitioners to distin-
guish each level clearly. This distinction of exploration levels, in turn, would facilitate the test teams to make informed decisions to more effectively test their software.

Exploratory testing can find critical and otherwise missed defects by utilizing the skill and creativity of the tester, while being insufficient for verifying conformance to requirements due to providing weak coverage of requirements. In contrast, scripted testing provides this and is a vital component in regression testing. However, the question about exploratory testing is not whether or not to apply it, but rather which levels of exploratory testing to apply to achieve the desired outcome.

We have identified five levels of exploratory testing (ET levels) from fully scripted to fully exploratory, or freestyle, and have explored factors that influence these level through a series of focus groups. Our research shows that the ET levels influence factors such as the ability to detect defects, test efficiency, learning and motivation, and that different outcomes are to be expected depending on the choice of ET level. This allows testers to select the ET level according to what they want to achieve with their testing. For example, testers operating at higher ET levels, e.g. freestyle, can expect to achieve improved defect detection, savings in time and effort, and facilitated management of evolution and change. They can also expect a positive impact with regards to learning and motivation. Though there are drawbacks too, since the higher ET levels are weak in supporting traceability and verification of quality requirements concerning conformance and performance. Another drawback with high ET levels is the weak reproducibility of defects, as the test steps are not clearly documented for developers to follow to reproduce the defect. However, we note that recent research studies provide solutions for tracking the testing session to later derive and repeat the test steps [2].

We encourage practitioners to consider striving for a combination of exploratory and scripted testing. In this way, testers can obtain the positive effects of the higher degrees of exploration, while not neglecting other types of testing. As one participant stated at the end of the focus group: we [now] think that we want both scripted and exploratory; a mix of both approaches, so that we can approach our testing in different ways. We also found that the test charters used to define the ET levels provided practical value to the participants. The practitioners quickly grasped the differences between the ET levels by viewing and applying these test charter types. We suggest that practitioners reflect on the ET levels by using a similar approach. They can explore and reflect on how the various ET levels could support them by rewriting existing charters or scripted tests according to the corresponding test charter types.
Chapter 5. Exploratory Testing: One Size Doesn’t Fit All

Acknowledgements

We would like to thank the participating companies and in particular the individuals for their active involvement in and support of this research.

This work was partly funded by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se).
5.7 References


REFERENCES


Chapter 6

A Decision Support Method for Recommending Degrees of Exploration in Exploratory Testing

Ahmad Nauman Ghazi, Kai Petersen, Claes Wohlin and Elizabeth Bjarnason
Submitted to EuroMirco SEAA 2017

Abstract: Exploratory testing is neither black nor white, but rather a continuum of exploration exists. In this research we propose an approach for decision support helping practitioners to distribute time between different degrees of exploratory testing on that continuum. To make the continuum manageable, five levels have been defined: freestyle testing, high, medium and low degrees of exploration, and scripted testing. The decision support approach is based on the repertory grid technique. The approach has been used in one company. The method for data collection was focus groups. The results showed that the proposed approach aids practitioners in the reflection of what exploratory testing levels to use, and aligns their understanding for priorities of decision criteria and the performance of exploratory testing levels in their contexts. The findings also showed that the participating company, which is currently conducting mostly scripted testing, should spend more time on testing using higher degrees of
6.1 Introduction

Exploratory testing (ET) is an approach to test software with a focus on personal freedom and the skills of the tester [1]. In ET, test execution is done without pre-defined scripted test cases. As the focus in ET remains on testing without pre-defined scripted tests, it is mostly considered an ad-hoc test approach. However, over the decades, ET has evolved to be a more structured approach without compromising the basic notions of personal freedom and individual responsibility of the testers. Session-based test management (SBTM) is an enhancement of ET. SBTM incorporates planning, structuring, guiding and tracking the test effort with good tool support [2]. Furthermore, SBTM reflects the concept of time boxing the test sessions to track testing where a test session is a basic testing work unit and an uninterrupted block of reviewable and chartered test effort [3]. Most practitioners claim that the test charter is an important element of SBTM [2], [3] and [4].

As pointed out by Itkonen et al. [5] exploratory testing has different degrees of exploration. The test charters are a means to steer the degree of exploration. Practitioners highlighted the need for support in selecting how much scripting testing to be conducted versus exploratory testing [6]. To date as illustrated in Itkonen et al. [5] only a few distinct levels of exploration are confirmatory (scripted) testing and exploratory testing, as well as pure exploratory testing (freestyle). In a previous study, Ghazi et al. [7] proposed five distinct levels of exploration in testing, namely: 1) Freestyle testing, 2) High degree of exploration, 3) Medium degree of exploration, 4) Low degree of exploration, and 5) Scripted testing.

The main contribution of this paper is the proposal of an approach for supporting practitioners in the decision of how to distribute their testing time between different levels of exploration. In particular, the approach aims at supporting the practitioners in their reflection and discussion.

The following sub-contributions are made:

- Describe the approach to facilitate its adoption in industry.
- Present factors relevant to consider when choosing between ET levels.
- Application of the approach in practice.

To achieve the contributions we adopted the repertory grid technique [8], which aids in group decision making. A central part of the repertory grid technique are decision criteria (so-called constructs), which were specifically identified for ET levels.
Focus groups were used to apply the approach in industrial practice. The application is illustrated and lessons learned from the application are presented.

The remainder of the paper is structured as follows: Section 6.2 presents the related work on exploratory testing and test charter design. In Section 6.3, we propose a decision support method for selecting levels of exploration. Section 6.4 presents the application of proposed solution in an industrial context. The results are discussed in section 6.5, and section 6.6 provides the conclusions from the research.

6.2 Related work

Exploratory testing is defined as simultaneous learning, test design and test execution [9]. ET is perceived to be flexible and applicable to different types of activities, test levels and phases [4]. Existing literature showcases a good amount of evidence regarding the merits of ET, such as its defect detection effectiveness, cost effectiveness and high performance for detecting critical defects [4], [10], [11] and [12]. During the exploratory testing process, the testers may interact with the application and take the information it provides to react, change course or explore the application’s functionality without any constraint [13]. ET is usually done in an iterative fashion [14] to facilitate continuous learning. The factors on which the effectiveness of ET depends are software maturity, the skills of the tester, the product being tested and the time required to test the product [9].

Session-based test management (SBTM) is an enhancement of ET that helps in tracking the individual tester’s ET progress. In SBTM, the test results are reported in a consistent and accountable way [14]. Session-based test management is a technique that helps in managing and controlling tests that are unscripted. It sets a framework around unscripted testing and builds on its strengths such as the speed, flexibility and range. These unscripted tests can be controlled. Thus, they form a powerful part of the overall test strategy [15], which is a set of ideas that guide the choice of test that in turn guide the test design. Also, the test strategy includes a set of ideas related to project environment, product elements, quality criteria and test techniques [2].

Test charters are a means to guide testers during the test session. Charters may include a range of information, such as the goals of the test session, prerequisites that need to be fulfilled before starting testing, or the acceptance criteria for the test session. Test charters are not static and may evolve over time [2]. In a recent study, Ghazi et al. [16] proposed checklists to support test charter design in ET. The study provides a list of contents to help practitioners design test charters, as well as factors that they ought to consider when designing their test charters. Although the usage of these checklists help providing some structure to exploratory testing, there is a trade-off
on the freedom of exploration. The more contents from the checklist that are included in the test charter, the less freedom for the tester to explore and execute the tests. In another study Ghazi et al. [7], suggest that exploration in ET is a continuum between two extremes: freestyle exploratory testing and scripted testing. Furthermore, they proposed a taxonomy of levels of exploration in ET where a clear distinction between the levels is made through characterising the levels with the elements in a test charter. This study serves as the foundation for our current work, proposed herein.

6.3 Solution proposal

6.3.1 General method

In the 1950s, George Kelly proposed the personal construct theory (PCT) and the associated repertory grid technique to elicit and analyse these personal constructs [8]. The basic idea behind the PCT is that individuals, based on their observations of their surroundings, have their own view of the world. Therefore, each person builds his/her own conceptual framework which results into having different opinions about the same problem [17]. Individuals constantly observe and react to understand their surroundings. In turn, they continuously construct and reform their personal theories and assumptions [18].

The repertory grid technique proposed by George Kelly [8], is an approach to elicit, evaluate and analyse the personal constructs of individuals. A typical repertory grid is based on the following three basic concepts:

- **Element Elicitation**: The individual aspects or objects of a topic people try to understand are termed as different elements. The technique itself provides freedom to the researcher to supply these elements to the participants so that participants can focus on eliciting only the constructs. Another way is to be more flexible and ask the participants to suggest the elements. However, this approach will inevitably affect the structure and standardisation of the grid [18].

- **Construct Elicitation**: Like element elicitation, constructs can also be either supplied by the researcher or may be elicited from the participants. Edwards et al. [18] suggest that for exploratory research, constructs must be elicited whereas for evaluative research constructs may be elicited.

- **Rating**: The elicited elements and constructs need to be evaluated. A construct is formed of two contrasting concepts that are weighted on a bipolar scale. The elements are then rated against each construct. An example is whether an element (such as a specific ET level) has a negative, neutral, or positive effect on a construct (e.g. facilitates learning or hinders learning).
6.3.2 Tailoring for ET

In previous work [7], we conducted focus groups to understand the advantages and disadvantages of different degrees of ET. In Figure 6.1, five levels of exploration in testing are presented. These provided the initial building blocks for constructing the repertory grid. The research was done in close collaboration with industry to ensure practical relevance. The industry partners include Sony Mobile Communications and Axis Communications.

Element elicitation: For element elicitation, we used the approach proposed by Edwards et al. [18] where they suggested to supply elements to the participants. For this purpose, we identified five distinct levels of exploration in exploratory testing ranging between freestyle ET and scripted testing. These levels of exploration are differentiated based on the details mentioned in a test charter. Furthermore, it is important to note that within exploratory testing, test charters play an extremely important role to drive the exploration as well as to provide structure to exploratory testing practices. Figure 6.1 provides a summary of the identified levels of exploration and a taxonomy of test charters.

Construct elicitation: Eliciting the personal constructs is considered the most important part of the repertory grid technique. We used two exploratory focus groups with six participants in the first group and four participants in the second group to elicit the constructs. These focus groups were conducted at Sony Mobile Communications and Axis Communications respectively [7]. The exploratory focus groups contained the following steps:

1. Basic concepts of exploratory testing are introduced to the participants,
2. A classification of exploration levels is presented,
3. Different examples of test charters for each level were shared with the participants,
4. The participants were asked to re-write an existing test case at the different exploration levels using the provided test charter types,
5. Open discussion of how each level and test charter type matches the context for their current test practices,
6. Elicitation of factors or personal constructs that affect the level of exploration in testing.

Based on the discussion with testers in the focus groups, a total of 17 constructs were elicited. Table 6.1 provides an overview of the elicited constructs.
Figure 6.1: The degrees of exploration [7] used as elements in the repertory grid for ET

<p>| Low degree of exploration: The tester is provided with medium degree of exploration, the tester is also expected to follow certain test steps, which further may bias the tester and reduce the exploration space. |
|-----------------|-------------------------------------------------------------|
| Medium degree of exploration: The tester is provided with one or more high goals for the test session. At the same time additional restrictions are required, that may bias and thus limit the tester in his/her testing session. |
| High degree of exploration: The tester can freely explore the system, besides that the tester can freely explore the system, knowing the test object. Besides that the tester can freely explore the system, the tester can freely explore the system, knowing the test object. |
| Fully scripted: The tester is provided with the test steps, but also with the test data, which does not provide room for exploration steps. |
| Freestyle: Only the test object is provided to the tester. The tester can freely explore the system. |</p>
<table>
<thead>
<tr>
<th>ID</th>
<th>Similarity pole</th>
<th>Contrast pole</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Better learning</td>
<td>Poor learning</td>
<td>Refers to the learning that occurs during the test session (including learning to test and learning the system).</td>
</tr>
<tr>
<td>C2</td>
<td>Easy to trace coverage</td>
<td>Hard to trace coverage</td>
<td>Ability to determine the coverage after testing has been concluded (e.g. coverage of functions/code etc.).</td>
</tr>
<tr>
<td>C3</td>
<td>Time efficient</td>
<td>Time inefficient</td>
<td>Resource efficiency (time to conduct the tests during the test session).</td>
</tr>
<tr>
<td>C4</td>
<td>Less effort to prepare tests</td>
<td>Effort intensive test preparation</td>
<td>Effort in preparing for prior to conducting the test session.</td>
</tr>
<tr>
<td>C5</td>
<td>Easy to design new tests</td>
<td>Difficult to design new tests</td>
<td>Perceived ease or difficulty of designing new tests.</td>
</tr>
<tr>
<td>C6</td>
<td>Easier/ provides freedom to change test cases</td>
<td>Resilient to change test cases</td>
<td>Perceived ease with which guiding test information can be modified prior to the test session (e.g. modification to charters).</td>
</tr>
<tr>
<td>C7</td>
<td>Less effort to maintain test cases</td>
<td>More effort to maintain test cases</td>
<td>Effort needed to maintain tests used in the test sessions.</td>
</tr>
<tr>
<td>C8</td>
<td>Easier to fill knowledge gap when adding new requirements</td>
<td>Difficult to fill knowledge gap when adding new requirements</td>
<td>Ease or difficulty to fill a knowledge gap about new requirements using testing.</td>
</tr>
<tr>
<td>C9</td>
<td>Easier to verify conformance/ legal requirements</td>
<td>Difficult to verify conformance/ legal requirements</td>
<td>Ability to verify conformance and legal requirements (e.g. fulfillment of standards).</td>
</tr>
<tr>
<td>C10</td>
<td>High relevance of bias</td>
<td>Low relevance of bias</td>
<td>Effect of biases (e.g. previous knowledge about the system and tests) during the test session.</td>
</tr>
<tr>
<td>C11</td>
<td>Efficient in checking verification of requirements</td>
<td>Inefficient in checking verification of requirements</td>
<td>Efficiency of determining to what degree requirements have been verified through the tests (confidence).</td>
</tr>
<tr>
<td>C12</td>
<td>Easier to reproduce defects</td>
<td>Difficult to reproduce defects</td>
<td>Ability to reproduce defects (e.g. in the development organization) to be able to debug and rework.</td>
</tr>
<tr>
<td>C13</td>
<td>Helps more to check performance issues</td>
<td>Does not help in checking performance issues</td>
<td>Ability to check performance related issues.</td>
</tr>
<tr>
<td>C14</td>
<td>Motivates critical thinking to challenge expected outcomes</td>
<td>Bounds the tester to follow the test plan</td>
<td>Degree by which the tester is triggered to think critical.</td>
</tr>
<tr>
<td>C15</td>
<td>Finds more significant/ critical defects</td>
<td>Finds less critical defects</td>
<td>Ability to detect critical defects.</td>
</tr>
<tr>
<td>C16</td>
<td>Helps to uncover unknown defects</td>
<td>Does not help finding unknown defects to a great extent</td>
<td>Ability to find unknown (new) defects</td>
</tr>
<tr>
<td>C17</td>
<td>Motivates the tester</td>
<td>Uninteresting</td>
<td>Degree of motivation.</td>
</tr>
</tbody>
</table>
Chapter 6. A Decision Support Method for Recommending Degrees of Exploration in Exploratory Testing

Prioritization: Once the elements and constructs are elicited, we designed a repertory grid where contrasting constructs were placed in a bipolar matrix. Each construct should be rated based on its effect on a scale of 1 to 3 (negative effect [=1], neutral [=2], and positive effect [=3]) against each element presenting a level of exploration identified through the test charter taxonomy. Furthermore, participants are asked to use cumulative voting (also referred to as the 100 dollar method) [19] to prioritise the most important constructs desired to be achieved in the context of testing the product they represented. That is, they distribute 100 dollars between the constructs giving more dollars to those constructs that are more important.

Calculate recommendation: The recommendation is given based on a combination of the priorities of the constructs (see Table 6.1) and the ratings for each decision option (see Figure 6.1), i.e. the level of exploratory testing. The following steps are followed in the calculation:

1. Multiply the priority of each construct with the corresponding ratings for each decision option.

2. For each decision option, calculate the sum of the multiplication of step 1, resulting in a score for each decision option.

3. For each decision option, calculate the ratio of the scores in relation to the total scores achieved by all options.

The result provides a percentage for each decision option (Figure 6.1), which can be interpreted as the distribution of testing time among the different levels of ET. We suggest to also ask the practitioners what the current distribution is, and have a discussion around a comparison between the current way the testing is conducted and the outcome of the decision support method.

6.4 Application

In this section, we explain the application of the repertory grid technique and the decision support method for exploratory testing. Any conflicting views regarding the priorities of the constructs and ratings of the alternatives were resolved through discussion between the participants and a consensus value was reported in the grid to calculate weighted priorities resulting in the percentage. The calculated percentages represent the time recommended for each level of exploration to achieve the prioritised constructs.
6.4.1 Evaluation method

To evaluate the decision support method, one focus group [20] with Ericsson AB was performed, which was represented by experienced exploratory testers working on different products. This focus group was performed in two instances at two different days. In the first instance of the focus group, a detailed discussion on the personal constructs, elicited earlier, was done followed by the consensus rating for each personal construct. In the second instance of the focus group, the repertory grid technique, the taxonomy for test charter designs, and the elicited personal constructs from the initial focus groups were presented and discussed. The evaluation is summarised as follows.

Goal

Our proposition for this research was that the solution proposed leads to reflections and consensus building among the participants. The research question was:

- RQ: What reflections are taking place when discussing the ET levels using the decision support method?

Participants

In Table 6.2, we present the context of the company and the number of participants in the evaluation. The proposed method was evaluated with the help of participants in the focus group from Ericsson. Each of the participants in this focus group had a strong understanding of ET although the current focus of the test practices at Ericsson is more towards scripted testing as compared to exploratory testing. The minimum experience of individuals in this focus group was 15 years in testing at Ericsson whereas the maximum experience reported was 24 years. For many years, these testers worked with a legacy product which is planned to be replaced with a new system currently under development and testing. The extensive experience of these testers from the legacy system enable them to understand which factors are most important when designing the test strategy for the new system. Therefore, the discussion in both instances of the focus group at Ericsson was around the legacy system and the new system and how the testing differs in between these two products. The factors prioritised by the participants were in line with their desired test focus for the new product.

Data collection

For validation of the decision support method, the following steps were taken during the research:
Table 6.2: Evaluation context

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Company</strong></td>
<td>Ericsson</td>
</tr>
<tr>
<td><strong>Domain</strong></td>
<td>Telecommunications</td>
</tr>
<tr>
<td><strong>No. of participants</strong></td>
<td>7</td>
</tr>
<tr>
<td><strong>Development process</strong></td>
<td>Agile</td>
</tr>
<tr>
<td><strong>Current test focus</strong></td>
<td>Mostly scripted</td>
</tr>
</tbody>
</table>

- Prior to the focus group the participants individually answered a survey where they provide a rating of the alternative ET levels for each construct. The rating was done individually on a scale of 1 to 3 (negative effect [=1], neutral [=2], and positive effect [=3]) for contrasting constructs in the repertory grid. The following steps are taken during the focus group.

- The taxonomy for test charter designs is presented to the participants.

- The participants introduce themselves (including their role, product, and current work activities).

- Discussion on how each exploratory testing level differs from the others and consensus building on the rating given individually in the survey.

- Personal constructs (cf. [7]) were presented and the coverage of the constructs were discussed with the participants.

- The participants were asked to apply the 100 dollar method to provide weighted priorities to the constructs in context of testing their product. A consensus rating was reached based on a discussion.

- The calculated time for each exploration level was presented. A comparison was done between the current distribution of exploration levels and the outcome of the decision support method. Based on this, future directions based on the findings were discussed with the practitioners.

**Threats to validity**

Four types of validity threats are commonly discussed, namely internal validity, external validity, construct validity and conclusion validity.

**Internal validity:** Internal validity threats are related to confounding factors that may have influenced the findings without the researcher’s knowledge. One threat to
validity is that participants misunderstand the constructs and elements. Hence, the constructs as well as elements were explained to them and were exemplified during the focus group to reduce the threat. During the focus group the moderators clarified and discussed them whenever there was a need to support the practitioners in their discussion.

Potential preferences and biases may influence the assessment provided by the practitioners. Even though having experience in ET, the current testing was primarily scripted. This threat was partially mitigated by having different views elicited individually, and thereafter having a dialog as there were different perspectives. However, a group that mainly conducted exploratory testing and very little scripted testing may have given different assessments or provided different reflections. This may not affect the general usefulness of the decision support method, i.e. being a reflective tool to have a discussion around ET levels and the time distribution between them in a specific context.

**External validity:** As mentioned above the company currently practice mostly scripted testing, even though having experience in ET as well. Therefore, the ratings are likely to be influenced by the specific experiences of the focus group participants. In addition, the constructs were elicited only from a few contexts. Given the flexibility of the approach to add or remove constructs, this should not influence the general applicability of the decision support method.

**Construct validity:** The ET levels and prioritised constructs are relevant to mention in the context of construct validity. They were elicited from industry practice and their interpretation may differ between contexts. The differentiation between test levels may only play a relevant role if the findings between the levels differ, as otherwise a distinction may be irrelevant. This was the case as the findings varied for sub-sets of variables (see [7] and Table 6.2).

**Conclusion validity:** A threat to conclusion validity is the bias of the researchers when interpreting or recording the findings. To reduce this threat multiple researchers (three) were present during the focus groups (observer triangulation). In addition, the focus groups have been recorded to confirm notes and thus reduce a bias during data collection.

### 6.4.2 Results

#### Consensus discussion

A total of five out of seven participants in the focus group at Ericsson, provided responses to the survey distributed to them earlier. Prior to the focus group, the repertory grid was filled by the authors and conflicts were identified and color coded.
### Chapter 6. A Decision Support Method for Recommending Degrees of Exploration in Exploratory Testing

<table>
<thead>
<tr>
<th>Practitioners (priority)</th>
<th>Freestyle</th>
<th>Gom</th>
<th>High exploration</th>
<th>Gom</th>
<th>Medium exploration</th>
<th>Gom</th>
<th>Low exploration</th>
<th>Gom</th>
<th>Scripted Manual</th>
<th>Gom</th>
<th>Cons</th>
<th>Contract pole (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Poor learning</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Hard to trace coverage</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>Time inefficient</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>Effort intensive test preparation</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>Difficult to design new tests</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Resilient to change test cases</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>More effort to maintain test cases</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Difficult to fill knowledge gap when adding new requirements</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>Difficult to verify conformance/ legal requirements</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>Low relevance of bias</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>Inefficient in checking verification of requirements</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>Difficult to reproduce defects</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>Does not help in checking performance issues</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Bounds the tester to follow the test plan</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>Finds less critical defects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Does not help finding unknown defects to a great extent</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>Uninteresting</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 6.2: Result of the repertory grid (Ericsson)
Figure 6.2 presents the repertory grid, which formed the basis for the consensus discussion to arrive at a common understanding on the ratings and priorities among the participants.

First, the participants prioritised the constructs. The 17 constructs in Table 6.1 were prioritised, and the priorities are presented in the first column in Figure 6.2. The participants focused on seven constructs, the three most important constructs being the ability to check the verification of requirements (20 points/dollars), the reproducibility of defects (20 points/dollars), and the ability to find new defects (20 points/dollars).

Secondly, the ratings of the elements (ET levels) were discussed based on the survey conducted prior to the focus group. The focus group participants answered the survey before the focus group individually. The following levels of agreement were defined:

- **Full agreement:** All respondents (five) provided the same rating (see light gray cells in the grid in Figure 6.2). For example, all participants agreed that freestyle, high and medium exploration lead to better learning.

- **Good to medium agreement:** The majority of answers (three to four) are the same (white cells).

- **Low agreement:** Two or less answers are the same (dark gray).

During the focus group, detailed discussions were conducted for each construct and its perceived influence on each level of exploration. All options and reasons were explored for both agreements and conflicts. All conflicts were resolved through discussion and a consensus value was added to the repertory grid to calculate recommendations (column “Cons” for Consensus).

For example, the participants disagreed on the suitability of the different ET levels when verifying conformance or legal requirements. The disagreement was a result of the fact that it may depend on the type of legal requirement (relevant types of requirements are related to duration of storage, type of data to be stored, or the type of encryption needed), as well as the background of the testers and their knowledge about the regulations. One respondent highlighted that a checklist is needed “unless you are a lawyer”. In consequence, the focus group participants agreed that freestyle, high and medium degrees are not suited due to the legal knowledge needed, hence clear instructions are needed to check the fulfilment.

Another example of disagreements is the detection of critical defects. The source of disagreements here was the interpretation of what a critical defect is. This relates to the type of defect (e.g. memory leak), but also the scope of the defect (affects entire system or only a part thereof). Significant defects are also often found in boundary
areas. In addition, the ability to find these defects depends on whether the tester may know where the critical defects may be, in particular in system testing. Often testers may also only know whether a defect is critical if they know the expected behaviour (test oracle).

From the discussion it was evident that the practitioners reflected on:

- Their understanding of the constructs (e.g. what is a critical defect) and the alignment thereof.

- the conditions under which a conclusion regarding the rating of elements holds (e.g. given a high level of experience and knowledge of test oracles, freestyle testing is good at finding significant faults).

Having reached a common understanding of the constructs and the conditions, the practitioners reached a consensus for every disagreement and provided a rational for each consensus.

**Recommendation calculation**

Figure 6.3 shows the results of the intermediate calculations combining the priorities of the constructs as well as the ratings. The values are compared using a heat-map. This allows to determine which ratings were particularly influential with regard to the overall results. For example, the recommendation of doing fully scripted testing for 21% of the available time was mostly attributed to the checking of the verification of requirements and defect reproducibility (highlighted with dark gray in Figure 6.3). The more exploratory levels were supported by their ability of finding unknown defects and motivating critical thinking (also highlighted with dark gray in Figure 6.3).

The current distribution is mostly on the lower end of the exploration scale. However, some exploration takes place (20%). Given the priorities of the constructions and the ratings of the elements a higher percentage is recommended (34%), which indicates that the practitioners ought to consider increasing the amount of exploratory testing taking place.

**6.5 Discussion**

The goal of the proposed decision support method was to provide guidance for reflection of what ET level to strive for, and hence designing the test charters guiding the testers accordingly. The goal of the research was to identify what reflections take place when discussing the ET levels by providing a method for decision support based on the repertory grid technique.
### Table 6.3: Recommendation Result (Ericsson)

<table>
<thead>
<tr>
<th>Similarity pole (1)</th>
<th>Free Style ET</th>
<th>High Exploration</th>
<th>Medium Exploration</th>
<th>Low Exploration</th>
<th>Fully Scripted</th>
<th>Contrast pole (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard to trace coverage</td>
<td>0,10</td>
<td>0,10</td>
<td>0,30</td>
<td>0,30</td>
<td>0,30</td>
<td>Easy to trace coverage</td>
</tr>
<tr>
<td>Resilient to change test cases</td>
<td>0,10</td>
<td>0,10</td>
<td>0,30</td>
<td>0,30</td>
<td>0,30</td>
<td>Easier/ provides freedom to change test cases</td>
</tr>
<tr>
<td>Difficult to fill knowledge gap when adding new requirements</td>
<td>0,10</td>
<td>0,20</td>
<td>0,30</td>
<td>0,20</td>
<td>0,20</td>
<td>Easier to fill knowledge gap when adding new requirements</td>
</tr>
<tr>
<td>Ineffecient in checking verification of requirements</td>
<td>0,20</td>
<td>0,20</td>
<td>0,40</td>
<td>0,60</td>
<td>0,60</td>
<td>Efficient in checking verification of requirements</td>
</tr>
<tr>
<td>Difficult to reproduce defects</td>
<td>0,40</td>
<td>0,40</td>
<td>0,40</td>
<td>0,40</td>
<td>0,60</td>
<td>Easier to reproduce defects</td>
</tr>
<tr>
<td>Bounds the tester to follow the test plan</td>
<td>0,30</td>
<td>0,40</td>
<td>0,30</td>
<td>0,20</td>
<td>0,10</td>
<td>Motivates critical thinking to challenge</td>
</tr>
<tr>
<td>Does not help finding unknown defects to a great extent</td>
<td>0,60</td>
<td>0,60</td>
<td>0,60</td>
<td>0,40</td>
<td>0,20</td>
<td>Helps to uncover unknown defects</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rel importance (%)</th>
<th>16</th>
<th>18</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>100,00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>20%</td>
<td>80%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.3: Recommended result (Ericsson)
Chapter 6. A Decision Support Method for Recommending Degrees of Exploration in Exploratory Testing

We conducted a focus group with practitioners at Ericsson AB.

**Common understanding:** Our first observation with regard to the reflections taking place was that a common understanding of the constructs as well as the ratings of the ET levels emerged when using the method. The disagreements while providing the assessments for the ratings of the elements individually (Table 6.2) showed that individual practitioners had different views of the elements. During the discussion the practitioners aligned their understanding and reached a consensus. The discussion revealed some conditions under which the ratings are valid. A key condition was the experience of the tester, as well as the definition of the constructs. An example that the participants discussed was the criticality of defects. Thus, to some degree the ratings are context dependent. This hinders in the construction for a pre-filled grid, and it is acknowledged that research findings depend on the context [21] and [22]. At this stage we also lack the empirical data and knowledge to completely populate the grid, in particular with regard to more fine-grained ET levels than used in the study presented. However, some initial evidence for the usefulness of this type of method has been provided. Afzal et al. [12] found that more significant defects are found using exploratory testing compared to scripted testing. Itkonen et al. (in [5] and [23]) highlighted the importance of experience, which was an important criterion when providing ratings for the ET levels.

**Flexibility:** The practitioners prioritised 17 different constructs to determine what is important to them when making a decision of how to distribute time between the different ET levels. The method is flexible in terms of adding further constructs for prioritisation. For example, a company may want to add specific quality requirements (such as security) if they are particularly important in their context. This may influence the specific recommendation given in a company, although the actual method and the way the practitioners may reflect and discuss should not be affected.

**Recommendation:** In the investigated companies, the recommendation was to increase the use of higher degrees of exploratory testing as the majority was currently spent on scripted testing. The recommendation gains in credibility as the decision support is driven by priorities as well as ratings originating from the practitioners. However, it should not be considered as the decision itself thereby indicating to the practitioners how much exactly to spend on different ET levels. Instead, it was used as an indication that the companies may want to strive for higher degrees of exploration (freestyle and high ET levels) given their priorities and ratings for constructs and elements respectively.
6.6 Conclusion

In this study, we propose a method for decision support to reflect on how to distribute time between five different exploratorion levels in testing [7], namely:

- Freestyle testing
- High degree of exploration
- Medium degree of exploration
- Low degree of exploration
- Scripted testing

The exploratory testing levels are achieved by using test charters, which determine the degree of freedom a tester has during the exploration. For example, in freestyle testing only the test object is provided, while in scripted testing the test steps as well as test data are defined.

The method proposed uses the repertory grid technique for group decision-making as a basis. The technique requires to define constructs (criteria for the decision) and elements (decision alternatives). Thereafter, the decision alternatives are rated with regard to the decision criteria. In this study, we provide the elements (exploratory testing levels) as well as the constructs (17 decision criteria). Among others, the decision criteria are related to learning, the ability to detect new defects, the ability to detect the most significant defects, etc.

The 17 criteria are based on focus groups conducted at Sony Mobile Communications and Axis Communications. Here, a focus group study has been conducted with Ericsson AB. The practitioners used the approach proposed. The key findings are that:

(a) The approach supported the practitioners in arriving at a common understanding of the criteria as well as the ratings for the exploratory testing levels. Given the limited empirical evidence with regard to the exploratory testing levels, we recommend that specific teams use the method for their particular context as a reflective tool.

(b) The companies were mostly using scripted testing (80-90% of all testing done), while the usage of the decision support method indicated that, given the priorities and ratings of the alternatives, they should at least conduct 40% or more using higher levels of exploration.

In future work, we recommend to conduct further studies using the decision support method. An interesting context for study is companies that currently conduct more exploratory testing than in the cases presented here. We also recommend studies investigating the actual effect of different exploratory testing levels on the constructs.
Further, exploring the role of contextual factors in context of exploratory testing, e.g. company size, degree of testing competence, etc., is another direction we recommend to investigate.

Acknowledgment

We would like to thank the participating companies and in particular the individuals for their active involvement in and support of this research.

This work was partly funded by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se).
6.7 References


REFERENCES


Chapter 7

Survey Research in Software Engineering: Problems and Strategies

Ahmad Nauman Ghazi, Kai Petersen, Sri Sai Vijay Raj Reddy and Harini Nekkanti
Submitted to e-Informatica Journal of Software Engineering

Abstract: Background: The need for empirical investigations in software engineering is growing. Many researchers nowadays conduct and validate their solutions using empirical research. Survey is one empirical method which enables researchers to collect data from a large population. Main aim of the survey is to generalize the findings.

Aims: In this study we aim to identify the problems researchers face during survey design, and mitigation strategies.

Method: A literature review as well as semi-structured interviews with nine software engineering researchers were conducted to elicit their views on problems and mitigation strategies. The researchers are all focused on empirical software engineering.

Results: We identified 24 problems and 65 strategies, structured according to the survey research process. The most commonly discussed problem was sampling, in particular the ability to obtain a sufficiently large sample. To improve survey instrument design, evaluation and execution recommendations for question formulation and survey pre-testing were given. The importance of involving multiple researchers in the
analysis of survey results was stressed.

**Conclusions:** The elicited problems and strategies may serve researchers during the design of their studies. However, it was observed that some strategies were conflicting. This shows that it is important to conduct a trade-off analysis between strategies.

## 7.1 Introduction

Surveys are a frequently used method in the software engineering context. Punter et al. [48] highlighted the increased usage of surveys over case-study and experiments. Surveys are one of the empirical investigation method which is used to collect data from a large population [32]. Surveys have been characterized by different authors: Pfleeger highlights that "survey is often an investigation performed in retrospection [46]"; Babbie adds that "surveys aim is to understand the whole population depending on the sample drawn? [2]. Fink [17] states that "surveys are useful for analyzing societal knowledge with individual knowledge". Wohlin et al. highlight that "many quantifiable and processable variables can be collected using a survey, giving a possibility for constructing variety of explanatory models" [60]; Fowler [18] states that "statistical evidences can be obtained in a survey". and Dawson adds that "surveys draw either qualitative or quantitative data from population" [11].

Stavru [55] critically reviewed surveys and found limitations in relation to the definition of the sampling frame, description of the sampling method and the definition of the actual sample. Furthermore, the response rate was rarely identified. Sampling-related aspects were most highly prioritized as issues[55]. Given the limitations in the agile literature there is a need to further explore the use of surveys and understanding how they were conducted in the software engineering context [55]. Stavru [55] also points to the need of frameworks to evaluate survey research as these were not available in the software engineering literature (cf. [55]). Researchers themselves recognize that they are facing problems when conducting surveys, highlighting problems such as limited generalizability, low response rate, survey reliability, etc.[13], [23], [63], [19], [45], [62]. The reason for researchers facing problems could be either he/she is unaware of the problems or they lack strategies to overcome the problems in the survey process. In both the cases the outcome of surveys is unreliable (cf. [47]).

Thus, in this study the main focus is on identifying the problems researchers face and document in the surveys they are executing and the mitigation strategies they report. In particular, the following contributions are made:

- C1: Identify the problems researchers in software engineering face when conducting survey research.
C2: Identify mitigation strategies.

The contributions are achieved through the review of literature combined with an interview-study has been conducted with nine subjects. In the literature review we focused on existing surveys and elicited problems observed as well as mitigation strategies reported in them. A traditional literature review has been used. The interview study was based on convenience sampling and face-to-face interviews. Thematic analysis has been used to analyze the results of the interviews.

The remainder of the article is structured as follows: Section 7.2 presents the background on survey research by explaining the general process of conducting survey research. Section 7.3 presents the related work where problems as well as strategies were elicited from existing guidelines as well as primary survey studies conducted in software engineering. Section 7.4 explains the research design for the interview study conducted. The interview results are thereafter shown in Section 7.5. Section 7.6 discusses the findings from the literature study and the interviews. Section 7.7 concludes the paper.

7.2 Background on the survey research method

Robson and McCartan[49] define the survey methodology as “a fixed design which is first planned and then executed”. Molleri et al. reviewed the steps of survey research guidelines for software engineering. Commonly defined steps are highlighted in Figure 7.1.

7.2.1 Research objectives are defined

The initial step is to identify the research objectives. They help to set the required research scope and context for framing the research questions. While identifying the research objectives it is essential to throw light on certain issues apart from just identifying the research questions. The following reflective questions should be checked when defining the research objectives [32]:

- What is the motivation behind Survey?
- What are the resources required to accomplish the survey’s goals?
- What are the possible areas which are close to the research objectives that were left uninvestigated?
- What is the targeted respondent population of survey?
- How will the data obtained from survey be used? [32] [36] [6]
While defining the research objectives for a survey, the related work pertaining to that particular field must be considered. The knowledge about similar research helps researchers to narrow down the objectives.

Wohlin et al. [60] clearly defines the purpose (objective or motive) for conducting a survey. Based on the objective, any survey falls into one of the below three categories:

- **Descriptive Surveys** are conducted with the intention of explaining traits of a given population. For example, they describe which development practices are used in practice.
- **Explanatory Surveys** investigate cause-effect relationships. For example, they try to explain why a specific software development practice is not adopted in practice.
• Exploratory Surveys helps the researcher’s to look at a particular topic from a different perspective. These surveys are generally done as a pre-study. They help to identify unknown patterns. The knowledge obtained from this pre-study will serve as a foundation to conduct descriptive or explanatory surveys in the future [60].

7.2.2 Target audience and sampling frame are identified

The identification of the target population implies the establishment of a targeted audience. The target audience selection must be driven by the research objectives. The survey instrument design must be designed from the respondent’s perspective, which requires a clear definition of the population and target audience. Similarly, the rule must be applied while selecting the method of surveying (questionnaire or interviews) [32].

The target audience is generally selected from the overall population, if they are attributed with distinct values. The sample is selected from the sampling frame comprising of the possible respondents from the population. Populations can be categorized into sub-populations based on distinguishing attributes, which may be utilized for stratified or quota sampling [56]. Four basic problems of sampling frames are identified in [33] which are: “missing elements, foreign elements, duplicate entries and group based clusters”.

7.2.3 Sample plan is designed

Sampling is the process of selecting a sample for the purpose of studying the characteristics of the population. That is, sampling is needed to characterize a large population [30]. Sampling is mainly divided in two types [37], namely probabilistic and non-probabilistic sampling.

Probabilistic Sampling: Each member of the population has a non-zero probability of being selected. Below are the three types of probabilistic sampling techniques [57]:

• Random Sampling: Members of the sampling frame are selected at random.

• Systematic Sampling: A sampling interval is determined ($k$) and every $kth$ element is chosen from the sampling frame.

• Stratified Sampling: The sampling frame is divided into different groups (e.g. based on experience level of developers in an experiment) and the subjects are chosen randomly from these groups.
Non-probabilistic Sampling: Member selection in this case is done in some non-random order. Below are the types of non-random sampling techniques [18], [32]:

- **Convenience Sampling**: Subjects are selected based on accessibility. Examples are the utilization of existing contact networks or accessing interest groups (e.g. LinkedIn) where subjects are available that are clearly interested in the subject of the survey.
- **Judgment Sampling**: The sample is selected through the guidance of an expert. For example, a company representative for a company-wide survey may choose the subject best suited to answer the survey due to their expertise.
- **Quota Sampling**: Similar to stratified sampling the sample is divided into groups with shared traits and characteristics. However, the selection of the elements is not conducted in a random manner.
- **Snowball Sampling**: Existing subjects of the sampling frame are utilized to recruit further subjects.

### 7.2.4 Survey instrument is designed

Survey outcomes directly depend on how rigorous the survey has been designed. Questions (such as open and closed questions) are designed, and different question types are available (e.g. Likert-scale based questions). The factors which need to be considered while designing surveys have been discussed by Kasunic [32].

### 7.2.5 Survey Instrument is Evaluated

After the Survey Instrument has been designed, it needs to be evaluated to find out if there are any flaws. To determine a questionnaire’s validity a preliminary evaluation is conducted. Examples of different evaluation methods are:

- Expert Reviews [54].
- Focus Groups [54].
- Cognitive Interviews [54][26][39].
- Experiment [43].

### 7.2.6 Survey data is analyzed

The obtained survey data is analyzed in this step. The data analysis depends on the type of questions used in the survey.
Common methods to analyze the results of open-ended questions are phenomenology, discourse analysis, grounded theory, content analysis and thematic analysis [15], [28], [3], [22], [51].

For closed-ended questions, quantitative analysis can be employed. Methods such as statistical analysis, hypothesis testing, and data visualizations can be employed to analyze the closed-ended questions [60].

With regard to the analysis process Kitchenham and Pfleeger [35] suggest the following activities:

1. **Data Validation:** Before evaluating the survey results, researchers must first check the consistency and completeness of responses. Responses to ambiguous questions must be identified and handled.

2. **Partitioning of Responses:** Researchers need to partition their responses into subgroups before data analysis. Partitioning is generally done using the data obtained from the demographic questions.

3. **Data Coding:** When statistical packages cannot handle the character string categories of responses, researchers must convert the nominal and ordinal scale data.

Wohlin et al.[60] describes the first step of quantitative interpretation where data is represented using descriptive statistics visualizing the central tendency, dispersion, etc. The next step is data set reduction where invalid data points are identified and excluded. Hypothesis testing is the third step.

### 7.2.7 Conclusions extracted from survey data

After the outcomes have been analyzed, conclusions need to be extracted from them. A critical review and an evaluation must be done on the obtained outcomes. Thus validity, reliability and risk management should be evaluated when presenting conclusions. Every research has threats, but the main motive is to identify them at the early stages and try to reduce them. Threats may be completely mitigated by research design decisions, while other threats remain open or may only be partially reduced. To handle such threats, it is advised that more than one method must be used to achieve a research objective for reducing the impact of a particular threat [6] [49].

### 7.2.8 Survey documented and reported

The documentation of the survey design is updated iteratively as the research process progresses. Different elements of documentation include RQ’s, objectives, activity
planning, sample method design, data collection, data analysis methods, etc. This documentation is referred to as “questionnaire specification” by [36], while it is named a “survey plan” by Kasuníc [32].

The last step is the reporting of the analysis and conclusion. Even though the survey methodology is administered sequentially, results reporting might vary depending on the targeted readers (e.g. researchers or practitioners). Since the interests of audiences differ, Kasuníc [32] recommend conducting an audience analysis. Stavru [55] evaluated existing surveys in software engineering and identified the most critical elements to be reported in surveys. The most critical elements were:

- The sampling frame and the number of elements in the sampling frame.
- The strategy of sampling from the sampling frame
- The size of the sample
- The target population
- The response rate
- Assessment of the trustworthiness of the survey
- Execution of the survey (research steps)
- Concepts and theories used (e.g. variables studied)
- The design of the survey

7.3 Related Work

7.3.1 Guidelines for survey research in software engineering

Molleri et al. [42] surveyed the literature to identify guidelines for survey research. Three literature sources [32, 38, 34] presented the overall survey process, while several studies focused on individual parts of the process (e.g. only planning and execution). Overall, Molleri et al.[42] found that the different processes comprise of similar steps, while they have different granularities.

The article by Kasuníc [32] described guidelines for conducting a survey. The author describes each step in the survey process and formed the basis to structure the background reported in this paper (Section 7.2).

In addition to overall processes prescribed for survey research several guidelines focused on specific aspects of survey research.

Punter et al.[48] presented guidelines focusing mainly on online-surveys. They have drafted a set of guidelines to perform online survey from their own experiences.
of conducting five on-line surveys. They highlighted that data obtained from online surveys is easy to analyze as it is obtained in the expected format while paper-based forms are error prone. Online surveys track the responses of invited respondents and log the details of those who actually answered the survey, which allows to more easily follow up and increase response rates. Punter et al.[48] argued that online surveys help to gather more responses and ease the disclosure of the results obtained.

Low response rates are a common problem for any survey, which was identified by Smith et al. [52]. Based on their expertise and the existing literature, they performed a post-hoc analysis on previously conducted surveys and came up with factors to improve participation rate. They even specified the limitations of the obtained results stating that “an increase in participation doesn’t mean the results become generalizable” [52].

Pertaining to the survey sampling, Travassos et al. [12] propose a framework consisting of target population, sampling frame, unit of observation, unit of attribute and an instrument for measurement. Ji et al. [29] have conducted surveys in China and addressed the issues relating to sampling, contacts with respondents and data collection, and validation issues. Conradi et al. [8] have highlighted the problem of method biases, expensive contact processes, problems with census type data, and national variations by performing an industrial survey in three countries; Norway, Italy and Germany. This is the first study in software engineering which used census type data. The problem of replications of surveys was highlighted by Rout et al.[4] who replicated a European survey, which was administered in Australian software development organizations.

7.3.2 Problems and strategies

The problems and strategies in literature are structured according to the steps presented in Figure 7.1. We first present the problems (LP**) and the strategies (LS**) mentioned in the literature that were directly linked to the problems by the authors.

Target audience and sampling frame definition and sampling plan

**LP01: Insufficient Sample Size:** Insufficient sample size is the major threat for any software engineering survey. Meaningful statistical evidences cannot be obtained even when the parametric tests are applied on to a particular sample due to insufficient size [41], [44]. One of the main aims of Surveys is to generalize findings to a larger population. Generalizability increases survey’s confidence. Small sample size is attributed as the main cause for the lack of generalizability. If generalizability is not possible then the whole aim of the survey is not achieved [13] [61] [23] [63] [5] [20]. As Kitchenham and Fleeger [35] describe, inadequate sample size negatively impacts the survey outcomes in two ways. Firstly, deficient sample size leads to results that do not show any
Chapter 7. Survey Research in Software Engineering: Problems and Strategies

statistical significance. Secondly poor sampling of clusters reduces the researcher’s ability to compare and contrast various subsets of the population.

Reasons are small sample sizes are busy schedules of the respondents [19][1], poorly designed survey layout, lack of awareness about survey and long surveys [19]. Conradi et al. [29] explained the impact of culture on response rates. They argued that socio-economic positions of the respondents might hinder their willingness to answers. Authors showed that collectivism had direct influence on the information sharing, where people are not interested in sharing information outside their group (i.e. with researchers). Several solutions have been proposed in the literature:

- **LS01: Use personal contact network:** The personal contact network is used to recruit respondents [19], [59], [45], [13], [1].
- **LS02: Cultural awareness:** This issue can be handled by carefully designing questionnaire being aware of the cultures of the respondents [29].
- **LS03: Use probabilistic sampling:** If researchers aim is to generalize to a target population, then probabilistic sampling must be considered [35].
- **LS04: Use of convenience sampling:** Garousi et al.[21] describe the motivation for researchers selecting convenience sampling over other techniques, highlighting that convenience sampling is less expensive and troublesome.
- **LS05: Evaluate the trustworthyness of the sample [55]:** Different ways for calculating the sample size depending on the size of the population have been proposed [32].
- **LS06: Reciprocity:** Researchers can induce reciprocity (respondents answer more than once, e.g. for different projects) by giving rewards. Smith et al [52] were not sure whether this practice was actually useful in software engineering domain as it may introduce a bias in the results.
- **LS07: Consistency:** It is the nature of humans to experience cognitive pressure when they are not performing the promised deeds. This characteristic can induce more responses for a survey [52].
- **LS08: Authority and Credibility:** The compliance for any kind of survey can be increased by the credibility of the person who is administering the survey. Researchers can utilize this benefit by providing the official designations like Professor or PhD in the signature of the survey request mail [52].
- **LS09: Liking:** Respondents tend to answer the surveys from known persons. The responsibility of gaining trust lies with the researchers [52].

148
- **LS10: Scarcity**: It is the human nature to react fast when something is scarce, research can increase the survey’s response rate by convincing about the survey’s uniqueness. [52].

- **LS11: Brevity**: Respondents tend to answer shorter surveys compared to lengthy ones. Researcher should address the number of questions at the start of survey, a progress bar must be placed to help respondents know the survey progress. Usage of close ended questions also helps to attract more respondents [52].

- **LS12: Social Benefit**: Authors describe that more respondents finish the survey if it benefits to a large group instead of a particular community. Researchers must convince the respondents that their survey benefits larger population [52].

- **LS13: Timing**: The time at which an email survey is sent also affects its response rate. A study shows that respondents tend to answer emails right after their lunch [52].

- **LS14: Define clear criteria for sample selection**: Selecting the respondents based on a set of criteria (that are defined at the survey instrumentation stage) can reduce the chances of improper selection [53].

- **LS15: Third party advertising**: Third party advertising can lead to more survey responses, Bacchelli et al[24] obtained a 25% increase in responses rate by following this process. Deuresen at al. [25] have used customized reports along with third party advertising to increase their response rate.

- **LS16: Use snowball sampling**: Respondents of the survey are asked to answer and forward it to their colleagues [21][20].

- **LS17: Recruit respondents from GitHub**: Testers and coders can be recruited for a survey using GitHub [7][25].

- **LS18: Provide rewards**: Researchers can attract the respondents by giving rewards like Amazon points or vouchers gifts. They have to be careful about the responses obtained, since respondents might just answer survey for sake of rewards or answer it twice [7][10].

**LP02: Confidentiality Issues**: In some case software engineering researchers would like to observe on-going trends in the industry or study about specific industrial issues. Though, the software companies do not allow the respondents to take the survey due to the issue of confidentiality. This is problem was faced by one of researchers in their survey “their companies would not allow employees to take this survey due to concerns about confidentiality” [29].

- **LS19: Personalized e-mails**: This threat could be mitigated by sending personal emails rather than system generated emails and by having a follow-up with all
those respondents till the survey ends [29]. Even if this does not handle the issue then it is better to have personal meeting to discuss about the survey.

**LP03: Gate Keeper Reliability:** A gate keeper (person having all the details of employees) from a particular company is contacted by the researcher. The questionnaire is then sent to the gatekeeper, then he/she forwards it to respondents in that company. Sometimes respondents do not receive questionnaire resulting in the a lower participation rate for a survey.

- **LS20: Use IT responsible for reliable distribution of invitations:** This issue was reported by Conradi et al. in their research. Authors mitigated this problem by contacting IT-Responsible for that particular company for getting respondent details [29].

**LP04: No Practical Usefulness:** Any surveys that does not prove to be useful to the respondents, chances are much likely to skip the survey. Authors of [58] clearly show this in the following lines “by far the study is interesting but to whom are the results useful for?”.

- **LS21: Explicitly motivate the practical benefit of the survey:** This issue can be handled by motivating the respondents by giving description about survey outcomes and need for answering survey.

**Survey instrument design, evaluation, and execution**

**LP05: Flaws in the wording of questions:** Sometimes questions are ambiguous, confusing or leading [24, 16]. When survey questionnaire is not clearly understood the respondents arrive at wrong conclusions about questions, as a result they answer incorrectly [58]. Respondents may give two contrary answers for the same question, i.e. being inconsistent within the same survey [62]. This problem can be handled by posing same question in different ways [62].

- **LS22: Survey pre-test:** Researchers [24, 16] pretested the survey with subjects (internally as well as externally with real subjects).

- **LS23: Expert discussions:** Discussions with colleagues and domain experts were also the part of pre-test process. Gorschek et al.[23] have also done redundancy check in addition pre-tests and expert discussion to handle the Survey Instrumentation Problems. Authors Travassos et al.[53] used external researchers that are not involved in the research and reformulated the questionnaire based on their reviews.
- **LS24: Ask the same question in different ways:** Lack of consistency and understanding can be handled by posing same question in different ways [62]

**LP06: Translation Issues:** Translation issue is one of the common problems faced in globally conducted surveys. Avgerio et al. [62] conducted a global survey in Europe and China. The authors posted a questionnaire after translation. As a result of a poor translation data loss occurred. It led to misinterpretation by the respondents leading to false answers.

- **LS25: Collaboration with international researchers:** This problem can be handled when researchers working same domain of the same origin are involved in translation process. Language issue like accent and sentence formulation can be handled in the same manner [24], [29]. Solutions are:

- **LP07: Biases due to Question-Order Effect:** Question-order effect [24] means that the order of the questions is a confounding factors influencing the answers by the subjects.

- **LS26: Order randomization:** This issue can be mitigated by the authors by randomizing the questions of the questionnaire [24].

- **LS27: Natural actions-sequence:** Designed the questionnaire based on a natural actions-sequence helping the respondents in recalling and understanding the questionnaire properly [25].

**LP08: Likert Scale Problems:** A Likert scale is one dimensional in nature, researchers mostly use this in surveys with an assumption that respondent’s opinions can be mapped well to a construct represented by the Likert scale (e.g. team motivation can be surveyed, but is a very complex construct). In a realistic scenario this is not true. Some respondents might get confused on what responses to pick, settling for the middle option. Analyzing the results obtained by higher order Likert scales for analysis posing a threat of misinterpretation or data losses [16].

- **LS28: Avoid two-point scales** Researchers should avoid two point Likert scales ?yes/no?, instead they are advised to use other multi-point scales [4].

**LP09: People Perceptions:** Perception of people answering the survey adversely impacts the survey outcome. In software engineering a survey is done to collect the attitudes, facts, and behaviors of the respondents. This issue cannot be mitigated or controlled completely [58].

**LP10: Lack of Domain Knowledge:** A posted survey could be answered by the respondents without proper domain knowledge. This leads to misinterpretation of the
questionnaire resulting in wrong answers [62], [4][41], [29]. Ji et al.[29] commented that “busy executives likely ignore the questionnaires, sometimes their secretaries finish the survey. In some case the responses obtained are filled with out by the respondents without domain knowledge”. One solution proposed was:

- **LS29**: Explicitly consider background knowledge in the survey: Gorschek et al. [23] stressed the need for considering the impact of background influence of the subjects on survey results while surveying.

**LP11: High drop-out rates**: Sometimes respondents start answering the surveys, but they lose interest after some time as the survey progresses; boredom leads to the low response rate. Lengthy surveys might a reason for the respondents to feel bored [19]. One obvious solution is:

- **LS11**: Brevity: Researcher should limit the number of questions.

**LP12: Time constraints of running the survey**: Time limitations put on surveys as a constraint limit the response rate. Smite et al.[45] showed that time limitation is the main factor for respondents not answering questionnaire or taking phone interviews. It can be clearly seen from these lines “all the 13 respondents were asked to take part, due to time limitation we obtained only 9 responses.” Sometimes researchers neglect the responses obtained from the actual subjects due to time limitation, following lines discuss about this issue “due to rather low response rate and time limits, we have stopped on 33 responses, which covers 13.58% of the Turin ICT sector” [14].

**LP13: Evaluation Apprehension**: People are not always comfortable being evaluated, which affects the outcome of any conducted study [60]. It is the same case with survey studies, sometimes respondents might not be in a position to answers all the questions, instead they shelter themselves by just selecting safer options. This affects the survey outcomes. The following solution has been proposed:

- **LS30**: Guarantee anonymity: Anonymity of subjects reduced this problem of evaluation apprehension [23].

**LP14: Common biases of respondents**: Bias or one-sidedness is a common problem during the survey process. Common types of biases are:

- **Mono-operation Bias**: Sometimes the instrument in survey process might under present the theory involved, this is called mono-operation bias [60]. Solutions are:

  - **LS24**: Ask the same question in different ways: Framing different questions to address the same topic [23], [41]

  - **LS31**: Source triangulation: Collecting data from multiple sources [23], [41]
Over-estimation Bias: Sometimes the respondents of the survey over-estimate themselves, introducing bias into survey results. Mello and Travassos [13] identified that “LinkedIn members tend to overestimate their skills biasing the results”.

Social Desirability Bias: There are situations where respondents tend to appear in the positive light. This might be due the fear of being assessed by the superior authorities. This has a lot of influence on survey outcomes. The following strategy is proposed:

- **LS30: Guarantee anonymity**: Maintaining the anonymity in responses and sharing the overall survey result after reporting [24].

**LP15: Hypothesis Guessing**: This is a construct validity threat where respondents guess the expected survey outcomes, they try to base that anticipation (hypothesis) towards answering questions either in a positive way or a negative way [60].

- **LS32: Stress importance of honesty**: Gorscheck et.al [23] tried to mitigate by stressing the importance of honesty in the introduction of the survey by means of a video and a web page.

**LP16: Respondent Interaction**: This is a conclusion validity threat. During the survey process the respondents might interact and thus influence each other. In small surveys this threat has a large impact on the survey outcome, but in case of surveys done at large scale the impact gradually decreases [23].

**Data analysis and conclusions**

**LP17: Eliminating invalid responses**: In large scale surveys, during analysis this problem poses a lot of work to the researcher as they need to eliminate all the incorrect responses. A strategy is voluntary participation.

- **LS27: Voluntary participation**: This problem can be reduced by making the survey strictly voluntary and only collecting data from the respondents who are willing to contribute [62].

**LP18: Response Duplication**: A major problem is faced in open-web surveys is response duplication, where the same respondent answers the questionnaire more than one time [40][16][25].

**LP19: Inaccuracy in data extraction and analysis**: Inaccuracy in the data extraction and analysis might arise when data extraction from the questionnaire and result reporting are done by an individual person [16].
Chapter 7. Survey Research in Software Engineering: Problems and Strategies

- **LS28**: *Multiple researchers conduct analysis*: Multiple researchers should be utilized when extracting and analyzing the data [16].

- **LS29**: *Check the consistency of coding between researchers*: Two researchers may check their inter-rater reliability through an analysis using the Kappa statistic [16].

**Reporting**

**LP20**: Lack of Motivation for sample selection: Many researchers fail to report their motivation for sample selection [55].

**LP21**: Credibility: For the survey methodology to be accepted as credible and trustworthy, the research method and results need to be clearly presented [55].

### 7.4 Research Method

#### 7.4.1 Research questions

We formulated a corresponding research question for each contribution.

- **RQ1**: Which problems do researchers in software engineering report when conducting surveys?
- **RQ2**: Which strategies do they suggest to overcome the problems?

#### 7.4.2 Selection of subjects

Initially a list of 20 software engineering researchers were chosen to be interviewed. We focused on people conducting empirical software engineering research and included early career researchers as well as senior researchers (PostDocs and professors). Request mails were sent stating the research purpose and the need for their appointment. We received nine positive replies stating their willingness for an interview. The interviews were conducted face-to-face. All the interviews were conducted for a time-span of 50 to 90 minutes. The subjects included four professors, two PostDoc researchers and three PhD students, as shown in Table 7.1. Overall, the table shows that the researchers have substantial experience.
### 7.4.3 Data collection

Generally, interviews are conducted either way individually or with a group of people, focus groups [50]. In this research, we have conducted individual interviews where interviews are done one person at a time. The characteristics of the interview that we have conducted are as follows [31]:

- **Use of open-ended questions:** Through these questions, we aimed for an extended discussion of the topic. In this way, interviewees had the freedom of expressing their opinions based on their experiences.

- **Semi-Structured format:** We focused on getting an in-depth knowledge of the topic thorough interviews. This can be achieved if the interviewer has a set of questions and issues that were to be covered in the interview and also ask additional questions whenever required. Due to this flexibility, we have chosen semi-structured interviews.

- **Recording of responses:** The interviews were audio recorded with interviewees consent. Field notes were maintained by the interviewer, which were helpful in the deeper meaning and better understanding of the results.

The aim of this interview questionnaire is to investigate the problems faced by the researchers while conducting surveys in software engineering. This questionnaire is divided into two sets of questions. The first set of questions mainly focuses on problems that are commonly faced by the researchers like cultural issues, instrument flaws, validity threats and generalizability issues. The interviewee is expected to answer these questions from a researcher’s perspective. The second set of questions mainly focuses on problems that a respondent faces while answering a survey. It also includes the questions asking for suggestions and recommendations regarding the questionnaire design. The interviewee (software engineering researcher) is expected to answer these questions from a respondent’s perspective.

<table>
<thead>
<tr>
<th>ID</th>
<th>Position</th>
<th>Research experience (years)</th>
<th>#Publications (DBLP)</th>
<th>Time taken (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Professor</td>
<td>32</td>
<td>170</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>Professor</td>
<td>16</td>
<td>73</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>Professor</td>
<td>12</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>Professor</td>
<td>15</td>
<td>37</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Post Doctoral Researcher</td>
<td>8</td>
<td>11</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>Post Doctoral Researcher</td>
<td>9</td>
<td>18</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>PhD student</td>
<td>4</td>
<td>4</td>
<td>90</td>
</tr>
<tr>
<td>8</td>
<td>PhD student</td>
<td>5</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>PhD student</td>
<td>5</td>
<td>17</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 7.1: Interviewee’s Details
Finally, the questionnaire ends by asking researchers for their strategies to address the problems raised earlier.

The complete questionnaire is can be found in Appendix ??.

7.4.4 Data analysis

We have chosen thematic analysis process to analyze the results obtained during the interviews. Although there are many other procedures that can be followed to analyze we have a strong reason for opting thematic analysis. The information which needs to be analyzed is the information obtained after conducting several interviews. Since, we were analyzing the results obtained from several interviews, we believed that thematic analysis will assist in analyzing the information very effectively. In the following part of this section, we are going to describe several steps performed during analysis[9].

Extraction of Information: In this stage, we collect all the data from the transcripts prepared from all interviews. As explained above, our transcripts were prepared immediately after the interviews. We have made field notes during each and every interview to make sure that all the interviewees exact view point and their suggestions about our research were penned down during the interview itself. We have collected all these information and documented as a part of this data extraction process. We have gone through all the interview transcripts several times in order familiarize ourselves about the information which we have extracted from our interviews both verbally and non-verbally. We made sure that we have a clear idea of all the information which we had extracted[9].

Coding of Data: As a process of coding our data, we have exclusive codes for all the interviews we conducted. We started with Interview1, Interview2 and so on. This will ensure that our information is segregated according to the interviews which will assist us during the later phases of analysis. We also provided coding few concepts which are similar for all interviews like Interview 1.1 and Interview 2.1 and so on.

Translation of Codes into themes: After all data was provided several codes we have generated. All the codes were translated into several themes according the information. Our main in translating the coded information into themes was to obtain all similar information under one theme. This will also help us in analyzing the information which we collected. Mapping of Themes: Mapping of themes is the process which acted as a check point for the standard of information which we have collected. This assisted to assessing if the amount of information is sufficient for our research and also checks on if we have missed out on any aspect during our process. All the themed information is mapped with the relevant codes during this process. Assess the trustworthiness of our synthesis: This process is to assess that if we had achieved our anticipated results and are the results obtained after the thematic analysis are in sync in
what we actually desired. This also helped us in gaining confidence when we know that our analysis came out well and this analysis is going to contribute us a lot in advanced stages of our research.

7.4.5 Threats to validity

Internal validity: Before designing the questionnaire, the objectives of conducting an interview have been clearly defined. The literature review was conducted prior to the interviews as input to the interview design. Interviews were recorded reducing the risk of misinterpretation or missing important information while taking notes. As the interview was semi-structured the risk of interviewees misunderstanding questions was reduced given the dialog that took place between interviewers and the interviewee.

External validity: A different set of researchers may have different experiences and views of how to conduct surveys. We reduced the threat by conducting an extensive review of the literature overall including more than 70 references. We assured that we included researchers of different experience levels included novice researchers (PhD students who had 3-4 years of experience); experienced researchers (8-10 years of experience) and very experienced researchers (who had 30 years of experience).

Construct validity: While coding interviews data, chances are that we might have wrongly interpreted and coded the results. To mitigate this threat, the data after coding was crosschecked with the actual descriptions from interviews. Furthermore, the coding and structuring into higher level categories were reviewed by multiple authors. This increased the trust in using and interpreting the constructs described in the interviews correctly.

Conclusion validity: Wrong conclusions may be drawn given the data. To reduce this threat multiple researchers were involved in the interpretation of the data. To also increase the reliability in the data we made sure that all the information obtained during interviews is documented immediately: “As soon after the interview as possible, to ensure that reflections remain fresh, researchers should review their field notes and expand on their initial impressions of the interaction with more considered comments and perceptions [27].”
7.5 Interview results

7.5.1 Target audience and sampling frame definition and sampling plan

IP01. Insufficient sampling: All the interviewers have one thing in common, they strongly believe that everyone who claims to use random and stratified sampling have actually done convenience sampling, the reason for this being infeasibility to get a representative sample of the population. The main reason behind this is researchers cannot explicitly define the target population as all relevant variables characterizing the population are high in number and possibly not obtainable. There is no hard and fast rule for determining the desired sample size of a survey. It depends on various factors like the type of research, the researcher, population size, and sampling method. Also the respondents selected using random sample lack motivation as they might not know what for the survey is being done, or they might misinterpret the survey. Similarly, stratified sampling is believed to be challenging, expensive and time consuming, as the theoretical basis for defining a proper “strata” from the given population is missing. Also the timing factor of when the sample is obtained plays a role as the applicability of the findings. Thus, the value of the survey diminishes over time, as survey is just a snapshot of a particular situation at a specific point in time. Multiple strategies for sampling and obtaining the responses have been presented during the interviews.

- IS01: Use random convenience sampling: Random convenience sampling was described as obtaining a sampling frame from personal contacts and randomly sampling form the frame.

- IS02: Use convenience snowball sampling: Due to self-selection process which is followed by them all of them recommended the usage of convenience snowballing. In convenience snowballing the population characteristics are known before-hand, researchers select respondents based on their choice. Questionnaire is then filled and the respondents are asked to forward it to their peers. This way responses of high quality responses are obtained. Convenience snowballing can facilitate an additional number of responses if extended to LinkedIn, most visited blogs and forums. Posting and re posting the survey link in such social networks will make it be on the top and helps to obtain diversified responses.

- IS03: Strive for heterogeneous sample: heterogeneous sample, based on existing literature and your requirements

- IS04: Characterize sample through demographic questions: Demographic ques-
tion helps to easily categorize the obtain data. Proper analysis method and reporting helps researcher to generalize the results involving some constraints.

- **IS05: Brevity:** A questionnaire should be short and precise. It must have a balance between time and number of questions. Interruptions might occur while answering the questionnaire, researchers should expect this while designing a survey. Survey time and questionnaire length must be specified beforehand. Interviews longer than 20 minutes fail to get responses. The interviewee suggested a length of 10-15 or less. They encouraged the inclusion of a feature where respondents can pause and continue the survey, while count-down timers should not be used.

- **IS06: Attend conferences:** Attending the conferences related to the survey domain can also increase response rate.

- **IS07: Guarantee anonymity:** Anonymity must be guaranteed and preserved.

- **IS08: Outcome accessibility:** Motivate the respondents by promising them to present the outcome of your research.

- **IS09: Avoid rewards:** Respondents must not be baited, instead they have to motivated on why they should perform the survey and the benefits they derive from the participation. Thus, it was recommended to not give rewards. If using rewards they should be given at the end of the survey study to assure only receiving committed responses. If rewards were to be given then the handover to each respondent should take place in person, though this might reduce the members of participants due to rewards.

### 7.5.2 Survey instrument design and evaluation

**IP02: Flaws in the wording of questions:** Respondents may misunderstand the context of questions, this is the common problem to every survey and cannot be neglected. Questions must be formulated with great care and must be understandable.

- **IS10: Consider question attributes:** Direct, consistent, non-contradictory, non-overlapping, and non-repeated questions must be asked to obtain vital information. A survey should have both open-ended and close-ended questions. Close ended save time and are easy for analysis, but open-ended give deeper insights about the study. Open ended answers also show the respondents commitment.
IS11: **Survey pre-test:** At first an internal evaluation of the survey with research colleagues should take place followed by piloting with practitioners. Piloting the survey with 5 to 10 people helps to design the survey clearly.

IS13: **Researcher accessibility:** Researcher must be approachable if there are any doubts about the questions that need to be clarified.

**IP03: Likert Scale Problems:** Improper usage of Likert scale confuses the respondents.

IS14: **Informed scale type decision:** Researchers need to investigate potential weaknesses of using different scales. Odd scales provide the respondent with the ability to be neutral by choosing the middle point of the scale, while even scales force the respondent to indicate a preference. The five-point Likert scale was suggested to be used due to its common usage in the information technology domain.

**IP04: Biases due to Question-Order Effect:** This effect should be addressed in a survey.

IS15: **Natural actions-sequence:** Randomizing the questions will not always work in software engineering because logical adherence might be lost. Only if the questions (or groups of questions in a branch) are self-contained then randomization can be done. Though, one should always consider that respondent might lose the context of the questions when randomizing.

**IP05: Evaluation Apprehension:** Respondents expect to be anonymous when answering surveys with questions focusing on their assessment or questions that are personal in nature. Respondents also check for credibility of source while answering these questions.

IS16: **Avoid sensitive questions:** Whenever possible these kind of questions must be generally avoided, if asked they should be placed at the end and be optional. Questions must be framed in such a way that the feeling of being assessed is masked for the respondents.

IS17: **Include “I do not know”-option:** By putting options like “I don’t know” or “I don not want to answer” will encourage respondents to be truthful, and also it helps to rule-out inconsistent responses.

**IP06: Lack of Domain Knowledge:** This problem cannot be eliminated completely, and is significant in the case of open web surveys where survey is being answered by many unknown individuals.
• **IS18: Define clear criteria for sample selection:** The target population should be clearly defined and communicated in the survey.

• **IS19: Stress the importance of honesty:** Explicitly motivate the respondents to be truthful about their experience when answering demographic questions.

**IP07: Hypothesis Guessing:** This is not a problem in case of explanatory surveys.

• **IS19: Stress the importance of honesty:** Respondents should not be influenced instead they should be motivated to be truthful on their part.

• **IS20: Avoid loaded questions:** Hypothesis guessing can be eliminated by not asking loaded questions.

**IP08: Translation issues:** The correct translation is one of the major problems when conducting global surveys.

• **IS21: Collaboration with international researchers:** It is recommended to consult senior researchers who can translate the survey into their mother tongue and are from the same domain.

• **IS22: Avoid Google Translate:** Google translate must not be used for language translations of surveys.

**P09: Cultural Issues:** Cultural issues may appear when conducting surveys globally, in particular the context may not be understood.

• **IS16: Avoid sensitive questions:** In particular in an unknown context it may be unknown how sensitive questions may be perceived, thus they should be avoided.

• **IS11: Survey pre-test:** Surveys should be pre-tested, and it may be recommended to use use face-to-face interviews to gain trust of the respondents and get better insights.

• **IS23: Use appropriate nomenclature:** Appropriate references and terms for things (e.g. concepts) should be used.

**P10: Reliability:** It is important to rule out the people with no hidden agenda or else they result in invalid conclusions.

• **IS4: Determine commitment:** In order to ensure reliability, the researchers must check whether the respondents are really committed towards the survey or not. One way of doing that is to use demographic or redundant questions, or to include open questions (see IS10).
7.5.3 Data analysis and conclusions

IP11: Response Duplication: Response duplication needs to be detected, and will result in wrong conclusions if remaining undetected.

- **IS25: Track IP address:** It can be identified and handled by crosschecking IP addresses. One-time links can be sent directly to the mails, survey tools monitor the duplication.

- **IS26: Session cookies:** Tracking session cookies may help in detecting duplicates as well as information about how many times did the respondent paused and resumed while answering.

IP12: Eliminating invalid responses: The respondents may contradict themselves, which puts the validity of the survey results into question.

- **IS27: Consistency checking:** During the analysis it is recommended to conduct a cross-analysis of questions using Cronbach’s Alpha.

7.5.4 Reporting

IP12: Incomplete reporting: Incomplete reporting will result in the inability to assess and thus trust the outcomes of the survey. Two reporting items were emphasized:

- **IS28: Report inconsistencies:** Inconsistencies and invalid responses should not just be discarded, they have to be reported.

- **IS29: Report biases:** The researcher needs to identify relevant biases and report them correctly in the study.

7.6 Discussion

7.6.1 Comparison with related work

Tables 7.2 and 7.3 present a comparison between the related work and the findings from the interviews. The problems and strategies are grouped by the phases of survey research (see Section 7.2). Whether a problem or strategy has been identified in either the literature or interview is indicated by stating the identifiers (LP** for findings from the literature and IP** for findings from the interviews). Problems and strategies not identified by either literature or interviews are marked as “red”; those identified by both are marked as “green”. The table shows that literature and interviews complement
Table 7.2: Comparison of findings between literature and interviews - Part 1

<table>
<thead>
<tr>
<th>Problems and strategies</th>
<th>Literature</th>
<th>Interview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insufficient Sample Size</td>
<td>L501</td>
<td>L501</td>
</tr>
<tr>
<td>Use personal contact network</td>
<td>L502</td>
<td>L502</td>
</tr>
<tr>
<td>Cultural awareness</td>
<td>L503</td>
<td>L503</td>
</tr>
<tr>
<td>Use probabilistic sampling</td>
<td>L504</td>
<td>L504</td>
</tr>
<tr>
<td>Use random convenience sampling</td>
<td>L505</td>
<td>L505</td>
</tr>
<tr>
<td>Use of convenience sampling</td>
<td>L506</td>
<td>L506</td>
</tr>
<tr>
<td>Use convenience snowball sampling</td>
<td>L507</td>
<td>L507</td>
</tr>
<tr>
<td>Strive for heterogeneous sample</td>
<td>L508</td>
<td>L508</td>
</tr>
<tr>
<td>Evaluate the trustworthiness of the sample</td>
<td>L509</td>
<td>L509</td>
</tr>
<tr>
<td>Reciprocity</td>
<td>L510</td>
<td>L510</td>
</tr>
<tr>
<td>Consistency</td>
<td>L511</td>
<td>L511</td>
</tr>
<tr>
<td>Authority and Credibility</td>
<td>L512</td>
<td>L512</td>
</tr>
<tr>
<td>Liking</td>
<td>L513</td>
<td>L513</td>
</tr>
<tr>
<td>Scarcity</td>
<td>L514</td>
<td>L514</td>
</tr>
<tr>
<td>Brevity</td>
<td>L515</td>
<td>L515</td>
</tr>
<tr>
<td>Social Benefit</td>
<td>L516</td>
<td>L516</td>
</tr>
<tr>
<td>Guarantee anonymity</td>
<td>L517</td>
<td>L517</td>
</tr>
<tr>
<td>Timing</td>
<td>L518</td>
<td>L518</td>
</tr>
<tr>
<td>Define clear criteria for sample selection</td>
<td>L519</td>
<td>L519</td>
</tr>
<tr>
<td>Characterize sample through demographic questions</td>
<td>L520</td>
<td>L520</td>
</tr>
<tr>
<td>Third party advertising</td>
<td>L521</td>
<td>L521</td>
</tr>
<tr>
<td>Use snowball sampling</td>
<td>L522</td>
<td>L522</td>
</tr>
<tr>
<td>Recruit respondents from GitHub</td>
<td>L523</td>
<td>L523</td>
</tr>
<tr>
<td>Attend conferences</td>
<td>L524</td>
<td>L524</td>
</tr>
<tr>
<td>Outcome accessibility</td>
<td>L525</td>
<td>L525</td>
</tr>
<tr>
<td>Provide rewards</td>
<td>L526</td>
<td>L526</td>
</tr>
<tr>
<td>Avoid rewards</td>
<td>L527</td>
<td>L527</td>
</tr>
<tr>
<td>Confidentiality issues</td>
<td>L528</td>
<td>L528</td>
</tr>
<tr>
<td>Personalized e-mails</td>
<td>L529</td>
<td>L529</td>
</tr>
<tr>
<td>Gate Keeper Reliability</td>
<td>L530</td>
<td>L530</td>
</tr>
<tr>
<td>Use IT responsible for reliable distribution of invitations</td>
<td>L531</td>
<td>L531</td>
</tr>
<tr>
<td>No Practical Usefulness</td>
<td>L532</td>
<td>L532</td>
</tr>
<tr>
<td>Explicitly motivate the practical benefit of the survey</td>
<td>L533</td>
<td>L533</td>
</tr>
</tbody>
</table>
Table 7.3: Comparison of findings between literature and interviews - Part 2

<table>
<thead>
<tr>
<th>Phase: Survey instrument design, evaluation, and execution</th>
<th>LP05</th>
<th>IP02</th>
<th>LS22</th>
<th>IS11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flaws in the wording of questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey pre-test</td>
<td>LS23</td>
<td>LS24</td>
<td>IS10</td>
<td>IS12</td>
</tr>
<tr>
<td>Expert discussions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask the same question in different ways</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consider question attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Researcher accessibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translation issues</td>
<td>LP06</td>
<td>IP08</td>
<td>LS25</td>
<td>IS21</td>
</tr>
<tr>
<td>Collaboration with international researchers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid Google Translate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Biases due to Question-Order Effect</td>
<td>LP07</td>
<td>IP04</td>
<td>LS26</td>
<td>IS15</td>
</tr>
<tr>
<td>Order randomization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Natural actions-sequence</td>
<td>LS27</td>
<td>IS19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Likert Scale Problems</td>
<td>LP08</td>
<td>IP03</td>
<td>LS28</td>
<td>IS14</td>
</tr>
<tr>
<td>Avoid two-point scales</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informed scale-type decision</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>People Perceptions</td>
<td>LP09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Domain Knowledge</td>
<td>LP10</td>
<td>IP06</td>
<td>LS29</td>
<td>IS18</td>
</tr>
<tr>
<td>Explicitly consider background knowledge in the survey</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define clear criteria for sample selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stress the importance of honesty</td>
<td></td>
<td>IS19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High drop-out rates</td>
<td>LP11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brevity</td>
<td>LS11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time constraints of running the survey</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation Apprehension</td>
<td>LP12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guarantee anonymity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid sensitive questions</td>
<td>LS30</td>
<td>IS16</td>
<td>IS17</td>
<td></td>
</tr>
<tr>
<td>Include &quot;I do not know&quot;-option</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common biases of respondents (mono-operation, over-estimation, social desirability)</td>
<td>LP13</td>
<td>LP05</td>
<td>LS30</td>
<td>IS16</td>
</tr>
<tr>
<td>Avoid sensitive questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source triangulation</td>
<td>LS34</td>
<td>LS31</td>
<td>LS30</td>
<td></td>
</tr>
<tr>
<td>Guarantee anonymity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypothesis Guessing</td>
<td>LP15</td>
<td>IP07</td>
<td>LS32</td>
<td>IS19</td>
</tr>
<tr>
<td>Stress the importance of honesty</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stress the importance of honesty</td>
<td>LS32</td>
<td>IS19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid loaded questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Respondent Interaction</td>
<td>LP16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cultural issues</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid sensitive questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey pre-test</td>
<td>LS27</td>
<td>IS28</td>
<td>IS29</td>
<td></td>
</tr>
<tr>
<td>Use appropriate nomenclature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>LP17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determine commitment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase: Data analysis and conclusions</td>
<td>LP18</td>
<td>IP11</td>
<td>LS27</td>
<td>IS24</td>
</tr>
<tr>
<td>Eliminating invalid responses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voluntary participation</td>
<td>LS27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Duplication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track IP address</td>
<td>LS27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session contexts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inaccuracy in data extraction and analysis</td>
<td>LP19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple researchers conduct analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check the consistency of coding between researchers</td>
<td>LS28</td>
<td>IS24</td>
<td>LS29</td>
<td>IS25</td>
</tr>
<tr>
<td>Phase: Reporting</td>
<td>LP20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Motivation for sample selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credibility</td>
<td>LP21</td>
<td>IP12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incomplete reporting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report inconsistencies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report biases</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
each other, as each has perspective (literature or interviews) clearly shows gaps. The table may be used as a consolidated view for strategies that researchers may employ to address the problems they face during the survey research process. However, it should be noted that (a) the strategies are not validated and their effect on the quality of surveys (e.g. insufficient sample sizes) is not quantifiable. Additionally, some strategies presented in the results (Section 7.5) and in Tables 7.2 and 7.3 are conflicting, and thus designers of surveys need to make trade-off decisions when planning their research (see Section 7.6.2). Researchers conducting interviews as well as interviewees discussed incomplete reporting and the lack of motivating the sample selection. Complementary to these findings, the contents to be reported in a survey as presented by [55] should be highlighted, which we summarized in Section 7.2.8.

### 7.6.2 Conflicting recommendations and trade-offs

Examples of conflicting strategies and the needs for trade-offs have to be highlighted considering the findings of the study.

To address the problem of small sample sizes it was recommended to have shorter surveys (Brevity, LS11), and as questionnaire attributes the interviewees recommended non-overlapping and non-repeated questions (IS10). However, using open questions helping to determine commitment and gathering qualitative information (IS4) will make the survey longer. In addition, asking questions to check the consistency of answers (IS27, LS24) leads to a longer survey. Hence, a trade-off between the survey length reducing the number of answers and the ability to check the consistency of the survey, and gathering qualitative information needs to be made. Also, the amount of demographic information to characterize the sample (IS04) is limited when aiming for a short survey.

Another decision concerns the type of sampling, namely probabilistic sampling (LS03) and the use of convenience sampling (LS04). As pointed out in the interviews, it is often challenging to sufficiently describe the characteristics of the population. The type of survey (exploratory versus explanatory) also influences the decision, and the degree of the ambition to generalize the survey to a population. Thus, the motivation of the sampling strategy and a clear definition of the sampling frame are essential [55]. During the interviews hybrid strategies were identified, namely using random convenience sampling, where the list of respondents comprises of the contact networks and accessible practitioners to the researchers. From this list a random sample is then selected to partially reduce biases.

Finally, rewards have been discussed as a strategy to increase the number of respondents. In the literature rewards were recommended as a strategy, while the risk of rewards has been pointed out (i.e. answering surveys multiple times for the sake
of rewards). In the interviews it was recommended not to give rewards if mitigation strategies for addressing the risk are not addressed (e.g. receiving the rewards in person).

7.7 Conclusions

In this study we identified problems and related strategies to overcome the problems with the aim of supporting researchers conducting software engineering surveys. The focus was on questionnaire-based research.

We collected data from multiple sources, namely existing guidelines for survey research, primary studies conducting surveys and reporting on the problems and strategies of how to address them, as well as expert researchers. Nine expert researchers were interviewed.

In total we identified 24 problems and 65 strategies. The problems and strategies are grouped based on the phases of the survey research process.

• **Target audience and sampling frame definition and sampling plan:** It was evident that the problem of insufficient sample sizes was the most discussed problem with the highest number of strategies associated with it (26 strategies). Example strategies are brevity (limiting the length of the survey), highlighting the social benefit, using third party advertising, and the use of the personal network to recruit responses. Different sampling strategies have been discussed (e.g. random and convenience sampling). In addition more specific problems leading to losses of responses were highlighted, such as confidentiality issues, gate-keeper reliability, and the lack of explicit motivations of the practical usefulness of the survey results.

• **Survey instrument design, evaluation, and execution:** The main problem observed was poor wording of questions, as well as different issues related to biases (such as question-order effect, evaluation apprehension, and mono-operation, over-estimation, and social desirability biases). The strategies were mainly concerned with recommendations for the attributes of questions and what type of questions to avoid (e.g. loaded and sensitive questions), as well as the need for pre-testing the surveys. It was also highlighted that expert discussions are helpful in improving the survey instrument.

• **Data analysis and conclusions:** For data analysis the main problems were the elimination of invalid and duplicate responses as well as inaccuracy of data extraction and analysis. Technical solutions were suggested for the detection of
detecting duplications. Invalid responses are avoided through consistency checking and voluntary participation. Finally, the importance of involving multiple researchers in the data analysis has been highlighted.

- **Reporting:** Missing information was highlighted as problematic, including the lack of motivation for the selection of samples. It was also highlighted to report inconsistencies and biases that may have occurred in the survey.

A high number of problems as well as strategies has been elicited. In future work a consensus building activity is needed where the community discusses which strategies are most important and suitable for software engineering research. In addition, in combination with existing guidelines the information provided in this paper may serve for the design of checklists to support the planning, conduct, and assessment of surveys.
7.8 References


REFERENCES


Appendix A
Appendix A

Appendix A: Test case template for TCT

Please use this template to design the test cases. Fill the fields accordingly.

- Date:
- Name:
- Subject ID:

<table>
<thead>
<tr>
<th>Test case ID</th>
<th>Function</th>
<th>Priority</th>
<th>Description</th>
<th>Inputs</th>
<th>Expected results</th>
<th>Result</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.1: Test case template.
Appendix B

Appendix B: Defect report

Please report your found defects in this document. Once you are done, please return the document to the instructor.

- Name:
- Subject ID:

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Critical: Prevents using the application, data loss, or serious crash</td>
</tr>
<tr>
<td>2</td>
<td>Normal: Prevents or seriously hinders using a feature</td>
</tr>
<tr>
<td>3</td>
<td>Minor: Hinders using a feature, but the effect is minor or cosmetic annoyance and work around is easy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Regression (Yes/No)</th>
<th>Test case ID</th>
<th>Title</th>
<th>Find time (hh:mm)</th>
<th>Severity (1, 2, 3)</th>
</tr>
</thead>
</table>

Detailed description: Description (how to reproduce, what was expected, what was the actual result)
Appendix C

Appendix C: ET – Test session charter

- Description: In this test session your task is to do functional testing for jEdit application feature set from the view point of a typical user. Your goal is to analyse the system’s suitability to intended use from the viewpoint of a typical test editor user. Take into account the needs of both an occasional user who is not familiar with all the features of the jEdit as well as an advanced user.

- What – Tested areas: Try to cover in your testing all features listed below. Focus into first priority functions, but make sure that you cover also the second priority functions on some level during the fixed length session.
  - First priority functions (refer to Section 3.3.5).
  - Second priority functions (refer to Section 3.3.5).

- Why – Goal: Your goal is to reveal as many defects in the system as possible. The found defects are described briefly and the detailed analysis of the found defects is left out in this test session.

- How – Approach: Focus is on testing the functionality. Try to test exceptional cases, valid as well as invalid inputs, typical error situations, and things that the user could do wrong. Use manual testing and try to form equivalence classes and test boundaries. Try also to test relevant combinations of the features.

- Focus – What problem to look for: Pay attention to the following issues:
Appendix C

- Does the function work as described in the user manual?
- Does the function do things that it should not?
- From the viewpoint of a typical user, does the function work as the user would expect?
- What interactions the function have or might have with other functions?
  Do these interactions work correctly as a user would expect?

• Exploratory log: Write your log in a separate document.
Appendix D

Appendix D: Interview guide for the study on survey research in software engineering

D.0.1 Researcher perspective

1. You have been doing research, publishing articles since long time, by looking at your publications it is visible that you have conducted multiple surveys. Can you explain in which context you think choosing survey as a research method is more beneficial than action research, case studies and experiments?

2. Surveys are used in social sciences and other disciplines including software engineering, do you think there are some special instructions to be followed while conducting surveys. How is it different in software engineering? What factors one shall consider while designing surveys in software engineering research?

3. When designing a survey what type of questions do you prefer asking, (open-ended or close-ended) and why? (Is the evaluation method your primary motivating factor for choosing it? Are evaluation methods one of the reasons for choosing the type of questions? what are the other factors that enable you to include both type of questions in your Survey)

4. In a survey one question may have provided context to the next one which may drive respondents to specific answers, randomization of questions to some extent
Appendix D

may reduce this question-order effect. Can you suggest some other techniques to deal with this question order effect?

5. How do you make sure that respondents understand the right context of your question, what measures do you adapt for making the questionnaire understandable?

6. Our literature analysis showed that 31.4% of primary studies used Stratified sampling technique, while only 15.7% of studies reported the usage of Snowball Sampling by researchers. (Literature describes that Snowball sampling leads to a better sample selection, where researcher has the freedom of choosing sample that suits to his/her requirements). Have you faced any situation where other sampling techniques were chosen over snowballing, what factors did you consider while making the selection?

7. Low response rates are common problem for any survey, how can the response rate be improved?

8. When a survey is posted, there are few respondents without a proper domain knowledge answering it. They might misinterpret data giving incorrect answers and this affects the overall analysis. In yours research how are such responses identified and ruled-out?

9. Our analysis showed that hypothesis guessing is an issue that can only be reduced to some extent rather than avoiding it completely. Explain how this problem is addressed in your work.

10. What measures do you take to avoid the duplication of responses in your surveys?

11. How do you overcome each of these common problems like bias, generalizability and reliability in the following cases? (a) Case A: Respondents answering the survey just for the sake of rewards. (b) Case B: Respondents answering surveys posted on social networks like LinkedIn and Facebook.

12. How do you mitigate the issue of inconsistency in responses? (Case-When a respondent is asked about his familiarity with non-functional requirements he chooses a Yes option. When asked to elaborate his opinion he just writes “No idea”, here comes the problem of inconsistency)

13. Assume you have conducted a global survey in Sweden, Norway, China and Italy collecting information from diverse respondents. How do you address the following issues in your research? (a) Issue: Questionnaire gets translated into
Chinese making it understandable to respondents over there, due to poor translation there might be an issue of data losses. How do you handle this language issue? (b) Issue: There might be some cultural issues where people of one country are more comfortable in answering an online questionnaire, while people of another country are more responsive to face-to-face interviews. In your opinion how can this kind cultural issue be mitigated?

14. In the review of the literature we found out that researchers used Likert scale as a question type. Even though its a commonly used we could obtain few problems like central tendency bias when a 4-point or 7-point Likert scale is used, respondent fatigue and interpretation problems have been identified when 9 or 10 point scales have been used. How do you address these kind of issues in your research?

15. How do you decide upon a particular sample size for your survey?

16. What motivates you to select a specific analysis technique for your research?

**D.0.2 Respondent’s perspective**

1. You must have answered many surveys till now, in your opinion how can a survey grab the attention of the respondent?

2. Does the questionnaire length affect your way of answering a survey?

3. What do you prefer mostly to answer, open-ended or close-ended questions?

4. Does time limitation affect your way of answering the Survey?

5. Would you be willing to disclose details about your research study when answering a survey? (Confidentiality issues, Intellectual theft)

**D.0.3 Concluding questions about survey guidelines**

- Do you think there is a need for the checklist of all the problems faced by the Software Engineering Researchers while conducting surveys?

- On the scale of 1 to 5, please rate the need for having such kind of checklist.
## List of Figures

1.1 Overview of the thesis .................................................. 10
1.2 Advantages and disadvantages of exploration levels ................. 16

2.1 Importance of test objectives ......................................... 38
2.2 Usage of Techniques in Heterogeneous Systems ....................... 39
2.3 Practitioners’ Perceptions of Testing Techniques for Heterogeneous Systems (1 = Strongly Disagree, 2 = Disagree, 3 = Uncertain, 4 = Agree, 5 = Strongly Agree) .......................... 40

3.1 Results of the multiple comparisons test for modes of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different modes of defects, i.e., in Fig 3.1(a) above, the vertical dotted lines indicate Mode 1 and 3 have mean ranks significantly different from Mode 0.) ............... 67
3.2 Results of the multiple comparisons test for different modes of defects detected by students using TCT. (The vertical dotted lines indicate differences in mean ranks of different modes of defects, i.e., in Fig 3.2(a) above, the vertical dotted lines indicate that mode 0 defects have mean rank significantly different from none other.) ..................... 69
3.3 Results of the multiple comparisons test for types of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.3(a) above, the vertical dotted lines indicate that documentation has mean rank significantly different from missing function, performance, technical defect and wrong function.) ............................ 71
3.4 Results of the multiple comparisons test for different types of defects detected by students using TCT. (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.4(a) above, the vertical dotted lines indicate that defects of type documentation have mean rank significantly different from defects of type wrong function.) .............................................................. 72

3.5 Results of the multiple comparisons test for different severity levels of defects detected by students and practitioners using ET (The vertical dotted lines indicate differences in mean ranks of different severity levels of defects, i.e., in Fig 3.5(a) above, the vertical dotted lines indicate that negligible defects have mean rank significantly different from normal and severe defects.) .............................................................. 74

3.6 Results of the multiple comparisons test for severity levels of defects detected by students and practitioners using TCT. (The vertical dotted lines indicate differences in mean ranks of different types of defects, i.e., in Fig 3.6(a) above, the vertical dotted lines indicate that defects of negligible severity level have mean rank significantly different from no other severity level.) .............................................................. 76

5.1 Classification of levels of exploratory testing; ET levels .................... 108
5.2 Example of test charters for the high, medium and low ET levels ...... 109
5.3 Overview of factors influencing the ET levels derived from the focus groups and the survey .............................................................. 112

6.1 The degrees of exploration [7] used as elements in the repertory grid for ET ......................................................... 124
6.2 Result of the repertory grid (Ericsson) ........................................ 130
6.3 Recommended result (Ericsson) ............................................ 133

7.1 Eight Steps of a Survey ...................................................... 142
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Mapping of methods, objectives, research gaps and research questions</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>Surveyed Variables</td>
<td>34</td>
</tr>
<tr>
<td>2.2</td>
<td>Roles of Subjects</td>
<td>36</td>
</tr>
<tr>
<td>2.3</td>
<td>Company Size (Number of Employees)</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>System Types</td>
<td>37</td>
</tr>
<tr>
<td>2.5</td>
<td>Development Models</td>
<td>37</td>
</tr>
<tr>
<td>2.6</td>
<td>Friedman test statistics</td>
<td>41</td>
</tr>
<tr>
<td>3.1</td>
<td>The division of subjects in experimental iterations and groups.</td>
<td>57</td>
</tr>
<tr>
<td>3.2</td>
<td>Average experience of subjects in software development and software testing in number of years.</td>
<td>58</td>
</tr>
<tr>
<td>3.3</td>
<td>Defect count data summary.</td>
<td>62</td>
</tr>
<tr>
<td>3.4</td>
<td>Description of the types of defects with examples.</td>
<td>65</td>
</tr>
<tr>
<td>3.5</td>
<td>Distribution of defects concerning detection difficulty.</td>
<td>66</td>
</tr>
<tr>
<td>3.6</td>
<td>Percentages of the modes of defects detected by students and practitioners applying ET.</td>
<td>68</td>
</tr>
<tr>
<td>3.7</td>
<td>Percentages of the different modes of defects detected by students and practitioners applying TCT.</td>
<td>68</td>
</tr>
<tr>
<td>3.8</td>
<td>Distribution of defects regarding technical type.</td>
<td>70</td>
</tr>
<tr>
<td>3.9</td>
<td>Percentages of the type of defects detected by students and practitioners applying ET.</td>
<td>70</td>
</tr>
<tr>
<td>3.10</td>
<td>Percentages of the different types of defects detected by students and practitioners applying TCT.</td>
<td>73</td>
</tr>
<tr>
<td>3.11</td>
<td>Severity distribution of defects.</td>
<td>73</td>
</tr>
<tr>
<td>3.12</td>
<td>Percentages of the severity level of defects detected by students and practitioners applying ET.</td>
<td>75</td>
</tr>
</tbody>
</table>
3.13 Percentages of the severity levels of defects detected by students and practitioners applying TCT.

4.1 Profile of the Interviewees
4.2 Factors influencing test charter design
4.3 Contents of test charters

6.1 Influence factors for degrees of exploration [7] used as constructs in the repertory grid for ET
6.2 Evaluation context

7.1 Interviewee’s Details
7.2 Comparison of findings between literature and interviews - Part 1
7.3 Comparison of findings between literature and interviews - Part 2

A.1 Test case template.
B.1 Defect types.
B.2 Defects in the experimental object.
“Whenever I set myself the task to learn, I realize how little I know and the more I learn, the more I realize how ignorant I am.”

–Imam Muhammad ibn Idris al-Shafi‘i (676 - 820 CE)
ABSTRACT

**Context:** Exploratory testing (ET) is an approach to test software with a strong focus on personal skills and freedom of the tester. ET emphasises the simultaneous design and execution of tests with minimal test documentation. Test practitioners often claim that their choice to use ET as an important alternative to scripted testing is based on several benefits ET exhibits over the scripted testing. However, these claims lack empirical evidence as there is little research done in this area. Moreover, ET is usually considered an ad-hoc way of doing testing as everyone does it differently. There have been some attempts in past to provide structure to ET. Session based test management (SBTM) is an approach that attempts to provide some structure to ET and gives some basic guidelines to structuring the test sessions. However, these guidelines are still very abstract and are very open to individuals’ interpretation.

**Objective:** The main objective of this doctoral thesis is to support practitioners in their decisions about choosing exploratory versus scripted testing. Furthermore, it is also aimed to investigate the empirical evidence in support of ET and find ways to structure ET and classify different levels of exploration that drive the choices made by exploratory testers. Another objective of this thesis is to provide a decision support system to select levels of exploration in overall test process.

**Method:** The findings presented in this thesis are obtained through a controlled experiment with participants from industry and academia, exploratory surveys, literature review, interviews and focus groups conducted at different companies including Ericsson AB, Sony Mobile Communications, Axis Communications AB and Softhouse Consulting Baltic AB.

**Results:** Using the exploratory survey, we found three test techniques to be most relevant in the context of testing large-scale software systems. The most frequently used technique mentioned by the practitioners is ET which is not a much researched topic. We also found many interesting claims about ET in grey literature produced by practitioners in the form of informal presentations and blogs but these claims lacked any empirical evidence. Therefore, a controlled experiment was conducted with students and industry practitioners to compare ET with scripted testing. The experiment results show that ET detects significantly more critical defects compared to scripted testing and is more time efficient. However, ET has its own limitations and there is not a single way to use it for testing. In order to provide structure to ET, we conducted a study where we proposed checklists to support test charter design in ET. Furthermore, two more industrial focus group studies at four companies were conducted that resulted in a taxonomy of exploration levels in ET and a decision support method for selecting exploration levels in ET. Lastly, we investigated different problems that researchers face when conducting surveys in software engineering and have presented mitigation strategies for these problems.

**Conclusion:** The taxonomy for levels of exploration in ET, proposed in this thesis, provided test practitioners at the companies a better understanding of the underlying concepts of ET and a way to structure their test charters. A number of influence factors elicited as part of this thesis also help them prioritise which level of exploration suits testing more in their product’s context. Furthermore, the decision support method provided the practitioners to reconsider their current test focus to test their products in a more effective way.