Poster: Is *Carmen* better than *George*? Testing the Exploratory Tester using HCI Techniques

Andrea Borg, Chris Porter, Mark Micallef
{andrea.borg.12 | chris.porter | mark.micallef}@um.edu.mt
University of Malta

***Abstract*—Exploratory software testing is an activity which can be carried out by both untrained and formally trained testers. In this paper, we propose using Human Computer Interaction (HCI) techniques to carry out a study of exploratory testing strategies used by the two groups of testers. This data will be used to make recommendations to companies with regards to the mix of skills and training required for testing teams.**

I. Introduction

Carmen and George are both employed as software testers. However, whilst George is formally trained and certified, Carmen has no training but got the job because she is *'good at finding bugs'*. A heated debate about whether or not software testers should be formally trained and certified was sparked by the announcement in 2011 of the development of a new ISO standard on software testing [1] and further fuelled by the perception that unlike other roles in software engineering, testing can be carried out by anyone, providing they understand an application's domain. One side argues that certified testers will approach testing with discipline and consistency whilst the other argues that much like a driving license does not make one a good driver, a testing certification does not guarantee a good tester. Furthermore, end-users regularly find and report bugs in systems even though they are not trained as testing professionals. Testing encompasses a wide range of skills (e.g., planning, design, automation, exploratory testing), proficiency in most of which requires a certain level of formal training. However, an interesting research opportunity presents itself when one considers that *exploratory testing* can be carried out by both trained and untrained testers. *Exploratory testing* involves a software tester interacting with a system in an unscripted manner guided mainly by her intuition and experience. Although it is a recognised approach, the technique is frequently referred to as *ad-hoc* testing and suffers from the reputation of delivering inconsistent results depending on the tester executing it. Despite the fact that there are documented exploratory testing strategies which can be utilised [2], effectiveness has been shown to be dependent on the tester's knowledge [3], learning style [4] and even personality [5]. The debate as to whether or not formal training is a positive or negative influence on the quality of exploratory testing motivates our hypothesis.

A. Hypothesis and Research Questions

If one partitions testers into two broad groups such that one group consists of testers with a formal qualification in software testing, and the second group consists of testers with no such formal training, then we hypothesise that the two groups of testers intuitively use different yet complimentary exploratory testing strategies. In order to explore this hypothesis, we propose to investigate three research questions: (RQ1) Which types of exploratory testing strategies are utilised by testers in each group?, (RQ2) Which types of bugs are found by testers in each group?, and (RQ3) Is there a link between the bugs found and the expected testing strategies adopted by the tester groups?

B. How can HCI techniques help?

Human Computer Interaction (HCI) is an interdisciplinary area of research and practice which has evolved throughout the years by attracting researchers and practitioners from various other disciplines, including cognitive sciences (e.g., human factors and psychology) as well as design and communications [6]. The field has produced tools and techniques which help requirements engineers, system designers and software developers steer their activities towards the production of usable and effective systems while encouraging positive user experiences.

In this study, we propose the use of HCI techniques to shed more light on the different strategies and associated testing rigour adopted by two different categories of software testers. We will mainly be utilising (1) *Personas* to represent tester archetypes; (2) *Eye Tracking*, a technology which is used to accurately record participant interaction with a system-under-test; (3) *Reflective Think-Aloud (RTA)*, a technique whereby the researcher takes note of points of interest (POIs) during an exercise and subsequently invites a participant back for a reflective discussion on those POIs; and finally (4) *Thematic Analysis* (TA), a technique in which all qualitative data generated throughout the study (i.e., via *RTA* sessions and semi-structured interviews) is systematically analysed to extract both implicit and explicit patterns (i.e., themes) from the corpus of data. The way in which these HCI tools will be incorporated into the study is discussed in Section II in which we present a data collection and analysis protocol which has been developed for this work.

C. Research scope

In this paper we outline our plan for a controlled case study amongst a population of testers uniformly divided into one of the two groups discussed above. The case study will require participants to exploratory test an e-commerce system with

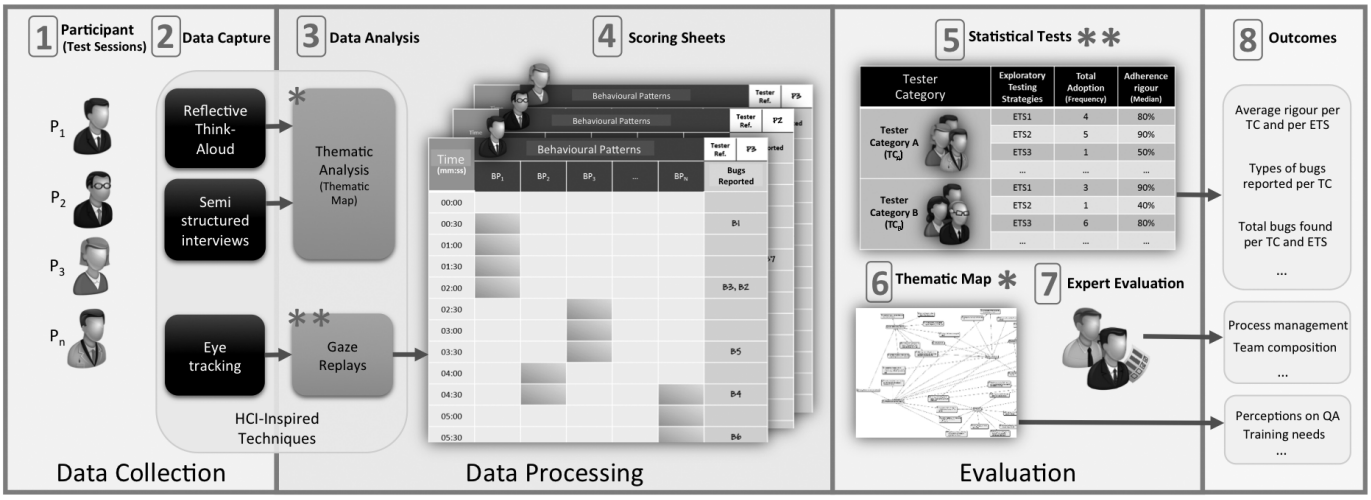

Fig. 1. Methodology for the proposed study

which they are not familiar. We will subsequently observe testers and collect data utilising HCI techniques in order to investigate the research questions outlined in Section I-A.

II. Method

In preparation, we firstly identified a number of exploratory testing strategies found in literature and subsequently designed a *scoring sheet* which maps a series of observed participant behavioural patterns to a probability that a particular strategy is being used during any particular timeframe. We then developed a fictitious e-commerce system within which several bugs were injected. A booklet containing background information about the system as well as fictitious usage statistics was produced for the system's various function points. This booklet will be sent to all recruited testers prior to their scheduled session for familiarisation purposes. This setup provides participants with the opportunity to execute any exploratory testing strategies known to them (explicitly or otherwise), while the purpose-built data collection and analysis protocol will allow us to monitor all strategies known to us at the time of designing the study. This data collection and analysis protocol was refined through a number of pilot sessions. Recruitment of participants was carried out during the protocol design phase through our links with the ICT industry.

As depicted in Figure 1, our data collection and analysis protocol consists of four main stages. These are (1) *data collection*, (2) *data processing*, (3) *evaluation*, and finally (4) the drawing of *conclusions*. During the *data collection* stage, participants come into our interaction design lab for a two-hour data collection session. On arrival, they are given a brief introduction to the study, the system-under-test and the eye-tracking equipment. Following that, they are allowed to exploratory-test the system for up to one hour whilst a researcher observes and takes notes unobtrusively. Finally, debriefing is conducted through a semi-structured interview which is potentially preceded by a *RTA* session in case POIs were identified. Following this stage we will then move on to the *data processing* stage. Here, transcribed *RTA* sessions as well as semi-structured interviews are analysed thematically following Braun and Clarke's TA process [7]. A visual representation of the underlying themes and patterns is also produced (i.e., thematic map). Also, data from gaze replays is analysed and scoring sheets are populated. We then move on to the *evaluation* and *conclusion* stages. Here, we leverage processed data in order to produce outcomes. Firstly, *statistical tests* on scoring sheet data will help us identify any correlations between tester categories, test strategies and the types of bugs found, whilst *thematic maps* will help us uncover underlying perceptions of participants on exploratory testing and testing in general. Finally, *expert evaluation* will consolidate our conclusions enabling us to produce our research outcomes as discussed in Section III.

III. Expected Research Outcomes

Although work is still in-progress, we expect results to enable us to make recommendations regarding *team composition* (the ideal mix of testers from the two categories), *training* (awareness of exploratory testing strategies) and *process management* (when are different types of testers most effective).

References

[1] ISO, *ISO Standard on Software Testing*, ISO 29 119:2013, 2013.

[2] J. A. Whittaker, *Exploratory software testing: tips, tricks, tours, and techniques to guide test design*. Pearson Education, 2009.

[3] J. Itkonen, M. V. Mantyla, and C. Lassenius, "The role of the tester's knowledge in exploratory software testing," *Software Engineering, IEEE Transactions on*, vol. 39, no. 5, pp. 707–724, 2013.

[4] A. Tinkham and C. Kaner, "Learning styles and exploratory testing," in *Proceedings of the Pacific Northwest Software Quality Conference*, 2003.

[5] L. Shoaib, A. Nadeem, and A. Akbar, "An empirical evaluation of the influence of human personality on exploratory software testing," in *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*. IEEE, 2009, pp. 1–6.

[6] J. M. Carroll, "Human computer interaction (hci)," ser. Encyclopedia of Human-Computer Interaction, W. Brainbridge, Ed. Thousand Oaks: Sage Publications, 2004.

[7] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.