Characteristics of High Performing Testers – A Case Study

Joonas Iivonen
Aalto University, SoberIT
P.O. Box 19210,
FI-00760 Aalto, Finland
-
joonas.iivonen@iki.fi

Mika V. Mäntylä
Aalto University, SoberIT
P.O. Box 19210,
FI-00760 Aalto, Finland
+358 50 577 1684
mika.mantyla@tkk.fi

Juha Itkonen
Aalto University, SoberIT
P.O. Box 19210,
FI-00760 Aalto, Finland
+358 577 1688
juha.itkonen@tkk.fi

ABSTRACT

Context: Previous research suggests a disconnect between testing research and industry practice. **Objective:** We studied what are the characteristics of high performing software testers in the industry. **Method:** We conducted an exploratory case study, collecting data through recorded interviews of one development manager and three testers in each of the three companies, analysis of the defect database, and informal communication within our research partnership with the companies. **Results:** We found that experience, reflection, motivation and personal characteristics were the top level themes. Experience related to the domain, e.g. processes of the customer, and on the other hand, specialized technical skills, e.g. performance testing, were seen more important than skills of test case design and test planning. In addition, inside reflection the ability to maintain the "big picture" and understanding the effect of defects in the production environment were seen among the characteristics of high performing testers. **Conclusion:** Future studies should look for ways how testers could acquire these characteristics or how people with the desired characterristics could be utilized in testing efficiently, for example, what are efficient ways for acquiring or utilizing domain knowledge in testing.

Categories and Subject Descriptors

D.2.9 [**Management**]: Software quality assurance (SQA)

General Terms

Management, Measurement, Reliability,Verification.

Keywords

Industrial case study, Testers, Performance, Characteristics, Traits, Competencies, Expertise

1. INTRODUCTION

For decades, there have been ideas on personality and skills affecting the efficiency and effectivity of software engineers [16]. Scientific studies on the topic, e.g. [14], have been conducted over the years that have focused to programmers and analysts. To our knowledge similar studies of software testers have not been performed. However, practitioner literature has listed respective *authors' opinions* on desirable characteristics of software tester [4, 10].

Software testing is a widely researched topic, with its own scientific journal and conference, and it is well represented in the empirical software engineering journals and conferences; however, a prominent gap exists between academic research and problems encountered by practitioners—an issue acknowledged on both sides [5]. In particular, there is limited number of industrial case studies of software testing. Martin et al. [8] argue that it is important to understand that software testing is a socio-technical rather than purely technical process greatly influenced by organizational realities and constraints. The researchers call for more work on "understanding testing as it happens". Beer and Ramler [3] found that domain knowledge was important in testing in addition to general testing knowledge. Furthermore, Itkonen et al. [7] observed techniques used by eleven experienced testers as they conducted exploratory testing. The techniques used were based on the testers' experience on the domain and in software testing.

In this paper, we define *software testers* broadly as *any employees detecting and reporting defects in running software*. Some may consider this approach unorthodox and in conflict with the more rigorous definitions of a software tester. We simply view the definition as practical: if one discovers a defect, then one has conducted testing. Our definition allows employees performing important quality assurance methods, e.g. internal alpha testing, to be considered as software testers regardless of their primary job duties. This paper is an industrial case study of understanding the characteristics of high performing software testers of three successful medium-sized software product companies.

2. METHODOLOGY

This section presents the methodology of this study. First, we present the research questions this study is about to answer. Second, we provide the case study method followed in this work. Third, we describe the case organizations. Finally, we present validity procedures and evaluation of the study.

2.1 Research question

Software testing is an area with high practical relevance and there is limited work on understanding high performing testers of the software industry. Our research question is as follows: *What are the characteristics of high performing testers?* We define high performing as effective in terms of the number of detected defects and, in addition, by the characteristics of testers seen important by managers and testers. The latter part of the definition is important as it allows inclusions of more qualitative issues, e.g. clearly written defect reports. On the other hand, relying solely on the number of detected defects could include testers that found numerous non-important bugs that are not considered worth fixing.

2.2 Case study method and interviews

This study was conducted as an *exploratory case study* [11] in three Finnish software product organizations. The case study is considered an embedded case study with three units of analysis, with each organization a unit of analysis. The research was conducted as follows. First, an initial analysis of the defect tracking database data was conducted to indentify high performing testers in terms of detected defects.

Second, to understand the characteristics of high performing testers we performed interviews of the product development manager and three high performing testers of each company. Complete interview guides are available online in Appendixes A and B of [6] and they were carried out mostly by two researchers using the interview guide approach p. 342 [9].

The first part of the manager interviews was common to all organizations. The second part of the interviews was unique for each case company and was based on the initial analysis of the defect database data that had been previously obtained from the company. The manager was inquired about his or her personal opinion as to who were the high performing testers in the company and what charactericss she/he possessed. Additionally the interviews of the managers covered issues such as testing process and defect reporting, that are outside the scope of this paper.

Three testers from each case company were selected for interviews. Selection in each case was made in co-operation with the interviewed manager based on their defect detection effectiveness and the manager's view of high performing testers. The basic idea was to interview the most effective tester from each type of testers in the company. Recent effectiveness was given more weight in order to get information from people who had current testing experience. The managers gave important additional information about the title and other work responsibilities of the testers in the organizations. Some of the high performing testers were not available for the interviews as they had left the company, or were too busy with current projects. Interviews of the testers covered such topics as the educational background and work experience, linkage of primary job to testing, conducting testing, and important characteristics of a good tester.

It should be noted that Companies B and C had no specialized testers on the payroll. Thus, the testers in the companies had other responsibilities and their respective job titles varied, e.g. customer consultant, developer, customer support, or manager. Company A had a group of specialized testers that tested the internal mainline releases. However, employees working in the customer projects that actually made the product deployments, i.e. project engineers, also carried significant testing load. Therefore, in this article the term *tester* means employees in the company who report defects of the product. The ones' selected for the interviews were effective in terms of reported defects and were seen by the managers as high performing testers.

The transcribed interviews were coded using the topics of the interview guides as preformed codes as well as codes emerging from the data; open coding in [13]. Codes were then analyzed, and similar codes were combined, called axial coding in [13]. This unification of coding was done in all cases in order to ease cross-case analysis. Coded transcriptions were then again analyzed. In the end, we had 1896 quotations with 182 codes that were grouped under 23 groups.

In addition, the researchers had other sources of information such as the defect reporting guidelines, organizational charts containing titles of employees of the companies, other research done with the companies, e.g. [15] and several informal discussions with personnel in the companies through a research partnership before and after the interviews. These additional sources of information helped us understanding the cases and interpreting their answers.

2.3 Case organization

All studied case organizations, see Table 1, were either major units of successful medium-sized software product companies (Case B and C) or consisted of the whole company (case A). Successful in this paper means financial success (data from the financial statements of 2005-2008): for majority of the years companies had profit margins of roughly 20% and during the period they had a revenue increase between 50 and 100%.

All the organizations had more than ten years of experience in their domains and had both domestic and foreign customers. The products in all organizations were relatively mature as they all had been under development ten years or more. In all cases, the customers of the studied organizations were companies in engineering, and the products were used by domain experts. Fictional but analogous software product to the ones of our cases would be software used to design airplanes. Case selection was based on accessibility through our research partnership. Information about the case organizations and the cases is summarized in table 1. In the table and in this description, some information is purposely unspecific to ensure the anonymity of the cases.

2.4 Validity and evaluation of the study

First author designed and performed all the interviews. Interview guides were reviewed by other researchers. This should ensure comparable interview data. Interviews were open and did not always follow same order of the topics, if the interviewee started to tell about issues elsewhere in the guide. This and the different phrasing of the questions might reduce the reliability and construct validity of the results [11]. Triangulation by sources [11] was used in this study in two levels. First of all there were three cases from three different companies and there were four interviewees per case. The case organizations were selected based on accessibility, but formed a good combined case in their similarity. Findings from all cases were similar and can so be seen internally consistent. All tester interviewees were top reporters and also seen high performing testers by their managers. The case descriptions were reviewed by the interviewed product development managers, also by the testers if possible, to check the interpretations done by the researcher.

Table 1. Summary of the case organizations

	Organization A	Organization B	Organization C
Personnel	>110 employees, organization = whole company	>60 in the studied organization (>300 in the whole company)	>70 in the studied organization (>100 in the whole company)
Customers	>200	>80 for the studied products	>300
Age	>10 years	>20 years	>20 years
Studied Product	-single product for engineering and billing -integrated directly into the customers' other business systems -many customization opportunities	- two products for engineering in different fields -integrated directly into the customers' other business systems -the products share a common core - some customization opportunities	-single product for engineering design -COTS type of software, i.e., not heavily integrated or customized product -product has a separate core that is also used for another product of the company

The study was done in limited context, which reduces the external validity of the results. The study consisted of three cases in three companies that had products aimed for professional engineering use. Only one of the cases used specialized testing team and also in that case project engineers had important role in testing. In all cases the application domain experts had great role in the product development and in testing.This makes it possible to predict that the results could probably be generalized to other contexts with products for domain expert users, e.g. software for airplane design. The findings cannot probably be transferred to other contexts, where the product is more general, e.g. calendar application.

3. RESULTS

This section presents the results of the study. Summary of the characteristics and the links to the individual cases are presented in

Table 2 Due to space restrictions, this paper presents only the cross case analysis but more detailed description is available [6]. We follow the model by Spencer and Spencer [12] and start with the most superficial characteristics that are the easiest to develop and move to the ones that are more difficult to develop. For example, learning how to use new software is considered superficial whereas changing one's personal characteristic like increasing patience is considered more difficult. The four themes found in the characteristics of high performing testers are experience, reflection, motivation, and personality. These themes are grounded to the data in a way that the "experience" theme covers four code groups (see Section 2.2) whereas each of the other themes covers only one. The rest of the 23 code groups link to either general information, e.g. background of the interviewee, issues not in the scope of this paper, e.g. defect reporting process, or are company specific codes. For summary of the characteristics and

3.1 Experience

Experience is generally seen to consist of knowledge and skill among other things. Most interviewees in the cases, reported that high performing testers should know the product under testing well, especially how it is supposed to be used, but also how it is implemented. Similarly, domain knowledge of customer processes and usage of the product was thought to be important. In most cases, there were limited amounts of documents available for the personnel performing the testing, or the available documentation could not always be trusted to be correct. Thus, domain knowledge was seen as necessary to understand requirements and to act as a test oracle. Knowledge in the implementation of the product was seen as useful during the testing to test boundary values and incorrect input data types. More traditional testing skills in test case design and test planning were not called for that often.

There were also several kinds of other special abilities and areas of knowledge that were considered important, in addition to the domain knowledge, e.g. testing of distributed systems, or performance testing. Thus, a high performing tester could also be a specialist in specific technical technique of testing. Several interviewees mentioned the designing and implementation of test automation as an important skill for a high performing tester. Linked to this was the characteristic of positive laziness. High performing testers would not perform the same testing task several times, at least not manually.

Good communication skills were seen as characteristics of high performing testers by the interviewees. There were two distinct aspects to this theme: first, overall communication, including the ability to express one's opinion about the issues and to criticize the product; and, secondly, the ability to write good and clear defect reports.

3.2 Reflection

Reflection can be seen as a special skill related to how well one is able to reflect oneself and the environment. Several interviewees commented on a high performing tester's ability to maintain the big picture and put the details in the correct context. Maintaining the big picture has been suggested by the practitioner literature of testers [4, 10] and studies of effective software engineers [14]. Maintaining the big picture allowed testers to focus on the most important parts of the software and understand what the user is trying to achieve. As a sort of sub-skill of maintaining the big picture, the interviewees saw that understanding the effects of defects in the production environment was important for focusing testing effort and prioritizing defects. Both of these characteristics were seen important for getting the most out of the domain specialist testers in terms of getting the right defects reported and fixed.

3.3 Motivation

Several interviewees told the researchers about the importance of motivation to test. Testers need to be motivated and have the correct attitude towards testing in order to be high performing. One of the interviewees said that a high performing tester "likes to find bugs". Several interviewees thought of testing as a tedious task but important task. It was not seen as creative. Some interviewees reported that, as the motivation varied, the efficiency of the testing also varied. Interviewees with closer customer contact reported that part of the motivation originates in fear of embarrassment if the product should fail while they were demonstrating it to customers or training users and from knowing how much it could cost the customers if it fails. It was believed that motivated testers probably use more time for testing and, tested more thoroughly.

Table 2. Characteristics of high performing testers found in different cases

Experience and Skills	Reflection	Motivation	Personal characteristics
Experience with the product (A, B C) Experience in the domain (A, B, C) Experience in programming (A, B, C) Experience in specific technical testing techniques (A, B, C) Writes good defect reports (A, B, C)	Maintaining the "big picture" (A, B, C) Understands the effects of defects in production environment (B, C) Independent and knows own skills and limits (B) Criticizes the product and process (C)	Has a mission to test "likes to find bugs" (A, B, C) Knows the importance of the testing (B, C)	Thoroughness, conscientiousness, patience or persistency (A, B, C) Accurateness (B, C) Creativeness (C)

There have been studies on motivation in software engineering, but whether it really results in top performance is not that clear [1, 2]. A couple of interviewees with more experience in testing and software development but less in the customer domains, told that there is a difference in the mindset, self image and skills between a good tester and a good software developer. And that software developer cannot well test code he has been developing himself.

3.4 Personal characteristics

The most common personal characteristics of high performing testers, as mentioned in the interviews, were thoroughness, carefulness, conscientiousness, patience, and persistency. It was noted that they should be precise and keen for details. Other characteristics mentioned were imagination, criticality in thinking, and autonomy. Some of these characteristics can be seen as slightly similar to the strict use of a methodological approach that was requested in [10]; at least, such traits can be seen as facilitating such an approach. The other personality characteristics reported by the interviewees – accuracy and creativity – are identical to those asked for by Pol et al. [10].

4. CONCLUSION

Domain knowledge and specialized technical skills are seen more important than skills in test case design and test planning. The fact that we only had three companies creating similar products, i.e. products for end-user domain experts, limits the generalizability of this result and calls need for further studies. Our finding is in contradiction with previous research [14] that did not find domain-knowledge as differentiating factor between exceptional and non-exceptional software engineers. On the other hand, our result is similar to Beer and Ramler [3] who reported need for domain experience in software testing. Although, skills of test case design and planning were seldom mentioned one could easily argue that domain knowledge is simply a prerequisite for effective test case design and that a true top tester possesses experience both in the domain and test case design. Perhaps future works of software testers should look for ways for acquiring domain knowledge effectively.

Ability to reflect is an important characteristic of high performing domain expert tester. In the interviews it appeared that domain experience is most useful when it is utilized for understanding what the user is trying achieve with the product and then to understand what parts of the software and what defects would be most crucial in the production environment.

Motivating non-specialized testers is highly important as they may view testing boring and tedious. Thus, one should look for ways on motivating the non-specialized domain expert testers. For example, could product demonstrations be used earlier in the development process as a motivating testing approach, since many relevant defcts are usually revealed when preparing and presenting demonstrations.

5. References

[1] Acuña,S.T., Gómez,M. and Juristo,N., 2009."How do personality, team processes and task characteristics relate to job satisfaction and software quality?" *Information and Software Technology,* vol. 51, no. 3, 627-639.

[2] Beecham,S., Baddoo,N., Hall,T., Robinson,H. and Sharp,H., 2007."Motivation in Software Engineering: A systematic literature review," *Information and Software Technology,* vol. 50, no. 9-10, 860-878.

[3] Beer,A. and Ramler,R., 2008, "The role of experience in software testing practice," In Proceedings of the 2008 34th Euromicro Conference Software Engineering and Advanced Applications.258-265.

[4] Black,R., 2002.*Managing the testing process: practical tools and techniques for managing hardware and software testing,* New York, New York, USA: Wiley-Interscience [Imprint]

[5] Glass,R.L., Collard,R., Bertolino,A., Bach,J. and Kaner,C., 2006."Software testing and industry needs," *Software, IEEE,* vol. 23; 23, no. 4, 55-57.

[6] Iivonen,J., *Identifying and Characterizing Highly Performing Testers – A Case Study in Three Software Product Companies,* Thesis M. Sc., Helsinki University of Technology, 2009, http://www.soberit.hut.fi/espa/publications/MSc-Thesis-Iivonen-2009-10-30.pdf

[7] Itkonen,J., Mantyla,M.V. and Lassenius,C., 2009, "How do testers do it? An exploratory study on manual testing practices," In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement.494-497.

[8] Martin,D., Rooksby,J., Rouncefield,M. and Sommerville,I., 2007, "'Good'Organisational Reasons for'Bad'Software Testing: An Ethnographic Study of Testing in a Small Software Company," In Proceedings of the 29th international conference on Software Engineering.602-611.

[9] Patton,M.Q., 1990."Qualitative evaluation and research methods,"

[10] Pol,M., Teunissen,R. and Van Veenendaal,E., 2002.*Software testing: a guide to the TMap approach,* Harlow, United Kingdom: Addison-Wesley Professional

[11] Runeson,P. and Höst,M., 2009."Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering,* vol. 14, no. 2, 131-164.

[12] Spencer,L. and Spencer,M., 1993."Competence At Work: Models for Superior Performance, NY: John Wily & Sons,"

[13] Strauss,A.L. and Corbin,J., 1990.*Basics of qualitative research: Grounded theory procedures and techniques,* Sage Newbury Park, CA

[14] Turley,R.T. and Bieman,J.M., 1995."Competencies of exceptional and nonexceptional software engineers," *The Journal of Systems & Software,* vol. 28, no. 1, 19-38.

[15] Vanhanen,J., Mäntylä,M.V. and Itkonen,J., 2009, "Lightweight Elicitation and Analysis of Software Product Quality Goals – A Multiple Industrial Case Study," In Proceedings of the third International Workshop on Software Product Management (IWSPM).42-52.

[16] Weinberg,G., M., 1998.*The Psychology of Computer Programing - Silver Aniversay Edition,* New York, USA: Dorset House Publishing