Bibliographic information:

Dietmar Pfahl, Huishi Yin, Mika V. Mäntylä, Jürgen Münch. How is Exploratory Testing Used? A State-of-the-Practice Survey. In Proceedings of the 8th ACM-IEEE International Symposium on Software Engineering and Measurement (ESEM 2014), Torino, Italy, September 2014.

How is Exploratory Testing Used? A State-of-the-Practice Survey

Dietmar Pfahl, Huishi Yin
Institute of Computer Science, University of Tartu
J. Liivi 2, Tartu 50409, Estonia
{dietmar.pfahl, huishi@ut.ee}

Mika V. Mäntylä
Department of Computer Science and Engineering, Aalto University
P.O. Box 19210, FI-00076 Aalto, Finland
mika.mantyla@aalto.fi

Jürgen Münch
Department of Computer Science, University of Helsinki
P.O. Box 68, FI-00014 Helsinki, Finland
Juergen.Muench@cs.helsinki.fi

ABSTRACT

Context: Exploratory Testing has experienced a rise in popularity in the industry with the emergence of agile development practices, yet it remains unclear, in which domains and how it is used in practice.

Goal: To study how software engineers understand and apply the principles of exploratory testing, as well as the specific advantages and difficulties they experience.

Method: We conducted an online survey in the period June to August 2013 among Estonian and Finnish software developers and testers.

Results: Our main findings are that the majority of testers, developers, and test managers using ET, (1) apply ET to usability-critical, performance-critical, security-critical and safety-critical software to a high degree; (2) use ET very flexibly in all types of test levels, activities, and phases; (3) perceive ET as an approach that supports creativity during testing and that is effective and efficient; and (4) find that ET is not easy to use and has little tool support.

Conclusions: The high degree of application of ET in critical domains is particularly interesting and indicates a need for future research to obtain a better understanding of the effects of ET in these domains. In addition, our findings suggest that more support to ET users should be given (guidance and tools).

Categories and Subject Descriptors

D.2.5 [**Software Engineering**]: Testing and Debugging.D.2.9 [**Software Engineering**]: Management – *Software quality assurance*.K.6.3.3 [**Management of Computing and Information Systems**]: Software Management – *Software development, Software process*.

General Terms

Management, Measurement, Experimentation, Verification.

Keywords

Exploratory Testing, Software, Survey.

1. INTRODUCTION

Testing is an important activity in the software development life cycle. Testing helps assess and improve the quality of software, and testers use many different ways to find more defects with the least possible effort. Exploratory Testing (ET) is a manual testing approach – or attitude as some proponents of ET would say – that was first presented by Cem Kaner in 1983 [10]. He defines ET as "A style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of his/her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project." [11]. James Bach gives a shorter definition of ET [2]: "Exploratory testing is simultaneous learning, test design, and test execution". Both definitions give an idea of what ET is but leave room for interpretation. Therefore, over the years many different descriptions and understandings of ET have merged. Some label ET is 'ad hoc' testing while others describe ET as a method of error guessing [6]. In addition, new approaches related to ET such as, for example, Session-Based Test Management (SBTM) have emerged to make ET more management compatible and provide guidance to testers using ET. James Bach describes SBTM as "A method for measuring and managing exploratory testing" [1].

Today we know little about how software developers actually interpret and apply the principles and strategies that constitute the essence of ET. However, it seems obvious that there are differences in the way ET is understood and applied in software industry. We conducted a survey to shed light on the issues involved so that we can better understand how ET is actually used in industrial practice, what its advantages and disadvantages are, what tool support is used, and how it can be improved. We were mainly interested in the views on ET in Estonia and Finland, not only because we have been working with many companies in these two countries but also because Finland and Estonia seem to be culturally susceptible to agile development practices, such as ET, due to their relatively low social power distance and high individualism[1] [12]. While both Estonia and Finland show a relatively low societal power distance (scores of 40 and 33, respectively) distance and a high degree of individualism (scores of 60 and 63, respectively), Estonians are on average more pragmatic than Finnish (scores of 82 and 38, respectively). Given the considerable difference in pragmatism, we were curious whether experience with and opinion about ET differs between the two countries. In the rest of this paper, all results reported relate to data collected in Estonia and Finland, without explicit mentioning of the two countries.

ESEM'14, September 18-19, 2014, Torino, Italy.

[1] Cf. Geert Hofstede's research on national and organizational culture: http://geert-hofstede.com/

Our paper is structured as follows. We present related work in Section 2. Then, in Section 3, we describe the research method and goals. Survey design and distribution are presented in Section 4. We present results in Section 5, followed by the discussion of results in Section 6. Limitations and threats to validity are discussed in Section 7. Conclusions and plans for future work are presented in Section 8.

2. RELATED WORK

ET was invented in the software industry and most of the original material produced describing the method first appeared in blogs or slide sets. Although ET has been covered in textbooks, e.g. [10] the first book exclusively focusing on ET appeared only in 2009 [20]. Yet, ET has been recognized in SWEBOK [3] which defines ET as simultaneous learning, test design, and test execution; that is, the tests are not defined in advance in an established test plan, but are dynamically designed, executed, and modified.

If credible industrial sources have been scarce, the same is true for academic studies. Experiments show that specified and documented test cases do not increase effectiveness of testing and decrease efficiency due to effort needed for test case creation [6][7]. This seems to heavily speak in favor of ET, which doesn't use specified and documented test cases. High efficiency of ET is supported by one of the earliest case studies of ET that showed how ET enabled highly efficient testing in a high time pressure situation [19]. Another case study also found support for high efficiency of ET in defect detection, but highlighted difficulties in test coverage management [9]. Proposals for ET use and improvement have also been made in empirical studies. Do Nascimento et al. [13] proposed that ET is used to acquire knowledge for model based testing. More recently, Itkonen et al. [8] showed how the application of the testers' knowledge during ET sessions can explain the high efficiency of ET. Tuomikoski and Tervonen [16] presented a case study of testing sessions to manage ET. The team ET sessions allowed people with different expertise to collaboratively test and to learn from each other.

In a recent article, Shah et al. conducted a comprehensive literature survey and interviews to investigate into strengths and weaknesses of ET and scripted testing [18]. Regarding strengths, the authors found ET to be cost-effective due to little and focused documentation, good resource utilization, rapid feedback, and quick familiarization with the product. As for testing quality, the authors found evidence for good overall defect detection effectiveness, and high performance for detecting critical defects. In addition, they found evidence that the flexibility of the ET process and it high degree of freedom as to how it is conducted helps testers utilize their skills better and makes them more responsible, engaged, motivated, and creative, while they are performing tests. Regarding weaknesses, the authors found that the unstructured and ad-hoc nature of the ET process causes difficulties in managing the testing process, in prioritizing and selecting the appropriate tests, and in repeating the tests. As for testing quality, the dependency on the skills, experience, and domain knowledge of the testers were among the major weaknesses identified, especially when the application to be tested is too complex. In addition, they found ET to be not suitable for acceptance, performance, and release testing, which in turn lowers the accountability and hence customer satisfaction. After comparing strength and weaknesses of ET with those of scripted testing, the authors suggest to combine both approaches into a hybrid approach in order to benefit from the complementary strengths of both thus improving the overall testing process.

3. RESEARCH METHOD AND GOALS

We conducted an online survey in June-August 2013 to investigate how ET is currently applied in Estonian and Finnish software companies and what software engineers think about ET. Thus, the objective of our research was to investigate the characteristics of those software companies that apply ET in Estonia and Finland and what experience these companies have with using ET. Also, we aimed at understanding which factors favor using ET in a company, for example, whether specific roles use ET more often than others or whether the maturity of an engineer or the age and size of the engineer's organizational unit influence whether ET is used more or less frequently. In addition, we tried to find how software engineers think about ET, for example, what characteristics they associate with ET, what advantages and disadvantages they observed when using ET, and what suggestions they have for improving the way how they are using ET. We formulated the following research questions:

RQ 1: What experience do respondents have with using ET?

- RQ 1.1 How frequently do software (sw) engineers use ET?
- RQ 1.2 When do sw engineers typically use ET?
- RQ 1.3 Do sw engineers use tools to support ET?
- RQ 1.4 In what testing context do sw engineers use ET?
- RQ 1.5 For what type of software do sw engineers use ET?
- RQ 1.6 Is the experience with using ET different between Estonia and Finland?

RQ 2: Which factors have an influence on using ET?

- RQ 2.1 Does the location (Estonia versus Finland) of the organizational unit have an effect on applying ET?
- RQ 2.2 Does the size of the organizational unit have an effect on applying ET?
- RQ 2.3 Does the age of the organizational unit have an effect on applying ET?
- RQ 2.4 Does the role of a sw engineer have an effect on applying ET?
- RQ 2.5 Does the time a sw engineer is working in his current role have an effect on applying ET?
- RQ 2.6 Does the type of test organization have an effect on applying ET?
- RQ 2.7 Do the characteristics of the tested software have an effect on using ET?

RQ 3: How do software engineers think about ET?

- RQ 3.1 What elements consider sw engineers as essential for defining ET?
- RQ 3.2 What characteristics do sw engineers think ET has?
- RQ 3.3 What do sw engineers think are advantages and disadvantages of ET?
- RQ 3.4 Do sw engineers want to improve ET, and how?

From the research questions, we formulated questionnaire items. The list of items and their mapping to the research questions is shown in Appendix A.

4. SURVEY DESIGN AND DISTRIBUTION

The design of the survey evolved in several iterations and involved reviews by external experts from industry. We advertised the survey through mailing lists, bulletin boards, blogs,

social media, and word of mouth, explicitly stating our focus on software industry in Estonia and Finland.

We published the survey on 10 June 2013 using the Diaochapai (http://www.diaochapai.com/) online survey system and left it open for access until 31 August 2013. We posted the survey link via several channels such as mailing lists, social networks (Linkedin, Twitter, Facebook, and Google) as well as via a professional blog related to testing (the Estonian Software Testing Club). In total, we received 61 complete responses. The distribution of responses with respect to visit resources is shown in Table 1. We see that mailing lists were the best way to promote our survey (47.54%) followed by Linkedin (22.95%).

Table 1.Visits per survey distribution channel

Visit Resource	Percentage
mailing lists	47.54 %
www.linkedin.com	22.95 %
www.softwaretestingclub.com	9.84 %
t.co	9.84 %
www.facebook.com	6.56 %
www.google.fi	1.64 %
www.google.com	1.64 %

5. SURVEY RESULTS

In the following sub-sections we first describe the demographics data of the responses received and then summarize the main findings for each research question.

5.1 Demographics

The tables presented below summarize demographic information of survey respondents:

- Geographical location, size and age of respondent's organizational unit (Tables 2 to 4)
- Current working role of respondent (Table 5)
- Time respondent spent working in the current role (Table 6)
- Type of test organization in respondent's organizational unit (Table 7)
- Typical characteristics of the tested software in respondent's organizational unit (Table 8)

Table 2.Geographical locations of survey respondents' organizational units

Geographical Location	Count	Percentage
Estonia	27	44 %
Finland	23	38 %
Other	11	18 %
Total	61	100%

Of the 61 responses received, 27 respondents located their organizational unit in Estonia, 23 in Finland, and 11 in other countries. In the following, we exclusively use the responses from Estonia and Finland as those two countries were in the focus of our research.

The sizes of the organizational units of respondents are shown in Table 3. The distribution over size categories is relatively uniform with a maximum for organizations with 20 to 49 employees. Large organizational units with more than 100 employees have the lowest share of all responses.

The ages of the organizational units of respondents are shown in Table 4. Only 8% of the respondents' organizational units were younger than 2 years. The biggest share (54%) had respondents' organizational units with an age or more than 5 years.

Table 3. Sizes of survey respondents' organizational units

Number of Employees	Count	Percentage
Less than 20	15	30 %
20 to 49	16	32 %
50 to 99	10	20 %
More than 100	9	18 %
Total	50	100%

Table 4. Ages of survey respondents' organizational units

Number of Years	Count	Percentage
Less than 2	4	8 %
2 to 5	19	38 %
More than 5	27	54 %
Total	50	100%

The roles that respondents currently assume in their respective organizational units are shown in Table 5. The vast majority of respondents are either testers (54%) or test managers (40%).

Table 5. Respondents' current roles

Current Role	Count	Percentage
Tester	26	52 %
Test Manager	20	40 %
Other role cooperating with Tester/Test Manager	4	8 %
Total	50	100%

The times that respondents have been working in their current roles are shown in Table 6. Almost half of the respondents (46%) have been working in their current role for more than five years.

Table 6. Respondents' times having worked in current roles

Time worked in Current Role	Count	Percentage
Less than 2 years	10	20 %
2 to 5 years	17	34 %
More than 5 years	23	46 %
Total	50	100%

Whether the respondents' organizational units have a separate test (or quality assurance - QA) organization is shown in Table 7. 48% of the respondents say their organizational unit has a separate test (or QA) organization and they are part of it. 10% of the respondents say that a separate test (or QA) organization exists in their organizational unit but they are not part of it. The remaining 42% of respondents say that their organizational unit doesn't have a separate test (or QA) organization.

Table 7. Type of test (or QA) organization in respondents' organizational units

Type of Test (or QA) organization	Count	Percentage
Separate test (or QA) organization – respondent is member	24	48 %
Separate test (or QA) organization – respondent is not member	5	10 %
No separate test (or QA) organization	21	42 %
Total	50	100%

The characteristics of the software dealt with in the respondents' organizational units are shown in Table 8. The characteristics are sorted according to frequency. When answering the questions in the related questionnaire item, respondents could check more than one characteristic. Therefore, the sum of responses is larger than the number of respondents. The most frequently mentioned software characteristic is 'usability-critical' (82%), followed by 'performance-critical' (72%) and 'high security demand' (64%).

Table 8. Software characteristics respondents are working with

Software Characteristics	Count	Percentage
It is usability-critical (e.g., it has a complex GUI which is important for the end user)	41	82 %
It is performance-critical	36	72 %
It has high security demand	32	64 %
It is safety-critical	24	48 %
None of above	2	4 %

5.2 Main Findings

In the following sub-sections we report the main results related to research questions RQ1 to RQ3. A complete report of all results can be found in [21].

5.2.1 RQ 1: What experience do respondents have with using ET?

In the following, we report the results related to RQ 1 for each sub-question separately.

RQ 1.1 How frequently do software engineers use ET? Table 9 shows the distribution between ET users and non-users received from respondents located in Estonia and Finland. For both countries, the vast majority of respondents (88%) claimed to be ET users. The highly unbalanced distribution between ET-users and non-users can be interpreted in several ways. If we assume our set of respondents to be representative for the set of software engineers and managers, then the numbers may indicate that ET is indeed a very popular approach utilized by many at least in some occasion. Alternatively, it is possible that the topic of the survey, clearly indicating that we were interested in finding out something about ET, induced a strong self-selection bias towards ET users.

Table 9. Frequency of using ET

ET Usage	EST	FIN	Total
Yes	25 (92.6%)	19 (82.6%)	44 (88%)
No	2 (7.4%)	4 (17.4%)	6 (12%)
Total	27	23	50

RQ 1.2 When do software engineers typically use ET? Table 10 shows data on the timing for the use of ET. In particular, we asked whether ET is used in early test activities, in late test activities, or at any time during testing. It turned out that the majority of respondents using ET stated they have no timing preference (72.7%). The indifference against any timing preference was more prominent among respondents from Finland (89.5%) than among respondents from Estonia (60%).

RQ 1.3 Do software engineers use tools to support ET? Table 11 shows the degree of tool support when using ET. It turned out that the majority of ET users (75%) didn't report any tool support. The lack of tool support was more prominent among respondents from Estonia (80%) than among respondents from Finland (68.4%).

Table 10. Timing for the use of ET

ET Timing	EST	FIN	Total
ET during early test activities	6 (24%)	1 (5.3%)	7 (15.9%)
ET during late test activities	4 (16%)	1 (5.3%)	5 (11.4%)
ET at any time during testing	15 (60%)	17 (89.5%)	32 (72.7%)
Total	25	19	44

Table 11.Tool support when using ET

ET Tool Support	EST	FIN	Total
Yes	5 (20%)	6 (31.6%)	11 (25%)
No	20 (80%)	13 (68.4%)	33 (75%)
Total	25	19	44

Table 12. ET supporting tools

Tools supporting ET		EST	FIN	Total
Software	Mind Maps (e.g. Xmind)	2	4	6
	Custom made tool	1	2	3
	Rapid Reporter	0	2	2
	Evernote	1	0	1
	Excel	0	1	1
	qTrace	0	1	1
	Vim-Editor	0	1	1
	Jira Test Sessions	1	0	1
	OneNote	1	0	1
	Perlclip	1	0	1
	IETester	1	0	1
	BB Flashback	0	1	1
	Total	*8*	*12*	*20 (80%)*
Non-Software	Literature	0	2	2
	PostIts	0	1	1
	Check lists	0	1	1
	Paper & pen	0	1	1
	Total	*0*	*5*	*5 (20%)*
Total		8	17	25 (100%)

Table 12 lists the types of supporting tools reported by ET users. Respondents could list as many tools as they wanted in free text format. The majority of tools are software-based (80%). Estonian ET users exclusively reported software-based tool support. Only 3 out of 20 ET users reporting software-based tool support listed custom-made tools for the specific purpose of supporting ET. None of the other tools listed are ET-specific tools. This result might indicate that there is either no need for or a lack of ET-specific tool support. From the table, we can see that Mind Mapping tools were the most popular (6/20) report using them. This suggests that ET specific tools might benefit from using mind map type of structure in the user interface.

RQ 1.4 In what testing context do software engineers use ET? Figure 1 and Figure 2 show the testing context in which ET is used. We listed nine typical test activities plus the item "automated testing" and asked whether this activity (or item) is performed, and if so, whether ET is used in this context. We

report only answers of those respondents who said they were using ET.

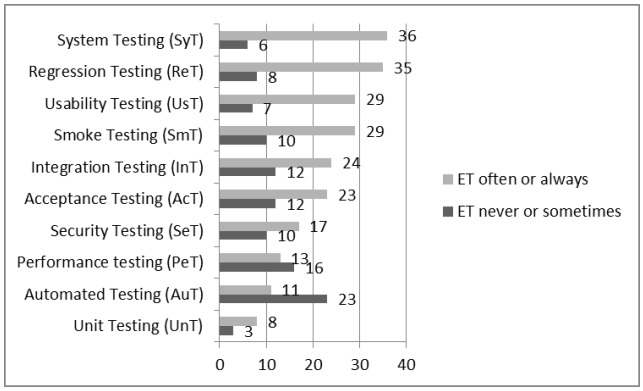

Figure 1. Testing context in which ET is used (absolute)

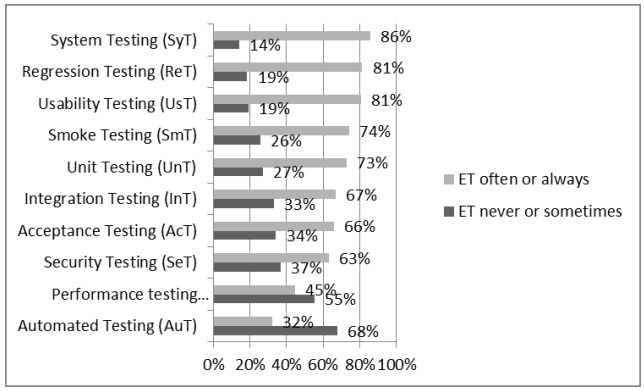

Figure 2. Testing context in which ET is used (relative)

It was surprising to see that most respondents (33 of 44) don't perform unit testing. This suggests that ET is mainly used by individuals who work on higher levels of testing, which is supported by the fact that programmers not testers are typically responsible for unit testing [15]. Another surprise was that more than 75% (34 out of 44) of the respondents did automated testing and that 50% of those who do automated testing use ET sometimes, often or always. It would be interesting to know more about how exactly those respondents combine automated testing with ET. Otherwise, the results show that whenever a test activity is performed, ET almost always used at least sometimes. This pattern seems to be consistent among both respondents from Estonia and Finland.

RQ 1.5 For what type of software do software engineers use ET? Table 13 shows the characteristics of the software tested by ET users. We offered four categories of software characteristics, i.e., usability-critical, security-critical, performance-critical, and safety-critical, where we assumed that ET would be frequently used. Respondents who were ET users could check one or more characteristics, or none of the offered characteristics. For example, a total of 13 respondents said they use ET for testing all four types of software (first row).

As expected, only two of the 44 responding ET users (both from Estonia) said that none of the four offered characteristics applied to their software. 26 of the 44 responding ET users (59%) reported that at least three of the four offered characteristics applied to their software. The most frequently mentioned characteristic was usability-criticality, followed by performance-criticality and security-criticality. No obvious differences between ET users in Estonia and Finland were visible.

Table 13. Characteristics of the software exposed to ET

Software Characteristics					Total / Estonia / Finland
Usability-critical	Security-critical	Performance-critical	Safety-critical	None of the mentioned	
x	x	x	x		13 / 6 / 7
x	x	x			6 / 5 / 1
x		x			6 / 3 / 3
x					4 / 2 / 2
x		x	x		3 / 1 / 2
x	x				3 / 1 / 2
x	x		x		2 / 2 / 0
	x	x	x		2 / 0 / 2
				x	2 / 2 / 0
	x		x		1 / 1 / 0
	x				1 / 1 / 0
		x			1 / 1 / 0
37 (84.1%)	28 (63.6%)	31 (70.5%)	21 (47.7%)	2 (4.5%)	44 /
20 (80%)	16 (64%)	16 (64%)	10 (40%)	2 (8%)	/ 25 /
17 (89.5%)	12 (63.2%)	15 (78.9%)	11 (57.9%)	0 (0.0%)	/ 19

RQ 1.6 Is the experience with using ET different between Estonia and Finland? To check whether there were differences between Estonian and Finnish users of ET, we tested the following null-hypotheses:

- H0-1: There is no difference in the size of ET users' organizational units between Estonian and Finnish ET users.
- H0-2: There is no difference in the age of ET users' organizational units between Estonian and Finnish ET users.
- H0-3: There is no difference in the ET users' current roles between Estonian and Finnish ET users.
- H0-4: There is no difference in the times that ET users' have worked in their current roles between Estonian and Finnish ET users.
- H0-5: There is no difference in the type of test (or QA) organization in ET users' organizational units between Estonian and Finnish ET users.
- H0-6: There is no difference in the timing when ET is used during testing between Estonian and Finnish ET users.
- H0-7: There is no difference in ET tool support between Estonian and Finnish ET users.

To test these hypotheses, we used both Pearson's Chi-square test (suitable for categorical data) and Fisher's exact test (suitable for small sample sizes, i.e., if table entries are smaller than 5). We used the R statistics package for all calculations, i.e. functions chisq.test(data) and fisher.test(data). We set the significance level alpha = 0.05. For the measurement of effect size, we report Cramer's V as we dealt with tables larger than 2 x 2.

In summary, regarding RQ 1.6 (differences between Estonian and Finnish ET users), it turned out that our tests didn't reveal any

significant differences between responses from Finnish and Estonian ET user for hypotheses H0-2 to H0-7. Only hypothesis H0-1 could be rejected, indicating that respondents from Finland who are using ET tend to work in larger organizational units than ET users from Estonia.

Table 14 shows the related data. Our Chi-square test ((df = 3, N = 44) = 11.68, p = 0.008545 < 0.05, Cramer's V = 0.515) showed that the effect size is greater than 0.5 and thus can be considered large. Due to the small entries in some table cells, we also used Fisher's exact test yielding p = 0.007582 and thus confirming the results produced by the Chi-square test. Note that the result is not significant when applying the Bonferroni correction.

Table 14. Size of ET users' organizational unit versus location

Number of Employees	EST	FIN	Total
Less than 20	12 (48%)	2 (10%)	14 (32%)
20 to 49	9 (36%)	5 (26%)	14 (32%)
50 to 99	2 (8%)	6 (32%)	8 (18%)
More than 100	2 (8%)	6 (32%)	8 (18%)
Total	25	19	44

5.2.2 RQ 2: Which factors have an influence on using ET?

In the following, we report the results related to RQ 2 for each sub-question separately.

To check whether there are factors that differ between respondents who are users of ET and respondents who don't use ET, related to each sub-question, we formulated and tested for each of RQ 2.1 to RQ 2.7 the following null-hypotheses:

- [RQ 2.1] H0-8: There is no difference in the respondents' locations between ET users and non-users.
- [RQ 2.2] H0-9: There is no difference in the size of the respondents' organizational units between ET users and non-users.
- [RQ 2.3] H0-10: There is no difference in the age of the respondents' organizational units between ET users and non-users.
- [RQ 2.4] H0-11: There is no difference in the respondents' current role between ET users and non-users.
- [RQ 2.5] H0-12: There is no difference in the respondents' times working in their current roles between ET users and non-users.
- [RQ 2.6] H0-13: There is no difference in the respondents' types of test organizations between ET users and non-users.
- [RQ 2.7] H0-14: There is no difference in the respondents' software characteristics between ET users and non-users.

The data related to H0-8 (RQ 2.1) was shown in Table 9 (Section 0). Table 16 to Table 21 show the data regarding H0-9 (RQ 2.2) to H0-14 (RQ 2.7), respectively. As we did for our hypothesis testing of RQ 1.6, for testing RQ 2, we used both Pearson's Chi-square test and Fisher's exact test. Again, we set the significance level alpha = 0.05. The test results are shown in Table 15. In summary, regarding RQ 2 (factors influencing the use of ET), it turned out that only hypothesis H0-12 could be rejected, indicating that the time respondents have been assuming their current role differs significantly between those respondents who use ET and those who don't use ET. The effect size was between 0.2 and 0.5 and thus can be considered moderate.

Table 15. Test results for RQ 2

	Pearson's Chi-square test					Fisher's exact test
	df	N	Chi-square	p-value	Cramer's V	p-value
H0-8	1	50	0.4175	0.5182	-	0.3946
H0-9	3	50	1.0206	0.7963	-	0.8811
H0-10	2	50	1.4378	0.4873	-	0.4104
H0-11	2	50	6.0023	0.0497	-	0.1029
H0-12	2	50	6.7179	0.0348	0.367	0.0156*
H0-13	1	50	0.0030	0.9859	-	1.0000
H0-14	4	135	0.6068	0.9623	-	0.9616

This finding is interesting in so far, as it corresponds to some of the findings related to RQ3 (opinion of ET users about ET), i.e. the opinion articulated by several respondents that ET imposes "high requirements on testers" (RQ3.3) and the relative low agreement to the characterization "ET is easy to use" (RQ 3.2). In other words, it might be easier to use ET for mature/experienced testers than for novices. Note that, as for H0-1, the result is not significant when applying Bonferroni correction.

For hypotheses H0-8 to H0-11 and H0-13 to H0-14, we couldn't find any significant differences between responses from ET users and non-users, thus indicating that location, size of organization, age of organization, current role, type of test organization and characteristics of the tested software don't differ between ET users and non-users.

Table 16. Usage of ET versus size of respondent's organizational unit

Number of Employees	Usage of ET		
	YES	NO	Total
Less than 20	14 (32%)	1 (16.7%)	15 (30%)
20 to 49	14 (32%)	2 (33.3%)	16 (32%)
50 to 99	8 (18%)	2 (33.3%)	10 (20%)
More than 100	8 (18%)	1 (16.7%)	9 (18%)
Total	44	6	50

Table 17. Usage of ET versus age of respondent's organizational unit

Number of Years	Usage of ET		
	YES	NO	Total
Less than 2 years	3 (7%)	1 (16.7%)	4 (8%)
2 to 5 years	16 (36%)	3 (50%)	19 (38%)
More than 5 years	25 (57%)	2 (33.3%)	27 (54%)
Total	44	6	50

Table 18. Usage of ET versus respondent's current role

Current Role	Usage of ET		
	YES	NO	Total
Tester	24 (55%)	2 (33%)	26 (52%)
Test Manager	18 (41%)	2 (33%)	20 (40%)
Other role cooperating with Tester/Test Manager	2 (4%)	2 (33%)	4 (8%)
Total	44	6	50

Table 19. Usage of ET versus respondent's time working in current role

Time worked in Current Role	Usage of ET		
	YES	NO	Total
Less than 2 years	7 (16%)	3 (50%)	10 (20%)
2 to 5 years	14 (32%)	3 (50%)	17 (34%)
More than 5 years	23 (52%)	0 (0%)	23 (46%)
Total	44	6	50

Table 20. Usage of ET versus type of test organization

Type of test (or QA) organization	Usage of ET		
	YES	NO	Total
Separate test (or QA) organization	25 (57%)	4 (66.6%)	29 (58%)
No separate test (or QA) organization	19 (43%)	2 (33.3%)	21 (42%)
Total	44	6	50

Table 21. Usage of ET versus characteristics of tested software

Software Characteristics	Usage of ET		
	YES	NO	Total
Usability-critical	37 (30%)	4 (25%)	41 (30.3%)
Security-critical	28 (24%)	4 (25%)	32 (23.7%)
Performance-critical	31 (26%)	5 (31%)	36 (26.6%)
Safety-critical	21 (18%)	3 (19%)	24 (17.7%)
None of the above	2 (2%)	0 (0%)	2 (1.7%)
Total	119	16	135

5.2.3 RQ3: How do software engineers think about ET?

In the following, we report the results related to RQ 3 for each sub-question separately.

RQ 3.1 What elements consider software engineers as essential for defining ET? To find out what definition of ET respondents that are using ET have in mind, we asked survey participants whether they think one or more of the offered statements about ET is correct (cf. Table 22). In addition, respondents could offer additional elements of ET under item "other" in a free text form. Table 22 shows the results. It turned out that a high percentage of respondents (79.5%) think that a defect log (or defect report/list) is an element of ET. ET Elements that were often agreed on (47%-57%) included "ET has a test log", "ET is session-based", "ET has a mission statement or charter" and "ET is time-boxed". This is interesting in so far as none of the ET elements that received high response rates are explicitly mentioned in the original definitions of ET by Cem Kaner. It seems that the more recent evolutions of ET towards session-based test management (SBTM) [1] have had a strong influence on what ET users think is part of ET.

On the other hand, there were several respondents who made it clear that they think elements of SBTM should not be considered part of the definition of ET. This can be seen from the list of "Other" answers shown below. Alternatively, some respondents clarified that for them ET can be anything. This is interesting in so far, as it seems to indicate that for those ET users it is difficult to explain what ET actually is and how it could be distinguished from other approaches to testing.

Table 22. Elements of ET

ET Element	Responses
ET has metrics	10 (22.7%)
ET has playbooks	7 (15.9%)
ET is time-boxed	21 (47.7%)
ET is session-based	24 (54.5%)
ET has a debriefing meeting	15 (34.1%)
ET has systematic coverage tracking	10 (22.7%)
ET has a mission statement or a charter	23 (52.3%)
ET has a defect log (or defect report/list)	35 (79.5%)
ET has a test log (recording of what was tested and/or how)	31 (57.4%)
Other [with free text input option]	9 (20.5%)
None of above	2 (4.5%)
Total	44

Other answers:

- 'Extra testing in addition to planned testing.'
- 'Don't mix ET with Session Based Testing. The plainest definition of exploratory testing is test design and test execution at the same time. ET is an approach, not another testing technique.'
- 'The statements above are about SBTM not ET. ET can be done without all of the above or with some of the above.'
- 'ET utilizes people and tools'
- 'ET can have all those things, but not necessarily always together'
- 'Based on oracles, skills, ideas etc.'
- 'Catches bug which other testing types misses.'
- 'Sapience'
- 'ET is about simultaneous exploration, observing, planning, experimentation and communicating your findings. All these running in a loop followed by each other. Purpose is to test by exploring.'
- 'Testing with up to date requirements'
- 'All of the options could be used with ET but they seem to be attributes of STBM, not ET in itself'
- 'ET can have anything you want. It's an approach to testing.'

RQ 3.2 What characteristics do software engineers think ET has? Table 23 shows the characteristics that respondents think ET has. Interestingly, 34 out of 44 ET users said that they either agree or strongly agree to any of the six characteristics offered. Moreover, a careful look at the table suggests that the seven characteristics can be classified in two groups, the first group containing those characteristics to which more than half of the ET users strongly agree ("ET supports creativity", "ET makes testing interesting and engaging", "ET is flexible"), and the second group containing those characteristics where the most frequent answer choice was a simple "agree" ("ET is easy to use", "ET is focused", "ET is efficient", "ET is effective"). Perhaps the most interesting aspect of the responses presented in Table 23 is the comparatively low agreement the characteristic "ET is easy to use".

RQ 3.3 What do software engineers think are advantages and disadvantages of ET? Table 24 and Table 25 list advantages and disadvantages, respectively, identified by respondents. Note that like for all other questions relating to RQ 3, due to the design of the survey questionnaire, we received answers exclusively from

ET users, and thus the responses shown in the tables correspond to those answers received from ET users in Estonia and Finland.

Table 23. Characteristics of ET

ET Characteristics	--	-	0	+	++	Total
ET is easy to use.	0	3	13	20	8	44
ET supports creativity.	0	0	2	13	29	44
ET is focused (goal-oriented)	0	2	11	21	10	44
ET makes testing interesting and engaging	1	0	2	13	28	44
ET is flexible (can be used in many different test situations)	0	1	1	17	25	44
ET is efficient (finds defects faster than other methods)	1	2	13	15	13	44
ET is effective (finds defects which other methods would not)	0	2	8	19	15	44

--: strongly disagree (-2)
-: disagree (-1)
0: neither agree nor disagree / balanced (0)
+: agree (+1)
++: strongly agree (+2)

Table 24. Ranked advantages of ET

Advantage (classified)	Adv 1 (* 3)	Adv 2 (* 2)	Adv 3 (* 1)	Total Score
Supports creativity	6	12	5	47
Efficient	10	6	1	43
Effective	7	3	5	32
Flexible	3	3	2	17
Supports learning	3	3	2	17
Time saving	4	1	1	15
Interesting	2	3	3	15
Easy	3	2	0	13
Emphasizes tester	2	1	1	9
Focused	1	0	1	4
Essential	1	0	0	3
Independent	0	1	0	2
Clear data	0	0	1	1
Create logs	0	0	1	1
Total	42	35	23	219

Each respondent could list up to three ranked advantages in a free format text fields. In order to make the rankings comparable, we weighted the first mentioned advantage (Adv 1) with factor 3, the second mentioned advantage (Adv 2) with factor 2 and the third mentioned advantage (Adv 3) with factor 1, and then summed up the weighted frequencies per mentioned advantage. As a result, "Supports creativity" (mentioned by 23 ET users) turned out to be the most important advantage of ET, followed by "Efficient" (mentioned by 17 ET users) and "Effective" (mentioned by 15 ET users).

The list of disadvantages of ET is longer than that of advantages and also less pointed. The most prominent disadvantage mentioned is "high requirement for tester" (mentioned by 14 ET users), followed by "inflexible" (mentioned by six ET users) and by "hard to record" (also mentioned by six ET users).

Table 25. Ranked disadvantages of ET

Disadvantage (classified)	Dis 1 (* 3)	Dis 2 (* 2)	Dis 3 (* 1)	Total Score
high requirement for tester	9	4	1	36
inflexible	5	0	1	16
hard to record	2	4	0	14
not good for complicated project	4	0	0	12
not all-inclusive	2	0	0	6
time consuming	2	0	0	6
no focus	1	1		5
confusing	1	0	0	3
hard to compare results	1	0	0	3
inefficient	1	0	0	3
stakeholders don't appreciate	1	0	0	3
time limit	1	0	1	4
too popular	1	1	0	5
unnecessary	1	0	0	3
unrepeatable	1	0	0	3
hard to report	0	4	0	8
ineffective	0	2	0	4
uncontrollable	0	2	0	4
energy consuming	0	1	0	2
inaccurate	0	1	0	2
Total	33	20	3	56

RQ 3.4 Do software engineers want to improve ET, and how? Table 26 shows suggestions to improve respondents' current practice of using ET. Again, respondents could make suggestions in a free format text field. They were allowed to make as many suggestions as they like. Overall, we received only 15 suggestions. Six of the 15 suggestions are related to improving the reporting or to creating a record for ET.

Table 26. Improvement suggestions for current ET practice

Respondent's plan for change	Frequency
Create a record for ET	3
Improve report	2
Find a better reporting system for ET	1
More risk-based testing.	1
Study more and have more experience	1
Use a good tool	1
Use ET more often	1
Use SBTM and TBTM together	1
Choose how to do ET according to project	1
Improve testing all the time	1
Do ET in the morning	1
Total	15

6. DISCUSSION

In this section, we discuss the results presented in the previous section.

Regarding RQ 1 we found that respondents from both Estonia and Finland using ET, reported the following experience:

- More than 70% of the respondents said they use ET at any time during testing (RQ1.2), and more than 60% of the

respondents said that they use ET at all test levels and in combination with all test approaches offered in the survey questionnaire, even together with automated testing (RQ 1.4). These results suggest that ET is highly flexible.

- 75% of the respondents don't have specific tool support when using ET (RQ 1.3).
- More than 60% of the respondents said they use ET for testing usability-critical, performance-critical, and security-critical software. More than 47% said they use ET for testing safety-critical software (RQ 1.5). Since the ET usage rates reported for the various types of software corresponds very closely with the relative occurrences of the respective types of software respondents said they normally work with (cf. demographic data, Table 8), we conclude that those respondents who use ET actually use it for any type of software they are working with almost always. These results suggest that ET is broadly applicable.
- With the exception of size of the organization ET users are employed with, we couldn't find any statistically significant differences between responding ET users in Finland and Estonia. The difference in company size might simply reflect the difference in size structure of companies between Finland and Estonia.

We also found that a vast majority of respondents use ET frequently (RQ 1.1). However, it is probable that this finding is a result of self-selection bias.

Regarding RQ 2 we found statistical support for only one factor having an influence on using ET, i.e., the longer our respondents said they were working in their current role the more frequently they said they were using ET. This suggests that more experience/maturity of testers, developers, or managers involved in testing are, the higher are the chances they use ET. This finding corresponds to some degree with the results found by Itkonen et al. [8] on the role of the tester's knowledge on ET.

Regarding RQ 3 we found the following:

- Almost 80% of the ET users said that a defect log (or defect reporting) is part of ET.
- More than 50% of ET users said that ET has a test log, is session-based and has a charter/mission statement. The high rate of responses stating that being session-based is an element of ET seems to indicate that many ET users do not make the distinction between session-based testing and (pure) ET as it is advocated by some of our respondents.
- Overall, as one would expect, respondents using ET mentioned more advantages (219) than disadvantages of ET (56).
- The three top-ranked advantages of ET mentioned by respondents are "Supports creativity", "Efficient", and "Effective". This largely corresponds to what other researchers have found, e.g., Shah et al. [18] and Itkonen et al. [7][9].
- The three top-ranked disadvantages of ET mentioned by respondents are "high requirement for tester", "inflexible", "hard to record", "not good for complicated project". The top-ranked disadvantage might explain why we found in RQ 2 that the experience of a test role (in terms of time having assumed such a role) correlates positively with the using ET. In addition, a relatively low share of responding ET users agreed to the statement that "ET is easy to use" (RQ 3.2). The relatively frequent mentioning of the disadvantage "inflexible" comes somewhat as a surprise. Perhaps, this finding is also related to the fact that ET puts high requirements on the tester. Another explanation could be that in order to do ET, the tester needs to know much about the software under test and thus ET is inflexible with regards to the persons who can perform the testing. The fourth highest ranked advantage was "flexible". Also, the answers by most ET users as to when and on what types of software they use ET seem to contradict its characterization as "inflexible".

7. THREATS TO VALIDITY

Every empirical study will have shortcomings. Here, we discuss the most pressing issues for our study.

7.1 Selection Bias

Selection bias refers to an error in choosing the individuals or groups to take part in a study. Since no survey similar to ours has been previously conducted, we do not know baseline characteristics of the population with which to compare our specific sample. We advertised our survey to practitioners through several channels. In the advertisement, we invited software engineers and managers to participate whether they use ET or not. However, the wording in our advertisements and in the survey itself may have encouraged engineers and managers who actually are using ET to participate. Given the small number of respondents who were not ET users there is a good chance that there was a self-selection bias in favor of ET users amongst our respondents.

7.2 Construct Validity

Construct validity pertains to how well the measures in an empirical study reflect the concepts under investigation, and also to how well-defined the concepts are. In our study, this translates to how meaningful our research questions are, how appropriate the derived hypotheses are, and to what extent the survey questionnaire items were appropriate for giving answers to the hypotheses and research questions. We made efforts to follow standard guidelines for designing survey questionnaires, e.g., [4][5][14], and we had external industry representatives review the questionnaire. Nevertheless, it is possible that some items were interpreted by respondents in an unintended way. For example, it is possible that the item asking about tool support might have been interpreted by some to be asking for software-based tool support only. Similarly, the fact that 75% of respondents using ET said they don't do unit testing but at the same time more than 75% of the same respondents said they were doing automated testing might indicate that automated unit testing is was checked by some respondents only once, either under unit testing or under automated testing. In other words, it might have been confusing that the items describing test context were not necessarily mutually exclusive. Finally, it should be pointed out that the whole concept of ET is vague as seen in the responses to RQ 3.1 which investigated elements of ET.

7.3 External Validity

External validity concerns the extent to which conclusions drawn on the study's specific operationalizations transfer to variations of these operationalizations [4]. Due to the low number of responses received, generalization of our findings might be questionable. This is particularly true when looking at RQ 2 where we tried to find factors that positively or negatively affect the use of ET. It is not plausible that the high imbalance between ET users and non-

users in our response set properly reflects the actual distribution between ET users and non-users among Estonian and Finnish software engineers and managers. On the other hand, we have no indication that the sets of respondents who are ET users are in any way non-typical for ET users from either country.

7.4 Statistical Conclusion Validity

Statistical conclusion validity pertains to the conclusions drawn from the statistical analyses, and the appropriateness of the statistical methods used in the analyses. Due to the relatively low number of responses, the statistical power of the tests conducted is low. However, in the two cases where we could reject the null hypothesis and the effects observed were high and medium. The statistical test we applied (Fisher's exact test for contingency tables) is suitable for small data sets and doesn't require the fulfilment of any specific assumptions about the data to which the test is applied.

8. CONCLUSIONS AND FUTURE PLANS

While there is a high risk that our survey results are partly influenced by a strong self-selection bias of respondents towards ET users, the consistency of many of our findings with those by other researchers suggest a reasonable degree of validity of our study results. Several of our findings suggest that more support to ET users should be given (guidance and tools). Some of our findings were surprising, e.g., the combination of ET with test automation as well as the high degree of usage of ET not only to usability-critical software but also to performance-critical, security-critical and safety-critical software. This suggests that we replicate the survey elsewhere and complement it with in-depth case studies in selected companies in Estonia and Finland.

9. ACKNOWLEDGMENTS

We would like to thank all survey respondents as well as several collaborators in industry who piloted our survey questionnaire. Dietmar Pfahl was supported by the Estonian higher education information and communications technology and research and development activities state program 2011-2015 (ICT program) - EU Regional Development Fund, and by the institutional research grant IUT20-55 of the Estonian Research Council.

10. REFERENCES

[1] J. Bach. Session-Based Test Management. Software Testing and Quality Engineering Magazine, 2000.

[2] J. Bach. Exploratory Testing. In: The Testing Software engineer, 2nd ed., E. van Veenendaal (Ed.) Den Bosch: UTN Publishers, pp. 253-265, 2004.

[3] P. Bourque, R. E. Fairley (ed.). SWEBOK V3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, 2014.

[4] F. J. Fowler, Jr. Improving Survey Questions. Design and Evaluation. Sage, 1995.

[5] F. J. Fowler, Jr. Survey Research Methods. Sage, third edition, 2002.

[6] J. Itkonen, M. V. Mäntylä. Are test cases needed? Replicated comparison between exploratory and test-case-based software testing. Empirical Software Engineering, pp. 1-40, 2013. DOI 10.1007/s10664-013-9266-8 (published online)

[7] J. Itkonen, M. V. Mäntylä, C. Lassenius. Defect Detection Efficiency: Test Case Based vs. Exploratory Testing, First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), pp. 61-70, 2007.

[8] J. Itkonen, M. V. Mantyla, C. Lassenius. The role of the tester's knowledge in exploratory software testing. IEEE Transactions on Software Engineering, 39(5): 707-724, 2013.

[9] J. Itkonen, K. Rautiainen. Exploratory testing: A multiple case study. Proceedings of the 4th International Symposium on Empirical Software Engineering (ISESE 2005). Noosa Heads, Queensland, Australia. IEEE, pp. 84-93, 2005.

[10] C. Kaner, J. Falk, and H. Q. Nguyen. Testing Computer Software (Second Edition), Van Nostrand Reinhold, New York, 1993.

[11] C. Kaner. A Tutorial in Exploratory Testing. Tutorial presented at QUEST2008. (Available online at: http://www.kaner.com/pdfs/QAIExploring.pdf, accessed: 26 Jan 2014)

[12] E. MacGregor, Y. Hsieh, P. Kruchten. Cultural patterns in software process mishaps: incidents in global projects. ACM SIGSOFT Software Engineering Notes, 30(4):1-5, 2005

[13] L. H. O. do Nascimento, P. D. L. Machado. An experimental evaluation of approaches to feature testing in the mobile phone applications domain. In: Workshop on Domain specific approaches to software test automation: in conjunction with the 6th ESEC/FSE joint meeting (DOSTA '07). ACM, New York, NY, USA, pp. 27-33, 2007.

[14] S. L. Pfleeger, B. A. Kitchenham. Principles of Survey Research, Part 1: Turning Lemons into Lemonade. Software Engineering Notes, 26(6): 16-18, 2001.

[15] P. Runeson. A survey of unit testing practices. IEEE Software, 23.4: 22-29, 2006.

[16] S. Saukkoriipi, I. Tervonen. Team Exploratory Testing Sessions. ISRN Software Engineering, vol. 2012, Article ID 324838, 20 pages, 2012. doi:10.5402/2012/324838

[17] W. R. Shadish, T. D. Cook, and D. T. Campbell. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin, 2002.

[18] S. M. A. Shah, C. Gencel, U. S. Alvi, K. Petersen. Towards a hybrid testing process unifying exploratory testing and scripted testing. Journal of Software: Evolution and Process, 26: 220-250, 2014.

[19] J. Våga, S. Amland. Managing high-speed web testing. In: Software quality and software testing in internet times, D. Meyerhoff, B. Laibarra, R. van der Pouw Kraan, A. Wallet (Eds.). Springer-Verlag New York, Inc., New York, NY, USA, pp. 23-30, 2002.

[20] J. A. Whittaker. Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design. Addison Wesley Professional, 2009.

[21] H. Yin. Survey on Exploratory Testing. Master's Thesis at University of Tartu, 2014. (Available online at: http://comserv.cs.ut.ee/forms/ati_report/datasheet.php?id=39151&year=2014)

APPENDIX A

http://figshare.com/articles/esem_2014_v5_appendix_pdf/971024