

Checklists to Support Test Charter Design in Exploratory Testing

Ahmad Nauman Ghazi, Ratna Pranathi Garigapati, and Kai Petersen

Blekinge Institute of Technology, Karlskrona, Sweden
nauman.ghazi@bth.se, pranathi.r8@gmail.com, kai.petersen@bth.se

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.

Keywords: Exploratory testing, session-based test management, test charter, test mission

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 2 presents the related work. Section 3 outlines the research method, followed by the results in Section 4. Finally, in Section 5, we present the conclusions of this study.

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.

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 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.

Table 1: Profile of the Interviewees

Interview ID	Role	Experience in testing	Organizational size
1	Senior Systems Test Engineer	4 years	More than 500
2	Test Quality Architect	10 years	50-500
3	Test Specialist	10 years	50-500
4	Test Consultant	12 years	More than 500
5	Test Strategist	3 years	Less than 50
6	CEO, Test Consultant	30 years	More than 500
7	Test Manager	20 years	More than 500
8	CEO, Test Lead	4 years	50-500
9	Test Quality Manager	13 years	50-500

Table 2: Factors influencing test charter design

Charter influence factors	Description
<i>F01</i> : Client Requirements	Requirements elicited from clients.
<i>F02</i> : Test Strategy	Set of ideas that guide the test plan.
<i>F03</i> : Knowledge of Previous Bugs	Knowledge regarding system related bugs that occurred in the past
<i>F04</i> : Risk Areas	Results of product risk analysis.
<i>F05</i> : Time-frame	Time needed for test mission execution, time constraints
<i>F06</i> : Project Purpose	Purpose of the project
<i>F07</i> : Test Function Complexity	Complexity of the tested functions.
<i>F08</i> : Functional Flows	Flow of data and functions
<i>F09</i> : Product Purpose	Principle goal(s) of the product
<i>F10</i> : Business Use-case	Business use-case for the system.
<i>F11</i> : Test Equipment Availability	Accessibility to tools and equipment needed for the software tests.
<i>F12</i> : Effort Estimation	Effort needed to carry out the test mission.
<i>F13</i> : Test Planning Checklist	Testing heuristics appointed for the particular test charter.
<i>F14</i> : Product Characteristics	Features of the product.
<i>F15</i> : Quality Requirements	Quality requirements of the product.
<i>F16</i> : Test Coverage Areas	Parts of the system to be tested.
<i>F17</i> : Test Team Communication	Means of communication between the testing team members.
<i>F18</i> : Project Plan	Plan for the project prior to its execution.
<i>F19</i> : General Software Design	Design of the system software.
<i>F20</i> : System Architecture	Structure, interfaces and platforms of the system.
<i>F21</i> : Process Maturity Level	Maturity of the process (e.g. CMMI levels).
<i>F22</i> : Product Design Effects	Impact of product design and features on other modules.
<i>F23</i> : Feedback and Consolidation	Feedback and consolidation of the test plan based on the comments of previous testers and clients.
<i>F24</i> : Session Notes	Notes filled during previous test sessions.
<i>F25</i> : SDLC Phase	Phase involved in the system development life-cycle
<i>F26</i> : Tester	Testers and their experience level
<i>F27</i> : Client Location	Location of the client, local or global
<i>F28</i> : System heterogeneity	Differences between interacting systems (different programming languages, platforms, system configuration)
<i>F29</i> : Project Revenue	Business returns for project
<i>F30</i> : User Journey Map	User interaction with the product over time.

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 2). The 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 Usecase, *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.
- *Product factors*: Product factors describe the attributes of the product under test, they include: "F08: Functional flows, *F09*: Product Purpose, *F14*: Product Characteristics, *F19*: General Software design, *F20*: System Architecture, *F22*: Product Design Effects, and *F28*: Heterogeneous Dimensions.

Table 3: Contents of test charters

Content type	Description
C01: Test Setup	Description of the test environment.
C02: Test Focus	Part of the system to be tested.
C03: Test Level	Unit, Function, System test, etc.
C04: Test Techniques	Test techniques used to carry out the tests.
C05: Risks	Product risk analysis.
C06: Bugs Found	Bugs found previously.
C07: Purpose	Motivation why the test is being carried out.
C08: System Definition	Type of system (e.g. simple/ complex).
C09: Client Requirements	Requirements specification of the client.
C10: Exit Criteria	Defines the “done” criteria for the test.
C11: Limitations	It tells of what the product must never do, e.g. data sent as plain text is strictly forbidden.
C12: Test Logs	Test logs to record the session results.
C13: Data and Functional Flows	Data and work flow among components.
C14: Specific Areas of Interest	Where to put extra focus on during the testing.
C15: Issues	Charter specific issues or concerns to be investigated.
C16: Compatibility Issues	Hardware and software compatibility and interoperability issues.
C17: Current Open Questions	Existing questions that refer to the known unknowns.
C18: Information Sources	Documents and guidelines that hold information regarding the features, functions and systems being tested.
C19: Priorities	Determines what the tester spends most and least time on.
C20: Quality Characteristics	Quality objectives for the project.
C21: Test Results Location	Test results location for developers to verify.
C22: Mission Statement	One liner describing the mission of the test charter.
C23: Existing Tools	Existing software testing tools that would aid the tests.
C24: Target	What is to be achieved by each test.
C25: Reporting	Test session notes.
C26: Models and Visualizations	People, mind maps, pictures related to the function to be tested.
C27: General Fault	Test related failure patterns of the past.
C28: Coverage	Charter’s boundary in relation to what it is supposed to cover.
C29: Engineering Standards	Regulations, rules and standards used, if any.
C30: Oracles	Expected behavior of the system (either based on requirements or a person)
C31: Logistics	How and when resources are used to execute the test strategy, e.g. how people in projects are coordinated and assigned to testing tasks.
C32: Stakeholders	Stakeholders of the project and how their conflicting interests would be handled.
C33: Omitted Things	Specifies what will not be tested.
C34: Difficulties	The biggest challenges for the test project.
C35: System Architecture	Structure, interfaces and platforms concerning the system, and its impact on system integration.

- *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.
- *Testing*: Testing factors include contextual information relevant for the planning, design and execution of the tests. They include: F02: Test Strategy, F03: Knowledge of Previous Bugs, F04: Risk Areas, F07: Test Function Complexity, F11: Test Equipment Availability, F13: Test Planning Checklist, F16: Test coverage areas, F23: Feedback and Consolidation , F24: Session Notes, and F26: Tester.

RQ2: What do practitioners include in their test charters? The interviews revealed 35 different contents that may be included in a test charter. Table 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, in-

frastructure, 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.
- *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.

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 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.

References

- [1] W. Afzal, A. N. Ghazi, J. Itkonen, R. Torkar, A. Andrews, and K. Bhatti. An experiment on the effectiveness and efficiency of exploratory testing. *Empirical Software Engineering*, 20(3):844–878, 2015.
- [2] J. Bach. Session-based test management. *Software Testing and Quality Engineering Magazine*, 2(6), 2000.
- [3] J. Bach. Exploratory testing explained, 2003.
- [4] J. Bach and M. Bolton. Rapid software testing. *Version (1.3. 2)*, *www.satisficc.com*, 2007.
- [5] R. E. Christ. Review and analysis of color coding research for visual displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 17(6):542–570, 1975.
- [6] A. N. Ghazi. Testing of heterogeneous systems. *Blekinge Institute of Technology Licentiate Dissertation Series*, 2014(03):1–153, 2014.
- [7] A. N. Ghazi, K. Petersen, and J. Börstler. Heterogeneous systems testing techniques: An exploratory survey. In *Proceedings of the Software Quality Days (SWQD), 20.01.-22.01.2015, Vienna, Austria*, pages 67–85. Springer-Verlag, 2015.
- [8] E. Hendrickson. *Explore it!* The Pragmatic Programmers, 2014.
- [9] J. Itkonen et al. Empirical studies on exploratory software testing. 2011.
- [10] J. Itkonen and M. V. Mäntylä. Are test cases needed? replicated comparison between exploratory and test-case-based software testing. *Empirical Software Engineering*, 19(2):303–342, 2014.
- [11] J. Itkonen and K. Rautiainen. Exploratory testing: a multiple case study. In *2005 International Symposium on Empirical Software Engineering, 2005.*, pages 10–pp. IEEE, 2005.
- [12] C. Kaner, J. Bach, and B. Pettichord. *Lessons learned in software testing*. John Wiley & Sons, 2008.
- [13] D. Pfahl, H. Yin, M. V. Mäntylä, J. Münch, et al. How is exploratory testing used? In *ESEM'14 Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2014.