Understanding the influence of human factors on testing activities in software producing organizations

Anca Deak

Thesis for the degree of

philosophiae doctor

Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology

July 31, 2015
Abstract
The view of software testing has, over the years, evolved into a constructive one. However, while testing has received an increased focus in the software engineering field, the literature has predominantly focused on new technologies and new methods for testing and little on how testing is conducted in real-world projects. Many of the problems encountered in testing are related to both work and organisation, and improving software testing should be seen as a socio-technical challenge.

This thesis aims to shed light on how software testers are organized within the companies and how they can be motivated in their work. We explored the policies and rules conceptualized and implemented in software development projects and searched for the most desirable set of traits and skills. We wanted to see when the perceptions of software testing are formed and how they evolve during the educational and professional years. We hope that the results of this thesis will contribute to the body of knowledge regarding testing research in an industrial setting.

This overall purpose will more specifically be pursued through the following three research questions:

- **RQ1**: Which contextual factors are salient in situations that can generate strong motivation for software testers?
- **RQ2**: How is the motivation of software testers encouraged in practice?
- **RQ3**: How does working experience in industry contribute to the perception of testing activities' importance?

In this research we have used a mix of quantitative and qualitative data analysis. The quantitative data were used to construct an overview of the internal organization of testing activities and for analysing the results from alumni survey data sets. Qualitative data have been used to understand and explore the many factors we observed during the stages with focus on motivation and characteristics of testing professionals.

The main contributions are:

**C1.** Identification of challenges encountered by testers, managers and developers during software testing activities.

**C2.** A classification framework for organizing and managing testing personnel.

**C3.** A description of factors that motivate software testing personnel as well as strategies deployed by the companies for stimulating their testers. New motivational and de-motivational factors were identified within testing context.

**C4.** A set of skills and characteristics considered suitable or desirable when hiring software testers.

**C5.** A multi-faceted perspective of the perception of a testing career among students and graduates as well as factors influencing the students in pursuing a software testing career.

This thesis offers insight into how organizations decided and adopted a guideline for testing activities. Furthermore, the research provides a comprehensive description of factors that influence the software testing personnel and a new set of motivation and de-motivation factors were introduced. In addition, a multi-faceted perspective of perception of the testing career among students and graduates emerged. Lastly, the results improve our understanding of the mechanisms influencing the students for pursuing a testing career.
Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Computer and Information Science, NTNU, Trondheim, under the supervision of Professor Tor Stålhane as the main supervisor and with Professor Guttorm Sindre and Adjunct Professor Torgeir Dingsøyr as co-supervisors.

This PhD thesis is financed by an internal scholarship from the Department of Computer and Information Science and the Faculty of Information Technology, Mathematics and Electrical Engineering at NTNU. The work also included a 25% teaching duty at the Department of Computer and Information Science.
Acknowledgements

I want to thank all the people who have made this work possible.

I would like to express my gratitude to my main supervisor Tor Stålhane for his advice and support and the freedom he gave me in pursuing the research in my area of interest. Thanks to my co-supervisor, Guttorm Sindre, for spending a large amount of time with my work and for providing me with valuable feedback. Thanks to my co-supervisor Torgeir Dingsøyr for providing useful inputs for my work.

Thanks to Torstein Hjelle, Gro Alice Hamre and Vigdis Heimly for sharing ideas and being patient and tolerant office mates and friends.

I also want to thank my family and friends and, specifically to my wonderful husband Ferenc Deak for being understanding and for providing me with support during this journey.
Abstract

Preface

Acknowledgements

List of Figures

List of Tables

Abbreviations

1 Introduction

1.1 Research Questions

1.2 Research Design

1.3 Papers

1.4 Contributions

1.5 Thesis Structure

2 State of the Art

2.1 Software development and testing: traditional and agile approach

2.2 Studies on testing practice

2.3 Concept of motivation

2.4 Studies on motivation in software engineering

3 Context and Research Design

3.1 Research methods

Surveys

Interviews

Analysis of qualitative data

3.2 Research Process

Stage 1

Stage 2

Stage 3

3.3 Validity Threats

4 Results

4.1 F1 Testing Team Structure

4.2 F2 Software tester concept

4.3 F3 Observation on testing management

4.4 F4 Automation

4.5 F5 Characteristics of a Software Tester

4.6 F6 Motivational and de-motivational factors for software testing personnel

4.7 F7 strategies deployed by the companies for stimulating their testing personnel
List of Figures
Figure 1: Research evolution.................................................................................................................. 3
Figure 2: Research design.......................................................................................................................... 4

List of Tables
Table 1: Correlations between research questions, contributions and papers ......................... 8
Table 2: Mapping between research questions, findings and papers ........................................... 29
Table 3: Testing activities - organizational categories ................................................................. 30
Table 4: Characteristics of software testers ......................................................................................... 34
Table 5: Relationships between codes and concepts for negative factors ........................................ 35
Table 6: Relationships between codes and concepts for positives factors ........................................ 38
Table 7: Positive factors for traditional and agile testers grouped by methodology ...................... 40
Table 8: Negative factors for traditional and agile testers grouped by methodology ...................... 40
Table 9: Interest in pursuing a testing career ...................................................................................... 43
Table 10: Negative aspects in pursuing a testing career ................................................................. 45
Table 11: Positives aspects of working in testing .............................................................................. 46
Table 12: Motivational factors for pursuing a testing career ......................................................... 47
Table 13: Mapping between papers, contributions and research questions ................................... 49

Abbreviations
NASA  National Aeronautics and Space Administration
QA    Quality Assurance
QC    Quality Control
RAD   Rapid Application Development
SE    Software Engineering
SLC   Software life cycle
ST    System Testing
TDD   Test Driven Development
UAT   User Acceptance Testing
V&V   Verification and validation
XP    eXtreme Programming
1 Introduction

Software testing is the main process for controlling the quality of a developed software product and as for any human-based activity; the outcome of the process is depended on human factors. While observing the management of conflicts in the world of software testing, Cohen et al. [1] concludes that the result of testing "ultimately depends on the interpersonal interactions of the people producing the software". The testing teams or the personnel involved in these activities encounter challenges in their daily work which can hinder their motivation. The focus of this thesis is to improve the understanding of the human factors which influence testing related personnel and impact their activities and their involvement in the testing field.

When forming a well functioning team, motivation, skill, satisfaction, values, and personality are factors which need to be taken into consideration. Motivation has been repeatedly cited as an important factor in productivity, quality and the successful delivery of a project within budget and time constraints [2] with several motivation theories emphasizing the importance of employee motivation, e.g. Herzberg [3] and Mayo [4]. Myers' work has pointed out the importance of psychology, intellectual manageability, economics and discipline as strong determinants for successful software testing. Despite his strong argument that the success of software testing has little to do with purely technical considerations, the nontechnical concerns have been neglected in favour of technological issues [5].

While software testing is a widely researched topic, and although there is an extensive literature on software testing as a generic activity, there is still a demand for more information on how software testing is performed in the real world [6]. Testing practices are built mostly on the experience gained in the domain, rather than on a rigorous and systematic approach [7], and although the testing problem tends to be perceived as mostly technical, research has documented a high impact of organizational factors in the testing process [8]. In order to improve the testing activities and tester satisfaction, it is important for researchers to improve their understanding of practices and challenges of testing in the software development industry [6]. Briand and Labiche's study on software testing techniques concluded that observing the activities of professional testers will enable us to retrieve the most applicable results [9].

The gap between software education and the industry needs was identified and reinforced in a series of worldwide conducted studies [10], [11], [12] and [13]. While
concomitant, the lack of professional expertise is considered to be the dominant factor in impeding the introduction and the adoption of systematic testing activities in software development companies. At the same time, the desired industry profiles for software professionals are not always consistent. Industry is often ambiguous about the responsibilities of testing positions, suggesting a wide variety of thoughts and perceptions surrounding the software tester role. The importance of establishing clear roles in software engineering was recognized from the beginnings of software development and reinforced in consecutive studies – see [14], [15]. Although there is a general awareness of the necessity of improved quality of software testing, there is low level of test maturity in most software development organizations [16].

This thesis will explore the evolution of views on testing from a broader perspective, starting from the young people which are at the beginning of their career, to those practitioners who have achieved a rich industrial experience.

1.1 Research Questions

The purpose of this PhD project is to observe and collect information about the challenges faced in the testing practice by testers and to propose empirically based improvements for the identified issues. This overall purpose will more specifically be pursued through the following three research questions:

- **RQ1** What contextual factors are salient in situations that can generate strong motivation for software testers?
- **RQ2** How is the motivation of software testers encouraged in practice?
- **RQ3** How does working experience in industry contribute to the perception of testing activities' importance?

Previous research indicates that motivation is a challenge for testers, and that this can cause poor quality software products in at least two different ways: one is that workers with low motivation perform more poorly, and workers with low motivation or job satisfaction will more often quit their jobs, causing more frequent changes of personnel [3], [4]. The situations will result in a bigger overhead in training new people and most likely having less experienced testers who are also less effective than the ones with more experience.

1.2 Research Design

The study was planned in several stages as illustrated in Figure 1, where each stage had studies conducted sequentially in time, building on the results from previous stages. For the first stage we constructed an overview of how companies organize testing activities by conducting a survey on the internal organization of the testers, working methodologies and the use of automation within the software companies who agreed to support us in our research. In order to provide an answer for RQ1 we need to understand the internal dynamics of a testing team or testing personnel and to construct an accurate overview of their daily activities.

A population of 19 companies participated at the survey. These represented various domains of business with products in different ranges of risks, and different sizes, from small companies to large international corporations. The first set of analysis indicated a
low preference for having a separate testing department or a dedicated testing staff as described in P1.

**Figure 1: Research evolution**

The second stage of the RQ1 study used a qualitative survey combined with semi-structured interviews which allowed us to retrieve more information about testing management and testing activities, and helped in building an answer to RQ2. For improving the knowledge about the human factors, we discussed with testing professionals about their career path, their choices which lead to the current testing position, the job satisfaction and the motivation behind a testing career choice, P5. We wanted to retrieve a description of the characteristics which are considered as the most suited or desirable when hiring software testers, P4. The data collection for the RQ2 stage was based on these interviews and overlapped with the data collection of the second stage of the RQ1 study.

To answer RQ3, we decided to first make an assessment of the perception and expectation from a testing career path among our students; hence we initiated a study in which we involved students from the third and fourth years. The results of the study, described in P2, indicated a low interest for a future career in testing with many of the participants showing little interest or having a negative attitude towards testing as a profession. The perception of testing activities seems to change once the students join the industry field. The results from alumni survey data, P3, shows that Software Testing scored high both when asking which topics they had needed knowledge about in their jobs, and which topics should be prioritized higher in education.

For the last stage of our research we conducted in-depth interviews with testing professionals, managers and developers over a two years period. In the study presented in P4 we created a profile with the qualities which were observed to be the best suited for a tester based on senior testers and direct managers’ point of view. We observed the motivational and de-motivational factors influencing the testers from a working methodology perspective, as described in P5. The last paper included in the thesis focuses on motivational factors and social aspects influencing the activities of the people involved with testing and provide a more detailed view on how professional testers are perceived and encouraged in a software development company. The detail of the research design is depicted in Figure 2, showing the connections between the research questions, the methodology and the included papers.
1.3 Papers

Six papers are included in this thesis, from which five have been published and one is under review for a journal. The relevance of the papers to this thesis and my contribution to each paper are described next.


The scope of the study is to explore how testing activities are organized in Norwegian software companies, and factors that influence the creation of a testing department, or encourage the investment in personnel dedicated to testing. The research was performed using a combination of qualitative and quantitative methods and started with an initial survey which comprised closed-ended as well as open ended questions. The next stage included interviews and an additional round of open ended questions. We wanted to retrieve the factors that influence and determine a certain organization of the testers in software producing companies.

**Relevance to this thesis:** This study provided an overview of the internal organization of testing activities in Norwegian software companies and resulted in four organizational categories in which testing activities are organized, combined with observations on the correlation between testing automation activities and testing team structure.

**My contribution:** As the principal author I was in charge of the study design. I conducted the data collection, data analysis and paper writing. Professor
Stålhane helped out with the data analysis, provided me with his feedback, and did a proofreading of the final paper.


In this paper we describe the result of a study on the evaluation of the interest and desire to work in software testing among engineering and computer science students from the Norwegian University of Science and Technology, and a survey of the factors motivating and de-motivating the students in engaging in a software testing career. The main goal was to investigate at which stage of the studies the opinion on a software testing career is formed, the evolution of this opinion during the study years, and the impact of dedicated testing course towards a software testing career. The results show a lack of interest and a tendency to associate testing activities with boredom for the students. Around half of the students who attended the testing course provided for the master level were interested in pursuing a testing career and their main request was for practical experience, combined with presentation of more innovative methods and tools for testing. Another essential result was the lack of information about the possibility of a testing career and the responsibilities and tasks of a tester. These results indicate the possibility of improving the perception of a testing career by changing the curriculum and reviewing some of the curriculum parts with testing professionals with the scope to retrieve the educational topics relevant to testing field of expertise.

Relevance to this thesis: This study provided an image of how testing jobs are perceived among master students likely to become future software professionals and the factors which influence the students when considering a testing career.

My contribution: As the leading author I contributed the research design, data collection, data analysis and paper writing. I was also responsible for analysing the data collected, by coding and re-coding it in NVivo. Professor Stålhane helped out with the data collection, provided me with his feedback, and did a proofreading of the final paper. Dr. Cruzes helped out with the data analysis, gave me her feedback, and proofread the final paper. Discussing the results and writing the paper was done collaboratively.

P3 Deak, A., Sindre, G. "Analyzing the importance of teaching about testing from alumni survey data" NIK Norsk Informatikkonferanse 2013 s. 100-109

This paper presents the result of two web based surveys, conducted by the Department of Computer Science (IDI) of the Norwegian University of Science and Technology (NTNU) among IDI alumni students in 2007 and 2011. The focus of the surveys was to identify the educational topics relevant to the participants' field of expertise based on the work experience accumulated after graduation and on the emphasis the IDI department is assigning on these topics. The data provided by this survey was analyzed from the perspective of software testing. Results shows that Software Testing scored high in the Topics ranked by perceived importance list as well as in Topics which should be higher prioritized list and that there is a need to increase its emphasis in the educational programs.
Relevance to this thesis: The study shows that the importance of software testing activities is perceived as higher once a person has acquired industry experience. These results can also lead to a better understanding on how motivational factors evolve over time.

My contribution: I wrote this paper in collaboration with Professor Sindre based on the data collected by him through the Alumni surveys. Discussing the results of the analysis and writing of the paper was done collaboratively.


We conducted a qualitative survey among four software development companies using interviews as the main method for data collection and the thematic-analysis approach for analysing the data. Our findings describe the characteristics which are perceived as most valuable for a tester based on the working experience accumulated by the participating testers and test managers. The study findings suggest that there are differences between the characteristics of a software engineer and a software tester, with communication and possessing an IT background being cited as the most frequently favoured characteristics for ensuring a high performing tester.

Relevance to this thesis: The aim of this study was to explore and identify the most favourable perceived characteristics of software testers and contrast them with the group of existing software engineers' characteristics provided by prior research.

My contribution: All work was done by the author.


This paper reports a qualitative study of motivational factors for software testers in pursuing their career path. In particular, the study attempted to provide evidence-based answers to the research question "How do motivational and de-motivational factors for software testers differ in agile environments versus traditional type environments?" The study interviewed 26 participants from six companies. Half of the participants worked in agile settings, the other half in traditional, plan-driven settings. Most participants were test managers and testers, while four of them were developers. The interview questions were open-ended and the process was semi-structured.

During the data analysis of the qualitative data suggested several codes emerged, which were linked by concepts (the factors) identified in two previous studies referenced in the paper. The concepts were divided into those considered positive (fostering motivation) and those considered negative (decreasing motivation). Using a qualitative survey among software development companies we retrieved information about the difference between the traditional and agile testers, which will help the companies in their recruiting processes and in the transition from traditional to agile development within a company.
**Relevance to this thesis:** This study observed and described the motivational and de-motivational factors influencing the testers from traditional and agile working methodology perspective.

**My contribution:** All work was done by the author.

**P6** Deak, A., Stålhane, T., Sindre G. "Challenges and strategies for motivating software testing personnel" Accepted at Information and Software Technology journal

This paper reports a qualitative study of motivational factors based on the entire data set collect in a two years period and focuses on motivational factors and social aspects influencing the activities of the people involved with testing and provide a more detailed view on how testers are perceived and encouraged in the company, by aggregating the motivational strategies developed internally.

**Relevance to this thesis:** The last paper included in the thesis provides in depth analysis on the factors influencing the people involved with testing and construct the perceived image of testing as seen from outside perspective.

**My contribution:** As the leading author I contributed with designing and conducting the qualitative study, gathering and analyzing data and writing the paper, Professor Stålhane and Professor Sindre provided me with their feedback, and preformed proof readings of the final paper.

Investing in software testing personnel: motivational factors and internal strategies

**1.4 Contributions**

We are confident that the results of this thesis will improve the knowledge about testing as a vital part of a software development process and will be a stepping stone in informing academia and industry on how to improve testers' motivation and job satisfaction.

C1. Identification of challenges encountered during software testing activities. (P1, P4, P5, P6)

C2. A classification framework for organizing and managing testing personnel. (P1, P4)

C3. A description of factors that motivates software testing personnel and strategies deployed by the companies for stimulating their testers. We identified new motivational and de-motivational factors within testing context (P1, P5, P6)

C4. The characteristics and skills considered most suitable or desirable when hiring software testers (P1, P4)

C5. A multi-faceted perspective of the perception of a testing career among students and graduates and factors influencing the students in pursuing a software testing career (P2, P3)
Table 1: Correlations between research questions, contributions and papers

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Contribution</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>C1, C2</td>
<td>P1, P2, P5, P6</td>
</tr>
<tr>
<td>RQ2</td>
<td>C3, C4</td>
<td>P1, P4, P5, P6</td>
</tr>
<tr>
<td>RQ3</td>
<td>C2, C5</td>
<td>P2, P3</td>
</tr>
</tbody>
</table>

1.5 Thesis Structure

The rest of the thesis is organized as follows:

Chapter 2 briefly presents the field of software testing and the state of the art and gives an overview of research performed on testing practices. In addition, it outlines the research done on motivational factors in the field of software engineering. Finally, the challenges faced in the context of software testing are presented.

Chapter 3 presents an overview of research methods and metrics, and challenges faced when selecting research methods.

Chapter 4 synthesizes the main results of the work. All the contributions from papers P1 to P6 are presented here.

Chapter 5 discusses the relationship between research questions, papers, contributions and their evaluations. Reflections on the research context are provided towards the end of this section.

Chapter 6 sums up the main findings from the discussion, limitations of the thesis, and proposes areas for focus in future work.

Appendix A contains the six papers included in this thesis.
2 State of the Art

The view of software testing has, over the later years, evolved into a constructive one. Previously testing was seen as an activity starting only after the coding phase was completed, with the sole purpose of detecting failures in the runnable code. In recent years, a more constructive view of testing has emerged, and testers are involved from the early stages of the software requirements process and throughout the entire development and maintenance life cycle. Software testing can be seen as a mean for evaluating the features and providing information about the functionality and quality attributes of a software item.

We are aware that a software product can still contain faults, even after completion of an extensive testing activity as expressed by Dijkstra's famous aphorism that “program testing can be used to show the presence of bugs, but never to show their absence” [5]. It is perhaps obvious but worth recognizing that complete testing is not feasible in software products of realistic industrial size. Because of this, the testing strategy can sometimes be based on risk based testing, i.e. you perform a risk analysis on the system in use, identify the errors that are most dangerous or costly and test these things first or most thoroughly.

An introduction to the software testing field and the main approaches used during this process are presented in Section 2.1, while Section 2.2 describes the research literature on testing practices. The concept of motivation and the research on classic motivational theory are provided in Section 2.3 and the state-of-the-art research on motivation studies in software engineering field are provided in Section 2.4.

2.1 Software development and testing: traditional and agile approach

Software testing is a process through which the functionality of a software program is assessed during and after the development process. According to IEEE 829 [17], Software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item. The investigation conducted for the assessment will focus both on verification and validation.
During the verification, the testers ensure that software correctly implements a specific function. In other words, it answers the question: are we building the product right? Does the product satisfy the requirements? Validation ensures that the software that has been built satisfy customer's requirements. Validation answers the question: are we building the right product? Do the requirements correspond to the users' needs?

We consider a tester to be a skilled professional who is involved in testing a component or a system and testing team as the team which focuses on testing activities. When referring to testing personnel we will consider the grouping of the testing staff. Integration testing, system testing and acceptance testing are considered the domains of the testers while unit testing falls under the developers' responsibility.

A developer is an individual that builds and create software and applications. He or she writes debugs and executes the source code of a software application. In this thesis we consider that a developer does not perform any testing outside unit testing, unless as a part of a specific testing assignment. In case a developer is involved in testing activities outside unit testing, we will consider him or her as being in a tester' role.

Traditional project management is based on a rational allocation, to carefully define and determine the future work goals, carefully plan the necessary work, and then implement the plan step by step and measure success in relation to the overseen advanced target and plans, rather than to the benefit of those actually produced results. One of the first models to be designed was the waterfall model (W-model), also known as the cascade model. This model was first presented by Benington [18] in 1956 and according to this framework, the software development lifecycle was constructed from a sequential set of stages: starting with a feasibility study at the top level and finishing with the implementation. One trait of this model was the placing of testing towards the end of the project life cycle, which leads to defect being discovered close to the production deployment stage. The first model was followed up with a modified version by Winston Royce [2] in 1970 which appended feedback loops from each stage to its anterior one, and an additional complex feedback loops between testing, design and requirements.

The waterfall model is the recommended framework for creating software which provides back-end functionality, meaning a software whose main scope is to provide a service for other applications [19].

The V-model, is a product-development process originally developed in Germany for government defence projects and used by NASA [20]. It was first presented in 1991 and is a variation of the waterfall model in a V shape folded at the coding level. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing, illustrating that testing activities should planned and designed as early as possible in the lifecycle. A more complex system, which can contain a high number of stages, can be represented by a deeper V shape, making it suitable for large projects which can include several subsystems and 3rd party systems.

As a paradigm for project management in general, this traditional approach was criticized by scholars such as Engwall et al. [21] and Hälgren et al. [22]. They point out the weaknesses in a plan established at the beginning of the project, when the knowledge of future work is at a minimum, and the target is fixed a long time before the results will be useful, instead of having a flexible target as the needs change over time.
as the project proceed. Success was defined in relation to the original target and plan, not in relation to the benefits that the results of the projects will provide in the end. Project tasks are characterized by both ambiguity and uncertainty [22], which implies complexity that it often is infeasible to foresee before implementing the project. An underlying implication of this is that the stakeholders, who lead and implement the project, are trying too hard to follow the original agreed plan rather than on creating useful results.

An alternative to plan everything and then to proceed, is to work evolutionarily, using iterations. This means that the IT system is developed little by little and that interim results and lessons learned are evaluated before proceeding, and then adjust both the content and the timetable. Iterative system development was been tried already in the 1960s, at NASA [23]. This meant that the system was not described in advance by comprehensive documentation in detail, without developers and clients consulting on the system's details throughout the project but was instead based on the evaluation of the achieved intermediate results. Another objection to the traditional waterfall thinking is more about issues related to the field than on the structuring of the work itself.

Developed mostly by James Martin in 1991, the Rapid Application Development (RAD) methodology [24] uses a large number of iterations, with every iteration being a complete development cycle. At the end of each iteration, a complete executable product is released. The iteration product is a subset of the complete desired product and it will be increased from iteration to iteration until the final product is released. These frequent releases are allowing the customers to receive concrete results and provide feedback regarding the product; the design and the requirements which the product is suppose to satisfy.

From the several approaches aligned under the RAD umbrella, XP and Scrum, both examples of agile software development approaches, are getting popular. Agile development can be described as a group of software development methodologies based on iterative and incremental development, where requirements and proposed solutions evolve thorough collaboration between self-organizing and cross functional teams. In the agile philosophy testing should take place at every stage. The term "agile software development" was created by the Agile Manifesto [25]. It was an attempt by the leaders of the agile movement to formulate a common basis for the above mentioned and other similar methods.

In an agile environment, testing is often included within development in the form of Test Driven Development (TDD), which is a programming technique that promotes code development by repeating short cycles, which combine Test-first development, whose consequences on productivity and errors are presented in [26], with refactoring. The TDD method involves four steps:

1. Write a test for an unimplemented functionality or behaviour;

2. Supply the minimal amount of code to pass the test;

3. Refactor the code;

4. Check that all tests are still passing after the changes were done.
TDD is not just a method of testing, but also a method for software design, which changes the testing last approach to a testing first approach. By developing only enough code to pass the tests and then using the refactoring as an improvement method for the design quality, TDD leads to better code quality and improves the confidence in the code [27], as well as increasing the productivity.

In an agile environment there would be more emphasis on collaboration and face to face interaction [26]. The testers will be involved earlier into the development process and the development team will have be coding to a test, rather than code first and then create a test plan, which is what tends to occur in traditional environments [27]. Members of an agile team are expected to be cross functional so they may have to write code or do requirements gathering, and work closely with the customer[28]. The people from an agile team do not separate in a QA/Testing and a Development group, they will all be organized as team and a person from the team involved in the testing activities will be seen as "a team member who has most testing experience" rather than "a tester".

If in previous times testing was seen as an activity starting only after the coding phase was completed with the purpose of detecting bugs. However, in modern development practices testers are involved throughout the entire development and maintenance life cycle as soon as the unit testing is finished. Integration testing, system testing and acceptance testing are still the domains of the testers.

In the requirement stage, software testers can check if the requirements are according to the client’s wishes, hence validating them. Another possible contribution of a tester in the requirements stage is to check if the requirements have been written in a testable manner (i.e., if it would be possible to write a test for the requirement).

During the design phase, the testers can verify if the design document covers all the requirements and review the design document from the architecture perspective. It is worth mentioning that although a software tester could perform the task in question, one could also imagine it performed by requirements analysts with no specific expertise in testing, but since the focus of this research is on testing and testers we will concentrate on their tasks, responsibilities and skills.

In the coding phase the testing team can execute test cases, as well as generating structural and functional testing data. In the testing stage, system test cases can be run to verify whether the system operates according to the stated requirements. Once the product reaches the maintenance stage, the tester can retest new fixes and patches and afterwards follow up with the regression testing to ensure that the new changes do not impact functionality in an unintended manner.

### 2.2 Studies on testing practice

Software testing and industrial surveys of testing practices such as [16] and [29] have been central themes in the specialized literature. Brain and Labiche [9] have emphasized the importance of research on testing in an industrial setting, by arguing that the human influence and experience are important factors to be considered when performing testing related research and that the most applicable results are the ones obtained by observing professional testers at work.
A certain ad-hoc practice was underlined in [29], while the importance of experience and domain knowledge in testing was emphasized by Beer and Ramler [30]. Their multiple-case study, covering three industrial software projects, classified two categories of experience: experience in testing and experience with the product domain. Having a degree of experience in software testing proved to be useful for those involved in general management of the testing and particularly for those working with test automation. Product domain knowledge also proved valuable when working with test case design, planning regression testing and requirements engineering. These results reinforce Turley and Bieman's conclusion that experience is a valued asset for software engineers [31], and those of the ethnographic study conducted by Martin et al. [32] on testing processes and practices in a small start-up company. The study conducted by Martin et al. [32], which focused on integration and acceptance testing done in the company, showed that testers working in contexts where requirements were not defined in detail and without any strict processes, needed to understand the business and experience in the domain and techniques that were used to test the product. In addition, testers were also required to possess good skills in test automation.

The perceptions of software testing were investigated by Causevic et al. [33] and by Shah et al. [34]. The survey conducted by Causevic et al. [33], which uses both qualitative and quantitative methods, organized the results into four distinct categories: safety-criticality, agility, distribution of development and application domain. Their findings revealed the discrepancies observed between the current practices and the perceptions of respondents which could prove beneficial in shaping future research on software testing. One notable result from the quantitative analysis on satisfaction level of practitioners is related to Test Driven Development (TDD) [35], which registered the most significant difference between the preferred practice and the current practice.

Among the findings of the empirical study conducted by Shah et al. [34], is the enthusiasm showed by the testers about their job and their positive attitude toward testing. This is the opposite of the common attitude towards testing, namely that a software development position is preferred over a testing position. A desire for innovation and a high value among the testers is also observed in the same study. In addition, this study shows that the quality of testing is affected by the motivation of the testers, and it emphasizes the need for appreciating testers' efforts. Taipale and Smolander conducted a qualitative study [36], which explored the software-testing practices and suggested improvements in this process based on the knowledge acquired during their study. Their improvement proposition includes adjusting testing according to the business orientation of the company, enhanced testability of the software components, efficient communication, early involvement of testers and increased interaction between developers and testers.

Dingsøyr, Dybå and Abrahamsson [37] conducted a research survey which concluded that agile methods largely reside on convincing arguments about perceived advantages rather than on empirical evidence of such advantages. Dybå and Dingsøyr [38] performed a systematic review of existing empirical studies of agile software development, and identified the need to increase the quality of studies as one of the major research challenges. From an initial sample of 270 articles, only 33 primary and 3 secondary studies, 13% of all articles, fulfilled the 11 criteria for quality which was based on terms of rigor, credibility and relevance. Furthermore, analysis of these studies
revealed that “methods were not well described, issues of bias, validity, and reliability were not always addressed; and methods of data collection and analysis were often not explained well” [38] (p.842). In a more recent study, examining a decade of agile methodologies Dingsøyr et al. [39] have analyzed the progress for research on agile software development and acknowledged the increase in the number of studies in the recent years, although emphasising that quality remains one of the major research challenges.

2.3 Concept of motivation

The term motivation derives from the Latin word "motivus" which corresponds to "causing movement" [40] (p. 18). According to Von Rosenstiel, motivation originates from specific situations in which a person notices incentives that triggers certain motives [41] (p. 217). The relationship between basic psychological needs and different kinds of motivation, as well as the relationship between different types of motivation and different kinds of performance are discussed further in this section. According to Ryan and Deci, motivation " concerns energy, direction, persistence and equifinality - all aspects of activation and intention" [42] (p. 69).

When referring to motivation it is sometimes necessary to explain and understand the differences between needs, drive, motivation and motives, e.g. Toates, [43], Deci and Ryan [44]. People have physical and psychological needs. The notion of drive or desire usually refers to the motivation linked to physiological needs, such as need of food, water, sleep, warmth and reproduction. Drive affect motivation in such a way that a physiological imbalance leads to motivation for activities which are likely to restore the balance. When the need is satisfied, the motivation will decrease. The rather general term needs is also used in conjunction with motivation related to the psychological needs [45].

Motivation triggered by physiological needs decreases when the individuals reach the satisfaction of their needs, while motivation triggered by psychological needs does not decrease when the individuals reach the gratification, such as experiencing unity or autonomy. Instead, the latter functions as a motivational factor, so that individuals seek out and stay in situations and activities where they get their psychological needs satisfied [44]. To exemplify, an individual may seek out and spend a lot of time in groups where he or she is experiencing togetherness. The motivation to be with the group does not decrease when the need is satisfied, as it does when a hungry person has eaten. Another way to put it is that people stay in the situation as long as their psychological needs are nourished.

This difference proves to be paramount for how the individual reacts in situations where obstacles are impeding the satisfaction of needs. Whenever there is an imbalance in relation to physiological needs (sleep), motivation increases (until the individual manages to restore balance). If an individual is unable to satisfy a psychological need (e.g., self-development and integrity), a defence mechanism may be developed instead, by relocating attention and effort to another area [44]. The motivation also is affected by social and other environmental factors [46].
Classic theories of motivation
The following examples of classic motivational theories illustrate the great research interest in the phenomenon of motivation for more than a hundred years.

**Behaviourism (Stimulus-Response Theory)**
A basic position of behaviourism is that the individual can only be viewed from outside. The phenomenon that refers to the individual's inner life, as "consciousness", goes either to define or to understand human behaviour, since it's just human behaviour that can be studied [47]. Behaviourism believes that learning is the result of individual experience and that behaviour has consequences. If the consequence is perceived as desirable it will increase the likelihood that the behaviour will be repeated. Behaviourism has provided important knowledge of how behaviour is affected by the consequences in the form of rewards over punishments (positive and negative reinforcement), as well as how the motivation for a certain behaviour is influenced by rewards strength, frequency, probability, temporal proximity between behaviour and consequence, and more [48].

**Maslow’s Theory of Needs**
Maslow proposes a theory in which motivation is based on assumptions about a small number of inherent general human needs. If these needs are not gratified then motivation to act to satisfy them is created [49]. The needs are arranged hierarchically where physical needs are at the bottom of the hierarchy and self-actualization at the top. Physical needs are more basic than others and therefore dominate motivation until they are satisfied. Only then can higher-lying needs dominate the individual's motivation and thus his behaviour. As needs are satisfied, new ones emerge to motivate behaviour.

**McClelland’s Needs Theory**
In contrast to Maslow hierarchy McClelland's Needs Theory [50] involves three motivational needs: perform ("achievement"), to exert power and influence ("power / influence"), and to experience connectedness ("affiliation"). The balance between the three needs of a particular individual decides how he will perceive and react in various situations. People biased to affiliation will be more objective, as this will increase their opportunity for bonding with the greatest number of people. Those with strong achievement needs tend to constantly challenge themselves and then they also require feedback on their achievements. People from this category find security and financial rewards less motivating than responsibilities and feedback. People with strong authority needs will be motivated by opportunities to conduct and organize others.

**Goal-Setting Theory**
Locke's Goal-Setting Theory [51] shows that in order for a case to serve as a motivating factor the individual must accept the goal. The individual increases his performance if the target is so specific that individuals can pursue their gradual progress and measure their effectiveness against a desired level.

**Equity Theory**
Adams Equity Theory [52] is about motivation in relation to perceived justice or injustice. The motivation will decrease if the individual feels unfairly treated in the situation relative to others. The theory is based on a set of inputs and outputs that must
be in balance to make people feel “equitable.” The inputs that people bring into a work context include experience, education, skills and seniority, which should be matched by outputs gained, such as, salary, recognition or opportunity for achievement.

*Expectancy Theory*
Expectancy Theory focuses on the cognitive processes that influence motivation, especially how the individual's interpretation of the situation affects the motivation. According to Vroom [53] three factors are important:  
- "Instrumentality: Performance to Outcome": To what extent the individual perceives a certain action to be relevant for achieving the desired result  
- "Expectancy: Effort to Performance": To what extent the individual expects to be able to perform the actions required to achieve the desired result.  
- "Valence": To what extent the reward is coveted by the individual, i.e. the result is seen as valuable.

*Job Characteristics Theory*
Job Characteristics Theory (JCT) [54] indicates that five factors together lead to meaningfulness of work: skill variety, task identity, task significance, autonomy and feedback. Skill variety refers to the degree to which the work involves variation between different kinds of activities. Task identity is related to the degree to which tasks constitute an identifiable whole, e.g., something from beginning to end with a clear end result, while task significance refers to the degree to which the work has impact on the lives or work within or outside the organization. The fourth factor, autonomy, affects the perception of responsibility for the end result, while the fifth factor, feedback, concerns the workers' access to informative feedback about the effectiveness of their performance. This way, they have better overall knowledge of the impact of their work activities, and what specific actions needs to be taken in order to improve their productivity. The average of the first three factors "multiplied" by the fourth and the fifth constitute the so-called work motivation potential.

*Herzberg: Motivation-Hygiene Theory*
Herzberg identified the types of job related factors that influence employee motivation (called attitude) to perform well [55], [3]. Dissatisfaction factors (also called hygiene factors) are a group of factors that can cause negative attitude. These include unfair rules, poor physical working conditions and poor relationship with supervisors. The opposite, e.g., rules that are perceived as fair, good physical working conditions, and good relationships with supervisors, do not lead to particularly positive job attitudes, but at best to the absence of dissatisfaction, i.e., a relatively neutral position. The attitude that will make the effort to do a good job is instead linked to the presence of another group of factors, such as responsibility, recognition, promotion and duties perceived as interesting. Herzberg called these motivational factors. Herzberg noted that motivational factors were primarily related to "the actual job", i.e., the work itself and that which is directly linked to the tasks, while the hygiene factors are more focused on "the job situation" [55].

The Motivation-hygiene theory claims that a person's motivation is significantly dependent on specific characteristic of a job and on the conditions in which it has to be
performed [3]. Herzberg's theory classifies motivation factors into extrinsic and intrinsic factors. The distinction between these two will be explained in the next paragraph.

In some situations, individual motivation is directed toward a certain goal and a particular activity is the necessary mean to achieve an individual and desirable results. This kind of motivation is termed extrinsic motivation. A child who performs a certain task for receiving an award, or an adult taking a job he dislikes because of financial benefits associated with that job. Extrinsic motivation means that the activity is necessary to achieve some desirable result, for instance material gains (e.g., salary, bonus) or increased status.

In other type of situations individual motivation is directed toward an activity that can provide an experience of the individual's own abilities. Examples of the latter are the child who plays a game, or an adult who solves crossword puzzles because they find the challenge fun and interesting. The reward lies in enjoying the activity itself; the instrumental outcome function only as an informative reference point for your own ability. This kind of motivation is termed intrinsic motivation.

Intrinsic motivation was described by Deci and Ryan as "the inherent tendency to seek out novelty and challenges, to extend and exercise one's capacities, to explore, and to learn" [44] (p.70). When people describe their experience of intrinsic motivation, they use emotionally charged words such as: interesting, exciting, exhilarating, energetic and enjoyable [45]. It is the emotional experience of the activity itself that is the reward, and what creates motivation. The experience of intrinsic motivation can be strong. Csíkszentmihályi has introduced the concept of "flow" for intense experiences in situations where the individual is completely absorbed by the targeted activity [56].

Csíkszentmihályi's concept of flow contains some, if not all, of the following components: the activity has a clear goal, it is close to the limit of what the individual can produce, and the individual can determine if the activity leads to the target or not. The individuals are completely concentrated on the task with no place for distracting thoughts and feelings, and the individuals experience full control of the situation. An important component of these flow stories is the experience of having forgotten about oneself and lost sense of time and space for the duration of the experience [56].

2.4 Studies on motivation in software engineering

The quality of a software product development is affected by the performance of the project participants, which depends on a number of simultaneously acting conditions, e.g., participants’ experience and domain knowledge, the historical context with other, past projects and events in the organization, participants' personalities, their mutual relations, status and employment conditions in the organization and the physical conditions in terms of premises and utilities [22]. But the factor that has the greatest impact on the process is the individual's motivation for the task. The individual's motivation is what ultimately determines "the difference between what a person can do and what that person will do" [57]. The motivation is in turn affected both by the success of a process or project, and by the individual's social environment, such as the work environment that is shaped by organizational culture, socializing standards, the physical workplace and interpersonal relationships with managers and colleagues.
Theory of motivation has been little used in research on software development. The software engineering literature has largely missed the opportunity to employ theories from psychology and cognitive science when dealing with human aspects. Motivation in software engineering was the scope of a systematic literature review conducted by Beecham et al. [58], in which 92 papers published between 1980 and June 2006 were analyzed. The result of this study provided 16 characteristics of the software engineer together with 21 motivators and 15 de-motivators identified in the literature. Another, subsequent study by Franca et al. [59] extended and updated this results by analyzing 53 papers published from March 2006 to August 2010. As a result, another 8 additional motivators were identified: team quality, creativity/innovation, fun, professionalism, having an ideology, non-financial benefits, penalty policies and good relationship with users/customer, as well as a new de-motivator: task complexity. The study also shows that two of the motivators discovered in [58] were no longer present: appropriate working conditions and sufficient resources. The change noticed in the motivators and de-motivators also illustrates the evolving nature in the motivation of software engineers and this is expected to change even more as the software engineering field is evolving further.

Most of the studies involved in these two literature reviews were quantitative survey studies which provided important insights into characterizing the factors and results related to motivation. One limitation of the mentioned studies, which we need to consider, is that the majority of the studies are referring to the job itself as being the main motivational factor. Since the title of software engineer can contain multiple roles and responsibilities which can greatly vary from one position to another, more information about the job that motivates the software engineer is required.

Based on the results presented in the systematic literature review by Beecham et al. [58], the same group of authors have studied several models of motivation and proposed a new model which was compared with the previous ones and then refined based on this comparison in Sharp et al. [60]. In an analysis of the motivation theory used in the identified publications, Hall et al. [61] found that studies of software engineers’ motivations are not explicitly underpinned by reference to the eight "classical" motivation theories from the social sciences. By far the most common is the Hackman and Oldham's Job Characteristic Theory [54] and Herzberg's Motivation-Hygiene Theory [3]. But nearly half of the publications used no "classical" motivation theory at all. Furthermore, despite the fact that in many cases there was a relationship between the findings and the theories, the studies seldom interpreted their findings in terms of those theories. The need for theory development with regard to motivation has also been identified by Agarwal et al. [62] in a study of employee turnover in IT.

Traditionally, software development companies have relied on three perspectives for motivating their employees: incentives, control, and coordination mechanisms. The first motivator is represented by payment and career incentives, and other benefits stipulated as part of employment contracts (e.g., Peters [63]). Regarding the use of incentives, software development firms implement control mechanisms, including behaviour or output based control (see Ouchi [64]). The third motivational perspective is related to the organizational hierarchy and centralized planning (see Cusumano and Selby [65]).

A systematic review of motivators in the agile context by de O. Melo et al. [66], highlights differences between the overall view of motivation in software development
and the motivation in an agile context. The study, which in addition includes three case studies in agile companies, suggests that certain motivators have an increased importance in agile teams and provides new motivators. The same study also claims that motivation seems to be higher for agile development teams which were previously exposed to other working methods.

Previous studies observing testing activities have examined how the organization and the relationships between the members of a software development team are impacting the testing process and implicitly the product quality. Although there is extensive work on motivation of IT personnel [67] and specifically on motivation in agile teams, e.g. [68] and [69], to our knowledge there is a lack of research focusing specifically on motivation in software testing.
Performing software testing activities involves adjustments of the external work environment, including both physical conditions and in social situations where people meet and create relationships. The physical and social environment affects the motivation of participants involved in these activities. Their motivation makes their behaviour more or less flexible and creative and affects their work performance. The good behaviour of a working system is characterized by flexible, creative behaviour and positive environment. All these factors are correlated with individual motivation. Therefore, the relationship between the factors involved in software testing activities and the motivation of participants are at the core of this thesis. Against this background, the following research questions are formulated:

- **RQ1:** Which contextual factors are salient in situations that can generate strong motivation for software testers?
- **RQ2:** How is the motivation of software testers encouraged in practice?
- **RQ3:** How does working experience in industry contribute to the perception of testing activities' importance?

According to Ryan & Deci [42], when we think about motivation we think about energy, direction, perseverance and goal focus and we are interested in factors that are strongly linked to action. Individuals acting under the influence of strong motivation perform close to their maximum capabilities. It is therefore reasonable to focus the investigation on the situations which give rise to particularly strong experiences of motivation: **RQ1** “Which contextual factors are the salient in situations that can generate strong motivation for software testers?” Similarly, we want to identify and analyse situations in which the individual feels a particularly weak motivation. Amabile and Kramer [70] have shown that the same general everyday events that have a positive impact on project participants' motivation also have negative counterparts. The negative perceptions have a stronger impact on individuals' inner work life than the positive ones. Just as important as identifying the social and contextual events that give rise to strong motivation, is to identify the events that can strongly discourage or frustrate workers.

Previous research indicates that motivation is a challenge for testers, and that this can cause poor quality software products in at least two ways: (1) workers with low motivation deliver more poorly, and (2) workers with low motivation or job satisfaction will more often quit their jobs, causing more frequent changes of personnel [3], [4].
These situations will result in a bigger overhead from training new people and on the average having less experienced testers who are likely less effective. Hence, it was imperative to observe the modalities in which these negative and positive factors are regarded and acted upon in software development industry by exploring the research question: **RQ2 How is the motivation of software testers encouraged in practice?**

Although the importance of testing is already acknowledged and established in many textbooks about software engineering, software testing or in research papers about testing, this acknowledgement is mainly academic, with limited impact on industrial practice and industrial attitudes. We were interested in following the evolution of testing perception from the early days of educational training until the point of becoming an experienced professional. This process was pursued with the research question: **RQ3: How does working experience in industry contribute to the perception of testing activities' importance?**

### 3.1 Research methods

Traditionally, researchers based in technology such as computer science, often incline towards quantitative methodologies while researchers based in behavioural science lean toward qualitative methodologies.

Providing an answer to RQs required first an overview of the knowledge gathered from many persons. Questionnaire surveys were chosen since they are an ideal solution for retrieving snapshots. In order to obtain answers for RQ1 and RQ2, deeper knowledge on how things are performed in organizations and why, qualitative data was required. It was decided that the use of in-depth interviews with fewer persons would be appropriate.

Since the focus of our research is situated between organizational behaviour and technology, a combination of quantitative and qualitative research methodologies is required. This has been recognized and throughout the years researches in Software Engineering have utilized approaches from multiple fields. Warfield [71] noted that there is a debate regarding which research methodologies are adequate for research within information systems and technology field.

In this research we have used the quantitative data for constructing an overview of the internal organization of testing activities and for analysing the results from alumni survey data sets. Qualitative data have been used to understand and explore the many factors we observed during the stages with focus on motivation and characteristics of testing professionals.

The two main data collection tools used in this research has been questionnaires and semi-structured interviews. In addition, we have also visited and observed some of our industrial partners on several occasions.

#### Surveys

A survey is a data gathering method that is utilized to collect and analyse "the characteristics, actions, or opinions of a large group of people, referred to as a population" [72] (p.78). Surveys are used in several fields of research, such as
sociology, marketing research, politics and psychology. In the field of software engineering, surveys are mostly used to determine how a technology, a method or even a change has been received or to capture a snapshot of large projects or group of organizations [73]. We opted for this type of survey to retrieve a snapshot of organization of testing activities which was the stepping stone for answering RQ1. RQ3 concerned the impact of experience in industry as a re-assessing factor for the perception of testing, hence using a survey allowed us to reach a reasonable number of participants both in academia and industry.

Surveys provide a high level of capability in representing a large number of participants. Due to the high number of respondents, the data being gathered possess a better description of the relative characteristics of the general population involved in the study, which will give us a better basis for generalization. Surveys gather quantitative and qualitative data and can be administered to the participants through a variety of ways mostly questionnaire and interviews [74].

**Interviews**

A personal interview is a data gathering method that is utilized when a specific target population is involved. The purpose of conducting a personal interview survey is to explore the responses of the people to gather more and deeper information. By interviewing the participants, the researcher can probe the answers of the respondents and at the same time, to observe the behaviour of the respondents, either individually or as a group and the context in which the interview is taken [75].

Interviews can have a range of designs, from structured and closed-ended to unstructured and open-ended. Structured interviews are based on a detailed interview guide, and the same questions are asked to all the informants. The results from this type of interview are often used in quantitative research and they can be performed by phone, email or regular post. Many times they might include multiple choice predefined answers or similar, and many of the questions can be answered with “yes” or “no”.

In case of a semi-structured interview, the interview guide will often include a list of topics to be raised during the interview. Questions will be more open than in structured interviews, and the intention is that the interviewed persons should formulate the answer using their own words. The interviewer is steering the interview, but the interview should ideally resemble a conversation between two equal partners. If new issues are raised during the interview, the interviewer would often follow a new thread with new questions.

If the interview is given a semi-structured format [76], the research can guide the focus of the respondent's story towards those aspects of perceived situations which are spontaneously selected as the most significant for his or her experience. The method has some traits of "critical incident technique" by Flanagan [77], which is based on critical events or episodes that the respondent chooses from his experience based on a question from the interviewer. The respondent recounts the incident or episode and the interviewer follows up the story with additional questions, for example, how the respondent was affected by the incident, how long the effect lasted, etc. A similar technique was used successfully by Herzberg [55] in his classic study on motivation at
work. A semi-structured interview was the method chosen for retrieving information on motivational factors affecting testers (RQ1) and exploring the use of these factors in practice (RQ2).

Analysis of qualitative data

Early forms of qualitative analysis originated in the natural sciences where the focus was on observing and measuring in order to better understand the explored phenomenon. Qualitative research developed alongside the social sciences where researchers found it difficult to measure human behaviour with simple quantitative methods. Although, there is no universally accepted method for the analysis of qualitative data as there is for the analysis of quantitative data, there are modalities in which qualitative data can be handled systematically. Robson [78] describe three different approaches for performing qualitative analysis:

Quasi-statistical approaches
Word or phrase frequencies and inter-correlations are used as key methods for determining the relative importance of terms and concepts.

Thematic coding approach
After the data are coded and labelled, codes with the same label are grouped together forming a theme. Codes and themes occurring within the data can be determined inductively by reviewing the data and/or by its relevance to the research questions, previous research or theoretical considerations. The themes can serve as a basis for further data analysis and interpretation.

Grounded theory approach
This is a version of thematic coding in which the codes arise from interaction with the data. Codes are based on the researcher’s interpretation of the meanings of the patterns within the text. The results are “grounded” in the findings from the data.

3.2 Research Process

Our study was planned in several stages where each stage had studies conducted sequentially in time, building on the results from studies from previous stage.

- **Stage 1**: Consisted of a quantitative survey P1 followed by qualitative semi-structured interviews related to organization of software testing activities in the industry and a qualitative study on testing career interest P2.
- **Stage 2**: Composed by qualitative semi-structured interviews focused on job satisfaction, motivational factors and characteristics for testing personnel P4, P5. In addition we analyzed data from two web based surveys from a testing perspective P3.
- **Stage 3**: Constructed mainly by an extended qualitative survey synthesizing motivational factors and stimulating strategies described in P6.
Stage 1

For the first stage we constructed an overview of the internal organization of testing activities by conducting a survey on the internal organization of the testers, working methodologies and automation involvement, within the software companies who agreed to support us in our research. We commenced with a cross-sectional designed survey, which is best suited when the goal is to describe a population and explore the difference in the subset of that population at a specific point in time [79]. This classic cross-sectional survey collected data about the organization of testing activities from software development companies at that point in time.

In order to provide an answer to RQ1 we need to understand the internal dynamics of a testing team or testing personnel and construct an accurate overview of their daily activities. A population of 19 companies which represented various domains of business, and different sizes, from small companies to large international corporations, participated in the survey. The first set of analysis indicated a low preference for having a separate testing department or a dedicated testing staff as described in P1.

The use of questionnaires for the first stage made it possible to quickly reach the companies and thereby get an overview of the situation in Norway. On the other hand, since questionnaires have shortcomings when it comes to providing deeper insights, we decided to use mainly interviews but also phone discussions and follow up emails for the second stage.

To answer RQ3 we decided to first make an assessment of the perception and expectation from a testing career path among our students, hence we initiated a study in which we involved students from the third and fourth years. The students answered the questionnaires separately after lectures, and they were filled in by hand. This was done to provide support for possible misunderstandings in answering the questionnaire. The results of the study, described in P2, indicated a low interest for a future career in testing with many of the participants showing little interest or having a negative attitude towards testing as a profession. The perception of testing activities seems to change once the students graduate and get jobs in industry.

Stage 2

The second stage of the RQ1 study used a qualitative survey with semi-structured interviews which allowed us to retrieve more information about testing management and testing activities, and help in building an answer to RQ2.

We chose this research method since it was clear that the information we sought could only be obtained directly from respondents rather than from the accumulated data. The questionnaire used in this study was based on previous published literature. For improving the knowledge about the human factors, we discussed with testing professionals about their career path, their choices which lead to the current testing position, the job satisfaction and the motivation behind a testing career choice, P5. We were interested in retrieving a description of the characteristics which are considered the most suited or desirable when hiring software testers, P4. The data collection for the RQ2 stage was based on these interviews and overlapped with the data gathering for the second stage for the RQ1 study.
In order to explore how the perception and expectation from a testing career path evolve over time and provide an answer for RQ3, we analysed from testing perspective two web based surveys, conducted by the Department of Computer Science (IDI) of the Norwegian University of Science and Technology (NTNU) among IDI master alumni in two separate years. The main purpose of the surveys was to retrieve the educational topics relevant to the participants' field of expertise based on the work experience accumulated after graduation and to compare them with the emphasis assigned on these topics during their studies.

The survey instrument was based on three sources: a candidate survey from a closely related study program in Communication Technology at the NTNU [80], the North American study conducted by Lethbridge [81] and [82]. Another source used for building this instrument is the ACM Computing curricula [83], while the 75 topics provided in this survey are inspired by SWEBOK (IEEE's Guide to Software Engineering Body of Knowledge [84]). The questionnaire's items enable the identification of topics which the alumni have been forced to learn after graduation and topics covered during studies but forgotten, since there was no need for participants to use this knowledge in their jobs. The questions did not only highlight what was beneficial for the alumni so far, it was also interesting to see what they thought would be important in the future based on trends they saw in the industry. The data already collected by these two surveys (for other purposes) was re-analysed from the perspective of the software testing topic. From an ideal viewpoint, it might have been better to perform a questionnaire study towards the alumni with an instrument designed especially for the purpose of this thesis, and thus having more questions about testing and fewer on other issues. However, this was not done due to the fact that the data was already available and we were reluctant to overload the alumni with yet another questionnaire, given the likelihood of a low response rate.

Stage 3

For the last stage of our research we extended the qualitative survey by increasing the number of interviews and building up the information gathered and analysed in Stage 2 and published in P4 and P5. The last paper, P6 included in the thesis, focuses on motivational factors and social aspects influencing the activities of the people involved with testing and provide a more detailed view on how professional testers are perceived and encouraged in a software development company.

The study was carried out using a qualitative approach through semi-structured and in-depth interviews. The focus of qualitative research is to investigate and understand phenomena within their real life context and provide a better view and understanding of a certain phenomenon [78]. The data collection included in-depth interviews with 36 practitioners from 12 companies and was performed over a two years period and investigates the strategies applied by the companies for stimulating their testers, while considering the motivational and de-motivational factors influencing the testing personnel.
3.3 Validity Threats

Validity of research refers to the many conclusions we reach about the quality of different parts of our research methodology. Cook and Campbell define validity as the "best available approximation to the truth or falsity of a given inference, proposition or conclusion" [85] (p. 37). Below we present the four types of validity and threats as listed in Wohlin et al. [74]. Validity threats are discussed in details in each paper and a more detailed discussion will be included in the Discussion chapter.

Construct validity

The construct validity is concerned with the relation between theories behind the research and the outcome. According to Runeson and Höst, "this aspect of validity reflect to what extent the operational measures that are studied really represent what the researcher have in mind and what is investigated according to the research questions" [86] (p. 153). The variables in our research are measured through semi-structured interviews, including open-ended questions where the participants are encouraged to express their own opinions. Most of the questions used in the semi-structured interview guidelines have their origin from the research literature. Further, through piloting the guideline among colleagues and industry partners, most of the questions were refined additionally.

Conclusion validity

Conclusion validity represents the degree to which conclusions we reach about relationships in our data are reasonable. A threat to conclusion validity is a factor that can lead you to reach an incorrect conclusion about a relationship in your observations and it is related to the ability to draw accurate conclusions.

Internal validity

Internal validity is relevant for studies that try to establish a causal relationship between treatment and outcome and not for those which are observational or descriptive. This type of validity is the primary consideration in studies that assess the effects of an alteration or intervention. In those contexts, you would like to be able to conclude that the treatment made a difference. Threats to internal validity include instrumentation, maturation, and selection threats.

External validity

The external validity is concerned with the ability to generalize the results, which for this study translates into the applicability of the findings beyond the survey and interview participants. Qualitative studies focus by nature on explaining and understanding the situation under study, rather than generalizing the results. The nature of qualitative design itself makes it impossible to replicate a study, due to the improbability to reproduce exactly the same context, but it can lead to a theory which can provide understanding for similar cases and situations.
This chapter presents a synopsis of the results, organized according to the three research questions. Each section gives an introduction explaining the research questions and the main results. Since this thesis is based solely on the included publications, no additional results or data are added in this chapter.

- **RQ1** What contextual factors are salient in situations that can generate strong motivation for software testers?
- **RQ2** How is the motivation of software testers encouraged in practice?
- **RQ3** How does working experience in industry affect the perceived importance of testing activities?

Further details and information about study approach, related research, analysis, discussions, and conclusions can be found in the original papers in part II of this thesis. Table 2 shows how the key findings are related to the research questions, and the mapping between the papers and findings. Important quotes from the interviews are highlighted in the text to illustrate the key findings. A description of each key finding starts with presenting the key finding itself, followed by an explanation of how the finding is related to the underlying studies.

### Table 2: Mapping between research questions, findings and papers

<table>
<thead>
<tr>
<th>RQ</th>
<th>Findings</th>
<th>Papers</th>
</tr>
</thead>
</table>
| RQ1 | F1. Empirically based research shows that testing activities tend to be organized in four categories  
F2. The role of software testers is not consistently defined  
F3. Testing activities are managed in different ways depending on the company profile and business  
F4. Test automation is considered beneficial and adopted to some extent, but is not utilized on a large scale | P1, P1, P2,    |
|     |                                                                                                                                                                                                          | P1, P5, P6      |
|     |                                                                                                                                                                                                          | P1              |
| RQ2 | F5. A set of characteristics and skills considered most suitable or desirable when hiring software testers  
F6. A comprehensive description of factors that motivate and stimulate the software testing personnel  
F7. Strategies deployed by the companies for stimulating their testing personnel | P1, P4, P1, P5, |
|     |                                                                                                                                                                                                          | P6              |
| RQ3 | F8. A multi-faceted perspective of the perception of a testing career among students and graduates  
F9. Motivating and de-motivating factors influencing the students in pursuing a software testing career | P2, P3         |
|     |                                                                                                                                                                                                          | P2              |
4.1 F1 Testing Team Structure

In order to suggest improvements we first need to understand the internal dynamic of a testing team and construct a reasonable overview of their daily activities. We initiated the research by conducting an introductory study P1 on the internal organization of the testers in the software companies that agreed to collaborate with us in our research. The study focused on how the testers are organized inside a company, and we were interested in information about the formal and informal position of the testing teams, the person in charge of the testers, the testers’ physical location, the software development lifecycle and the methodology used in their activities.

The results indicated a low preference for having a separate testing department or a dedicated testing staff and showed us that the participating companies had four ways of organizing their testing, as displayed in Table 3. In addition, none of the participant companies are outsourcing testing activities to external companies, although four companies prefer to have remote testing teams onshore or offshore.

Table 3: Testing activities - organizational categories

<table>
<thead>
<tr>
<th>Organizational categories</th>
<th>Percentages</th>
</tr>
</thead>
<tbody>
<tr>
<td>A dedicated software testing department</td>
<td>21.05%</td>
</tr>
<tr>
<td>Combination of testing department and testers divided among application groups.</td>
<td>10.53%</td>
</tr>
<tr>
<td>Testers divided among the application groups</td>
<td>42.11%</td>
</tr>
<tr>
<td>Developers responsible for all the testing activities with no formal testing staff in the company.</td>
<td>26.32%</td>
</tr>
</tbody>
</table>

A dedicated software testing department

Within companies with a dedicated testing department, the recognition of the importance of testing has grown over time and the centralization of the testers was done as an upgrade from the previous organization, where testers were located in several groups and departments: A separate testing department was created with the aim to improve their performance and organization and the focus on testing activities. We were a large group working for a customer who was focused on quality. We had frequent releases and we were good at it. Before, testers were located in groups and departments and in order to improve our performance and focus on testing we created groups with project managers and testers. (Testing Manager) The same company has a policy for organizing informal and formal training sessions internally and externally with regular meetings solely for knowledge improvement. In another company, the creation of a dedicated testing department was the result of lobbying activities by testing managers, combined with a motivation to improve product quality, which had proven unsatisfactory in previous projects.

Combination of testing department and testers divided among application groups

Two of the participating companies had a combination of testing departments and testers being divided among the application teams. One of the companies is using Scrum, and in this case, testing is the responsibility of the development team.
**Results**

*Testers divided among application groups*

The largest category contained the companies with testing staff distributed among the applications teams. By following this organizational model the testers can work closely with the developers and this configuration was the preferred choice in small and medium sized companies. Two of these companies have express awareness of the need to create a separate testing team in the future, but no concrete plans were made so far. In the meantime, hiring temporary testers for larger projects is considered an alternative. The use of agile methodology and Scrum motivate another company to organize their tester as part of the application based in the “Whole team” agile practice.

*No dedicated testing staff*

Five of the participating companies decided not to invest in dedicated testing personnel. We asked these companies what lead to this decision. Two of the participants reported having insufficient amount of testing work to justify a dedicated testing position, while for the young companies testing was seen as a future component, which will be formally organized once they reach a more mature stage.

**4.2 F2 Software tester concept**

The concept of software tester is not as consistently defined and cemented as the one of software developer. The responsibilities and assignments differ based on the company policies and business profile.

One of the companies considers Release Engineering and Documentation as the responsibility of the testers although there is no testing team and the testers are part of the application groups (P1).

Other companies specifically look for software engineers and not testers. There is no general consensus on the testers' responsibilities, with some putting a strong emphasis on automation experience and programming skills while others are more concerned with the business domain knowledge (P4).

**4.3 F3 Observation on testing management**

Based on our studies P1, P5 and P6, we extracted some observations related to the management of the testing activities.

*The responsibility of testing management varies between development, testing or product groups*

In our introductory study P1 we found that of the 19 participant companies, 10 had an Application Development Manager or Product Manager as the person in charge of the testing activities. The interviews for the studies presented in P5 and P6 gave similar findings.
The ratio between the number of developers and testers has a wide range

The developer-tester ratio varies greatly from 2:1 to 1:5, sometimes even in the same company, where the ratio can differ from one project to another, with more testers being assigned in case of a new project or product.

Contradictions between goals and implementations

We found discrepancies between the expressed goals and the concrete implementation and organization of testing activities. All the participant companies have a strongly expressed interest in ensuring and strengthening the quality of the product or service provided. They have developed internal rules and guidelines in order to ensure that the quality requirements are honored. However, the manner in which they choose to invest in the testing activities and implement these guidelines can lead to unsatisfactory results.

Not looking for testing skills as a mandatory or at least an essential requirement during the recruitment process might result in a team lacking the strength or motivation for efficient and thoughtful testing.

"Usually we don't look for dedicated testers so testing is seen as an extra skill" (Section Manager)

"Within the R&D department we recruit new developers and if they are accustomed to testing or test procedure that will be a plus in the hiring process. It is more important that they fit in the organization and their personal skills rather than specific skills in testing." (Section Manager)

Combined responsibilities were seen as a way to prevent a sense of monotony, but it can also be perceived as a way of reducing testing costs by having a person fill several roles. Having 50% development work was considered a good way of keeping people assigned to test activities while providing them with development responsibilities. In case a person is mainly interested in development responsibilities, he may execute his testing assignments without much enthusiasm or desire to be proactive and improve the testing process. This strategy can backfire on the quality of the product, when most of the people involved in the testing process perceive it just as an additional part of the job or a temporary assignment.

When interviewing managers and testers from the same company we observed differing opinions about the resources or effort required to run testing activities, with managers often arguing there was too little testing work to justify hiring more fulltime testers, while the testers expressed a need for more testing resources to be assigned to their team or project.

Outsourcing of testing related assignments, entirely or partially, typically to lower cost countries, proved to be an established practice. The managers involved in this type of process mentioned problems with the quality of the service provided by the offshore testing partner, or the necessity of setting aside additional time for communication and trips abroad, to facilitate knowledge transfer and process improvement.
4.4 F4 Automation

Automation plays an important role in producing a consistent testing base with reproducible results and testing steps. Thus, we asked the companies about their interest in test automation, the time they invested in it and the tools that were used for these activities. The results described in P1 showed that around 75% of the companies were involved in test automation activities.

We asked the companies involved in test automation about the experience, expressed as number of years they have in implementing automation for testing activities and the percentage of the testing activities which are automated. Although 64% of companies had more than two years of experience in automation, half of the companies have automated 25% or less of their testing activities. This result confirms the findings of previous studies on test automation that although it is considered beneficial and adopted to some extent, it is not utilized on a large scale in the software companies [87].

4.5 F5 Characteristics of a Software Tester

Since the job as software engineer can contain multiple roles and responsibilities which can vary from one position to another, more information about the characteristics which influence the motivation of software engineers is required. Our studies focus on software testers, who are often considered a subset of software engineers in the job title terminology, but may have different responsibilities than their fellow developers.

The job categories used by the US Bureau of Labor Statistics does not mention Software Tester as a job type, but lists Software Developer as an occupation [88]. Moreover, under the detailed description of Software Developer [89] it appears that testing responsibilities are included in this category, e.g. "Analyze users’ needs, then design, test, and develop software to meet those needs", "Ensure that the software continues to function normally through software maintenance and testing". It is worth mentioning though, that the US BLS categories distinguish between software developer and computer programmer, a distinction which will not apply in many smaller companies. And programmers in the US Bureau of Labor Statistics terminology also seem to include testing activities [90], e.g. "Debug programs by testing for and fixing errors"

Our findings describe the characteristics which are perceived as most valuable for a tester, based on the working experience accumulated by the participant testers and testing managers. The findings of P1 and P4 suggest that there are differences between the characteristics of a software engineer and a software tester, with communication skills and an IT background being cited as the most frequently favored characteristics when employing testers.
### Results

<table>
<thead>
<tr>
<th>Most wanted tester characteristics</th>
<th>Rank</th>
<th>Most wanted tester characteristics</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication skills</td>
<td>7</td>
<td>Diplomacy</td>
<td>3</td>
</tr>
<tr>
<td>IT background</td>
<td>6</td>
<td>Need for challenge</td>
<td>3</td>
</tr>
<tr>
<td>Need for variety</td>
<td>6</td>
<td>Domain knowledge</td>
<td>3</td>
</tr>
<tr>
<td>Details oriented</td>
<td>5</td>
<td>Courage</td>
<td>3</td>
</tr>
<tr>
<td>Curious</td>
<td>4</td>
<td>Creative</td>
<td>2</td>
</tr>
<tr>
<td>Focus on quality</td>
<td>4</td>
<td>Proactive</td>
<td>2</td>
</tr>
<tr>
<td>Patient</td>
<td>4</td>
<td>Structured</td>
<td>2</td>
</tr>
<tr>
<td>Testing experience</td>
<td>4</td>
<td>Team player</td>
<td>2</td>
</tr>
<tr>
<td>Achievement orientated</td>
<td>3</td>
<td>Mind set to find bugs</td>
<td>2</td>
</tr>
</tbody>
</table>

#### IT background

Possessing an IT background refers to having an education and/or experience within the computer science/software engineering field. Interviewees in this study held programming knowledge and experience as valuable assets due to the expertise accumulated which can help in certain testing activities. "A development experience can help a lot. All the knowledge I gain while working as a developer, helped me in my testing work." (Tester)

An IT background was considered an important characteristic for a tester by our interviewees. The IT background proved valuable when communicating with the developers. "In order to speak to a developer and understand what they are talking about, an IT background is a very good thing". (Tester)

#### Need for variety

In all participant companies the **Need for variety** characteristic was often cited as a positive trait due to the varying nature of tasks in which a tester could be involved. A tester who enjoys a variety of tasks could adapt better to different responsibilities throughout the project lifecycle, starting with the design phase and ending with the maintenance stage. "The biggest factor for me is that you do different things, it's very varied and you get to see the whole picture. You can participate from the start of a project to the end doing various things, that's the biggest thing for me, the different things I can do." (Testing Manager)

#### Details oriented

Being a details oriented person was considered an advantageous skill for a tester among the study participants. "You have to have an eye for details... you cannot just check the normally expected results. You cannot just test the happy path scenarios". (Testing Manager)

#### Curious

Some of the participants considered curiosity trait as an incentive for continuously improving the understanding of the product and for pursuing alternative testing scenarios. Curiosity was also advantageous for coming up with unusual testing scenarios. "You don't have to be afraid to click on new things and that you are going to break something. That is part of being curious" (Testing Manager). "Curiosity. You should want to learn new things because you will get new software, new projects, new teams all the time" (Tester)
Results

Focus on quality
Some of the participants have expressed their desire to improve the quality of the products they are working on and taking pride in participating in the delivery of a high quality product. They talked about their joy in finding defects, which will lead to a better product. "I do have a passion for improving the quality and finding defects... I'm happy when I find bugs. Of course, I'm also happy when things are working." (Tester)

Patience
Patience was mentioned as being an important requirement for software testers. The testers may need to work outside of regular office hours with little warning in advance. "you say: <<today I’ll leave at 4>>, but at 3:45 they say <<you have to do this testing before you leave today>>, and you have to work 2 or 3 hours more. That’s a part of it" (Testing Manager)

4.6 F6 Motivational and de-motivational factors for software testing personnel

Our studies identified a set of motivational and de-motivational factors for software testing personnel, and insight into strategies deployed by the companies for stimulating their testing personnel. As any human-based activity, the outcome of the final software product is dependent of human factors and an essential challenge for software development organizations is to find effective ways to enhance the motivation and job-satisfaction of their testers. The results indicate that the job-satisfaction of testing personnel can be increased by combining testing responsibilities with development, and ensuring a variety of engaging and challenging tasks and products.

F6.1 De-motivational factors

In Table 5 we show a list presented in descending order starting from the de-motivational factor which was most frequently mentioned in the interviews:

Table 5: Relationships between codes and concepts for negative factors

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
<th>Mentioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative factors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of influence</td>
<td>late involvement in the project, testing is underestimated in the company, afraid of opening defects, no control over the schedule</td>
<td>21</td>
</tr>
<tr>
<td>and recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unhappy with</td>
<td>insufficient resources, unrelated tasks</td>
<td>19</td>
</tr>
<tr>
<td>management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical issues</td>
<td>versioning, insufficient number of test environments, poor quality, integration issues with simulators</td>
<td>16</td>
</tr>
<tr>
<td>Lack of organization</td>
<td>lack of clear processes, tasks, redundant meetings</td>
<td>13</td>
</tr>
<tr>
<td>Time pressure</td>
<td>squeeze, long days, short periods, overloaded schedule</td>
<td>13</td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boredom</td>
<td>routine, repetitive tasks, unchallenging work</td>
<td>12</td>
</tr>
<tr>
<td>Poor relationships with developers</td>
<td>bugs related friction, stereotypic view of testing, slow defect fix rate, late changes to the code</td>
<td>7</td>
</tr>
<tr>
<td>Working environment issues</td>
<td>colleagues with no social antenna, open plan landscape related issues</td>
<td>5</td>
</tr>
</tbody>
</table>

**Concept - Lack of influence and recognition**

The concept which appeared most often as a factor with negative impact was the lack of influence and recognition. Under this concept we gathered interview segments referring to the irregular working flow and lack of control over an unstable schedule. Testers’ late involvement in the development cycle, together with the struggle for recognition is also frequently cited by the participants: "When a tester or test manager enters a project it's too late in the process to get a reasonable contribution to the quality with the testing" (Tester). When the focus of testing activities is more on testing issues, like retesting defects, rather than testing the product or requirement, testers are not provided with a sense of accomplishment, but rather with a frustration of not performing their real job. Under the same concept we aggregated the worries for a low likelihood of promotion, in comparison with other roles such as developers. "The developing projects and the daily operations have to realize how important software testing is. The testing area has to be lifted up as an important part of the company’s work." (Testing Manager)

**Concept - Unhappy with management**

Another concept mentioned is the participants’ dissatisfaction with the management related policies, the unrealistic schedules and the scarcity of resources. An unsupportive management can make testers reluctant to log in new defects: "testers use a lot of time, they are afraid of opening defects". (Testing Manager) Opening a critical bug can be a stressful scenario even for an experienced tester; it can lead to frictions with the developers or conflicts with the management: "I found bugs which stopped or hold a release. On one hand is a good thing, because if the bug went into production it could have created serious problems, but in the same time, is also a little bit like putting your reputation on the line. The release is stopped because of you" (Tester). Raising invalid defects can be detrimental for a tester. These actions can lead to lack of respect from developers and pressure from managers.

**Concept - Technical issues**

Technical issues within the testing context are referring to problems with testing tools, development environments or a weak infrastructure. An insufficient number of test environments, poor quality or insufficient fidelity to the actual system being tested, together with integration with 3rd party tools or simulators were mentioned as hindering factors of a technical nature. "It takes a lot of time to get the tests started, not everything works correctly, setting up an environment and also installing the software on our test servers" (Tester). In some companies the participants complained about the weak infrastructure which turned out to be the root cause in many false defects and required time and effort in investigations. “My main frustration is that we don’t have good
enough tools to do our work and we have to use tools that make our work a lot more difficult than it should be” (Developer).

**Concept – Lack of organization**

The interviewees were not pleased with the continuously changing plans or bad initial planning. In addition, some of the participants had an increasing number of tasks which were not related to testing or outside their focus area. “We fill a lot of time until we don’t have any left space [in the project schedule], but often we want to update the plan” (Tester). Participants considered the lack of organization or poor planning as a strong source for the repeating problems encountered in previous releases with regard to members of the testing team. A high number of meetings which were considered redundant or irrelevant to their work tasks were also mentioned as a negative and time consuming factor.

**Concept - Time pressure**

Another concept which appeared often as a factor with negative impact was the time pressure under which testing is performed. Traditional development teams often delay testing until the end of the projects, squeezing the calendar time available. Unfortunately, projects often fall behind schedule, so the testing teams need to compress and sacrifice activities due to their shrinking time frame. "I've been in this business for many years and testing is at the end of this lifecycle, and always pressed to so short periods, long days, and shortcuts. It's always like that" (Testing Manager). Testing time is sacrificed to recover the delays in other processes and by doing so there is often a compromise on the quality of the delivered product. "I don't like that we are the last link in the chain, and we don't always get the time that was promised in the beginning. Give us more time to finish our testing and do it properly" (Tester).

This concept appears also in the interview with testers from agile teams where the testing is occasionally facing similar time pressure. The company has sprints with unbreakable deadlines, but since the first half is allocated to test case designs, issues are often discovered late in the sprint. This situation gives little time to fix the issues. "Sometimes it's difficult to plan because they [the developers] don't really know when they are ready. They want testing done immediately as they are ready, but they themselves don't really know when they are ready" (Tester).

**Concept - Boredom**

Some of the participants mentioned the routine of some testing activities and the feeling of boredom associated with maintenance testing. "Everything is routine, there are no surprises after the system is in production" (Testing Manager)

**Concept - Poor relationships with developers**

Another de-motivational concept was represented by the poor relationship between testers and developers, which can be problematic. Most of these frictions result from discussions related to bugs. "I do remember having discussions about bugs: Is it really a bug, or is it [the bug fix] really important enough to be included in the release" (Tester). Another factor quoted by many participants was the stereotypic view of testing by the developers, "the classical view that they are developing and finally we are testing
and then it's coming back with us saying <<that is not good, that is not good>>" (Testing Manager).

Two testers from different companies described their co-workers' view of testing as "a necessary evil". The slow defect fix rate and developers making unannounced late changes to code were also mentioned as a source of concern and conflict between developers and testers. "It's a lot of things, challenges that take time, sometimes it can take time to get testing environments, sometimes you raise bugs and they don't take them quickly enough" (Testing Manager)

**Concept - Working environment issues**

Several participants complained about working in open space landscapes, noisy due to the nature of the office design and sometimes with colleagues with no social antenna. “When it comes to office conditions it can be quite noisy in this open landscape thing” (Tester).

**F6.2 Motivational factors**

The codes and concepts related to positive factors derived from the study are presented in Table 6. The positive factors are presented in descending order starting from the one which was most frequently mentioned in the interviews.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
<th>Mentioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enjoy challenges</td>
<td>Enjoy challenging yourself, every day you never know what's coming up, like the chaos, need challenges</td>
<td>11</td>
</tr>
<tr>
<td>Focus on improving the quality</td>
<td>finding bugs, to investigate, making things better, personal goal on improving the quality</td>
<td>11</td>
</tr>
<tr>
<td>Variety of work</td>
<td>work variation, combine testing and programming.</td>
<td>11</td>
</tr>
<tr>
<td>Recognition</td>
<td>ensure that testing tasks are important in the company, send testers to courses and conferences, get the support I need to do a good job</td>
<td>9</td>
</tr>
<tr>
<td>Good relationships with colleagues</td>
<td>good communication in the team, with developers, enough resources</td>
<td>7</td>
</tr>
<tr>
<td>Technically challenging work</td>
<td>technically challenging work</td>
<td>6</td>
</tr>
</tbody>
</table>

**Concept - Enjoy challenges**

Most of the interviewed participants enjoyed the challenges represented by the testing activities, challenging themselves or simply thriving on the chaos which can sometimes accompany the daily activities of a tester. "When I perform my test and it works, I’m thinking: Am I doing something wrong? Is the test doing what it’s supposed to? When it fails, I’m also thinking: Is it really doing things correctly?" (Tester in an enthusiastic voice) and "There is always something new, new challenges towards different test scenarios" (Tester).
Concept - Focus on improving the quality

The second most occurring motivational concept related to testers' passion for improving the quality of the software, the pleasure of investigating and finding defects, which will lead to a better product. "I do have a passion for improving the quality and finding defects. And there I have learned that I have different focus than the developers, maybe the right focus for testing. I'm happy when I find bugs. Of course, I'm also happy when things are working" (Tester).

Concept - Variety of work

On several occasions the concept variety of work was mentioned, referring to being included in the testing activities associated with the whole development cycle, not just a specific phase. Another contribution to the variety was to have a combination of programming and testing tasks as part of job responsibilities. "The biggest factor for me is that you do different things, it's very varied and you get to see the whole picture. You can participate from the start of a project to the end doing various things, that's the biggest thing for me" (Tester).

Concept – Recognition

The concept of recognition included awareness of the importance of testing in the company from both management and development teams as well as positive feedback from developers in relation to discovering and fixing bugs. "When we heard feedback from engineers, when we hear they say <<thank you, this test helped us to fix something that is wrong>>” (Tester). In the same category we included participants expressing the pride they experience by working in a company known for delivering high-end products. “I believe I work in a company that is delivering high end embedded software worldwide. I want to make sure that the software we deliver has high quality” (Testing Manager).

Concept - Good relationships with colleagues

Under this concept we aggregated all the positive references to relations and communication with the managers, between the testers and with developers. "I think is important to be on good terms with the developers; if they are having some agile approach, you as a tester or test manager will get invited to their daily Scrum, so you get a feel for the modules they are struggling with and so on. It can help you prioritize, when you start to test" (Testing Manager).

Concept - Technically challenging work

Another positive concept, Technically challenging work, was associated with the participants' need to have allocated tasks reflecting their technical competencies. “The most interesting thing that you can have is interesting technology to work with” (Developer).
Results

Contrast between the agile and traditional testers

When exploring the difference between the traditional and agile testers, interviews indicated a higher time pressure factor for the traditional testers, while the Lack of organization tends to score higher in the agile teams. The lack of influence and recognition is present in both type of working environments with the traditional teams having a slightly higher occurrence. When discussing with general managers in companies working in the traditional way, they signalled several problems with the testing position, such as a struggle for recognition as a valuable team member and also frustration coming from the lack of influence when suggesting recommendation or requests related to their working activities. Whatever methodology is followed, all the participant companies are interested in providing Product Quality. What differs from traditional to agile is that testing is started early in the sprint and the emphasis on testing has improved with practices such as TDD.

<table>
<thead>
<tr>
<th>Positive factors group by methodology</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enjoy challenges</td>
<td>3</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Focus for improving the quality</td>
<td>7</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>Variety of work</td>
<td>6</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Recognition</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Good management</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Technically challenging work</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Testers working in agile do not belong to a separate testing group, but work within the development team. They consider testing an ongoing process throughout the development process, not just something that happens in a separate phase after development is done. Another point is that testing is done by the whole team, rather than just by testers and the relationship between testers and non-testers tends to be collaborative rather than adversarial. It was interesting to notice that more agile testers were unhappy about their relationship with developers since testers get more respect on agile teams where they are seen as colleagues, and are involved much earlier in the process, making it easier to ensure the testability of the system under development. The relationship problems might be related to situations where companies apply customized version of agile methods “for good organizational reasons” [32]. Participants from both categories complained about the heavy load and unrealistic schedules which is in concordance with earlier research results [67].

<table>
<thead>
<tr>
<th>Negative factors</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of influence and recognition</td>
<td>9</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>Unhappy with management</td>
<td>9</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>Technical issues</td>
<td>9</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>Lack of organization</td>
<td>8</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Time pressure</td>
<td>3</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Boredom</td>
<td>7</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Poor relationships with developers</td>
<td>5</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Working environment issues</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
4.7 **F7 strategies deployed by the companies for stimulating their testing personnel**

In this section we present the strategies adopted by companies for motivating their employees and stimulating their productivity. Several combinations or rotation of responsibilities and products are presented in the Combined responsibilities section. Under the Open office area segment we present the pros and cons of adopting an open office landscape. The Singular strategies part presents policies which we found implemented in one or two companies, while the policies presented in the first two sections were applied in all the participant companies.

**F7.1 Combined responsibilities**

The term combined responsibilities refers to the situation where a person, often a self-declared developer at heart, is in charge of several roles, one of which is the tester. This strategy is used as a method for personnel keen on development work, desiring a type of position which do not imply just testing. Another reason provided for applying this strategy is to provide more variety for testing personnel, hence avoiding the sense of monotony and routine which tends to be associated with testing activities, especially in the case of regression testing.

Although providing variety of work should be seen as a positive trait, it can also be an indicator of unwillingness to invest too much time in testing and one of financial pressure. Hiring personnel who can switch between several roles and fulfil different tasks is more cost-efficient than having a dedicated person for each type of responsibility. From the employee perspective, a downside of this approach is the energy and time spent on changing the working context and hence the time restriction that must be applied for each task.

One problem which might arise for employees having "several hats" is the difference of perspective between a tester and a developer. The testers tends to have a customer view when looking at the scenarios they are testing and they are interested in how these scenarios are integrating with the overall business concept. On the other hand, developers tend to concentrate on building the product from a technical design perspective, or to the provided requirements, with less concern for the overall business concept [91], [92].

"For me personally it's a bit difficult to change hats. It is too easy for the testers to get involved with development, and that can be both good and bad...If you write system tests, it might be that you are a developer involved in actually developing a feature and if you are also involved in writing tests for that feature you might not test it enough or you might test things that...You may use the knowledge about code... maybe you don’t test the code enough. It’s easy to hang on the knowledge that you have and you do not see the defects that otherwise you might detect" (Developer).

Under this category we can also add the situations where a group of developers are assigned for testing each other's code. The managers are aware of the possible lack of objectivity which might arise from testing one's own code, so this rotation is seen as a risk reduction measure and fulfilling quality related requirements.
F7.2 Open office area

Having an open landscape or co-locating the testers in the same area or room was seen as a way of encouraging communication and building a sense of unity. The open office space is considered one of the most productive working environments for software development [93] and it is supposed to increase the amount of informal communication inside the team, stimulate discussion and improve the problem solving pace [93], [94].

However, there were also complaints about this type of landscape with several participants describing work in open space landscape as noisy, which concords with the results reported in [93] where the developers perceived open office spaces as distracting, especially when people doing programming were in the vicinity of other developers holding a meeting at the same time.

In one company there were strong complaints about cases where co-located colleagues had no social antenna. "One of them has a cold and sneezes and coughs and it infects the other three. And it’s happened during a release time. He can bring down the whole team" (Testing Manager).

Another reason given for using an open landscape was that co-locating the testers and developers might lead to better communication and relationships between the two groups, who are prone to having a difficult relationship [95]. The tester-developer relationships can easily deteriorate due to defects related friction and discussion, late and un-reported changes to the code or due to the stereotypic view of testing by some of the developers. Communication among the stakeholders in a software development project is often of informal nature, mostly technology-based with daily meetings and discussions taking place over the phone, chat and video conference [96], [97]. Face to face communication is still considered the best way for building trust among people belonging to the same group and implicitly ensure a higher productivity in comparison with distributed teams [96]. "In our team we communicate very well and a lot; mainly because we sit in the same room" (Tester).

F7.3 Singular strategies

In this section we will present a collection of concrete changes which were applied in a single instance in different companies. The trigger for these changes was a change of management or a change of technology which led to the implementation of concrete and effective changes for improving the way testing is perceived in those companies."(The changes started) with the change of project lead. A couple of new project managers came into the project couple of years ago "(Testing Manager)

One preconception related to the (low) importance of testing is the view of the testing process as an activity that starts only after the coding phase is complete and which has a limited purpose, mainly to detect failures. We found mangers ready to challenge this prejudice by making substantial effort to have testing members involved all the way, starting with the software requirements process.

Since testing time pressure was a recurring issue in the testing activities, a buffer period was allocated for testing from the planning stage to ensure that delays in the development schedule would not reduce testing focus or give too little time for testing.
"Also we have planned the testing period with a slacking period. If they take from the time allocated for testing is not a crisis. Generally we have improved in that field." (Testing Manager)

The companies that wanted to improve testing decided that they need to have people who are motivated by that type of work. In order to motivate, they decided to make a priority of communicating the importance of quality and to emphasize that testing is one of the ways to achieve it. Their effort focused on ensuring that testing considerations was present throughout the entire development and maintenance life cycle and that the test personnel were included in the daily communication, and in the feedback from the customer. "I think we solved the motivation factor by having testers as part of the core team that delivers ... the tester receives the feedback from the customer now, bad or good" (Section Manager).

"We have activities for testing on the sprint. When a module is finished we don't put it to Done we put it To test. Testing is part of our daily tasks” (Section Manager).

4.8 F8 A multi-faceted perspective of the perception of a testing career among students and graduates

In order to raise awareness and improve the image of testing in academia we need to retrieve and explore the factors that lead people to this position in the first place. We conducted a study on a population formed by bachelor and master level students from the Norwegian University of Science and Technology (groups S1 and S2), together with computer engineering students from Trondheim University College (group S3). The survey was conducted towards the end of the semester in order to be able to assess the impact of the taught curriculum on the students’ perception. A total of 161 participants responded to our survey representing 100% of the number of approached students.

All the participants were asked if they had experience in professional testing. The purpose of this question was to determine the exposure level of the participant to professional software testing activities and to determine the impact of industrial experience on forming an opinion about software testing. From the 161 participants to our survey only 21 students (14 from group S1 and 7 from group S2) had previous testing experience such as a part time job in software companies performing testing activities. No student from group S3 had previous experience in professional testing.

Table 9 for presents the results our first question: Are you interested in working as a tester in a software development company after graduation? A vast majority of the students belonging to groups S1 and S3 were not interested in pursuing a career in testing while in the S2 group we noticed a balance between the students who were interested (17 students) and those who were not (18 students).

<table>
<thead>
<tr>
<th>Table 9: Interest in pursuing a testing career</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Undecided</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>
The Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU), conducted a survey among its master alumni in 2007 and 2011, with the purpose of investigating which educational topics had been most relevant to the alumni’s professional careers after graduation. The results enabled the identification of topics which they had to learn after graduation, and the two sets of data registered four years apart, provided us with an overview of the trends and a certain shift in the importance of the most favoured topics. The data available from these two surveys were re-analysed from the perspective of software testing and the results show that the importance of software testing activities are perceived as more important once a person has acquired industry experience.

In both the alumni surveys, Testing and QA were rated among the top ten relevant topics from around 63 proposed ones, gaining the 7th place in 2007 and the 8th place in 2011, while Programming came first in both surveys. The last question of the survey asked the participants "What three topics should have higher priority in the education [that the department offers], based on the trends you see in the workplace?" Testing came in second place on the list of topics that 10 or more people suggested as requiring more focus in the 2011 survey while it was mentioned eight times during the 2007 survey.

4.9 F9 Motivating and de-motivating factors influencing the students in engaging in a software testing career

In the study that focused on bachelor and master level students we asked the participants to provide arguments for their answer which enabled us to retrieve positive and negative factors influencing the students in their career choice. In addition we created separate groups for people who were undecided about their choice or who express interest in testing under certain conditions. This group represented 23.6% of the participants, from which 17 required more information in order to be able to reach a decision and another 17 of the participants were interested in working in testing as long as the testing responsibilities were part of a varied task set and not the sole responsibility of the job. The type of software to be tested was a condition mentioned by some participants from S1 with game design mentioned as a favourite.

F9.1 Negative aspects of working in software testing from students perspective

From the negative factors associated with working in software testing, the element of boredom stands out in all the three categories of participants. In the Boredom category we included all references to boredom, boring, monotonous, not exciting, one sided job and repetitive. Table 10 illustrates the most frequent negative factors associated with the work responsibilities of a tester. We decided to leave out factors which were quoted less than eight times over all sources.
For group S1, representing the Bachelor students, we observe that almost half of them (48.61%) consider testing work to have an element of boredom. Around a fifth of the participants (18.06%) are more interested in core development as we can see from the Rather writing code category, a factor which is followed closely by the lack of creativity presented in Not creative category (16.67%). Low status and Unrewarding aspects of a tester's responsibilities are quoted by 8.33% of S1 participants. One participant argued his lack of interest by the statement "testing can often have little impact on the finished product" (student, group S2) This opinion was corroborated by a series of statements from other students which were in concordance with the preconception of testers being seen as second class citizen in the software development world: "does not seem interesting to not be able to influence the development" (student, group S1) and "test is completely at the bottom of the ranking" (student, group S1). Another uninspiring aspect aggregated under the Unrewarding was the assumption that a software tester will be paid on a lower scale than a software developer. The financial aspect was raised again in the section of motivational factors.

The majority of people in the group S2 (56.25%) was more interested in writing their own code (Rather writing code), which they found incompatible with the testing job responsibilities, combined with concerns of not being able to gain a position as a developer once they join the working market as testers "Becoming tester tends to make it less likely to become part of the core department" (student, group S2). The lack of creativity was not mentioned by the participants from the second group, while the issue of Status was mentioned by three students "it looks like the testers work at the end of the project with limited time and get blamed for every defect they fail to find" (student, group S2). The Unrewarding category followed by the Unrewarding category.

Compared to the previous groups, in S3 there is no clear dominating negative factor with the two main ones coming in at a similar level: Boring (41.67%) and Rather writing code (37.5%). One participant mentioned the responsibility for the software which comes with testing as a deterring factor for him "Testing means a lot of responsibility, if something does not work after testing, it is your responsibility" (student, group S3). The issue of No creativity was also mentioned by 13.04% of the group. One student mentioned Low Status and Unrewarding as negative aspects of software testing. "I like to write it on my own. Others' code doesn't give you credit and thankfulness” (student, group S3).
F9.2 Positive aspects of working in software testing from students perspective

In comparison with the group of negative factors which were present in all student groups, the positive factors (see Error! Reference source not found.) were not homogeneously distributed. The Interesting category includes all the statements which mention software testing as an interesting occupation. Under the Importance category we included all the statements which provided acknowledgement of the importance of testing, such as "Improving a system", while under the Experience category we aggregated the references of gaining experience and/or knowledge during testing "It would be interesting and instructive to work with other people's code and think about it in a non-traditional way" (student, group S1), "such an experience helps make me a better developer" (student, group S3). We noticed that a high percentage of the participants acknowledges the importance of testing, especially group S2 (20%), the participants of the Testing Course. Some of the participants consider working with popular methodologies such as Test-driven development, (under Interest in TDD category), an alluring factor for the software testers." It is another way to think about how to develop a software when you use TDD" (student, group S3).

<table>
<thead>
<tr>
<th>Table 11: Positives aspects of working in testing</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interesting</td>
<td>13(13.27%)</td>
<td>7(17.5%)</td>
<td>0</td>
</tr>
<tr>
<td>Experience</td>
<td>3(3.06%)</td>
<td>0</td>
<td>2(8.70%)</td>
</tr>
<tr>
<td>Importance</td>
<td>5(5.10%)</td>
<td>8(20%)</td>
<td>2(13.04%)</td>
</tr>
<tr>
<td>Interest in TDD</td>
<td>2(2.04%)</td>
<td>2(5%)</td>
<td>0</td>
</tr>
</tbody>
</table>

F9.3 Motivational factors for working in software testing from students perspective

Concerning the motivational and de-motivational factors towards a software testing career we asked the participants what will motivate them to get engaged in professional software testing. By retrieving the factors which can trigger a change in the attitude towards testing we can improve the motivation among the students towards choosing the software testing profession. Also, this question allows us to perceive further motivational and de-motivational factors towards software testing.

The results show that sources S1 and S3 consider they need more information on software testing to have a qualified opinion about the possible improvements in software testing. All these statements were collocated under More Information category. "Lecture on testing / real life view of the workplace" (student, group S1), "Get a bit more about what the testing entails. Certainly been demonstrated in practice by some who work with it" (student, group S3). The fact that only one person from S2 had provided this answer conforms with our expectation, since they had attended a course about software testing and should thus have more than sufficient information already.

Group S1 consider it important to have a variety of tasks while doing testing activities which, in their opinion will also make the testing responsibilities more interesting "If you get a variety of test tasks and feel that you participate in developing a product" (student, group S1).

For group S3, the second motivator is the financial aspect of the position (Money), with 30% of the students assuming that a tester will automatically receive a lower pay than a
Results

developer. Also, for group S1 the financial aspect is considered as important, with 10 students invoking it. "Higher salaries for testers. That testing is more appreciated. The testers will be included in other parts of the process as well" (student, group S1). "Decent pay, being involved in developing the software I am testing" (student, group S3).

The issue of the Status was also mentioned by students from all three groups. Several students from groups S1 and S2 requested a better marketing for the testers’ position and reputation as well as increasing emphasis of Testing importance: "increase reputation, normal SW developers look down on testers. Include testers better in agile groups, most companies have testing still separate from developers which use Scrum/X" (student, group S3). "More emphasis on testing feedback and not only finding out if something works as it should" (student, group S1).

Get testing experience category was dominant among students from group S2, while the second place for the same group was taken by Improved testing methods category which contains the reference about innovative or new testing methods such as TDD as well as an involvement in testing automation.

<table>
<thead>
<tr>
<th>Motivational factors</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>More information</td>
<td>18(18.37%)</td>
<td>1(2.5%)</td>
<td>7(30.43%)</td>
</tr>
<tr>
<td>Variety of tasks</td>
<td>16(16.33%)</td>
<td>1(2.5%)</td>
<td>2(8.70%)</td>
</tr>
<tr>
<td>Money</td>
<td>10(10.20%)</td>
<td>1(2.5%)</td>
<td>7(30.43%)</td>
</tr>
<tr>
<td>Status</td>
<td>8(8.16%)</td>
<td>2(5%)</td>
<td>3(13.04%)</td>
</tr>
<tr>
<td>Get testing experience</td>
<td>2(2.04%)</td>
<td>10(25%)</td>
<td>0</td>
</tr>
<tr>
<td>Improved testing methods</td>
<td>4(4.08%)</td>
<td>6(15%)</td>
<td>1(4.35%)</td>
</tr>
<tr>
<td>Testing Importance</td>
<td>4(4.08%)</td>
<td>1(2.5%)</td>
<td>0</td>
</tr>
</tbody>
</table>

This thesis aims to shed light on how software testers are organized and structured within the companies and how they can be motivated in their work. We explored the policies and rules conceptualized and implemented in software development projects and we searched for the most desirable set of traits and skills. We sought to investigate when the perceptions of software testing are formed and how they evolve during the educational and professional years. We hope the results of this thesis will contribute to the body of knowledge regarding testing research in an industrial setting and provide recommendations both for industry and educational institutions.
5 Evaluation and discussion of results

This chapter contains an evaluation of the contributions of this thesis in relation to the software engineering topic. The following sections assess the implications of the results for the research community and provide a set of recommendations for the practitioners. Lastly, we discuss the main threats to validity for this study.

5.1 Evaluation of Contributions

This section presents a synopsis of the results, organized according to the contribution to this thesis. Further details and information about study approach, related research, analysis, discussions, and conclusions can be found in the original papers in part II of this thesis. Table 13 summarizes the contributions and shows how they are related to the research questions, and also shows the mapping between the papers and the contributions. Since this thesis is solely based on the supporting publications, no additional results or data are added in this chapter.

<table>
<thead>
<tr>
<th>Contributions</th>
<th>Papers</th>
<th>RQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1. Identification of challenges encountered by testers, managers and developers during software testing activities.</td>
<td>P1, P5, P6</td>
<td>RQ1</td>
</tr>
<tr>
<td>C2. A classification framework for organizing and managing testing personnel.</td>
<td>P1, P2,</td>
<td>RQ1</td>
</tr>
<tr>
<td>C3. A description of factors that motivates software testing personnel and strategies deployed by the companies for stimulating their testers. New motivational and demotivational factors were identified within testing.</td>
<td>P1, P5, P6</td>
<td>RQ2</td>
</tr>
<tr>
<td>C4. The set of characteristics and skills considered the most suitable or desirable when hiring software tester.</td>
<td>P1, P4,</td>
<td>RQ2</td>
</tr>
<tr>
<td>C5. A multi-faceted perspective of the perception of a testing career among students and graduates and factors influencing the students in pursuing a software testing career</td>
<td>P2, P3</td>
<td>RQ3</td>
</tr>
</tbody>
</table>
5.2 Results vs. Existing literature

C1. Identification of challenges encountered during software testing activities

Previous studies observing testing activities examined how the organization and the relationships between the members of a software development team are impacting the testing process and implicitly the product quality. We wanted to build on this research by determining additional factors influencing the dynamics of the testing team and the motivation behind different types of testers’ organization.

Testing practices and testers' work in real world environments have been the subject of study of a large body of literature [6], [13], [14] and [15]. A testing services survey conducted on more than 400 European organizations’ employees [27], stated as one of the main challenges the view of quality assurance as an additional cost and an activity which increases the duration of the development lifecycle. Another survey which compared testing practices from 2004 and 2009 in Alberta, Canada [20], although limited to the local IT industry, shows that testing practices in relation to separation of test and development environments has not changed much in that period. A survey of software testing practices in Australia provided good insights of software testing practices for testing methodologies, tools, metrics, standards and education [21]. This survey concluded that the biggest impediments of developing testing practices were the lack of knowledge in adopting new techniques and methods together with the costs of testing tools.

Zhang and Dhaliwal [98] looked at how the quality of the software is influenced by the relationships between testers and developers and observed that pairing one to one between developers and testers leads to positive changes like higher job satisfaction for the people involved. This finding is applicable to both agile and waterfall software development methodologies. Thus, having testers and developers knowing each other personally will bring benefits to the entire team and for the software products itself, regardless of the development methodology used.

Taipale and Smolander conducted a study [23], which looked at the software testing practices, and suggested improvements in this process based on the knowledge gathered from their study. Their improvement proposition included adjusting testing according to the business orientation of the company, enhanced testability of software components, efficient communication, and early involvement of testers [23]. The predominance of companies with no dedicated testing personnel in the organizational categories presented in this study shows a movement towards a working environment in which being a dedicated tester will automatically situate him or her at a disadvantage. The current trend of using agile development is shifting the balance towards a situation where the testers are part of the development team, or in some cases being a tester is just a secondary role for the developer.

C2. A classification framework for organizing and managing testing personnel

The study performed by Zhang and Dhaliwal [98] showed an increase in the quality of software products for organizations where testers are organized in a separate
department, within the development organization. A testing services survey conducted on more than 400 European organizations in 2011 [10] presents the results of a comprehensive application testing survey conducted across Europe, and focused on understanding the internal set-up and organization of QA and testing teams. Among the results presented in the report is the fact that one in five organizations has a decentralized and ad hoc approach to application testing.

During our research we observed that for small and medium sized companies, there is not enough ongoing work to justify the existence of a testing department or, in some cases, a dedicated testing staff. Most of these companies decided to divide the testers among the application groups. This category seems to be the middle way between having a testing department and having no dedicated testing staff. For companies that do not need a large testing department due to a low amount of testing work needed, this might be seen as a simple solution. For companies with no formal testing staff, it will represent the first stage on the road to having more organized testing activities. Regardless of the size of the company, the organization of testing activities was not always implemented in a predefined or structured way. Based on the amount of testing resources were assigned on a limited time or developers allocate certain section for their time.

C3. A description of factors that motivates software testing personnel and strategies deployed by the companies for stimulating their testers

The theory of motivation has been too little used in research on software testing and the software engineering literature has largely missed the opportunity to employ theories from psychology and cognitive science when dealing with human aspects. However, motivation in software engineering was the scope of a systematic literature review conducted by Beecham et al. [45], and in a subsequent study by Franca et al. [46]. The latter study also shows that two of the motivators discovered in [45] were no longer present: appropriate working conditions and sufficient resources. The change noticed in the motivators and de-motivators illustrates the evolving nature of the motivation of software engineers and this is expected to change even more as the software engineering field is evolving.

We provided a set of factors with negative and positive influence on the daily activities of software testers and added additional categories to the ones already presented and published in the software engineering world. We look at the differences between testers working in traditional and agile development and noticed a higher degree of stress and a more positive approach towards the challenges of testing activities for those engaged in the waterfall approaches, while the agile testers, although expressing more problems in communication with developers seemed to be better integrated into their teams.

The thesis shows that testers enjoy having variety in their work, while the lack of influence and recognition was considered a major negative factor for most of the participants involved in this study. Challenges specific to the nature of testing activities where emphasised in our research. Technical issues occurring within testing context involve large quantities of effort and time invested in items which should be readily available at the beginning of testing. The Time pressure challenge refers to the tendency
of the time available for testing to shrink from the original estimate until the actual testing execution period is taking place.

C4. The set of characteristics and skills considered most suitable or desirable when hiring software testers

Pettichord provides a list of desired traits compared between testers and developers [99], while [100] suggests concrete, desired skills. Both studies acknowledged the need for strong knowledge both of software development and the application domain of the product. The importance of experience in testing was also underlined by Beer and Ramler [30] and Taipale and Smolander [36]. Domain knowledge and specialized technical skills were considered more relevant than test case design and test planning skills in a study concerning high performing testers [101].

The systematic literature review conducted by Beecham et al. [58] indicates that several factors influence the software engineers' motivation and one of these factors is related to the personal characteristics of the software engineer. Our findings describe the characteristics which are perceived as most valuable for a tester, based on the working experience accumulated by the participating testers and testing managers.

In contrast to existing work on software engineers' characteristics it is noticeable that the most frequent cited characteristics for software engineers were not applicable to software testers. A strong emphasis was laid on communication skills and IT background which implied programming knowledge and certain technical capabilities. Testing knowledge and domain knowledge were also valued but not as strongly as found in previous research conducted by Beer and Ramler [30]. Testing experience was considered to be an advantage especially for the testers involved in domains with volatile requirements. This finding seems to be in concordance with the result of previous studies [30], [32].

C5. A multi-faceted perspective of the perception of a testing career among students and graduates and factors influencing the students in pursuing a software testing career

The gap between software education and the industry needs was identified and reinforced in a series of worldwide conducted studies. Analysis of the practice in the US Academia conducted by Astigarraga [102] and Lethbridge [9] emphasise the need for improvement of the current curriculum and more focus on software testing. The study performed by Astigarraga et al. [102] combines a presentation of testing as an industry profession, a survey of the software testing curriculum in US and a discussion on the efforts involved in increasing the status of testing in the academic curriculum. The study also highlights the low number of Software Testing journals and societies in comparison with other areas of software engineering.

The results of this thesis indicate the need to improve the perception of a testing career by changing the curriculum and reviewing some of the curriculum parts with testing professionals with the goal to retrieve the educational topics relevant to the field of testing expertise. By organizing guest lecturers with software professionals having
strong testing careers and being involved in challenging projects, we can increase the awareness and stimulate interest with testing activities.

However, we should be aware of the differences between academia and industry, and accept that it will be nearly impossible to definitively close the gap between software education and the industry needs. Universities should strive to provide the best education in accordance with the latest technologies and updates from science, but the core of the education will be principles and theory. Another aspect that needs to be taken in consideration is the lengthy process of implementing a change in the curriculum in academia.

An additional way of improving the curriculum is to regularly run surveys among university alumni where they are asked to rate the relevance of the topics thought in the university. The reports based on this survey can help in improving the curriculum by identifying the educational topics which proved to be of high importance and reducing the curriculum for certain courses.

### 5.3 Recommendations to practitioners

The results obtained from this research aimed to shed light on how testing activities are organised and how professional software testers and computer science students can be motivated. We also explored the policies and rules created and implemented inside software development projects.

**Accurately assess the required testing specialty**

The list of characteristics suited for a software testers provided by our study should not be seen as exhaustive or definitive but rather as a starting point. Each company should consider the type of knowledge they value and require and, at the hiring stage, assess the ability to provide internal training for testing methodology, testing tools or domain knowledge relevant for their business.

**Invest in experience and professional testers**

Testing should not be seen as a secondary skill for a developers but a profession in its own right and with specialization in sub domains, such as performance testing or usability.

In order for a project to be successful, managers need to ensure smooth collaboration between requirement engineers, testers and developers. It is of paramount importance that testers are seen as belonging to a specific field and not a mobile unite who can shift from requirement engineering trough testing and development.

**Establish a significant place among the stakeholders**

Although there is an increasing awareness on testing benefits such as cost reducing for finding defects at an early stage, testers are still seem as a cost centre rather than beneficial teams.
5.4 Limitation and threats to validity

Human factors cannot be considered in isolation and on the individual level but are affected by sociological, organizational factors and other environmental factors [46]. The results of our study should be treated with some caution since there are other factors which may impact the motivation of a tester such as the organization structure, internal policies and processes. In addition, motivation can be influenced by human factors such as personality types [103], and individual characteristics such as age [104].

Conclusion validity

Threats to conclusion validity arise from the ability to draw valid conclusions based on the collected data. The qualitative and quantitative surveys were conducted in several companies and in case there were a group of people to be interviewed, all the interviews were performed in one session. Hence, we avoided that answers might be influenced by internal discussions. To ensure interviews of high quality, several pilot interviews were conducted at the study initiation, in order to avoid poor phrasing or structure for the interview or survey guideline. After each study stage we provided feedback to the participant companies and discussing the results helped validating my conclusions.

Internal validity

Internal validity threats are related to matters that may affect the causal relationship between treatment and outcome. Threats to internal validity include instrumentation, maturation, and selection threats. The potential problem of instrumentation threats was attenuated by developing a research instrument with close reference to the literature which relates to quality requirements, and was influenced by a previously validated interview instrument and a previously piloted interview study [58]. By collecting background information about the participants we were able to perform the interview session in approximately 60 minutes, which alleviates maturation threats.

Threat to selection bias is always present when a study subjects are not fully randomly selected. Engagement of industrial partner for research is a typical challenge in the SE community when engaging industrial partners for participating in research programmes. One of limitation in this research is the small number of participant companies and the selection criteria.

A bias which is associated with qualitative data collection is selecting relevant candidates for the scope of the study. In order to avoid the threats to internal validity we used data source triangulation by interviewing multiple roles at a company.

Construct validity

The construct validity is concerned with the relation between the theories behind the research and the outcome. In our research we used semi-structured interviews, including open-ended questions where the participants are encouraged to express their own opinions. The interview guidelines were developed in several iterations, and based on previous literature [58]. During the qualitative survey interviews, the respondents were encouraged to express their opinions freely, by guaranteeing their anonymity and assuring them that the records will be accessible only to the researchers involved in this study, hence alleviating the potential problem of evaluation apprehension [105].
In order to mitigate the risk of identifying incorrect factors in this kind of research, we ensured observer triangulation by having the data analysed by three researchers in all the studies involved, see Robson [78].

**External validity**

The external validity is concerned with the ability to generalize the results [86], which for this study translates into the applicability of the findings beyond the participating interviewers, companies or specific time frames. Qualitative studies focus by nature on explaining and understanding the situation under study, rather than generalizing the results. The nature of qualitative design itself makes it impossible to replicate a study, due to the difficulty with reproducing the same context, but it can lead to a theory which can provide understanding for similar cases and situations. Generalizability can be difficult to achieve in software engineering due to the nature of the context [106], however, our qualitative analysis spanned companies using traditional and agile methodologies, performing functional and non-functional testing, which could give better generalizability than performing interviews in just one company. However, the findings may be generalized to companies with similar characteristics as the participant companies by theoretical generalization [75].

Since several of the participants from our studies pointed out the same set of the discovered factors, it increases the possibility of transferring the results to other situations or companies. Even if other companies in other locations, at a later time might encounter different challenges with testing, and use different strategies for motivating testers, it still could be useful also for these companies to be aware of the experience faced by companies which participated in our studies. To avoid the interaction of selection and treatment, interviewees were selected according to their roles within the company by an internal representative; the researchers did not select the subjects themselves. Moreover, the companies selected belonged to different geographical locations and different industry areas.
6 Conclusion

This thesis aims to shed light on how software testers are organized and structured within companies and how they can be motivated in their work. We explored the policies and rules defined and implemented in software development projects and searched for the most desirable set of traits and skills. We wanted to understand when the perceptions of software testing are formed and how they evolve throughout the educational and professional years.

In this research we have used a mix of quantitative and qualitative data analysis. The quantitative data have been used for constructing an overview over the internal organization of testing activities and for analyzing the results from alumni survey data sets. The qualitative data analysis has been utilized to understand and explore the many factors we observed during the research on motivation and characteristics of testing professionals. We hope the results of this thesis will contribute to the body of knowledge regarding testing research in an industrial setting.

The purpose of this chapter is to briefly outline the contributions of the reported research. In addition, potential directions for future research are outlined.

6.1 Contributions

This research focused firstly on improving our understanding of the human factors which testing related personnel faces during their activities and explore the motivational factors behind a testing career. The research explored the elements impacting the testing activities and examined how testing activities are organized and conducted. The result offers insight on how organizations decided and adopted a guideline for testing activities.

Furthermore, the research provides a comprehensive description of the factors that influence the software testing personnel. In addition, a new set of motivation and de-motivation factors were introduced which allowed us to provide an answer for RQ1. Through the second research questions RQ2, we aimed to assess how the motivational factors for software testers are used by software companies and we explored and presented the tactics adopted by companies for motivating their employees and stimulating their productivity.
In addition, we performed a multi-faceted perspective of perception of the testing career among students and graduates, and enabled us to retrieve an answer for RQ3. Lastly, the results improve our understanding of the mechanisms influencing the students for pursuing a testing career.

The main contributions of this work are the following:

C1. Identification of challenges encountered by testers, managers and developers during software testing activities (P1, P4, P5, P6)

C2. A classification framework for organizing and managing testing personnel (P1, P4)

C3. A description of factors that motivate software testing personnel and strategies deployed by the companies for stimulating their testers. We identified new motivational and de-motivational factors within the testing context (P1, P5, P6)

C4. A set of characteristics and skills considered suitable or desirable when hiring software testers (P1, P4)

C5. A multi-faceted perspective of the perception of a testing career among students and graduates and factors influencing the students in pursuing a software testing career (P2, P3)

6.2 Future Work

Based on the results from this thesis we recommend a set of proposals for further study in academia and industry.

Extended or longitudinal study of factors motivating testing personnel

The research on testing personnel motivation is in its budding stage. The findings of our work should be used as a basis for a broader questionnaire survey which could reach a high number of software testers in Norway or even expand to an international level. It will be interesting to investigate whether companies have used any of the strategies for motivating testers or not, and whether testers in the companies that had used such strategies were better motivated than others.

Another possibility will be to use action research, by trying to go into selected companies with a perceived need to improve testers’ job satisfaction, using some of the suggested strategies, and then evaluating whether they were successful. This work has highlighted differences between testers’ job satisfaction in agile and traditional teams. Future studies should target the differences in motivational factors between agile and traditional teams. A longitudinal study may provide further insights into the motivational and de-motivational factors of software testing personnel.

Integrate testing curriculum in software development courses

Both academia and industry must work to raise the interest for software testing to young people and by redefining the definition of testers’ responsibilities and highlight the
career path opportunities. The curriculum should increase the focus on testing tasks and responsibilities which combine classical development roles such as creating and writing testing automation tools. If testing is emphasised from the early stages of education, testing activities will be perceived as a natural part of a software development lifecycle.

**A focused study on alumni’s need for testing competence**

The alumni surveys conducted at NTNU were not originally made to investigate the relative importance of testing in the careers of the alumni, but rather to investigate their needs for a wide range of competencies, including testing. Although the surveys had more than 100 respondents each, we recommend strengthening and expanding the investigation of alumni's need for testing competence with a targeted questionnaire survey made for this specific purpose.

A specific survey will give the opportunity to ask more detailed questions related to the participant background, and job responsibilities. Another advantage of a more targeted survey is that it might ask more specialized questions on testing itself, distinguishing between different types of testing competence, test management, usage of specific testing tools or testing methods.

**6.3 Concluding remarks**

Testing has received increased attention in the software engineering field. However, the literature has predominantly focused on new technologies and new methods for testing and little on how testing is conducted under real-world circumstances. Many of the problems encountered in testing are related to both work and organisation, so improving software testing should be seen as a socio-technical challenge.

This thesis offers insight on how organizations decided and adopted a guideline for testing activities. Furthermore, the research provides a comprehensive description of factors that influence the software testing personnel and a new set of motivation and de-motivation factors were introduced. Our study shows that the perception of success and motivational factors varies between stakeholders; however, constructing a strong team requires members with diverse experience and complementary skills.

In addition, a multi-faceted perspective of perception of the testing career among students and graduates emerged. Lastly, the results improve our understanding of the mechanisms influencing the students for when deciding whether to pursue a testing career. Also they could serve as a guideline for deciding and implementing changes in curriculum while taking into consideration the motivators and de-motivators of students.

In recent years, software testing has gained acknowledgement in industry and focus in the academia. However, as software grows more complex so do the challenges and despite the awareness of this process’ necessity and importance, more emphasis is laid on minimizing project costs and duration. Hence, it is essential that both industry and academia come together and agree on enhanced research and acknowledge the testing field as an intricate social-technical process.
7 References


34. Shah, H., Harrold, M.J., Sinha, S.: Global software testing under deadline


46. Amabile, T.M.: Motivational synergy: Toward new conceptualizations of


94. Mann, C., Maurer, F.: A case study on the impact of scrum on overtime and
References

customer satisfaction. Agil. Dev. Conf. 70–79.


Appendix A: Selected papers

P1  Deak, A., Stålhane, T. "Organization of Testing Activities in Norwegian Software Companies", 2013 IEEE Sixth International Conference on Software Testing,


P3  Deak, A., Sindre, G. "Analyzing the importance of teaching about testing from alumni survey data" NIK Norsk Informatikkonferanse 2013 s. 100-109


P6  Deak, A., Stålhane, T., Sindre G., "Challenges and strategies for motivating software testing personnel" Accepted at Information and Software Technology journal
P1 Deak, A., Stålhane, T. "Organization of Testing Activities in Norwegian Software Companies", 2013 IEEE Sixth International Conference on Software Testing,
Organization of testing activities in Norwegian Software Companies

Anca Deak, Tor Stålhane,
Norwegian University of Science and Technology
Trondheim, Norway
deak@idi.ntnu.no, stalhane@idi.ntnu.no,

Abstract — Software testing plays a major role in determining the quality of software products. Testing activities are influenced by the available methods and tools, but also by the non-technical aspects of the software development process. The scope of the study is to explore how testing activities are organized in Norwegian companies, and factors that influence the creation of a testing department, or encourage the investment in personnel dedicated to testing. The research was performed using a combination of qualitative and quantitative methods and started with an initial survey which comprised questions with multiple choices as well as open ended questions. The next stage included interviews and an additional round of open ended questions. Our study found four organizational categories in which testing activities can occur, combined with observations on the correlation between testing automation activities and testing team structure. We were interested in retrieving the factors that influence and determine a certain organization of the testers in software producing companies. This study provides a snapshot of the current trends of organizing software testing activities and it is a stepping stone into research of the testing practices and testers’ experience.

Keywords — software testing, testing practices, test automation, testing organization

I. INTRODUCTION

The role and importance of software testing as a critical stage in the development of any useful software system is being today acknowledged by the industry [1]. While software testing field is a widely researched topic and although there is extensive literature on software testing as a generic activity, there is still a demand for more information on how software testing is performed in the real world, as underlined in a study performed by Rooksby et al. [2].

Testing practices are built mostly on the experience gained in the domain, rather than on a rigorous and systematic approach as stated in [3] study. Although the testing problem tends to be perceived as mostly technical, Ahonen et al. [4] presents three case studies which show that organizational factors have a high impact in the testing process. In order to improve the testing activities, it is important for the researchers to understand better the practices and challenges of testing in software development industry [2]. Briand and Labiche's study on software testing techniques concluded that observing the activities of professional testers will enable us to retrieve the most applicable results [5].

Myers has pointed out the importance of psychology, intellectual manageability, economics and discipline as strong determinants for successful software testing [7]. Despite his strong case, that the success of software testing has little to do with purely technical considerations, the nontechnical concerns have been left neglected in favor of those focused on technological issues. In a study looking at the management of conflicts in the world of software testing, Cohen et al. stipulates that the result of testing "ultimately depends on the interpersonal interactions of the people producing the software." [6].

Test personnel and test automation have been suggested as the most costly items for testing activities [8]. Improving the testing activities has positive economic outcomes by spending less on defects at later stages as mentioned in [9] and [10]. Although there is much awareness for the necessity of improved quality of software testing, studies such as [11] indicate a low level of test maturity in software development organizations.

The objective of this study is to identify the daily challenges encountered in real world of Norwegian software organizations and by better understanding the difficulties faced by testers in their activities, to derive at further research hypothesis on how these difficulties can be addressed. We are focusing on the relationships and organization of the testing personnel by investigating several factors that influence the work and the general perception of what a tester’s work might imply. In order to suggest improvements, it is mandatory to first see and understand what is happening today in the software development industry, and to retrieve a snapshot of the current trends of organizing software testing activities.

An additional aim of this study is to observe the possible connections between automation of software testing activities [12] and testing organization. By test automation we refer to the use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions [28]. Exploring the possibility of a link between testing activities and automation allows us to retrieve the factors which are impeding a high degree of automated process, while improving the automation process will lead to shorter testing times and consequently to lower costs.

The rest of the paper is organized as follows: Section 2 presents the background literature, while the research questions, research design and data collection process are described in Section 3. The results of the study are presented and examined in Section 4. In Section 5 we examine the
findings of the study and discuss the implications and the future work for this research.

II. RELATED RESEARCH

Previous studies observing the testing activities examined how the organization and the relationships between the members of a software development team are impacting the testing process and implicitly the quality of the product. We wanted to build on this research by determining additional factors influencing the dynamics of the testing team and the motivation behind different types of testers’ organization.

Testing practices and testers’ work in real world like environments have been the subjects of study of an increasing body of literature [6], [13], [14] and [15]. However, many of these studies tend to focus on the technical perspective of testing, by analyzing or demonstrating the benefits of specific methods or approaches to testing. In some case, the focus of the studies is mainly on theoretical aspects of software testing practices [1], [16].

Automation, testing usability and testability are nominated as the main challenges for good testing practices [10] while an emphasis on automation as an important element for the improvement of testing activities is reconfirmed in [3]. A testing services survey conducted on more than 400 European organizations with more than 250 employees [27], stated as one of the main challenges the negative view of quality assurance as an additional cost and an activity which increases the development life.

There are not many studies focusing on the testers or on testing teams, although it is already acknowledged that some of the problems associated with software testing in the industry are not technical, but originate from other sources, such as the socio-technical environment and organizational structure of the company [2], [6], [7] and [13]. Given the nature of the software testing and development work, the conflicts between software testers and developers are inevitable and widespread, impacting the software quality and job satisfaction [6], [17]. The source of these conflicts and their implications for research and practice are discussed in [17].

The continuous trend of developing embedded software and security related web application is increasing the number of products which require high quality [18]. This trend should push the software development world into developing better practices, but an industrial survey of aspects of testing presents differences between the current practices and the perceptions of participants [19]. Another survey which compared testing practices from 2004 and 2009 in Alberta, Canada [20], although limited to the local IT industry, shows that testing practices in relation to separation of test and development environments has not changed much in 2009. A survey of software testing practices in Australia provided good insights of software testing practices for testing methodologies, tools, metrics, standards and education [21]. This survey concluded that the biggest impediments of developing testing were the lack of knowledge in adopting new techniques and methods together with the costs of testing tools.

A study performed by Zhang and Dhaliwal [22] showed an increase in quality of software products for organizations where testers are organized in a separate department, within the development organization. The same study looked at how the quality of the software is influenced by the relationships between testers and developers and observed that pairing one-to-one between developers and testers leads to positive changes like higher job satisfaction for the people involved. This finding is applicable to both agile and waterfall software development methodologies. Thus, having testers and developers knowing each other personally will bring benefits to the entire team and for the software products itself, regardless of the development methodology used.

Taipale and Smolander conducted a study [23], which looked at the software-testing practices, and suggested improvements in this process based on the knowledge gathered from their study. Their improvement proposition included adjusting testing according to the business orientation of the company, enhanced testability of software components, efficient communication, early involvement of testers and increased interaction between developers and testers.

III. RESEARCH QUESTIONS AND METHODS

The primary goal we explore in this study is to retrieve the factors that influence and determine the organization of the testers in the software company, while the second objective was to explore the possible correlation between the testing organization and automation activities. By addressing the following two main research questions this study aims to achieve the research objectives mentioned as our main goal. Additional questions were added for deeper analysis of the result.

- **RQ1:** Why have the companies chosen to organize testing in a particular way?
  - **RQ1.1:** What different ways of organizing the testing personnel can be found in Norwegian software companies?
  - **RQ1.2:** Who is responsible for performing testing activities?

- **RQ2:** What is the relation between the organization of testing activities and the testing automation process?
  - **RQ2.2:** What is the degree of involvement and experience in testing automation?

The study was performed in two stages using a combination of qualitative and quantitative methods where each method supported the other. During the first stage we initiated contact with the software development companies, and started our study with an introductory survey. The use of questionnaires for the first stage made it possible to quickly reach a larger number of companies and thereby get an overview of the situation in Norway. On the other hand, since questionnaires have shortcomings for providing deeper insights, we decided to use mainly interviews but also phone discussions and follow up emails for the second stage. The data collected was analyzed by organizing it from several perspectives, with our main criteria being the organizations of the testers, while the second criteria related to testing managers and automation.

Since our study includes companies within several business domains, ranging from small companies to multinational ones, we decided to use the organizational unit as a measuring unit.
An organizational unit is defined as a part of an organization that deploys one process, has a coherent process context and set of business goals [24]. We will use the term company as a synonym for organizational unit in the remaining of this paper.

A. Participant selection

From an initial sample of 22 companies we collected data through a total of 23 participants from the 19 organizational units involved, by using interviews, surveys and follow up emails. Three companies declined to participate due to internal policy and security related regulations. The participants from the companies involved included project managers, testing managers, testing leaders and consultants. A categorization of the participating companies based on size and on the business in which they are involved is presented in Table 1. The classification of the companies’ sizes was done according to the SME Definition [25].

<table>
<thead>
<tr>
<th>Table 1: Classification of companies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Large/international companies</strong></td>
</tr>
<tr>
<td>Org. A Testing manager</td>
</tr>
<tr>
<td>Org. B Senior Software Engineer</td>
</tr>
<tr>
<td>Org. C Project Manager</td>
</tr>
<tr>
<td>Org. E Test Lead</td>
</tr>
<tr>
<td>Org. F Test Lead</td>
</tr>
<tr>
<td>Org. G Software producer</td>
</tr>
<tr>
<td>Org. H Software producer</td>
</tr>
<tr>
<td><strong>Large/national companies</strong></td>
</tr>
<tr>
<td>Org. I Software producer</td>
</tr>
<tr>
<td>Org. J Development Manager</td>
</tr>
<tr>
<td><strong>Medium/international companies</strong></td>
</tr>
<tr>
<td>Org. K Testing manager(2)</td>
</tr>
<tr>
<td>Org. L Development Manager</td>
</tr>
<tr>
<td>Org. M Quality Assurance Lead</td>
</tr>
<tr>
<td><strong>Medium/national companies</strong></td>
</tr>
<tr>
<td>Org. N Testing managers (3)</td>
</tr>
<tr>
<td>Org. O System Development Manager</td>
</tr>
<tr>
<td><strong>Small/national companies</strong></td>
</tr>
<tr>
<td>Org. P Senior Software Engineer</td>
</tr>
<tr>
<td>Org. Q Project Manager</td>
</tr>
<tr>
<td>Org. R Development Manager</td>
</tr>
<tr>
<td>Org. S Project Manager</td>
</tr>
</tbody>
</table>

The first sets of questions of the introductory survey are intended to establish a profile for each participating company. In this section we enquired upon their working platforms and the products and/or services offered. We were interested to find out if they decided to use only one software development method or a collection of methods adopted based on the company’s needs.

The objective of the Testing organization and management section is to determine the representative of the testers inside the company and the parties involved in the management of different testing stages. The testers sections of our questionnaire is meant to retrieve information about who is performing the testing activities in each company and the person(s) responsible for performing testing activities, the roles and responsibilities of these persons. Testing activities can be performed by testers, developers or in some cases persons with business knowledge rather than technical background. Also, the questions from the section allowed us to retrieve the testing teams or of the testing pool size.

We wanted to determine if the company is using automation of testing activities and for those who were involved in automation process we asked information about the tools and the experience they have with automation. We considered experience to be based on the number of years they were involved in automation and the percentage of the software testing which was automated. For the companies involved in automation process we enquired on how much of their testing is automated and from how long they have been involved with testing automation. The list of questions from the questionnaire is available in Appendix A.

C. Interview study design

The follow-up stage was performed using mainly interviews as well as phone discussions and follow up emails with open ended questions. After analyzing the results from the first stage we identified further topics of interest and four organizational categories, see Table 3, in which the companies could be classified. Based on these results, we constructed an interview guideline for on each organizational category.

The mapping between Research Questions, topics and questionnaire’s questions

<table>
<thead>
<tr>
<th>Table 2: Mapping between RQs, topics and questionnaire’s questions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RQs</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>RQ1.1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>RQ1.2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>RQ2.2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The mapping between Research Questions, topics and questionnaire’s questions referring to the evaluation of testing activities and testers and handling of challenging situations. The questions from Testing career path sectioned focus on the options and opportunities of a career path in testing on a managerial or technical path and
the training opportunities available to the testers in the company, while the questions from testing activities aimed towards building a picture of the daily life of the tester and the relationship with the development team.

The second stage interviewees were the same persons who had responded to the questionnaire. In some cases the participant in the first round suggested additional people for the interviews, which were considered in closer relation to testing activities from their company or having relevant knowledge and experience in testing field and could provide valuable feedback.

IV. RESULT AND OBSERVATIONS

The results of this study are organized and presented based on the research questions. The first section presents the retrieved organizational categories of the testers, followed by section dedicated to automation of testing activities.

A. Testing Team Structure

The study provided us with an answer for RQ1.1, which showed us that the members of a company who are performing testing activities tend to be organized in one of the four categories shown in Table 3. Our results showed that none of the participant companies is outsourcing testing activities to external companies, although four companies prefer to have remote testing teams onshore or offshore.

Table 3: Testing activities - organizational categories

<table>
<thead>
<tr>
<th>Organizational categories</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A dedicated software testing department (Test Dept.)</td>
<td>4(21,05%)</td>
</tr>
<tr>
<td>Combination of testing department and testers are divided among application groups (Mixed Org.)</td>
<td>2(10,53%)</td>
</tr>
<tr>
<td>Testers are divided among application groups (App.Gr. Testers)</td>
<td>8(42,11%)</td>
</tr>
<tr>
<td>Developers are solely responsible for testing activities with no formal testing staff in the company (Dev. As Testers)</td>
<td>5(26,32%)</td>
</tr>
</tbody>
</table>

a) A dedicated software testing department

Four participating companies have a dedicated testing department. In one of these companies the recognition of the importance of testing has grown over time and the centralizing of the testers was done as an upgrade from the previous organization, where testers were located in several groups and departments. To improve their performance and organization and the focus on testing activities, a separate testing department was created. *We were a large group working for a customer who was focused on quality. We had frequent releases and we were good at it. Before testers were located in groups and departments and to improve our performance and focus on testing we created groups with project managers and testers.* (Testing Manager 1, Org. M.) The same company is organizing informal and formal training sessions internally and externally with regular meetings organized solely for knowledge improvement. This company has a dedicated position for organizing and promoting the training sessions and events. *Our company has a sharing philosophy. We organize brown bag lunches and we have regular sharing knowledge short presentations.* (Testing Manager 3, Org. M.) Part of our company culture is to attend conferences and courses. (Testing Manager 2, Org. M.)

In another company, the creation of a dedicated testing department was the result of lobbying activities by testing managers, combined with a motivation to improve product quality, which had not been satisfactory in previous projects.

Two of the companies which have a dedicated testing department have developed a knowledge sharing culture by promoting informal and formal knowledge sharing sessions. One of these companies is supporting the testers in obtaining testing certification by ensuring training sessions and courses and encouraging the testers to take the exams.

b) Combination of testing department and testers divided among application groups

Two of the participating companies have a combination of testing departments and testers being divided among the application teams. The first company is using Scrum, and in this case, testing is solely the responsibility of the development team.

c) Testers are divided among application groups

Eight of the participant companies have their testing staff distributed among the applications teams. In this way the testers can work closely with developers and it's a preferred choice in small or medium sized companies.

Two of the companies where testers are divided among the application groups, have express awareness of the need of a separate testing team and have plans of creating one in the future. They also indicated that they have plans for improvement in this direction, but no concrete dates or information was provided. The same company currently chooses to hire temporary testers for larger projects. *We are planning to have a testing team sometime in the future. We will hire a consultant for test activities in connection with one of our larger projects* (Development Manager, Org. K.) The use of agile methodology and scrum motivate another company to organize their tester as part of the application based in the "Whole team" agile practice.

d) No dedicated testing staff

Five of the participating companies decided not to invest in dedicated testing personnel. We enquired these companies upon the factors which lead to this decision. Two of the participants invoked insufficient amount of testing work to justify a dedicated testing position. *We have not enough testing activities to establish a testing department* (Systems Development Manager, Org. N.)

For a young company testing was seen as a future component, which will be formally organized when they reach a more mature stage. *We are a startup company, and still a rather little group of Research & Development* (Quality Assurance Lead, Org. L.) One of the managers of a small size company was reluctant to have testers since this will lead to a more relaxed approach to code debugging from the developers side.

e) Observation on testing management

Based on our studies, we extract some observations related to the management of the testing activities. From the 19 participant companies, 10 had Application Development
Manager or the Product Manager as the person being in charge of the testing activities. This finding confirms the result presented earlier in Table 3, that the majority of testers are not organized in a centralized form. One of the companies considers Release Engineering and Documentation as the responsibility of the testers although there is no testing team and the testers are part of the application groups.

While hiring temporary testers is not a preferred practice, the developer-tester ratio varies greatly from 2:1 to 1:5. Even in the same company, the ratio can differ from one project to another, with more testers being assigned in case of a new project or product.

B. Automation

Since automation has an important role in producing a consistent testing base with reproducible results and testing steps, we asked the companies about their interest in testing automation, the time they invested in it and the tools that were used for these activities. Table 4 indicates that a high number of companies are involved in the automation process with only 5 out of 19 companies not being involved in testing automation activities.

Table 4. Results for experience vs. percentage of tests automated

<table>
<thead>
<tr>
<th>Experience/Percentage</th>
<th>less than one year</th>
<th>one to two years</th>
<th>more than two years</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dev. As Testers</td>
<td></td>
<td>Dev. As Testers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>App.Gr. Testers</td>
</tr>
<tr>
<td>50 %</td>
<td></td>
<td>Dev. As Testers (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dev. As Testers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100 %</td>
<td></td>
<td>Mixed Org.</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

For companies who are performing testing automation, we asked them about the experience, expressed as number of years, they have in implementing automation for testing activities and the percentage of the testing activities which are automated. From Table 4 we see that, although 9 of 14 companies have more than two years of experience in automation, half of the companies have automated 25% or less of their testing activities. This result is confirming the findings of previous studies on automation that although test automation is considered beneficial and adopted, it is not utilized on a large scale in the software companies [26].

Further research is needed to discover the factors that are encouraging the automation of testing activities in software development.

V. DISCUSSIONS ANF FURTHER RESEARCH

The objective of this study is to build a foundation for further research in exploring the factors that influence testing processes and through that understanding to suggest process improvement propositions that could reduce development and testing costs and improve the quality of the developed product.

As seen in the studies referred to earlier like [6], [22] and [23] and the predominance of companies with no dedicated testing personnel in the organizational categories presented in this study, we can see a move towards a working environment in which being a dedicated tester will automatically situate him or her at disadvantage. The current trend of using agile development is shifting the balance towards a situation where the testers are part of the development team, or in some cases being a tester is just a secondary role for the developer.

In response to our first main research question we observed that for small or even medium sized companies, there is not enough ongoing work to justify the existence of a testing department or, in some cases of a dedicated testing staff. From a certain perspective, we can even consider having a testing team as a luxury asset. Most of these companies decided to divide the testers among application groups. This category seems to be the middle way between having a testing department and having no dedicated testing staff. For companies that do not need a large testing department due to low amount of testing work needed, this might be seen as a simple solution. For companies with no formal testing staff, it will represent the first stage in having more organized testing activities.

The automation process leads to high costs for maintaining and updating the tests, while another downside is the cumbersome process of maintaining and updating the documents of a software process. This high cost was one of the reasons why the automation process has not advanced more in the participating companies.

The growing complexity of the automation tests over time, can lead to a decrease in the enthusiasm for automation. No clear, co-relation between the organization of testing activities and the testing automation process has been detected in this study, but given the small sample size which is a clear limitation of the study is, caution should be taken in concluding it as an answer to the second main research question. Regarding the sub-question referring to the degree of involvement and experience in testing automation the need of automating testing activities is acknowledged but more involvement seems to be required. Most of the participant companies kicked off this process but lost their enthusiasm rather quickly with only a small percentage of automation being achieved. For all the companies who do not have a specialized testing staff and are involved in automation, the testing is done by developers, which find it was easy to adopt strategies like TDD or to build their own automation scripts, in comparison with a testing team where not everyone has a technical background. Three of the companies which have a dedicated testing staff were also involved in automation. These companies are part of large organizations which permitted the creation of teams specialized in automation of tests.

We observed the current trends in the field of software testing organizations and we are interested in what will happen with the traditional organization of testers in testing departments. The findings of this study might not be generalized to a higher level due to the small sample size, but given the reduced number of companies with dedicated testing personnel, it indicates that in overall, testing still seems to be considered as a secondary position among software developers.

This study is the start of a wider and deeper research into the factors which influence testing teams' organizations, and to understand the elements which contribute to creating the image of the testers inside a software development team organization.
REFERENCES


APPENDIX A: SURVEY INSTRUMENT

First and last name:
Your position in the company:
Can you describe your responsibilities?
1. Development methodology used:
2. On which platforms do you test?
3. Can you describe the types of software products developed and/or tested at your company?
4. Who is primarily responsible for testing activities inside the company? A) A dedicated software testing department B) Testers are part of the application groups C) No formal software testing staff or department D) Testing activities are outsourced
Additional comments:
5. Which job functions is your software testing group responsible for?
A) Software testing B) Quality control and assurance C) Software development D) Other (please specify): 6. To whom does the testing team of a project report? A) Application Development Manager B) Product Manager C) Senior Executive D) Other (please specify):
7. Who is primarily responsible for prioritizing software defects?
A) Testing team and testing manager B) Application development manager C) Customers D) Other (please specify):
8. Which metrics are you using to determine the state/quality of the software?
The testers
9. What is the number of testers belonging to the testing group?
10. How many temporary testers are employed?
11. What is the number of developers who perform testing activities in addition to development?
12. Do the testers receiving formal training in?
A) The technologies/ systems used at your company B) Testing methodology/processes/tools C) Troubleshooting/problem solving
13. What are in your opinion the five most important qualities for a software testing person?
14. What is the ratio for testers-developers in a typical project?
15. Are testers assigned to a project in the same location as the developers?
Automation
16. Is your company using automated testing?
17. Which tools are used for automated testing?
18. For how long have you used these automated tools?
A) Less than one year B) One to two years C) More than two years
19. What percentage of your testing process is automated?
FACTORS INFLUENCING THE CHOICE OF A CAREER IN SOFTWARE TESTING AMONG NORWEGIAN STUDENTS

Anca Deak, Tor Stålhane, Daniela Cruzes
Norwegian University of Science and Technology
Trondheim, Norway
deak@idi.ntnu.no, stalhane@idi.ntnu.no, dcruzes@idi.ntnu.no

ABSTRACT
There is an identified need of software testers in the industry with the required number of testers being expected to increase, yet the future candidates looking for positions in software industry tend to lean towards software development coding rather than software testing. In the same time, there is a gap between the education and industry needs with graduating students not being fully prepared or having the required skills when entering the testing industry. In addition to these factors, we noticed a reluctance towards this professional choice among students. In this paper we look into the elements influencing the students when making a career choice in the direction of software testing.

KEY WORDS
Software testing, testers, career, education.

1. Introduction
The continuing growing complexity of IT delivered systems triggers a stronger request for professionals and highly performing software testers. However, at the same time the industry is impacted by the lack of skills and insufficient formal education of the young graduates who enter the software engineering working field. The gap between software education and the industry needs was identified and reinforced in a series of worldwide conducted studies in Australia [1], South Africa [2], Hong Kong [4], Canada [3] and more recent Finland [6], UK [8] and Spain [7]. The lack of professional expertise is considered to be the most dominant factor in impeding the introduction and the adoption of systematic testing activities in software development companies.

A testing services survey conducted on more than 400 European organizations in 2011 [10], found an increased focus on quality assurance combined with a desire of the participant organizations to drive validation to the front of the software development life cycle. One of the main challenges expressed by this study is the view of quality assurance as an additional cost and an activity which increases the development life cycle of the application development process. In addition, there are the barriers imposed by the top management which sees testing units as "cost centres" rather than "service centres".

At the same time, the desired industry profiles for software professionals are not always concise, with industry openings for tester positions asking for different or various sets of skills [5], [11]. This ambiguity in the definition of the companies about the responsibilities of testing positions, suggests a wide view of thoughts and perceptions surrounding the software tester role. The importance of establishing clear roles in software engineering was recognized from the beginnings of software development and reinforced in several studies among which we can mention Acuna and Juristo [14] and Zhu [17]. Although there is a general awareness for the necessity of improved quality of software testing, there is low level of test maturity in most software development organizations [12].

Fernández-Sanz et. al [16] published the results of a survey looking at the human factors which have a negative influence on real practice of software testing in software companies in Spain. The factors related to instability of testers positions (48 %), lack of attractiveness of testing (48 %) and poor career development for testers (41,7 %) were mentioned by less than half of the participants. Even so, the study advises us to seriously take into consideration these factors due to the high percentage of respondents. The human and social aspects of working in testing inside a testing team, as well as the attitude towards the testing team in a company, were studied from the testers’ perspective in Shah and Harrold case study [13].

The definitions of software testing and the responsibilities of a software tester seem to be unclear also among students. Analysis of the current practice in the US Academia conducted by Astigarraga [5] and Lethbridge [9] emphasises a need for improvement of the current curriculum and of focus on software testing.

The Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU), conducted an internal survey among its master alumni in 2007 and 2011, with the scope to retrieve the educational topics relevant to their field of expertise based on the work experience accumulated after graduation. In both surveys, Testing and QA were rated among top eight relevant topics from around 63 proposed ones. The lack of
knowledge and necessary skills can become an additional expense for the companies which find themselves in the need of providing additional education for the young testers [11].

In this paper we describe the result of a study on the evaluation of the interest and desire to work in software testing among engineering and computer science students from Norwegian University of Science and Technology, and a survey of the factors motivating and de-motivating the students in engaging in a software testing career. Our goal is to investigate in which stage of the student studies, is the opinion on a software testing career formed, the evolution of this opinion during the study years and the impact of dedicated testing courseware towards a software testing career.

The rest of the paper is organized as follows. Section 2 reviews related work on software testing in industry and education, Section 3 presents the research questions and the survey design. Section 4 describes the data collection process and the results, while Section 5 discusses the result of the survey. Section 6 closes the paper with a discussion of the ongoing efforts to promote and improve the image of a software testing career as well as software testing activities.

2. Related Work

A testing services survey conducted on more than 400 European organizations in 2011 [10] presents the results of a comprehensive application testing survey conducted across Europe, and focused on understanding the internal set-up and organization of QA and testing teams. Among the results presented in the report is the fact that one in five organizations has a decentralized and ad hoc approach to application testing. The most predominant challenges in the European companies in the domain of process, tools and methodology are: the slow speed of testing combined with a lack of test case automation, limited information sharing across projects, as well as access to real-time metrics regarding the quality of the system under testing. Among resource and expertise setbacks many organizations highlighted the deficit of knowledge transfer to the testing team, lack of early sight of user requirements by testers, and a general lack of collaboration combined with insufficient domain knowledge to understand user requirements and business risks. Quality Assurance has received a stigma of additional cost and an activity that increases the development time cycle with further impediments imposed by the top management that sees testing units as "cost centres" rather than "service centres". Currently, one popular solution is to use third-party providers for testing activities. According to the authors of the report, while this approach can provide momentarily cost reduction and an increase in software quality, it does not have a positive effect in planning of the requirements and knowledge sharing. The authors also conclude that the trend of using 3rd party units for testers can stimulate the need for professional testers on the market.

The study performed by Astigarraga et al. [5] combines a presentation of testing as an industry profession, a survey of the software testing curriculum in United States and a discussion on the efforts made with the scope of increasing the status of testing in the academic curriculum. The study also highlights the low number of Software Testing journals and societies in comparison with other areas of software engineering. These results show a minimal content for the field of software testing in the undergraduate students, the topic being generally included in software engineering or programming courses. We noticed the same trend in our early educational programs. Astigarraga et al. make an interesting observation by pointing out that "software testing work is often not seen as glamorous or intellectually stimulating to the average Ph.D.-level researcher who might rather pursue novel development-oriented challenges instead". This lack of glamour of software testing in comparison with software development is extended beyond the research area towards the industry as well. The study advises on changes to the curriculum which will lead to more focus on software testing courseware and opportunities to obtain and improve education for software test engineering.

The attitude towards testing inside a software company was studied from testers’ perspective by Shah and Harrold case study [13]. The study reveals a difference in perspective between senior and junior testers. A tester was considered a junior tester, if he or she had less than two years of experience. The senior testers considered testing to be an important but boring activity. However, when mentoring the younger members of the team they chose not to share their view, and rather emphasise the advantages of being involved in testing such as: gaining a good quick understanding of the product and learn what good practice development is. In addition, the study describes the efforts made by a senior manager from the Quality Assurance team, who was aware of the testers being considered as “second-class citizens”, to raise awareness about the importance of the testing activities, to include testing team in the communication loop with the customer and to assure the recognition of the testing effort from the customers at the end of the development cycle. One down side of a being a tester proved to be the monotonous nature of the testing activities, especially for those working with manual testing. This factor was cited by the vast majority of the study participants as the main reason for not being interested in staying in the testing department, and preferring to move to other positions like development or business analyst. An important agent in keeping the testers motivated about the testing activities was the feeling of responsibility with power. By giving individual testers ownership and responsibility of testing modules, the testers felt more enthusiastic about their activities.

Although there is extensive research on motivation for software engineers [18], as per our knowledge, no empirical research was done so far with the scope of retrieving the motivation behind the attitude towards
software testing. There is little research on why people chose to become software testers. Our research is focusing on the motivation of choosing Software Testing as a profession.

3. Research questions and survey design

In order for us to raise awareness and improve the testing image in academia we need to retrieve and explore the factors that leads people to this position in the first place. We created a short questionnaire, which could be filled in less than 20 minutes in order to encourage students to participate. The research questions which determined the design of this survey are:

RQ1: What is the image of software testing among computer science students?
RQ2: What motivates and de-motivates students to choose a career path in software testing?

The questionnaire included 4 questions (See Appendix A). Besides demographical information, we were also interested in the level of professional experience in testing among the students in order to see if they had any exposure to testing outside the one provided by the curriculum. In this paper we focus on analysing two main questions that will help us to answer our research questions:

The first question enquires if students are interested in working as a tester after graduation and also we asked them to explain their answer. By explaining their answers we expect that they will also provide us with their perception regarding testing activities. In case of a negative answer the respondent will provide the de-motivating factors of a tester career while in case of a positive answer we will be able to gather the positive aspects of testing:

Q1: Are you interested in working as a tester in a software development company after graduation? Please explain why.

The second question is designed to determine the factors which can influence a change in the attitude towards testing and will help improve the motivation towards choosing the profession. Also, this question will allow us to perceive further motivation and de-motivation factors towards software testing:

Q2: What will motivate you to get engaged in professional software testing?

4. Data collection and results

4.1 Data collection

The study was conducted on a population formed by bachelor and master level students from Norwegian University of Science and Technology (groups S1 and S2), together with computer engineering students from University College (group S3). The survey was conducted towards the semester end in order for us to be able to assess the impact of the curriculum on students’ perception. A total of 161 participants responded to our survey (as shown in Table 1), representing 100% of the number of approached students.

Group S1 represents 2nd or 3rd year students attending a Software Engineering course, they are required to have taken courses Object-Oriented Programming and Algorithms and Data Structures prior attending this course. At the end of this course students are expected to be able to specify, design, implement and test software systems of sizes that require team work and co-operation. The second group (S2) represents master level students, from 3rd or 4th year, attending a Requirements and Testing focused course. Students taking this course are expected to have passed the course taken by students from S1 group. During this course the students learn techniques for testing IT systems and the relations between testing and other activities in the systems development process while keeping focus on requirements specification.

A third group of respondents were recruited from another education institution, namely Sør-Trøndelag University College (HiST), more specifically 3rd year students from their program in Computer Engineering. These students receive training in computer programming but they do not have any specific dedicated lectures or courses for software testing in their curriculum, although it is expected to receive knowledge in testing from their programming classes. For this reason we decided to add an additional question in their survey regarding the sources they are using for obtaining information on software testing (See Table 2). Half of the participants indicated the curriculum as being their main source of information about software testing followed closely by books and internet. Some participants proved to be very enthusiastic about guest lecturers presented by local representatives of software developing companies.

<table>
<thead>
<tr>
<th>Sources</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curriculum</td>
<td>12(52.17%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Books</td>
<td>5(21.74%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet</td>
<td>5(21.74%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guest lectures</td>
<td>3(13.04%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All the participants were asked if they have experience in professional testing. The purpose of this question is to determine the exposure level of the participant to professional software testing activities and to determine the impact of industrial experience on forming an opinion about software testing. From the 161 participants to our survey only 21 students (14 from group S1 and 7 from group S2) had previous testing experience such as part time job in software companies performing testing activities. No student from group S3 had previous experience in professional testing.

4.2 Results

Our research was performed using a combination of qualitative and quantitative methods where each method supported the other. The collected data was analyzed by organizing it from several perspectives, with our main criteria being the motivational and de-motivational factors for each participant group. The open-ended questions were coded and analyzed using the qualitative data presentation and analysis methods. For the qualitative analysis, in order to avoid researcher bias, the generated categories were reviewed and discussed in meetings between three researchers. Portions of the data was reanalyzed and recoded until an agreement was reached.

Table 3 and Figure 1 presents the results for our first question: Are you interested in working as a tester in a software development company after graduation? A vast majority of the students belonging to groups S1 and S2, is not interested in pursuing a career in testing industry while in S2 group we noticed a balance between the students who are interested (17 students) and those who are not interested (18 students) to work in testing industry.

Table 4

<table>
<thead>
<tr>
<th>Reasons for indecision about working in testing</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional information</td>
<td>13(13.27%)</td>
<td>4(10%)</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Not just testing</td>
<td>8(8.16%)</td>
<td>5(12.5%)</td>
<td>4(17.39%)</td>
<td>17</td>
</tr>
<tr>
<td>Depending on software</td>
<td>4(4.08%)</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

A numerical comparison between negative and positive factors for pursuing a testing career (Table 5) highlights a dominance of negative ones in sources S1 and S2 while in S2 both negative and positive factors are equally present.

We asked the participants to provide arguments for their answer which enabled us to retrieve positive and negatives factors influencing the students in their career choice (Table 4). In addition we created separate groups for people who were undecided about their choice or who express interest in testing under certain conditions. This group represented 23.6% from the total participants, from which 17 required more information in order to be able to reach a decision and with another 17 of the participants being interested to work in testing as long as the testing responsibilities were part of an variation of tasks and not the sole responsibility of the job. The type of software to be tested was a condition mentioned by 4 participants from S1 with game design mentioned as a favourite.

4.2.1 Negative aspects of working in software testing

From the negative factors associated with working in software testing the element of boredom stand out in all the three categories of participants. In the Boring category we included all references to boredom, boring, monotonous, not exciting, one sided job and repetitive. Table 6 and Figure 2 are illustrating the most frequent negative factors associated with the work responsibilities of a tester. We decided to leave out factors which were quoted less than eight times over all sources.

Table 5

<table>
<thead>
<tr>
<th>Numerical comparison between negative and positive factors for pursuing a testing career</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negatives factors</td>
<td>135</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>Positives factors</td>
<td>42</td>
<td>36</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th>Negative aspects in pursuing a testing career</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boring</td>
<td>35 (35.71%)</td>
<td>3(7.5%)</td>
<td>10(43.68%)</td>
</tr>
<tr>
<td>Rather writing code</td>
<td>13(13.28%)</td>
<td>9(22.5%)</td>
<td>9(39.13%)</td>
</tr>
<tr>
<td>Not creative</td>
<td>12(12.24%)</td>
<td>-</td>
<td>3(13.04%)</td>
</tr>
<tr>
<td>Status</td>
<td>6(6.12%)</td>
<td>3(7.5%)</td>
<td>1(4.35%)</td>
</tr>
<tr>
<td>Unrewarding</td>
<td>6(6.12%)</td>
<td>1(2.5%)</td>
<td>1(4.35%)</td>
</tr>
</tbody>
</table>
For the first group S1, see Figure 3, which is representing the young students we observe that almost half of them (48.61%) considers testing work to be of a boring nature. Around fifth of the participants (18.06%) are more interested in core developing as we can see from the Rather writing code category, factor which is followed closely by the lack for creativity presented in Not creative category associated with testing position (16.67%). Status and Unrewarding aspects of a tester's responsibilities are also quoted, each by 8.33% of S1 participants. One participant argument his lack of interest by the fact "testing can often have little impact on the finished product" (student, group S2) This opinion was corroborated by a series of statements from other students which were in concordance with the preconception of testers being seen as second hand citizen in the software development world: "does not seem interesting to not be able to influence the development" (student, group S1) and "test is completely at the bottom of the ranking" (student, group S1). Another uninspiring aspect from aggregated under the Unrewarding was the assumption that a software tester will be paid on a lower scale than a software developer. The financial aspect was raised again in the section of motivational factors.

The majority of the group S2 (56.25%) was more interested in writing their own code (Rather writing code), aspect which they found incompatible with the testing job responsibilities, combined with concerns of not being able to gain a position as a developer once they join the working market as testers "Becoming tester tends to be less likely to become core department" (student, group S2). The lack of creativity was not mentioned by the participants from the second group, while the issue of the Status was mentioned by three students "it looks like the testers work at the end of the project with limited time and get blamed for every defect they fail to find" (student, group S2) followed by the Unrewarding category.

In comparison with the previous groups, in S3 there is no clear dominating negative factor with the two main ones coming in at a similar level: Boring (41.67%) and Rather writing code (37.5%). One participant mentioned the responsibility for the software which comes with testing as a deterring factor for him "Testing means a lot of responsibility, if something does not work after testing, it is your responsibility" (student, group S3). The issue of No creativity was also mentioned by 13.04% of the group. One student mentioned Status and Unrewarding as negative aspects of software testing. "I like to write it on my own. Others' code doesn't give you credit and thankfulness." (student, group S3)

4.2.2 Positive aspects of working in software testing

In comparison with the group negative factors which were present in all sources, the group of positive factors (see Table 7) were not homogeneously distributed as seen from Figures 4 and 5 which show the results for the positive aspects of working in testing according to the three groups.

The Interesting category includes all the statements which mentions software testing as an interesting occupation. We included under the Importance category all the statements which provided an acknowledgement of the importance of testing including the "Improving a system", while under the Experience category we aggregated the references of gaining experience and/or knowledge during testing "It would be interesting and instructive to work with other people's code and think about it in a non-traditional way" (student, group S1), "such an experience helps make me a better developer" (student, group S3). We noticed that a high percentage of the participants acknowledges the importance of testing, specially group S2 (20%), the participants of the Testing Course.

Four of the participants consider working with popular methodologies such as Test-driven development, (under Interest in TDD category), an alluring factor for the software testers. "It is another way to think about how to develop a software when you use TDD" (student, group S3).
Although they are not qualifying over the threshold of minimum four citations, we found two factors which were cited both negative and positive. First is referring to testing as not being hard coding, which proved in some cases a deficit for people interested in core development while for others it was consider a nice alternative for those who are not fond of just coding. The second factor was related to having fixed routines, which was associated with testing responsibilities while for few of participant a fixed routine was a positive trait for a work position.

### 4.2.4 Motivational factors for working in software testing

In order for us to understand the motivational and de-motivational factors towards a software testing career we asked the participants what will motivate them to get engaged in professional software testing. By retrieving the factors which can trigger a change in the attitude towards testing we can improve the motivation among the students towards choosing the software testing profession. Also, this question allows us to perceive further motivation and de-motivation factors towards software testing.

In the result of second question (see Table 8 and Figure 6), sources S1 and S3 consider they need more information on software testing in order to construct an image and provide an opinion about the possible improvements in software testing. All these statements were collocated under More Information category. "Lecture on testing / real life view of the workplace" (student, group S1), "Get a bit more about what the testing entails. Certainly been demonstrated in practice by some who work with it" (student, group S3). The fact that only one person from S2 had provided this answer is conform with our expectation, since the image of testing should be formed after attending a course about software testing. Group S1 considers important to have a variety of tasks while doing testing activities which, in their opinion will also make the testing responsibilities more interesting "If you get a variety of test tasks and feel that you participate in developing a product" (student, group S1).

For group S3, the second motivator is the financial aspect of the position (Money), with 30\% of the students assuming that a tester will automatically have a lower pay than a developer. For group S1 also, the financial aspect is considered as important, with 10 students invoking it. "Higher salaries for testers. That testing is more appreciated. The testers will be included in other parts of the process as well" (student, group S1), "Decent pay, being involved in developing the software I am testing" (student, group S3).

The issue of the Status was also mentioned by students from all three groups. Several students from groups S1 and S2 requested a better marketing for the testers’ position and reputation as well as an increase on the emphasis of Testing importance: "increase reputation, normal SW developers look down on testers. Include testers better in agile groups, most companies have testing still separate from developers which use Scrum/X" (student, group S3). "More emphasis on testing feedback and not only finding out if something works as it should" (student, group S1).

Get testing experience category was dominant among students from group S2, while the second place for the same group was taken by Improved testing methods category which contains the reference about innovative or new testing methods such as TDD as well as an involvement in testing automation.

<table>
<thead>
<tr>
<th>Motivational factors for pursuing a testing career</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>More information</td>
<td>18(18.37%)</td>
<td>1(2.5%)</td>
<td>7(30.43%)</td>
</tr>
<tr>
<td>Variety of tasks</td>
<td>16(16.33%)</td>
<td>1(2.5%)</td>
<td>2(8.70%)</td>
</tr>
<tr>
<td>Money</td>
<td>10(10.20%)</td>
<td>1(2.5%)</td>
<td>7(30.43%)</td>
</tr>
<tr>
<td>Status</td>
<td>8(8.16%)</td>
<td>2(5%)</td>
<td>3(13.04%)</td>
</tr>
<tr>
<td>Get testing experience</td>
<td>2(2.04%)</td>
<td>10(25%)</td>
<td>0</td>
</tr>
<tr>
<td>Improved testing methods</td>
<td>4(4.08%)</td>
<td>6(15%)</td>
<td>1(4.35%)</td>
</tr>
<tr>
<td>Testing Importance</td>
<td>4(4.08%)</td>
<td>1(2.5%)</td>
<td>0</td>
</tr>
</tbody>
</table>
5. Interpretation of the results and discussion

Most participants from groups S1 and S3 were not interested in pursuing a career in testing. Students from group S1 were very enthusiastic and opinionated in their answers and portrayal of software testing and provided the most diverse set of negative factors associated with a testing career. On the second question of the survey, the S1 group, despite having the largest number of participants provided the lowest number of requests for more information on software testing, which leads us to believe that there is a certain level of overconfidence associated with a mix of young age and lack of experience.

The number of citation between negative and positive factors for pursuing a testing career was compared between the three sources and a dominance of the negative ones in sources S1 and S2 was observed. As for the S2 group, the negative and positive factors were equally represented. The group S2 was more uniform on the answers to the questions. This difference in result can be attributed to the difference in the level of knowledge about software testing, where is it safe to assume that S2 is excelling in comparison with the other groups. It is encouraging to see the positive effects of a dedicated testing courseware and it is a stimulus for further research in this area.

The S3 group, although manifested a strong desire not to pursue a testing career, was more soft-spoken in their arguments and had the highest percentage of students asking for more information for Q2 question.

From the participants that responded with undecided at Q1, 17 stated they require more information on software testing in order to decide to pursue a testing career. This number, although representing a small fraction from the total participants combined with high number of participants necessitating for more information on testing for Q2, is an indication of a strong need for a change and restructure in the teaching methods applied in testing.

6. Conclusion

The study shows a lack of interest and a tendency to associate testing activities with boredom for the students not attending a software testing courseware. Around half of the students who attended the testing course were interested in pursuing a testing career and their main request was for practical experience, combined with presentation on more innovative methods and tools for testing. These results indicate the possibility of improving the perception of a testing career by changing the curriculum and reviewing some of the curriculum parts with testing professionals with the scope to retrieve the educational topics relevant to testing field of expertise. In parallel with these activities, we can discuss the dedicated testing courses with students upon course completion, and capture the topics they did not consider interesting or relevant together the ones they liked most.

All three groups that we surveyed, seem to equal testing activities with not writing code, or checking others’ code which indicates a misunderstanding of the responsibilities of testers. We need to redefine and enhance the definition of testing and combat the prejudices which are surrounding the tester role such as: no creativity, no coding, unrewarding and low status. We need to increase the focus on testing tasks and responsibilities which combine classical development roles such as: creating and writing testing automation tools. By organizing guest lectures with software professionals having strong testing careers and pointing out the writings, even the blogs of recognized professional testers such as Cem Kaner, Michael Bolton and James Bach we can improve the low status associated with testing activities.

There is a gap between the industry needs and the education the students are receiving and we should strive to decrease this gap but at the same time we should be aware of the differences between the academia and industry, and accept that it will be nearly impossible to definitively close the gap. Universities should strive to provide the best education in accordance with the latest technologies and updates from science, but the core of the education will be principles and theory.

Obtaining an university degree will take four to five years, if not more, and if the tools and methods that are taught, are not even the latest in the software industry, by the time of graduation these methods will be even more outdated. In Norway, the students studying at a university college can obtain their qualification in three years, so they are more equipped with the right knowledge, which makes them more desirable for the companies.

An additional way of improving the curriculum, which is applied at our university, is the survey performed among university alumni in which they are asked to rate the relevance of the topics learned in the university. The reports based on this survey can help in improving the curriculum by retrieving the educational topics which
proved to be of high importance and reducing the curriculum for certain courseware.

The availability of a large number of free software tools worldwide and the fast pace with which these tools are adopted, changed, upgraded and even dropped makes it difficult or even futile for a university program to establish a curriculum for these tools and implement it. Due to the long process of implementing a change in the academia we risk providing outdated material. One plausible solution in regard to testing tools will be to provide testing tool oriented topics in the last year of university to get hands-on experience so valued in industry. Our future work will focus on deciding and implementing changes in curriculum while taking into consideration the motivators and de-motivators of students, as well as presenting practices from the real world.

References

[16] Factors with Negative Influence on Software Testing Practice in Spain: A Survey Luis Fernández-Sanz1, M. Teresa Villalba, José Ramón Hilera, and Raquel Lacuesta

Appendix A

Software testing survey questions

1. Are you interested in working as a tester in a software development company after graduation? Please explain why:
2. What will motivate you to get engaged in professional software testing?
3. Do you have professional testing experience?
4. What were the sources you used for obtaining information on software testing?
P3 Deak, A., Sindre, G. "Analyzing the importance of teaching about testing from alumni survey data" NIK Norsk Informatikkkonferanse 2013 s. 100-109
Analyzing the importance of teaching about testing from alumni survey data

Anca Deak, Guttorm Sindre
Department of Computer and Information Science
Norwegian University of Science and Technology, Norway
{deak, guttors}@idi.ntnu.no

Abstract
The Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU), conducted a survey among its master alumni in 2007 and 2011, with the scope to retrieve the educational topics relevant to their field of expertise based on the work experience accumulated after graduation. The results enabled the identification of topics which proved mandatory to learn after graduation, and the two sets of data registered four years apart, provided us with an overview of the trends and a certain shift in the importance of the most favored topics. The data available from this two surveys was analyzed from the perspective of software testing topic and the results of this analysis are presented in this paper.

1 Introduction
"An investment in knowledge pays the best interest" (Benjamin Franklin), and those preparing the future software professionals are responsible for providing them with the skills required and desired in software industry. Many of the young graduates who are entering the work field of software engineering are lacking knowledge in key topics such as requirements engineering, software testing, human related factors and project management [1].

The mismatch of expected knowledge between academic curricula and industry has its roots in the set of acquired skills which accompanies the students at the moment of graduation. This knowledge gap affecting the industry needs was highlighted by [2], [3], [4] and reconfirmed in various studies conducted worldwide [5], [6], [7] and [8]. In order to decrease the existing gap it is mandatory for the universities and colleges to adapt their educational programs while listening to the industry requirements and also ensure that it is conveyed in a manner allowing future professionals to handle correctly the problems that they are most likely to face during their professional career.

An essential part of any software engineering process which ensures the high quality of the software products developed is software testing. Analysis of the current practice and education programs in the US Academia presented by [9] and [2] emphasizes a need for improvement of the current curriculum and of focus on software testing. The lack of knowledge and necessary skills in this domain for the young graduates can be translated as an additional expense for the companies which find themselves in the need of providing additional training for the young testers [10]. Since many companies spend more than 60% of the development cost on testing, and testing is the main process of ensuring a high quality system, as described by [11], it is imperative to search for improvement of the methods, not only among testing techniques, which are extensively covered in the literature, but also on those performing the testing activities, namely the testers.

This paper presents an analysis with the main focus on software testing on the results, of the two web based surveys, conducted by the Department of Computer Science (IDI) of the Norwegian University of Science and Technology (NTNU) among IDI alumni students in
2007 and 2011. The focus of the surveys is to retrieve the educational topics relevant to the participants' field of expertise based on the work experience accumulated after graduation and on the emphasis the IDI department is assigning on these topics. The data available collected by this two surveys was analyzed from the perspective software testing topic at the stage where but the design and collection processes were already completed.

2 Related work

The gap between industry expectation and the knowledge and skills provided by the academia has been the focus of various studies conducted worldwide. The study performed by [9] combines a presentation of testing as an industry profession, a survey of the software testing curriculum in United States and a discussion on the efforts made with the scope of increasing the status of testing in the academic curriculum. The study also highlights the low number of Software Testing journals and societies in comparison with other areas of software engineering. These results show a minimal content for the field of software testing in the undergraduate students. The testing topic is generally included in software engineering or programming courses.

The study conducted by Astigarraga, [9], makes an interesting observation by pointing out that "software testing work is often not seen as glamorous or intellectually stimulating to the average Ph.D.-level researcher who might rather pursue novel development-oriented challenges instead". This lack of glamour associated with software testing in comparison with the software development topic is extended beyond the research area towards the industry as well. The study advises on changes to the curriculum which will lead to more focus on software testing courseware and opportunities to obtain and improve education for software test engineering.

Lethbridge conducted 2 surveys [2] and [12], with software professionals to identify the areas which would benefit from an improved education curriculum. The scope of these surveys was to provide information and data to educational institutions and companies which will allow them to adjust their curricula and training programs. The survey described in [12], presented a list of 57 topics related to software (31 topics), mathematics (9 topics), engineering (4 topics) and other concerns (13 topics) with four questions described in section 3.1 for each topic and answers organized on a six-point ordinal scale. The result of the survey emphasized the lack of necessary education for practitioners and the necessity of updating the curricula.

An updated version of the survey which was repeated later by Lethbridge provided congruent results [2]. A customized version of Lethbridge survey was run in a similar study on recent graduates by [4] in UK universities. The small survey of Finnish IT professionals, professors and lecturers, and students presented in [5] confirmed the results of Lethbridge's and Kitchenham's studies. In addition, the topic Testing obtained a higher scoring in comparison with the scores from Lethbridge's study. The software developers, academics and students ranked the importance of testing with respectively 3.5, 3.8 and 3.3 while in Lethbridge's survey, the topic of testing scored 3.0.

While Lethbridge's survey is looking at professionals with experience in industry and Kitchenham’s focuses on recent graduates only, the survey performed by IDI comprise both professionals with experience, as well as recent graduates giving us a broader picture of the knowledge gap between the education received at university level and its relevance in the industry.
3 Survey instrument and data collection

3.1 Survey instrument

The Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU), conducted an internal web based survey among its master alumni in 2007 and 2011, with the scope to retrieve the educational topics relevant to their field of expertise based on the work experience accumulated after graduation.

The survey was based on three sources: a candidate survey from a closely related study program in Communication Technology at the NTNU [13], the North American study conducted by Lethbridge [2] and [4]. Another source used for building this instrument is the ACM Computing curricula [14]. The NTNU Communication Technology survey had a simple structure containing few questions asking whether the graduates had a job or not and whether in that case they had a job even before submission of the master thesis or not. Other questions included in this survey are enquiring where in the country they have received a position and what were the responsibilities of that position. The survey by Lethbridge is far more detailed, but does not cover the entire IDI professional profile as well, as it has a main focus on software and system development. The 75 topics provided in this survey are inspired by SWEBOK (IEEE's Guide to Software Engineering Body of Knowledge [15]). For each of these topics the Lethbridge survey includes four questions, all answered on a scale from 0 (nothing) to 5 (very much):

i. How much did you learn about this topic in your formal education (college / university)?

ii. How much do you know about this now?

iii. What are the direct benefits, this topic had for you in your work?

iv. What were the indirect utility, this topic had (even if you have not directly used it, so it is believed that it has affected one's ability, so that one does a better job, for example, ability to analytical thinking)?

Using this enables the identification of topics which alumni have been forced to learn after graduation and topics covered during studies but forgotten, since there was no need for participants to use this knowledge in their jobs. The results provide a list of topics of great usefulness, and the subjects suspected of small utility. This type of questions above will only highlight what was beneficial for the alumni so far, but since it may also be interesting to hear what they think will be important in the future based on trends they see in the industry, a final question was added: "In light of the trends you see in the workplace, mention three topics that you think should be given higher priority in education (need not be obtained from the above list)"

The ACM Computing curricula is not a survey, but advice on what different programs should contain. The overview report distinguishes between different disciplines: Computer Engineering, Computer Science, Software Engineering, Information Technology and Information Systems. The report provides an overview of the 40 data items and 17 subjects with different weights, which indicate how relevant these topics will be mentioned in the various degree programs.

During the survey of 2007 the participants were asked to state how much they needed to know to perform their work task from a topics list, regardless of whether these subjects were covered during their study periods or not. The available response options were "no direct requirements", "little need", "moderate needs ", "strong need", "very strong need" which were scored with 1 to 5 points response analysis. During the 2011 survey, a 3-points scale was used.
with the following options: very helpful, useful, not helpful options plus a N/A option for those who found the topic inapplicable to their work responsibilities.

The IDI survey topics list contains 63 items based on Lethbridge topics with local adaptation which are a better reflection of the IDI educational program. The available topics are not the equivalent of established university courses, they can be thought among a full course, a partial course or they can be taught in different layers in a range of courses. The survey was web based, conducted in Norwegian and could be completed in less than 30 minutes.

3.2 Data collection

The purpose of the survey was to retrieve feedback from alumni with regard to the quality and the relevance of the education provided by IDI. In order to not only capture short-term first job experiences, the primary audience for the 2007 survey were people who graduated 5-10 years back, and therefore may have gained experience from several jobs, as seen in Table 2, and experienced the need for professional development. For the 2011 survey, a question asking for the graduation year was included with the results for this question being available in Table 1:

<table>
<thead>
<tr>
<th>Year</th>
<th>No</th>
<th>Year</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>3</td>
<td>2004</td>
<td>6</td>
</tr>
<tr>
<td>2010</td>
<td>10</td>
<td>2003</td>
<td>11</td>
</tr>
<tr>
<td>2009</td>
<td>4</td>
<td>2002</td>
<td>12</td>
</tr>
<tr>
<td>2008</td>
<td>6</td>
<td>2001</td>
<td>3</td>
</tr>
<tr>
<td>2007</td>
<td>12</td>
<td>2000</td>
<td>6</td>
</tr>
<tr>
<td>2006</td>
<td>16</td>
<td>90-99</td>
<td>26</td>
</tr>
<tr>
<td>2005</td>
<td>10</td>
<td>68-89</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: Number of jobs obtained after graduation

<table>
<thead>
<tr>
<th>Nr. of jobs</th>
<th>2007 graduates</th>
<th>2011 graduates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>68</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>6+</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>

4 Results and observations

4.1 Demographic results

The information presented in the following tables provides a profile of the participants' with regard to education in Table 3 and the type of employment is available in Table 4. The total number of participants was 202 for the 2007 survey and 133 participants for the one conducted in 2011.

<table>
<thead>
<tr>
<th>Education</th>
<th>2007</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doctor / PhD</td>
<td>0</td>
<td>9 (6,77%)</td>
</tr>
<tr>
<td>Master Technology</td>
<td>173 (85,64%)</td>
<td>99 (74,44%)</td>
</tr>
<tr>
<td>Bachelor</td>
<td>7 (3,47%)</td>
<td>3 (2,26%)</td>
</tr>
<tr>
<td>Master Informatic</td>
<td>20 (9,90%)</td>
<td>22(16,54%)</td>
</tr>
<tr>
<td>Other</td>
<td>2 (0,99%)</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Education</th>
<th>2007</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private</td>
<td>146 (72,28%)</td>
<td>92 (69,17%)</td>
</tr>
<tr>
<td>Public</td>
<td>34 (16,83%)</td>
<td>25 (18,80%)</td>
</tr>
<tr>
<td>Self employed</td>
<td>8 (3,96%)</td>
<td>4 (3,01%)</td>
</tr>
<tr>
<td>Education</td>
<td>8 (3,96%)</td>
<td>2 (1,50%)</td>
</tr>
<tr>
<td>Available</td>
<td>2 (0,99%)</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>4 (1,98%)</td>
<td>10 (7,52%)</td>
</tr>
</tbody>
</table>
4.2 Software Testing's perspective results

In both years alumni surveys, Testing and QA were rated among top ten relevant topics from around 63 proposed ones, gaining the 7th place in 2007 and the 8th place in 2011, while Programming remains on the top as seen in Table 5. The largest increase in importance was registered by Software Engineering Methods, Software architecture and Algorithms and Complexity each of them climbing 5 places in the hierarchy Presentation Technique and Interpersonal communication have registered the higher decline in the perceived importance by decreasing with 8 and respectively 7 places in the top 10 topics.

<table>
<thead>
<tr>
<th>Topics ranked by perceived importance</th>
<th>2007</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming</td>
<td>89</td>
<td>90</td>
</tr>
<tr>
<td>Database Systems</td>
<td>77</td>
<td>86</td>
</tr>
<tr>
<td>Software Engineering Methods</td>
<td>75</td>
<td>80</td>
</tr>
<tr>
<td>Software architecture</td>
<td>75</td>
<td>80</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>81</td>
<td>76</td>
</tr>
<tr>
<td>Algorithms and Complexity</td>
<td>63</td>
<td>75</td>
</tr>
<tr>
<td>Project Management</td>
<td>80</td>
<td>70</td>
</tr>
<tr>
<td>Testing</td>
<td>77</td>
<td>69</td>
</tr>
<tr>
<td>UI / MMI</td>
<td>66</td>
<td>68</td>
</tr>
<tr>
<td>Presentation Technique</td>
<td>82</td>
<td>67</td>
</tr>
<tr>
<td>Interpersonal communication</td>
<td>81</td>
<td>60</td>
</tr>
</tbody>
</table>

The last question of the survey asked the participants "What three topics should be higher prioritized during education, based on the trends you see in the workplace?" Testing came on the second place on the list of Topics that 10 or more people have responded as requiring more focus in our education for the 2011 survey and was mentioned 8 times during the 2007 survey as seen in Table 6 and illustrated in Figure 1.

<table>
<thead>
<tr>
<th>Topics</th>
<th>2007</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming</td>
<td>29 (14,36%)</td>
<td>27 (20,30%)</td>
</tr>
<tr>
<td>Testing</td>
<td>8 (3,96%)</td>
<td>22 (16,54%)</td>
</tr>
<tr>
<td>Project Management</td>
<td>16 (7,92%)</td>
<td>21 (15,79%)</td>
</tr>
<tr>
<td>Software architecture</td>
<td>2 (0,99%)</td>
<td>20 (15,04%)</td>
</tr>
<tr>
<td>Software Engineering Methods</td>
<td>9 (4,46%)</td>
<td>15 (11,28%)</td>
</tr>
<tr>
<td>UI / MMI</td>
<td>7 (3,47%)</td>
<td>13 (9,77%)</td>
</tr>
<tr>
<td>Algorithms and Complexity</td>
<td>5 (2,48%)</td>
<td>13 (9,77%)</td>
</tr>
<tr>
<td>Agile methods</td>
<td>4 (1,98%)</td>
<td>13 (9,77%)</td>
</tr>
<tr>
<td>Security</td>
<td>7 (3,47%)</td>
<td>12 (9,02%)</td>
</tr>
<tr>
<td>Distributed systems</td>
<td>1 (0,50%)</td>
<td>11 (8,27%)</td>
</tr>
<tr>
<td>Requirements engineering</td>
<td>4 (1,98%)</td>
<td>11 (8,27%)</td>
</tr>
<tr>
<td>System integration</td>
<td>1 (0,50%)</td>
<td>10 (7,52%)</td>
</tr>
<tr>
<td>Negotiation skills</td>
<td>0</td>
<td>10 (7,52%)</td>
</tr>
</tbody>
</table>
For the participants who stated Testing as a very relevant topic for their work, we selected the next first 15 topics rated. In case the topic mentioned in the next first 15 topics rated in one year was present on a lower scale in the other year we decided to include both years’ data. For the 2007 survey 50 participants out of 202 who stated testing as being very much needed in their work, while in 2011 survey, 52 out of 133 participants stresses the importance of the testing knowledge for their workplace. In Table 7, we can see that, both years programming was mentioned as the most important topic, which is much needed in their work assignments with around 80% of the participant stressing its importance alongside testing knowledge.

### Table 7: Top topics rated as relevant for the participants requiring testing knowledge

<table>
<thead>
<tr>
<th>Topics</th>
<th>2007</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Architecture &amp; Design</td>
<td>18 (36%)</td>
<td>43 (82.69%)</td>
</tr>
<tr>
<td>Programming</td>
<td>41 (82%)</td>
<td>42 (80.77%)</td>
</tr>
<tr>
<td>Software Engineering Methods</td>
<td>25 (50%)</td>
<td>40 (76.92%)</td>
</tr>
<tr>
<td>Database Systems</td>
<td>18 (36%)</td>
<td>39 (75.00%)</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>22 (44%)</td>
<td>38 (73.08%)</td>
</tr>
<tr>
<td>Project Mgmt. &amp; Estimation</td>
<td>18 (36%)</td>
<td>30 (57.69%)</td>
</tr>
<tr>
<td>Presentation Technique</td>
<td>23 (46%)</td>
<td>27 (51.92%)</td>
</tr>
<tr>
<td>Webdevelopment</td>
<td>0</td>
<td>27 (51.92%)</td>
</tr>
<tr>
<td>Interpersonal communication</td>
<td>17 (34%)</td>
<td>26 (50.00%)</td>
</tr>
<tr>
<td>Security</td>
<td>0</td>
<td>26 (50.00%)</td>
</tr>
<tr>
<td>Algorithms and Complexity</td>
<td>6 (12%)</td>
<td>25 (48.08%)</td>
</tr>
<tr>
<td>UI / MMI</td>
<td>7 (14%)</td>
<td>23 (44.23%)</td>
</tr>
<tr>
<td>System integration</td>
<td>10 (20%)</td>
<td>22 (42.31%)</td>
</tr>
<tr>
<td>Management</td>
<td>13 (26%)</td>
<td>21 (40.38%)</td>
</tr>
<tr>
<td>Technical Writing</td>
<td>11 (22%)</td>
<td>21 (40.38%)</td>
</tr>
<tr>
<td>Negotiation Skills</td>
<td>10 (20%)</td>
<td>16 (30.77%)</td>
</tr>
<tr>
<td>Software Maintenance</td>
<td>16 (32%)</td>
<td>0</td>
</tr>
<tr>
<td>Software Process and Quality</td>
<td>14 (28%)</td>
<td>0</td>
</tr>
</tbody>
</table>
We can notice in Figure 2 a certain shift in the importance of the top topics selected by those who required more emphasis on software testing education. We notice a tremendous increase for Software Architecture and Design (46.69%), Database Systems (39%), Algorithms and Complexity (36.08%), UI / MMI (30.23%), Requirements Engineering (29.08%) and Software Engineering Methods (26.92%) topics, while Programming has a slight decrease (-1.23%) although overall it still keeps a strong position.

**Figure 2: Top topics rated as relevant for the participants requiring testing knowledge**

For the participants who stated Testing as not relevant topic for their work 32 (15.85%) in 2007 and 7 in 2011 (5.26%), we selected the 3 most topics rated. The participants of the 2007 survey who stated testing as not being in their work stated Interpersonal communication (14 participants), Management (10 participants) and Presentation Technique (11 participants) as the most relevant areas for their work responsibilities.

### 4.2 Validity threats

The questionnaire used in this set of surveys was based on an instrument previously developed and used by [2] and [4] at a later stage, which serves a strengthening factor for the surveys’ internal validity while the reliability is assured by the same standardized questions presented to all participants [16]. The completion of the survey was done voluntarily from a participant population with a graduation year covering a white range of 40 years from 1968 to 2011. The participants answered questions about each of 63 topics which were much inspired by the topic list used in the [2] survey, but with some local adaptations.

A possible limitation of the survey with respect to testing professionals is the selection of participants with mandatory technological background which is not allowing us to make an
assessment for the professional testers which might have a different educational background such as a financial one. For future it will be interesting to see the result of such a survey at an entire university level, but still we consider the population involved to be the representative one since a computer science based education is the most recommended one for a professional tester. One downside regarding our methodology is related to the survey as not being designed with main focus of software testing. The data collected from the 2 surveys was ready and available, and analyzed from the testing perspective, but the design and collection processes were already completed.

Some typical threats to validity that are often considered for questionnaire surveys like this one are clearly relevant to this study:

- **Sampling error.** For the questionnaire survey as it was originally made, the target group was all candidates having graduated from a certain university department. Of more general interest, however, would be the competence needs of testers regardless of what department they had graduated from.

- **Coverage error.** Even within the more limited pool of graduates from IDI, far from all graduates had registered themselves in the alumni database, and some were registered with email addresses that were no longer valid, thus not receiving any offer to participate in the survey.

- **Non-response error:** Not all who got the offer to participate did so. In total there would be more than 2000 alumni from the department aggregated over the years, meaning that coverage and non-response problems taken together, only around 10% of the target population responded (or even less for the 2011 survey). This is normal for such surveys, and would only pose a serious problem to the study if the non-respondents were very different from the respondents, e.g., if non-respondents were generally much less interested in testing and more interested in other things. However, since the questionnaire was not specifically targeting testing but asking about all kinds of courses, there is no particular reason this should have happened.

- **Measurement error.** Candidates might answer inaccurately as may be hard to estimate on a scale how much need one has had for various course topics in the job. In particular, there might be a tendency that the answer is dominated either by the most recent needs, since these are freshest in memory at the time of response, or by needs in the very first job obtained, since this maybe feels most important for getting started in the career at all (as the first job is probably the one where the recruitment decision is most heavily based on the university diploma, whereas for later jobs references and previous job experience and performance gradually diminish the impact of the university diploma.

All these threats are of course relevant, and it would ideally have been better if all the alumni had responded to the surveys (which would also have given much more data to work with). For comparability between the two surveys, it would also have been an advantage if exactly the same questionnaire had been used twice, avoiding changes to the scale and some changes of questions. Still, it is hard to see how any of this would cause a particular bias for software testing, whether positive or negative. For the issue of measurement error, an important point to notice is that the survey investigates the need for testing (and other subjects) over time, throughout the respondents’ careers, which may be different from the current need in industry. However, considering the fact that test-driven development has become a very hot topic in industry recently, it seems unlikely that the importance of testing now would be less than it has been over time. Also, the extra question where respondents were asked to consider trends that they now see in work-life and suggest which topics should
be prioritized due to this partly compensates for the retrospective angle of the other questions and testing came out as important also here.

5 Discussion and future work

Based on IDI survey results which showed that Software Testing topic scored high in the Topics ranked by perceived importance list as well as in Topics which should be higher prioritized list, there is an obvious need to increase its emphasis in the educational programs. Our results corroborate with previous results published by Lethbridge's survey which showed that Testing is one of the topics requiring greatest on-the-job learning effort. Another similarity with Lethbridge's finding is the strong status of programming which is maintaining its top position among topics ranked by perceived importance classification. One positive results of the two survey is the obvious decrease in numbers from 32 (15.85%) in 2007 to 7 in 2011 (5.26%), for the participants which do not consider software testing knowledge relevant to their work.

Teaching students about testing in high level or master study can be ineffective since they have already acquired a foundation in programming. If the testing skills are built into the students’ education from the early levels, these skills will be perceived as a mandatory knowledge and the testing practice will be incorporated in the programming style of the student rather than be seen as an additional skill. Educating software testers and providing them with the necessary knowledge and skills to perform their jobs efficiently and properly remains an important goal to be reached by the academia while keeping in mind the trends and practices from the IT world.

Informing students about the possibility and the options of a career in software testing should also be taken in consideration. Not many students are focused on a testing career in their final years of studies, but since software tester is also an entry level position on the job market the probability for a beginner to enter in a testing position is quite high.

There are two directions in which the results of this study can be applied. From the perspective of high level educational organization the conclusions of these surveys can be applied to retrieve the areas where industrial requirements are not or partially addressed and new curriculum can be developed or the existing one can be redesigned. The results of the study can also be used by students when selecting their computer Science courses, especially for those who are considering an industry career path and/or who would like to pursue a Master Program which is oriented towards the industry. From another point of view, the industry can also be a beneficiary of this study by identifying possible skill gaps among future software testers and graduates and utilize this information useful for planning and organizing the relevant industrial training.

References


What Characterizes a Good Software Tester? – A Survey in Four Norwegian Companies

Anca Deak

Department of Computer Science,
Norwegian University of Science and Technology (NTNU),
Trondheim, Norway
deak@idi.ntnu.no

Abstract. The aim of this paper is to identify the most favourable perceived characteristics of software testers and contrast them with the group of existing software engineers' characteristics provided by prior research. We conducted a qualitative survey among four software development companies using interviews as a main method for data collection and the thematic-analysis approach for analyzing the data. Our findings describe the characteristics which are perceived as most valuable for a tester based on the working experience accumulated by the participant testers and testing managers. The study findings suggest that there are differences between the characteristics of a software engineer and a software tester, with communication and possessing an IT background being cited as the most frequently favoured characteristics for ensuring a high performing tester.

Keywords: software testers, software testing, software engineers, characteristics, human factors.

1 Introduction

Successful software projects outcomes are highly dependent on the skills and commitment of the testers involved. Having good testers will significantly improve the odds of project success and the delivery of high quality products. Since testing projects often occur under tight deadlines, budget constraints and organizational challenges, the success of a project can be highly dependent on the employee’s characteristics.

There are not many studies focusing on the testers or on testing teams, although it is acknowledged that some of the problems associated with software testing in the industry are not technical, but originate from other sources, such as the socio-technical environment and organizational structure of the company as presented in Rooksby, Rouncefield and Sommerville study [1]. Although there is extensive work on characteristics of software engineers, to our knowledge there is a lack of research focusing specifically on the characteristics of software testers. The goal of this study is to empirically determine specific traits for software testers and contrast them with software engineers' characteristics provided by prior research. A better understanding
of these traits could assist in the identification of individuals with potential to be high performing testers and help the software companies within their recruiting process.

The remainder of this article is structured as follows. A literature background on characteristics of software engineers and software testers is presented in Section 2. We describe the research questions and design of the survey in Section 3. In Section 4, we present the survey’s results. Section 5 discusses the lessons learned while Section 6 summarizes the findings and suggest areas for further research.

2 Characteristics of Software Engineers and Software Testers

2.1 Research on Characteristics of Software Engineers

Software Engineers characteristics was the scope of a systematic literature review conducted by Beecham et al. [2], in which 92 papers published between 1980 and June 2006 were analyzed. The result of this study provided 16 characteristics of the software engineer together with a group of motivators and de-motivators which were identified in the literature. The study conducted by Beecham et al. [2] indicates that several factors have influence on a software engineers' motivation. One of these factors is related to the personal characteristics of the software engineer. The most cited characteristics of software engineers were: Growth orientation, Need for independence and Low need for social interaction which is associated with introverts.

These characteristics have two set of determinants: the control factors and the moderators. The control factors group relate to individual’s personality, their career paths and their competencies. The second group, the moderators, are considered external factors that can influence the characteristics, such as: career stage and culture which are most often cited, followed by the type of job and the type of organisation which have less influence. A subsequent study by Franca et al. [3] extended and updated this research by analyzing 53 papers published between March 2006 to August 2010. As a result, another three characteristics were identified: Competent in Management, Flexible and Have fear of punishment. Most of the studies involved in these two literature reviews were quantitative survey studies and they provided important insights into characterizing the factors and results related to motivation. One limitation of the two literature reviews, is that the majority of the studies are referring to the job itself as being the main motivational factor.

Since the position of software engineer can contain multiple roles and responsibilities which can vary from one position to another, more information about the characteristics which influence the motivation of software engineer is required. Our studies focus on software testers, who are often considered as software engineers in job title terminology, but may have different responsibilities than their fellow developers.

Lack of motivation has been seen as an important factor for failure in software development projects [4] and a recent study of a vendor organization’s testing show that the quality of testing is affected by the motivation of the testers [5]. Previous studies observing the testing activities examined how the organization and the relationships between the members of a software development team are impacting the
testing process and implicitly the quality of the product [6]. We wanted to build on this research by determining the characteristics which influence the motivation of different types of testers’.

2.2 Research on Software Testers' Characteristics

Testing literature written by practitioners proposed skills and characteristics which they considered most important for software testers. Pettichord provides a list of twelve desired traits which he compares between testers and developers [7]. Good testers are expected to have broad knowledge of many domains, learn new things quickly, focus on user needs and behaviour, think empirically, tolerate tedium, be sceptical, concentrate on reporting problems and be comfortable with conflicts. Black [8] suggests concrete desired skills, such as the ability to interpret the product requirements and other types of available material as well as the ability to write good defect reports. The skills are also described in Pol [9] study, but they are not so strongly emphasized. Both studies acknowledged the need for strong knowledge both of software development and application domain of the product.

An ethnographical study of a small start-up software product company conducted by Martin, Rooksby et al. [10] studied how testing is performed in an environments with extremely volatile requirements. Their study focused on integration and acceptance testing done in the company. The results showed that testers working in contexts where requirements where not defined in detailed and without any strict processes, needed understanding of the business and experience in the domain and techniques that were used to test the product. Testers were also required to possess good skills in test automation.

The importance of experience in testing was also studied by Beer and Ramler [11] through a multiple-case study, which covered three industrial software projects. The results of the study identified two kinds of experience: experience in testing and experience with the domain of the product. Experience in testing was used in general management of the testing and especially in automation of the testing. Testers who have accumulated experience in the business domain and have strong business knowledge and expertise proved valuable when performing test case design, regression test case selection, or facing incomplete or evasive requirements and specifications.

A case study on high performing testers reported by [12], was conducted in three Finnish software product companies. The aim of the study was to identify the characteristics which were considered important for testers with high performance. Among the findings were that experience, reflection, motivation and personal characteristics had the highest importance. The results also strengthen the findings of the previous studies by stating that domain knowledge and specialized technical skills were considered more relevant that test case design and test planning skills.

Taipale and Smolander conducted a qualitative study [13], which explored the software-testing practices and suggested improvements in this process based on the knowledge acquired during their study. Their improvement proposition include adjusting testing according to the business orientation of the company, enhanced
testability of software components, efficient communication, early involvement of testers and increased interaction between developers and testers. The same group of authors conducted another study, focusing on the conditions that influence software testing as an online service, looked among other at the skills required for software testing [14]. The results of the study showed that in order to assure a successful execution of an online testing process, both technical and soft skills are required. An ability to adjust to different working methods, increased understanding of customer’s business needs and requirements, communication, project management, technical skills alongside with a flexibility to learn and adapt to evolving technologies were considered the required testing skills.

An empirical study of a vendor organization’s testing team conducted by Shah, Harrold and Sinha [5], studied the perceptions the participants had about testing. The testers showed enthusiasm about their job and they have a positive attitude toward testing, which is the opposite of the common attitude towards testing, as being not so high valued as a software development job. A desire for innovation and a high values among the testers were also observed in the same study. In addition, the study shows that the quality of testing is affected by the motivation of the testers and emphasizes the need for appreciating testers’ efforts.

3 Research Questions and Method

The aim of this paper is to study the characteristics of software testers (RQ1), and to contrast the group of characteristics derived from our study with the group of existing characteristics for software engineers provided by two previous systematic literature reviews of motivation in software engineering, [2] and [3], (RQ2). Thus, we investigated the following two research questions:

- **RQ1** What are the characteristics perceived as important of software testers?
- **RQ2** Are there specific characteristics that separates software testers from a software engineer?

The target population for our study is represented by software testing professionals. In the testers’ category we will consider all software engineers who have software testing as their main job responsibility. A total of 14 participants from four companies were interviewed, with seven interviews being performed in agile working teams from two companies, while the other seven interviews were done in two companies following the traditional development methodology. The participants included testers and testing managers who face the daily problems of software testing activities. The companies and interviewees are described in Table 1.

During the interviews we used semi-structured questions in order to encourage the respondents to provide us with their own reflections and use their own terms. During the interviews the respondents were encouraged to freely express their opinions, by guaranteeing their anonymity and assuring them that the records will be accessible only to the researchers involved in this study. The interview guideline included both closed questions, for which responses will be easier to analyze and compare, and open
questions which will allow the participants to point out issues that were not mentioned in the closed-form questions. The open questions were theme based and concerned problems of testing, collaboration within their team, relationships with fellow colleagues, plus positive and negative aspects of their daily activities. In addition, we enquired about their working environment, schedules and the influence of the business domain. The same set of questions will enable us to see if there is a difference between the characteristics of a tester and those of the more general category of software engineers.

Table 1. Companies and interviewees

<table>
<thead>
<tr>
<th>Business</th>
<th>Size</th>
<th>Methodology</th>
<th>Testing</th>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software producer and service provider</td>
<td>medium international</td>
<td>Agile, TDD</td>
<td>Functional</td>
<td>Testing manager Tester (3)</td>
</tr>
<tr>
<td>Company B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software producer and testing provider</td>
<td>medium national</td>
<td>Agile, Scrum</td>
<td>Non functional</td>
<td>Testing manager Tester (2)</td>
</tr>
<tr>
<td>Company C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software producer</td>
<td>medium national</td>
<td>Traditional</td>
<td>Functional</td>
<td>Testing manager Tester (2)</td>
</tr>
<tr>
<td>Company D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software producer</td>
<td>large international</td>
<td>Traditional</td>
<td>Functional</td>
<td>Testing manager (2) Tester (2)</td>
</tr>
</tbody>
</table>

The interviews varied between half an hour and one hour and a half, and were performed on the company premises, in quiet meeting rooms. Each participant was interviewed individually. During the interviews the respondents were encouraged to express their opinions freely and as recommended by Myers and Newman [15], we used a mirroring technique in questions and answers in order to encourage the respondents to share their stories. During the interviews we asked the participants to talk about both current events and to reflect retrospectively on previous scenarios. All interviews were recorded and transcribed, and the transcriptions were sent to each participant for final checking and approval. Notes were also taken with the leading issues for each interview. The transcribed interviews were coded in several rounds.

Starting the process of analyzing the research data available, we first identified the segments of text relevant to the two research questions and discarded those bearing no relation to it. The transcripts were reexamined several times, and the coding process was performed in repeated rounds and the results were peer reviewed and discussed. Afterwards, we proceeded with the coding phase and labelled each segment by means of one or more easily recognizable terms or categories, using a software tool designed for qualitative analysis (NVivo 10) in which we imported all the transcripts. Afterwards the codes were analyzed and similar codes were aggregated into more general codes in order to reduce the number of codes utilized and retrieve the emerging categories. We derived the categories using the framework provided in the
studies by Beecham et al. [2] as a model for constructing a list of characteristics for software testers. Each category and code can be linked to quotations from the interviews and these are used to strengthen and enhance the results. The list combining characteristics from both studies [2] and [3] and our proposal is available in Appendix A.

4 Results

In this section we present and describe the characteristics of software testers derived from the study during the qualitative analyze process. Table 2 shows the characteristics sorted in quantitative descendent order starting from the characteristics mentioned by the higher number of participants:

<table>
<thead>
<tr>
<th>Most wanted tester characteristics</th>
<th>Rank</th>
<th>Most wanted tester characteristics</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication skills</td>
<td>7</td>
<td>Diplomacy</td>
<td>3</td>
</tr>
<tr>
<td>IT background</td>
<td>6</td>
<td>Need for challenge</td>
<td>3</td>
</tr>
<tr>
<td>Need for variety</td>
<td>6</td>
<td>Domain knowledge</td>
<td>3</td>
</tr>
<tr>
<td>Details oriented</td>
<td>5</td>
<td>Courage</td>
<td>3</td>
</tr>
<tr>
<td>Curious</td>
<td>4</td>
<td>Creative</td>
<td>2</td>
</tr>
<tr>
<td>Passion for quality</td>
<td>4</td>
<td>Proactive</td>
<td>2</td>
</tr>
<tr>
<td>Patient</td>
<td>4</td>
<td>Structured</td>
<td>2</td>
</tr>
<tr>
<td>Testing experience</td>
<td>4</td>
<td>Team player</td>
<td>2</td>
</tr>
<tr>
<td>Achievement orientated</td>
<td>3</td>
<td>Mind set to find bugs</td>
<td>2</td>
</tr>
</tbody>
</table>

Communication

Good communication skills is the most cited characteristic. Participants mentioned that during the communication process the testers must be able to know how to provide the right information and how to communicate it properly to the parties involved. "The communication characteristic, both oral and by writing is very important in a team because we often give some bad news (defects or problems) and it must be communicated in a way that do not blame anyone for it." (Tester Manager, Company C)

A tester is required to communicate with different project team members, managers, sometimes clients, to report defects, discuss and follow them up with the developers: "The tester needs communication skills in order to communicate with the project manager, developers and the users, and he has to make them understandable to each other" (Tester, Company C)

IT Background

Possessing an IT background refers to having an education and/or knowledge within computer science fields. Interviewees in this study held programming knowledge and experience, as valuable assets due to the expertise accumulated which can help in
certain testing activities. "A development experience can help a lot. All the knowledge I gain while working as a developer, helped me in my testing work.” (Tester, Company A)

An IT background was considered an important characteristic for a tester by our interviewees. The IT background proved valuable when communicating with the developers. “In order to speak to a developer and understand what they are talking about, an IT background is a very good thing”. (Tester, Company B)

**Need for Variety**
In all participant companies the **Need for variety** characteristic was often cited as a positive trait due to the varying nature of tasks in which a tester could be involved. A tester who enjoys having a variety of tasks could adapt better to the different nature responsibilities alongside a project lifecycle starting from the design phase until the maintenance stage. "The biggest factor for me is that you do different things, it's very varied and you get to see the whole picture. You can participate from the start of a project to the end doing various things, that's the biggest thing for me, the different things I can do." (Tester Manager, Company B).

**Details Oriented**
Being a details oriented person was considered an advantageous skill for a tester among the study participants. “You have to have an eye for details… you cannot just check the normally expected results. You cannot just test the happy path scenarios”. (Tester Manager, Company B)

**Curious**
Some of the participant considered that being curious was an incentive for continuously improving the understanding of the product and for pursuing several testing scenarios. "You don’t have to be afraid to click on new things and that you are going to break something. That is part of being curious." (Tester Manager, Company D) "Curiosity. You should want to learn new things because you will get new software, new projects, new teams all the time” (Tester, Company B)

**Passion for Quality**
Some of the participants have expressed their desire to improve the quality in the products they are working on and taking pride in participating in the delivery of a high quality product. They talked about their joy in investigating and finding defects which will lead to a better product. "I do have a passion for improving the quality and finding defects... I'm happy when I find bugs. Of course, I'm also happy when things are working.” (Tester, Company A)

**Patient**
Patience was mentioned is being an important requirement for software testers. The testers may need to work outside of regular office with little warning in advance. "you say: &lt;&lt;today I’ll leave at 4&gt;&gt;, but at 3:45 they say &lt;&lt;you have to do this testing before you leave today&gt;&gt;, and you have to work 2 or 3 hours more. That’s a part of it” (Tester Manager, Company D)
Patience was considered critical in situations where often and recurrently the planned scheduled was discarded and new plans were proposed. "I think you have to be patient, because usually you are at the end of the cycle and your plans and timetables are going to change, a lot." (Tester, Company A)

Testing Experience
The interviewees thought that testing experience and testing knowledge could improve the design skills such as writing Test Cases: "Defining the TC's is very important and you find all kind of situations where you need quite a lot of fantasy, since you have to go further than the developers go. And that has a lot to do with experience" (Tester Manager, Company A)

Experience with requirements engineering was also valued among the interviewed testing participants: "I would like them to have some competency in requirements, how should requirements be, because often they will be asked to review requirements and provide feedback." (Tester Manager, Company D)

5 Discussion

The answer to RQ1, “What are the characteristics perceived as important of software testers?” can be constructed by starting from the information provided in Table 2. Communication skills are seen as a valuable characteristic both from the participants of this study and from the research literature. During the communication process the testers must be able to know how to provide the right information and how to communicate it properly to the parties involved. For example, management may not be interested in the details of every defect that has been located. They are interested in the overall quality of a release and if the release can go in production. Testers have to be able to write clear defects reports and they also must be able to describe a problem while taking into consideration the type of interlocutors: “The communication characteristic, both oral and by writing is very important in a team because we often give some bad news (defects or problems) and it must be communicated in a way that do not blame anyone for it.” (Tester Manager, Company C).

One contrasting results with [2] and [3] is that the three most rated characteristics, growth orientation, need for independence and low need for social interaction associated with introverts did not score high with our participants. The characteristic of being Introverted was not mentioned at all in our interviews. This seems to be one of the strong difference between a tester and a developer. If you are introverted, then spending most of your workday behind a computer screen alone, as a developer, is a good choice, but not necessarily for a tester. Testers often have to report problems and they have to be comfortable with conflict [7], which lead to a vast amount of communication on daily basis. They have to learn a new product or feature in short time and posses or develop an ability, so a certain need for variety and curiosity might prove the right incentive. A note of caution should be taken for this characteristic, the need of variety is a useful trait in a domain where you might often need to acquire new knowledge of a new topic, but a strong need for variety may clash with the
possible monotonous nature of some testing activities and desired characteristics such as patience and tolerance to tedium which corroborate the findings of Pettichord [7].

If we compare our results with the characteristics derived from studies for Software Engineers [2] and [3], we notice that communication is not mentioned among the study's result, which leads us to the conclusion that communication skills is a specific valuable trait for a tester, hence proving an answer for our second research question: "Are there specific characteristics that separates software testers from a software engineer?".

Other characteristics which were not present in [2] and [3] are patience and curiosity. Patience is a virtue, especially among testers. They might invest a lot of effort in testing a new release and it may turn out that an incorrect version was deployed, so all their test results are worthless. During the testing process, testers might encounter opposition, aggression or debating situations which require patience. A tester might need to listen to the comments, criticism and deal with frustrated colleagues or managers on a daily basis. Software Testing can and should be seen as a journey for discovering and exploring freshly delivered software. A curious and inquisitive mind will be important to understand the why and how of products, their interactions and dependencies.

In most of the existing studies on testing domain knowledge and testing experience are ranked highly, but our participant emphasised the need for an IT background. This type of background can provide knowledge and language which will improve the tester's ability to write technical reports, relate and discuss with developers.

One company valued the domain knowledge above the technical ones due to the sensitivity and complexity for the of business they were involved in. One possible explanation is that in some of the participant organizations, parts of testing are done by non-specialized software testers alongside their other activities, or as a temporary task according to the projects' needs. In these cases, the people performing the testing usually lack testing skills, experience, and training which can lead to difficulties when communicating with more technical member of the team and when reporting defects. This situation was also described in Pol's study [9], and one of the consequences is that testing is seen as a job anyone can do.

As explained by Black [8], we have to remind ourselves and be aware that different levels of testing requires different skills. Unit testing requires programming and software development skills, while specialized tester are doing functional testing. When the process reach the level of acceptance or beta testing the most fitted testers are the one with best user knowledge such as people from customer support or business analysts. One negative aspect of this practice will be to rely too much on borrowing people from support group for testing, since they might not have sufficient knowledge for testing at earlier stages.

Limitations and Threats to Validity
In order to avoid threats to validity presented by Robson [16] in this kind of research, we ensured observer triangulation by having the data analyzed by three researchers, while the collected data and the results of this study were compared with our earlier quantitative study [17], which allowed us to apply data and method triangulation.
We are aware that the low number of participants is a limitation and given the high number of variables playing an important role in the survey, the results of this study should be considered as preliminary. However, since the focus was on depth rather than breadth we still think that the participants provided a typical sample giving us a lot of input and perspectives. Since increasing the number of participants could reveal more details and strengthen the conclusion of this study, our plan is to further expand our research by inviting new companies and increase the total number of interviewees. In addition, we will enhance our target population by inviting developers and project managers to take part our study. This augmentation could provide us with valuable information on the desirable characteristics for testers as seen from the developers perspective and from a higher managerial angle.

In addition to the small sample research, other factors such as the organizational mode or the type of product might influence the characteristics which were most valued or encountered in the participant companies. Nevertheless, our qualitative analysis spanned across four companies with mixed sizes and methodologies, involved in national and international business, handling data with medium or high risk. The participants were involved in performing functional and non-functional testing, and were coming from a mixed background with some having good technical skills and an IT background, while other were poses more strength in the domain business knowledge. All the mentioned factors could give better generalizability of findings than performing interviews in just one company [15].

6 Conclusion

In this paper we present the results of a qualitative survey among four Norwegian companies and the characteristics which were perceived as most valuable for a software tester. We contrasted our results with exiting work on software engineers' characteristics and noticed that the most frequent cited characteristics for software engineers were not applicable to software testers. A strong emphasis was laid on communication skills and on IT background which implied programming knowledge and certain technical capabilities. Testing knowledge and domain knowledge were also valued but not as strongly as found in previous research conducted Beer and Ramler [11]. Testing experience was considered to be an advantage especially for the testers involved in domains with volatile requirements which seems to be in line with Martin, Rooksby et al. [10] earlier study.

The list of characteristics provided by our study should not be seen as exhaustive or definitive but rather as a starting point and each company should consider the type of knowledge they value at hiring and the ability to provide internal training for testing or domain knowledge for their relevant business.

Acknowledgment. We are very grateful to the companies and their employees for their contribution to this project. Also, we would like to thank to Torgeir Dingsøyr at the Norwegian University of Science and Technology, for comments of versions of this article.
References

## 7 Appendix

### Characteristics for software engineers

<table>
<thead>
<tr>
<th>Proposing study</th>
<th>Software Engineer characteristics</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Beecham 01</strong></td>
<td>Growth orientated</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Introverted</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Autonomous</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Need for stability</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Technically competent</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Need to be sociable</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Achievement orientated</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Need for competent supervising</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Need for variety</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Need for challenge</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Need to make a contribution</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Need for feedback</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Marketable</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Creative</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Need for involvement in personal goal setting</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Need for Geographic stability</td>
<td>1</td>
</tr>
<tr>
<td><strong>França01</strong></td>
<td>Competent in Management</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Flexible / Team worker (easy to work with)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Have fear of punishment</td>
<td>1</td>
</tr>
<tr>
<td><strong>Our study</strong></td>
<td>Communication</td>
<td></td>
</tr>
</tbody>
</table>
A Comparative Study of Testers’ Motivation in Traditional and Agile Software Development

Anca Deak

Department of Computer Science
Norwegian University of Science and Technology Trondheim, Norway
deak@idi.ntnu.no

Abstract. The future software engineers looking for positions in the software industry tend to lean towards software development/coding rather than software testing. Our study investigates what factors cause software testing professionals working both in agile and traditional methodologies, to choose and remain in this career path. Using a qualitative survey among software development companies we retrieve information about the difference between the traditional and agile testers. In addition we identify information about the motivating and de-motivating factors in current testing practices. The results could help the companies in their recruiting processes, in the transition from traditional to agile within a company and in motivating their testers, which will lead to better job satisfaction and productivity.

Keywords: software testing, agile, waterfall, motivation, testers, human factors.

1 Introduction

The aim of this paper is to investigate the motivational factors impacting a software tester, observing them from the perspective of working methodology and comparing them with the existing results developed for the software engineer category. Motivation has been repeatedly cited as an important factor in productivity, quality and the successful delivery of a project within budget and time constraints [1] with several motivation theory emphasizing the importance of employee motivation such as Herzberg [2] and Mayo [3]. In this study we are looking at the positive and negative factors which influence professional software testers' motivation when working in traditional and agile methodologies. The subject of motivation within software engineers was the scope of an extensive systematic literature review performed by Beecham et al. [4] and updated by Franca et al. [5]. The two studies provided us with groups of motivators and de-motivators for a software engineer, as they were identified in their literature reviews.

Although there is extensive work on motivation in IT personnel [6] and on motivation agile teams [7] and [8], to our knowledge there is a lack of research focusing specifically on motivation in software testing. A tester's position can be similar to a software engineer, but there is a particularity of the testing jobs which is...
prone to situations where discussions or diplomacy might be required. The results of a survey looking at the human factors which have a negative influence on real practice of software testing in software companies in Spain identified the following factors: instability of testers positions (48 %), lack of attractiveness of testing (48 %) and poor career development for testers (41.7 %), [9]. The study advises us to seriously take into consideration these factors due to the high percentage of respondents. The human and social aspects of working in testing, or inside a testing team, as well as the attitude towards the testing team in a company, were studied from the testers’ perspective in a case study by Shah and Harrold [10]. These lead us to believe that it is worthwhile to make specific investigations about motivation among software testers. The research will be guided by the investigation of these factors, their relationships and effects on job outcomes.

While there is a documented lack of interest in choosing and pursuing a testing career [11] with development positions seen as more rewarding from career and financial perspective, there is still little research on why professional software testers choose to remain in their position. Our research is focusing on retrieving the motivation for those who chose to remain in software testing as a profession, while observing them in two different working environments, the agile and traditional settings. The final results will be compared to the factors in [4] and [5], which will enable us to observe if there are specific motivational factors for a software tester. With the framework of these criteria in mind we retrieved information about the difference between the traditional and agile testers, which could help the companies in their recruiting processes or in the transition from traditional to agile within a company. The study can also provide recommendations to companies and management for motivating the testing personal, which will lead to a better job satisfaction, productivity and quality of the developed product.

The rest of the paper is organized as follows: Section 2 presents the related research while the research question and methods used are described in Section 3 together with the research design and data collection process. In Section 4 we described and examined the results, while in Section 5 we examine the findings of the study and discuss the implications and the future work for this research.

2 Research Context

2.1 Traditional and Agile Testing

Software testing is a process through which the functionality of a software program is assessed during the development process. The investigation conducted for the assessment will focus both on verification and validation [12]. During the verification, the testers ensure that the software correctly implements a specific function and otherwise formulated, that it answers the question: are we building the product right? Validation ensures that the software has been built in order to satisfy customer's requirements. Validation answers the question: are we building the right product? The software testing process is also used to evaluate the nonfunctional quality of a system, by assessing aspects such as performance, security or usability.
For the traditional methodology, the software development lifecycle is constructed from a sequential set of stages: starting with a feasibility study at the top level and finishing with the implementation of the product. One trait of this methodology is the placing of testing towards the end of the project life cycle, which leads to defects being discovered close to the production deployment stage. In the requirement stage, the software tester can check if the requirements are according to the client’s wishes, while during the design phase, the tester can verify if the design document covers all the requirements and review the design document from the architecture perspective. In the coding phase the testing team can execute test cases, as well as generating testing data. In the testing stage running the system test cases can verify whether the system operates according to the stated requirements. Once the product reaches the maintenance stage, the tester can retest new fixes and patches and afterwards use regression testing to ensure that the new changes do not impact functionality in an unintended manner.

In an Agile Environment there would be more emphasis on collaboration and face to face interaction. The testers will be involved earlier into the development process and the development team will have write unit tests first and then code, rather than code first and then create a test plan which tends to occur in traditional environments. Members of an agile team are expected to be cross-functional so they may have to write code, do requirements elicitation or work closely with the customer. No QA/Testing department will be present and the person involved in the testing activities will be seen as "a team member who has most testing experience" rather than "a tester".

2.2 Studies on Testing Practice

Software testing and industrial surveys of testing practices such as [13] and [14] have been central themes in the specialized literature. Brain and Labiche [15] have emphasized the importance of testing research in an industrial setting, by arguing that the human influence and experience are important factors to be considered when performing testing related research and that the most applicable results are the ones obtained by observing professional testers at work.

A certain ad-hoc practice was underlined in [14], while the importance of experience and domain knowledge in testing was emphasized by Beer and Ramler [16]. Their multiple-case study, covering three industrial software projects, classified two categories of experience: experience in testing and experience with the product domain. Having a degree of experience in software testing proved to be useful for those involved in general management of the testing and particularly for those working with test automation. Product domain knowledge also proved valuable when working with test case design, planning regression testing and requirement's engineering. Those results reinforced Turley and Bieman's conclusion that experience is a valued asset for software engineers [17], and those of the ethnographic study conducted by Martin et al. [18] on testing processes and practices in a small start-up company. The Martin et al. [18] study, which focused on integration and acceptance testing done in the company, showed that testers working in contexts where
requirements were not defined in detail and without any strict processes, needed understanding of the business and experience in the domain and techniques that were used to test the product. In addition, testers were also required to possess good skills in test automation.

The perceptions of software testing were in the focus of an industrial survey conducted by Causevic et al. [19] and in that of the empirical study of a testing team in a vendor organization conducted by Shah, Harrold and Sinha [20]. The survey conducted by Causevic et al. [19], which uses both qualitative and quantitative methods, organized the results into four distinct categories: safety-criticality, agility, distribution of development and application domain. Their findings revealed the discrepancies observed between the current practices and the perceptions of respondents which could prove beneficial in shaping future research on software testing. One notable result from the quantitative analysis on satisfaction level of practitioners is related to Test Driven Development (TDD), which registered the most significant difference between the preferred practice and the current practice.

Among the findings of the empirical study conducted by Shah, Harrold and Sinha [20], is the enthusiasm showed by the testers about their job and their positive attitude toward testing, which is the opposite of the common attitude towards testing: where a software development job is preferred over a testing one. A desire for innovation and a high value among the testers were also observed in the same study. In addition, the results of this study show that the quality of testing is affected by motivation of testers and emphasizes the need for appreciating testers’ efforts. Taipale and Smolander conducted a qualitative study [21], which explored the software-testing practices and suggested improvements in this process based on the knowledge acquired during their study. Their improvement proposition include adjusting testing according to the business orientation of the company, enhanced testability of software components, efficient communication, early involvement of testers and increased interaction between developers and testers.

2.3 Existing Motivation Models in Software Engineering

Motivation in Software Engineering was the scope of a systematic literature review conducted by Beecham et al. [4], in which 92 papers published between 1980 and June 2006 were analyzed. The result of this study provided 16 characteristics of the software engineer together with 21 motivators and 15 de-motivators identified in the literature, which are available in the Appendix. Another, subsequent study by Franca et al. [5] extended and updated this results by analyzing 53 papers published between March 2006 to August 2010. As a result, another 8 additional motivators were identified: team quality, creativity/innovation, fun, professionalism, having an ideology, non-financial benefits, penalty policies and good relationship with users/customer, as well as a new de-motivator: task complexity. The study also shows that two of the motivators discovered in [4] were no longer present: appropriate working conditions and sufficient resources. The change noticed in the motivators and de-motivators also illustrates the evolving nature in the motivation of software engineers and this is expected to change even more as the software engineering field is evolving.

Most of the studies involved in these two literature reviews were quantitative survey studies and they provided important insights into characterizing the factors and
results related to motivation. One limitation of the mentioned studies, which we need to consider, is that the majority of the studies are referring to the job itself as being the main motivational factor. Since the title of software engineer can contain multiple roles and responsibilities which can greatly vary from one position to another, more information about the job that motivates the software engineer is required. Our study focuses on software testers, who are often considered as software engineers in job title terminology, but have different responsibilities than developers. Based on the results presented in systematic literature review conducted by Beecham et al. [4], same group of authors have studied different models of motivation and proposed a new model which was compared with the previous models and refined based on this comparison in Sharp et al. [22]. A systematic review of motivators in the agile context conducted by de O. Melo et al. [23], highlights differences between the overall view of motivation in software development and the motivation in an agile context. The study, which in addition includes three case studies in agile companies, suggests that certain motivators have an increased importance in agile teams and provides new motivators. The same study also claims that motivation seems to be higher for agile development teams which were previously exposed to other working methods.

3 Research Method

The scope of this paper is to investigate the motivational factors impacting a software tester, observing them from the perspective of traditional and agile working methodology and comparing them with data analyzed for the software engineer category. These objectives are reflected by the following research question:

- **RQ**: How do motivational and de-motivational factors for software testers differ in agile environments versus traditional type environments?

3.1 Survey Design

The population of our study is made by software testing professionals with testing experience. In the software testers' category we will refer to all software engineers who have software testing as their main job responsibility. In addition, we discussed with few developers who were involved with testing as part of their responsibilities.

A total of 26 participants were interviewed from six companies, from which 13 interviews were performed in agile working teams, while the other 13 interviews occurred in teams following the traditional development methodology. The interviewees included testers and testing managers who face the daily problems of software testing activities. In company F we talked with members of two teams involved in testing activities, one agile working team and one team following the traditional development methodology. The companies and interviewees are described in Table 1.

During the interviews we used a semi-structured guideline and open questions to encourage the respondents to provide us with their own reflections and use their own terms. The interview guideline included both closed questions, for which responses
will be easier to analyze and compare, and open questions which will allow the participants to point out issues that were not mentioned in the closed-form questions. The open questions were themes’ based and concerned problems of testing, collaboration within their team and relationships with fellow colleagues. In addition we enquired about positive and negative aspects of their daily activities, working environment, schedules and the influence of the business domain orientation. In parallel with this process we will check if all the motivators and de-motivators of software engineers present in literature can be applied for software testers as well. The same set of questions will enable us to see if there is a difference in the priority of these motivators and de-motivators between traditional and agile testers and for testers in relation to the more general category of software engineers.

Table 1. Companies and interviewees

<table>
<thead>
<tr>
<th>Company</th>
<th>Business</th>
<th>Size</th>
<th>Methodology</th>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Software producer &amp; service provider</td>
<td>medium international</td>
<td>Agile, TDD</td>
<td>Testing manager(1) Tester (2) Developer (1)</td>
</tr>
<tr>
<td>B</td>
<td>Software producer &amp; testing provider</td>
<td>medium international</td>
<td>Agile, Scrum</td>
<td>Testing manager(1) Tester (2)</td>
</tr>
<tr>
<td>C</td>
<td>Software producer</td>
<td>large national</td>
<td>Traditional</td>
<td>Section manager (1) Testing manager (1) Tester (2)</td>
</tr>
<tr>
<td>D</td>
<td>Software producer</td>
<td>large international</td>
<td>Traditional</td>
<td>Section manager (1) Testing manager (2) Tester (3)</td>
</tr>
<tr>
<td>E</td>
<td>Software producer</td>
<td>medium international</td>
<td>Agile</td>
<td>Testing manager (1) Tester (1) Developer (1)</td>
</tr>
<tr>
<td>F</td>
<td>Software producer</td>
<td>large international</td>
<td>Agile/Traditional</td>
<td>Testing manager(1) Tester (3) Developer (2)</td>
</tr>
</tbody>
</table>

The duration of the interviews varied between 30 minutes and 90 minutes, and they were performed on the premises of each company, in quiet meeting rooms where each participant was interviewed individually. During the interviews the respondents were encouraged to express their opinions freely, by guaranteeing their anonymity and assuring them that the records will be accessible only to the researchers involved in this study. As recommended by Myers and Newman [24], we used a mirroring technique in questions and answers in order to encourage the respondents to share their stories. During the interviews we asked the participants to talk about both current events and to reflect retrospectively on previous scenarios. All interviews were recorded and transcribed, and the transcription was sent to each participant for final checking and approval. Notes were also taken with the prominent issues for each interview. The transcribed interviews were coded in several rounds. All data has been anonymised, which included changing names and removing unnecessary details.
Starting the process of analyzing the research data available, we first identified the segments of text relevant to the research question and discarded those having no relation to it. Afterwards, we proceeded with the coding phase and labeled each segment (or sub-segment) by means of one or more easily recognizable terms or categories, using a software tool designed for qualitative analysis (NVivo 10). The codes were analyzed and similar codes were aggregated into more general codes in order to reduce the number of codes utilized and retrieve the emerging categories. The transcripts were revisited several times, and the coding process was performed in repeated rounds and the results were reviewed and discussed with my senior colleagues. Each category and code can be linked to quotations from the interviews and these are used to strengthen and enhance the results. The categories were derived based on the results provided in the studies by Beecham et al. [4] and [5] as a model for constructing a list of motivators and de-motivators for software testers. Two tables, one combining the de-motivators, and another one combining the motivators from both studies and the ones emerging from our study, are available in the Appendix.

4 Results and Discussion

In this section we present and describe the concepts for negative and positive factors, and we present a comparison between these factors based on the working methodology for traditional and agile testers.

4.1 Concepts for Negative Factors

In Table 2 we can observe the relationship between codes and concepts for negative factors derived from the study after the qualitative analyze process. The negative factors are presented in descending order starting from the one who was most frequently mentioned in the interviews:

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of influence and recognition</td>
<td>late involvement in the project, testing is underestimated in the company, afraid of opening defects, no control over the schedule</td>
</tr>
<tr>
<td>Unhappy with management</td>
<td>insufficient resources, unrelated tasks</td>
</tr>
<tr>
<td>Technical issues (NEW)</td>
<td>versioning, insufficient number of test environments, poor quality, integration issues with simulators</td>
</tr>
<tr>
<td>Lack of organization</td>
<td>lack of clear processes, tasks, redundant meetings</td>
</tr>
<tr>
<td>Time pressure (NEW)</td>
<td>squeeze, long days, short periods, overloaded schedule</td>
</tr>
<tr>
<td>Boredom</td>
<td>routine, repetitive tasks, unchallenging work</td>
</tr>
<tr>
<td>Poor relationships with developers</td>
<td>bugs related friction, stereotypic view of testing, slow defect fix rate, late changes to the code</td>
</tr>
<tr>
<td>Working environment issues</td>
<td>colleagues with no social antenna, open plan landscape related issues</td>
</tr>
</tbody>
</table>
Concept - Lack of influence and recognition
The concept which appeared most often as a factor with negative impact was the lack of influence and recognition. Under this concept we gathered the segments referring to the irregular working flow, and lack of control over an unstable schedule. Testers' late involvements in the development cycle, together with the struggle for recognition are also frequently cited by the participants "When I as a tester or test manager enter a project too late in the process to get a reasonable contribution to the quality with the testing." (Tester, Company C). When the focus of testing activities is more on testing issues, like retesting defects, rather than testing the product or requirement, testers are not provided with a sense of accomplishment, but rather with a frustration of not performing their real job. Under the same no sense of recognition concept we aggregated the worries for an unattractive career path development, with a low likelihood of promotion, in comparison with other roles, such as the ones for developers. "The developing projects and the daily operations have to realize how important software testing is. The testing area has to be lifted up as an important part of the company’s work." (Testing Manager, Company C)

Concept - Unhappy with management
The second most mentioned concept addresses participants’ dissatisfaction with the management related policies, the unrealistic schedules and the scarcity of resources. An unsupportive management can lead to tester being reluctant when they need to log in new defects: "testers use a lot of time, they are afraid of opening defects". (Testing Manager, Company D) Opening a critical bug can be a stressful scenario even for an experienced tester, it can lead to frictions with the fellow developers or with conflicts with the management: "I found bugs which stopped or hold a release, which on one hand is a good thing, because if the bug will have go into production it will have created serious problems, but is also a little bit like putting your reputation in line. The release is stopped because of you." (Tester, Company A) Raising defects which prove to be invalid can be detrimental for a tester but it is a natural part of his or hers career but can lead to a lack of respect from the developers or pressure from managers.

Concept - Technical issues
Technical issues within testing context are referring to problems with testing tools, development environments or a weak infrastructure. An insufficient number of test environments, poor quality or insufficient fidelity to the actual system being tested, together with integration party with 3rd party tools or simulators were mentioned as hindering factors of a technical nature. "It takes a lot of time to get the tests started, not everything works correctly, setting up an environment and also installing the software on our test servers." (Tester, Company B). In some companies the participants complained about the weak infrastructure which was proving to be the root cause in many false defects and required time and effort in investigations. “My main frustration is that we don’t have good enough tools to do our work and we have to use tools that make our work a lot more difficult than it should be.” (Developer, Company F)

Concept – Lack of organization
The interviewees were not pleased with the continuously changing plans or bad planning from the beginning. In addition some of the participants were having an increasing number of tasks which were not related with testing or outside their focus
area. “We fill a lot of time until we don’t have any left space, but often we want to update the plan.” (Tester, Company F). Participants related the lack of organization or carefully planning as a strong source for the repeating time pressure problem for members of the testing team. A high number of meetings which were considered redundant or irrelevant to their work tasks were also mentioned as a time consuming negative factor.

Concept - Time pressure
Another concept which appears often as a factor with negative impact was the time pressure associated with testing execution. Traditional working teams often delay testing until the end of projects, squeezing it in the process. Unfortunately, projects often fall behind schedule, so the testing teams need to compress and sacrifice the activities due to their shrinking time frame. "I've been in this business for many years and testing is at the end of this lifecycle, and always pressed to so short periods, long days, and shortcuts. It's always like that." (Testing Manager, Company D). Testing time is sacrificed to recover the delays in other processes and by doing so there is often a compromise on the quality of the delivered product. "I don't like that we are the last link in the chain, and we don't always get the time that was promised in the beginning. Give us more time to finish our testing and do it properly.” (Tester, Company D).

The concept appears also in the interview with testers from agile teams where the testing is occasionally facing similar time pressure. The company has sprints with unbreakable deadlines, but since the first half is allocated to test case designs, issues are often discovered late in the sprint. This situation gives little time to fix the issues. "Sometimes it's difficult to plan because they don't really know when they are ready. They want testing done immediately as they are ready, but they themselves don't really know when they are ready." (Tester, Company A)

Concept - Boredom
Some of the participants mentioned the routine of some testing activities and the feeling of boredom associated with maintenance testing. "Everything is routine, there is no surprises after the system is in production" (Testing Manager, Company D)

Concept - Poor relationships with developers
The second most mentioned concept is the relationship between testers and developers, which can be problematic at times. Most of these frictions results from discussions related to bugs. "I do remember having discussions about bugs: Is it really a bug or is it really important enough to be included in the release." (Tester, Company A). Another factor quoted by many participants was the stereotypic view of testing by the developers, "the classical view that you are developing and finally we are testing and then it's coming back with us saying <<that is not good, that is not good>>” (Testing Manager, Company A).

Two testers from different companies described their co-workers' view of testing as "a necessary evil". The slow defect fix rate and developers making unannounced late changes to code were also mentioned as a factor of concern and conflict between developers and testers. "It's a lot of things, challenges that take time, sometimes it can take time to get environments, sometimes you raise bugs and they don't take them quickly enough” (Testing Manager, Company D)
Concept - Working environment issues
Several participants complained about working in open space landscape which is considered noisy due to the nature of the office design but also due to colleagues with no social antenna. “When it comes to office conditions it can be quite noisy in this open landscape thing.” (Tester, Company F)

4.2 Concepts for Positive Factors

The relationship between codes and concepts for positive factors derived from the study are presented in Table 3. The positive factors are presented in descending order starting from the one which was most frequently mentioned in the interviews.

Table 3. Relationships between codes and concepts for positives factors

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive factors</td>
<td></td>
</tr>
<tr>
<td>Enjoy challenges</td>
<td>Enjoy challenging yourself, every day you never know what’s coming up, like the chaos, need challenges</td>
</tr>
<tr>
<td>Focus for improving the quality</td>
<td>finding bugs, to investigate, making things better, personal goal on improving the quality</td>
</tr>
<tr>
<td>Variety of work</td>
<td>work variation, combine testing and programming,</td>
</tr>
<tr>
<td>Recognition</td>
<td>ensure that testing tasks are important in the company, send testers to courses and conferences, get the support I need to a good job</td>
</tr>
<tr>
<td>Good management</td>
<td>good communication in the team, with developers, enough resources</td>
</tr>
<tr>
<td>Technically challenging work</td>
<td>technically challenging work</td>
</tr>
</tbody>
</table>

Concept - Enjoy challenges
Most of the interviewed participants enjoyed challenges represented by the testing activities, challenging themselves or simply thriving on the chaos which can sometimes accompany the daily activities of a tester. “When I perform my test and it works, I’m thinking: Am I doing something wrong? Is the test doing what it’s supposed to? When it fails, I’m also thinking: Is it really doing things correctly?” (Tester, Company A) and “There is always something new, new challenges towards different test scenarios.” (Tester, Company D)

Concept - Focus for improving the quality
The second most occurring concept related to testers passion for improving the quality of the software, the pleasure in investigating and finding defects which will lead to a better product. “I do have a passion for improving the quality and finding defects. And there I have learned that I have different focus than the developers, maybe the right focus for testing. I’m happy when I find bugs. Of course, I’m also happy when things are working.” (Tester, Company A)
Concept - Variety of work
On several occasions the concept of **variety of work** was mentioned and it referred to being included in the testing activities associated with the whole development cycle, not just a specific phase. Another contribution to the variety was considered having a combination of programming and testing tasks as part of job responsibilities. “The biggest factor for me is that you do different things, it’s very varied and you get to see the whole picture. You can participate from the start of a project to the end doing various things, that's the biggest thing for me.” (Tester, Company B)

Concept – Recognition
The concept of recognition included awareness of testing importance in the company both from management and development teams as well as positive feedback received from developers in relation to discovering and fixing bugs. “When we heard feedback from engineers, when we hear they say <<thank you, this test helped us to pick something that is wrong>>” (Tester, Company F) Under the same category we included participants expressing the pride they experience by working in a company known for delivering high-end products. “I believe I work in a company that is delivering high end embedded software for the worldwide. I want to make sure that the software we deliver has high quality.” (Testing Manager, Company E)

Concept - Good management
Under the concept **Good management** we aggregated all the positive references to relations and communication with the managers, between the testers and with developers. "I think is important to be on good terms with the developers; if they are having some Agile approach, you as a tester or test manager will get invited to their daily Scrum, so you get a feel for the modules they are struggling with and so on. It can help you prioritize, when you start to test.” (Testing Manager, Company D)

Concept - Technically challenging work
Another positive concept **Technically challenging work** was associated with the participants need to have allocated tasks reflecting their technical competencies. “The most interesting thing that you can have is interesting technology to work with.” (Developer, Company D)

If we look at the list of concepts from which we derived the factors available in Table 4 and Table 5, we see that while both types of testers enjoy having a degree of variety in their work, the lack of influence and recognition is a major negative factor for most of the participants involved in this study. If we compare the concepts emerging from this study with the list of de-motivators and motivators available in the Appendix we noticed that several new factors emerged from our study: **Time pressure** and **Technical issues** within testing context for the negative factors. On the positive side we identified new factors **Enjoy challenges** and **Focus for improving the quality**. All these concepts are specific to the nature of testing activities with **Technical issues** within testing context involving large quantities of effort and time invested in items which should be readily available at the beginning of testing. The **Time pressure** concept is referring to the tendency of testing time to shrink from the original estimate until the actual execution period is taking place.

Table 4 and Table 5 show a comparison of positive and negative factors between the testers from the two groups, based on the number of respondents mentioning these factors. If we look at how the factors are distributed among traditional and agile...
testers, we easily observe a higher time pressure factor for the traditional testers, while the Lack of organization tends to score higher in the agile teams. The lack of influence and recognition is present in both type of working environments with the traditional teams having a slightly higher occurrence. When discussing with general managers in companies working in the traditional way, they signaled several problems with the testing position, such as a struggle for recognition as a valuable team and also frustration coming from the lack of influence when suggesting recommendation or requests related to their working activities. Whatever methodology is followed, all the participant companies are interested in providing Product Quality. What differs from traditional to agile is that testing is started early in the sprint and the emphasis on testing has improved with practices such as TDD.

Table 4. Positive factors for traditional and agile testers grouped by methodology

<table>
<thead>
<tr>
<th>Positives factors</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enjoy challenges</td>
<td>3</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Focus for improving the quality</td>
<td>7</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>Variety of work</td>
<td>6</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Recognition</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Good management</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Technically challenging work</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5. Negative factors for traditional and agile testers grouped by methodology

<table>
<thead>
<tr>
<th>Negatives factors</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of influence and recognition</td>
<td>9</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>Unhappy with management</td>
<td>9</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>Technical issues</td>
<td>9</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>Lack of organization</td>
<td>8</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Time pressure</td>
<td>3</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Boredom</td>
<td>7</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Poor relationships with developers</td>
<td>5</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Working environment issues</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Testers working in Agile do not belong to a separate testing group, but work within the development team. They consider testing an ongoing process that happens throughout the development process, not just something that happens in a separate phase after development is done. Another point is that testing is done by the whole team, rather than just by testers and the relationship between testers and non-testers tends to be collaborative rather than adversarial. It was interesting to notice that more agile testers were unhappy about their relationship with developers since testers get more respect on agile teams where they are seen as colleagues, and are involved much earlier in the process, making it easier to ensure a system is produced that's easy to test. It might be related to a situation where a company applies customized version of agile methods “for good organizational reasons” [18]. Participants form both categories complained about the heavy load and unrealistic schedules which is in concordance with earlier research results [6].
Both categories of testers face the time pressure issue and although the initial model proposed by Beecham includes stress as a strong de-motivational factor, we feel that time pressure is such a specific and persistent problem during testing activities that we can assign it a separate category. A complete list of motivating and de-motivating factors for software testers, including the ones proposed during this study are available in the Appendix.

Limitations and Threats to Validity
The results of our study should be treated with some caution since there are other factors which may impact the motivation of a tester such as the organization structure, internal policies and processes. In addition, motivation can be influence by human factors such as personality types [25], and individual characteristics such as age [26]. In order to avoid the threats to validity presented by Robson [27] in this kind of research, we ensured observer triangulation by having the data analyzed by three researchers. In addition, the collected data and the results of this study were compared with our earlier quantitative study [28], which allowed us to apply both data and method triangulation. We are aware that the low number of participants is a limitation and given the high number of variables playing an important role in the survey, the results of this study should be considered as preliminary, but since the focus was on depth instead of breadth we still think that the participants provided a typical sample giving us with a lot of inputs and perspective. Since increasing the number of participants could reveal more details or strengthen the conclusion of this study, our plan is to further expand our research by engaging with other companies and increase the total number of interviewees. A longitudinal study may provide further insights into the motivational and de-motivation factors of software testing personal. Our qualitative analysis spanned across six companies using traditional and agile methodologies, performing functional and non-functional testing, which could give better generalizability than performing interviews in just one company [24].

5 Conclusion and Further Work

The extensive research about motivation in software engineering has added to the body of knowledge characterizing the factors behind the motivation at the workplace. In this study, we looked at a specific branch of software engineering, namely software testing and we presented the main results of a qualitative study about motivation of testers in four software development companies.

We provided a set of factors with negative and positive influence on the daily activities of software testers and added additional categories to the ones already presented and published in the software engineering world. We look at the differences between testers working in traditional and agile development and noticed a higher degree of stress and a positive approach towards the challenges of testing activities for those engaged in the waterfall approaches, while the agile testers, although expressing more problems in communication with developers seemed to be better integrated into their teams. To further our research we plan to extend this study by involving more companies and in addition to look into the characteristics of testers and the relationships with their fellow coworkers.
References


### Appendix

**Table 6. List of de-motivating factors from previous work**

<table>
<thead>
<tr>
<th>Nr.</th>
<th>De-motivating factors for Software Testers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Proposing study Beecham et al.</strong></td>
</tr>
<tr>
<td>1</td>
<td>Stress</td>
</tr>
<tr>
<td>2</td>
<td>Inequity</td>
</tr>
<tr>
<td>3</td>
<td>Interesting work going to other parties</td>
</tr>
<tr>
<td>4</td>
<td>Unfair reward system</td>
</tr>
<tr>
<td>5</td>
<td>Lack of promotion opportunities</td>
</tr>
<tr>
<td>6</td>
<td>Poor communication</td>
</tr>
<tr>
<td>7</td>
<td>Uncompetitive pay/unpaid overtime</td>
</tr>
<tr>
<td>8</td>
<td>Unrealistic goals/phony deadlines</td>
</tr>
<tr>
<td>9</td>
<td>Bad relationship with users and colleagues</td>
</tr>
<tr>
<td>10</td>
<td>Poor working environment</td>
</tr>
<tr>
<td>11</td>
<td>Poor management</td>
</tr>
<tr>
<td>12</td>
<td>Producing poor quality software</td>
</tr>
<tr>
<td>13</td>
<td>Poor cultural fit/stereotyping/</td>
</tr>
<tr>
<td>14</td>
<td>Lack of influence/</td>
</tr>
<tr>
<td></td>
<td><strong>Proposing study Franca et al.</strong></td>
</tr>
<tr>
<td>15</td>
<td>Task Complexity (too easy or too difficult)</td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th>Nr.</th>
<th>Motivating factors for Software Testers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proposing study Beecham et al.</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Rewards and incentives</td>
</tr>
<tr>
<td>2</td>
<td>Testing needs addressed (training opportunities; opportunity to specialize)</td>
</tr>
<tr>
<td>3</td>
<td>Variety of work</td>
</tr>
<tr>
<td>4</td>
<td>Career path (opportunity for advancement, promotion prospect, career planning)</td>
</tr>
<tr>
<td>5</td>
<td>Empowerment/responsibility (responsibility is assigned to the person not the task)</td>
</tr>
<tr>
<td>6</td>
<td>Good management (sr. management support, team-building, good communication)</td>
</tr>
<tr>
<td>7</td>
<td>Sense of belonging/supportive relationships</td>
</tr>
<tr>
<td>8</td>
<td>Work/life balance (flexibility in work times, caring manager/employer)</td>
</tr>
<tr>
<td>9</td>
<td>Working in successful company (e.g. financially stable)</td>
</tr>
<tr>
<td>10</td>
<td>Employee participation/involvement/working with others</td>
</tr>
<tr>
<td>11</td>
<td>Feedback</td>
</tr>
<tr>
<td>12</td>
<td>Recognition (for a high quality, good job done based on objective criteria)</td>
</tr>
<tr>
<td>13</td>
<td>Equity</td>
</tr>
<tr>
<td>14</td>
<td>Trust/respect</td>
</tr>
<tr>
<td>15</td>
<td>Technically challenging work</td>
</tr>
<tr>
<td>16</td>
<td>Job security/stable environment</td>
</tr>
<tr>
<td>17</td>
<td>Identify with the task (clear goals, personal interest, know purpose of task)</td>
</tr>
<tr>
<td>18</td>
<td>Autonomy</td>
</tr>
<tr>
<td>19</td>
<td>Appropriate working conditions/environment/good equipment/tools/physical space</td>
</tr>
<tr>
<td>20</td>
<td>Making a contribution/task significance</td>
</tr>
<tr>
<td>21</td>
<td>Sufficient resources</td>
</tr>
<tr>
<td><strong>Proposing study Franca et al.</strong></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Team quality</td>
</tr>
<tr>
<td>23</td>
<td>Creativity/Innovation</td>
</tr>
<tr>
<td>24</td>
<td>Fun (playing)</td>
</tr>
<tr>
<td>25</td>
<td>Professionalism (high professional environment)</td>
</tr>
<tr>
<td>26</td>
<td>Having an Ideology</td>
</tr>
<tr>
<td>27</td>
<td>Non-financial benefits</td>
</tr>
<tr>
<td>28</td>
<td>Penalty Policies</td>
</tr>
<tr>
<td>29</td>
<td>Good relationship with users/customers</td>
</tr>
</tbody>
</table>
Deak, A., Stålhane, T., Sindre, G. "Challenges and strategies for motivating software testing personnel" Accepted at Information and Software Technology journal
Challenges and strategies for motivating software testing personnel

Anca Deak, Tor Stålhane, Guttorm Sindre
Department of Computer Science
Norwegian University of Science and Technology Trondheim, Norway
{deak, stålhane, guttorm.sindre}@idi.ntnu.no

Abstract. Context Software testing is the key to ensuring a successful and reliable software product or service, yet testing is often considered uninteresting work compared to design or coding. As any human-based activity, the outcome of the final software product is dependent on human factors and an essential challenge for software development organizations is to find effective ways to enhance the motivation and job-satisfaction of their testers.

Objective Our study aims to cast light on how professional software testers can be motivated and we explore the policies and rules conceptualized and implemented inside software development projects.

Method This paper presents the results of an empirical study that collected data through semi-structured and in-depth interviews with 36 practitioners from 12 companies in Norway. The data collection was performed over a two years period and investigates the strategies applied by the companies for stimulating their testers, while considering the motivational and de-motivational factors influencing the testing personnel.

Results Our results provide a set of motivational and de-motivational factors for software testing personnel and present the strategies deployed by the companies for stimulating their testing personnel.

Conclusions The study shows that combining testing responsibilities with development and ensuring a variety of engaging and challenging tasks and products does increase the satisfaction of testing personnel. However, despite the systematic and sincere effort invested in recognizing the importance of testing and motivating the testers, heavy emphasis is laid on minimizing project costs and duration. The results could help the companies in organizing and managing processes and stimulate their testing personnel, which will lead to better job satisfaction and productivity.

Keywords: software development, software testing, motivation, testers, human factors, management
1 Introduction

Software testing is a crucial activity in the quality assurance of most software products. In spite of a wide range of available tools, it is still an activity requiring a lot of human labor, i.e., a socio-technical rather than purely technical activity [1], for which the outcome will be highly dependent on the performance of the involved employees. Unfortunately, findings both in industry [2] [3] and among IT students [4] indicate that many current and future software professionals consider testing as unattractive work. This may cause problems in recruiting and retaining testers, and low motivation can lead to poor testing, overlooking software defects [5]. Such problems are especially worrying in a time when the relative importance of testing to e.g., coding is increasing, due to more system integration projects and fewer green-field development projects, and relying gradually more on available components and services rather than coding entirely from scratch [6]. Though we have found few academic publications claiming shortage of skilled personnel for testing, except ([7], [8]) specifically for China, there are signals from industry in many countries (e.g., Britain, Australia, India)\(^1\) that this is becoming a key challenge, and that there is an increased need for testing-related training [9].

Massive automation of testing work might alleviate some personnel shortage, but as observed in [10], this is a long-term research goal, not a solution of the near future. Outsourcing and offshoring will work only if a shortage of skilled testers in some companies or countries is compensated by a surplus elsewhere, but as argued in the previous paragraph the shortage might instead be global. Hence, testing jobs must be made more attractive.

The research on human factors in software testing has been limited. Bertolino [10] gives a good overview of a variety of research challenges in software testing, outlining different angles by means of the question words why, how, how much, what, where, and when. A word lacking from her analysis framework, though, is who. This seems to be symptomatic of the research she is mapping. As also observed by Kanij et al. [69], testing as a human activity tends to be under-researched relative to technical issues.

Although there is extensive work on motivation of IT personnel in general [11] and specifically on motivation in agile teams, e.g., [12] and [13], to our knowledge there is a lack of research focusing specifically on motivation in software testing. There was a survey in Spain with 127 respondents [2], looking at human factors negatively affecting the practice of software testing. The factors reported by the most respondents were instability of tester positions (48 %), lack of attractiveness of testing (48 %), and poor career development for testers (42 %). Similar problems were also identified in a study by Shah and Harrold [3], investigating human and social aspects of working as a tester, or inside a testing team, as well as the attitude towards the testing team in a company. Our own investigations on student attitudes towards the prospect of a future testing career [4] revealed a similar image of low status for testing

---

work, with most students seeing development positions as more rewarding from a career and financial perspective.

Given the limited research on the topic so far, we find it premature for this article to focus on establishing advice to the industry on how to make testing more attractive. A necessary first step is rather one of gaining understanding: What aspects of testing work makes it unattractive? Positive aspects to testing work must also be investigated. After all, in spite of the mentioned problems in recruitment and motivation, it is also possible to find many professionals who actively pursue careers in testing, who really enjoy that kind of work [14] and stay with it for a long time. A prerequisite for making advice on how to make testing work more attractive is therefore both to understand its problematic issues, which must somehow be reduced, and its good sides, which should be kept or strengthened in spite of proposed changes. Hence, this paper poses the following research questions:

- **RQ1**: Which motivational and de-motivational factors influence testing personnel in their daily activities?
- **RQ2**: Which strategies are applied by companies to encourage their testers?

The idea behind RQ1 is to capture both the positive and negative aspects of testing work, as argued above. For RQ2, given the limited research on the topic so far, it seems natural to elicit descriptive knowledge of what companies are currently doing to encourage their testers, rather than jumping directly to the task of giving advice to companies on what should be done.

Other research questions could also have been justified by the observed problems related to recruitment, retention and motivation in software testing. For instance, the problem could be mitigated by improved education in testing, or improved recruitment strategies. Both these have, however, received some attention. Testing education is addressed by ([15], [16], [17], [18]) and several other publications. Related to recruitment strategies, there has for instance been work on what personality types are most suitable for testing work [19], [20]. Although interesting, these topics are out of scope for this article, whose focus is on factors that motivate or de-motivate testers, and what companies do to encourage their testers.

The rest of the paper is organized as follows: Section 2 presents the related research while the research methods are described in Section 3 together with the research design and data collection process. In Section 4 we present and analyze the results, while in Section 5 we examine the findings of the study and discuss the implications. Directions for future work are presented in Section 6.

## 2 Research context

Based on the research questions posed in the previous section, there are three topics we need to cover: (i) The concept of motivation, to establish a theoretical underpinning for the investigation. This makes it necessary to look at theory of work-
related motivation which is, however, not specifically targeting software testing. (ii) Findings from empirical software engineering related to industrial practice in testing, or related to motivation of software engineers / software testers. The last category relates most directly to our research questions, and if there was already a large body of theory and empirical findings here, we might not have needed to cover so much of the broader background. However, the limited literature makes it necessary to look broader to have a sufficient basis for the investigation. The two items presented above will be discussed in subsequent subsections below.

2.1 The concept of motivation

According to Ryan and Deci, motivation "concerns energy, direction, persistence and equifinality – all aspects of activation and intention" [21] (p. 69), while Robbins [22] stated that motivation is the willingness to do a certain action and is conditioned by this action’s ability to satisfy needs for the individual. When referring to motivation it is necessary to understand the differences between needs, drive, motivation and motives, e.g. Toates, [23], Deci and Ryan [24].

One main theory of motivation is Herzberg’s two-factor theory, also known as “Motivation-Hygiene Theory”. Herzberg identified the types of job related factors that influence employee motivation (he called it attitude) to perform well [25], [26]. Dissatisfaction factors (also called hygiene factors) are a group of factors that can cause negative attitude. These include unfair rules, poor physical working conditions and poor relationship with supervisors. The opposite, fair rules, good physical working conditions, and good relationships with supervisors, do not lead to particularly positive job attitudes, but at least gives absence of dissatisfaction, a neutral position. Motivation to do a good job is instead linked to another group of factors, such as responsibility, recognition, promotion and duties perceived as interesting. Herzberg called these motivational factors. Herzberg noted that motivational factors were primarily related to "the actual job", while the hygiene factors are more focused on "the job situation" [25].

The Motivation-hygiene theory classifies motivational factors into extrinsic and intrinsic factors. Extrinsic motivation means that the activity is necessary to achieve some desirable result, for instance material gains (e.g., salary, bonus) or increased status. Intrinsic motivation instead means that the reward lies in enjoying the activity itself. Intrinsic motivation was described by Deci and Ryan as "the inherent tendency to seek out novelty and challenges, to extend and exercise one's capacities, to explore, and to learn" [24]. Csikszentmihályi's concept of flow [27] also relates closely to intrinsic motivation.

De Jonge et al. [28] and Sargent and Terry [29] found that a work situation having both high job demands and job control was related to a high degree of work motivation and job satisfaction. Similarly, in a study of Swedish IT consultants, Wallgren and Hanse [30] found that influence on and variety in tasks contributed much more strongly to motivation than monetary incentives or company norms. High job demands, however, are positive only as long as they can be handled by the employee. If demands exceed abilities, time or resources, or employees do not feel
valued for their efforts, high demands will instead lead to stress, as observed by Kalimo and Mejman [31]. Connell and Wellborn [32] stated that people’s motivation to engage in a task is influenced by the extent to which the task provides opportunities to fulfill their basic needs for autonomy, social relatedness and a sense of competence. Elliot and Dweck [33] add the individuals’ need to feel valued and respected by their social group. Smith and Shields [34] investigate the characteristics related to job satisfaction among social service workers based on Herzberg’s concept of motivation to work. Their results, which support Herzberg’s insights, show that variety and creativity have the most predictive value and that experiences with supervisors also predicts job satisfaction.

2.2 Related work from empirical software engineering

Empirical studies of industrial testing practices such as [35], [6] and [36] have given important insights on how testing work is actually done, although they have not made a detailed inquiry into motivation of the employees. Briand and Labiche [37] have emphasized the importance of testing research in an industrial setting, by arguing that the human influence and experience are important factors to be considered when performing testing related research, and that the most applicable results are the ones obtained by observing professional testers at work.

A certain ad-hoc practice was found in [36], while the importance of experience and domain knowledge in testing was emphasized by Beer and Ramler [38]. Their multiple-case study, covering three industrial software projects, defined two categories of experience: experience in testing and experience with the product domain. The former proved to be useful for those involved in general management of the testing and particularly for those working with test automation. The latter also proved valuable when working with test case design, planning regression testing and requirements engineering. The study by Martin et al. [39], which focused on integration and acceptance testing in the company, showed that testers working in contexts where requirements were not defined in detail and without any strict processes, needed understanding of the business and experience in the domain. In addition, testers also needed good skills in testing techniques and test automation.

The perceptions of software testing were investigated by Causevic et al. [40] in an industrial survey and by Shah, Harrold and Sinha [14] in an empirical study of a testing team in a vendor organization. The survey conducted by Causevic et al. [40] revealed discrepancies between the actual practices and the perceptions of respondents. In particular, Test Driven Development (TDD) had the most significant difference between the preferred practice and the current practice among practitioners. Shah, Harrold and Sinha [14] found many testers being enthusiastic about their job, contradicting the common attitude that a software development job is preferred over a testing job. A desire for innovation and a feeling of high value among the testers were also observed in the same study, concluding that the quality of testing is affected by testers’ motivation and the importance of appreciating testers’ efforts. Taipale and Smolander [41] conducted a qualitative study which explored the software testing practices and suggested improvements in this process based on the knowledge
acquired during their study. They propose to adjust testing according to the business orientation of the company, enhance the testability of software components, and strive for efficient communication, early involvement of testers and increased interaction between developers and testers.

Motivation in software engineering was the scope of a systematic literature review conducted by Beecham et al. [42], in which 92 papers published between 1980 and June 2006 were analyzed. The result of this study provided 16 characteristics of the software engineer together with 21 motivators and 15 de-motivators. Another, subsequent study by Franca et al. [43] extended and updated this result by analyzing 53 papers published from March 2006 to August 2010. As a result, another eight additional motivators were identified: team quality, creativity/innovation, fun, professionalism, having an ideology, non-financial benefits, penalty policies and good relationship with users/customer, as well as a new de-motivator: task complexity. The study also indicated that two of the motivators discovered in [42] were absent in the newer literature: appropriate working conditions and sufficient resources. Possibly these are now taken for granted, thus not emphasized by respondents in recent years. Based on the results presented in a systematic literature review conducted by Beecham et al. [42], the same group of authors have studied several models of motivation and proposed a new model which was compared with the previous models and refined based on this comparison in Sharp et al. [44]. It is also worth mentioning, that in an analysis of the motivation theory used in the identified publications, Hall et al. [45] found that eight ”classical” motivation theories are represented. Lenberg et al. [46] argue that a synthesized view of the emerging human-focused Software Engineering research is needed and propose the concept of Behavioral Software Engineering (BSE) which focuses on the notions of human nature in software development work.

The recent literature on agile development includes studies of motivation and cohesion [47] and the motivational needs of extreme programming developers [48]. A recent systematic review of motivators in the agile context conducted by de O. Melo et al. [49], highlights differences between the overall view of motivation in software development and the motivation in an agile context. The study, which in addition includes three case studies in agile companies, suggests that certain motivators have an increased importance in agile teams. The authors also claim that motivation seems to be higher for agile development teams previously exposed to other working methods, than to those who used agile from the start. The study conducted by Tessem and Maurer found high motivation and job satisfaction in large agile teams [50]. The reviews [42] [43], and the papers surveyed therein, investigated the more general job category of software engineer, which can include multiple roles and tasks. As software testers have different responsibilities from developers, and may sometimes be in conflict with developers, this mandates specific investigations of motivation for testers. For agile development, the distinctions between developer and tester may sometimes be more blurred. Nevertheless, even if employees are heavily involved both in development and testing, it is interesting to explore their motivation for the testing work in particular. Our study thus adds to the existing body of knowledge by giving deeper insight into motivational challenges specifically for software testers, and company strategies to deal with such challenges, than what other studies with a more generic software engineering perspective have provided.
We have published some work on similar topics already. Most directly related, one of our earlier studies [51] take the motivational and de-motivational factors found in [42] [43] as a starting point, and then make a comparison of motivational challenges for traditional and agile testers. Here, several new factors emerged: Time pressure and Technical issues as negative factors and Enjoy challenges and Focus for improving the quality as positive. That work, however, was a more limited investigation looking specifically for differences between agile and traditional testers. The current paper is based on more data, and looks deeper into the motivational issues of testers regardless of development paradigm. Also, the current paper investigates company efforts to encourage testers, which was not addressed in [51].

2.3 Summary of the research context

Quickly summing up what to distill from this section, an study of testers’ motivation will need to investigate both extrinsic and intrinsic motivation, hygiene factors and motivation factors – as indicated by well-established theory on motivation. The general empirical findings on work motivation as presented above suggest that a sense of job control and high demands (as long as they can be met, thus giving a feeling of competence), variety of tasks, creativity, and a feeling of being respected in the work-place will contribute positively to motivation. On the other hand, stress, poor working conditions, lack of respect and poor relation to supervisors will likely cause low motivation.

As for the empirical literature in software engineering, it should be noted that both testing knowledge and domain knowledge is highly relevant for testers and can contribute to the above-mentioned feeling of competence. Also, placement of testing work in the development process, and the relationship between testers and developers, may affect motivation. Motivational factors found in previous works, e.g., [45], [46], [47], can form a good basis for our investigation. However, while there is much overlap between the factors found in those three studies, there are also some differences, indicating that one should also have the eyes open for discovery of new factors (i.e., there is no agreed, stable set of motivational and de-motivational factors to build on).

3 Research Methodology

3.1 Overall approach

Our research questions RQ1 and RQ2 as given in section 1 are about the motivating and de-motivating factors of software testers, as well as company strategies to encourage testers. The goal is to understand the industrial situation, not to improve it, although the results of this work can hopefully contribute to improvement in the future. This indicates an empirical approach, typical alternatives being
quantitative investigations such as questionnaire surveys or qualitative investigations such as case study. The advantage of a questionnaire survey is that input could be quickly collected from a broad range of respondents, and statistical analysis could uncover causal effects if quantitative questions are given. However, findings may be shallow and of limited value unless there is a sound theory behind the survey. Given the limited research in the area we found it more fruitful to use a qualitative approach, as our focus was to investigate and understand phenomena within their real life context and provide a better view and understanding of a certain phenomenon [52].

In a qualitative study, several data collection methods could be possible, e.g., observation or interviews. Observation would be highly relevant if investigating industrial practice (e.g., what employees do, which could differ from what they say they do). Our research questions about motivation would however be hard to answer from observing actions, unless having mind-reading abilities. Hence, interviews were more appropriate. In some cases, interviewing people in groups (e.g., focus group sessions) can be effective. For our research questions, this appeared less appropriate, as job motivation may be a quite personal matter that informants would speak less freely about in the presence of colleagues. Hence, individual in-depth interviews were chosen, using a semi-structured approach because this was found to strike a good balance between comparability of informants (asking everybody some identical questions), and possibility to elicit personal viewpoints "outside the box". With fully open interviews, findings might have been more difficult to synthesize, and with fully structured interviews, it would be easy to miss interesting informant viewpoints that somehow did not fit under any particular item in the interview guide.

3.2 Survey Design and Data Collection

Planning and selection

The interview guide was constructed by the first author with inputs from the second author, and it was reviewed by a fellow researcher. The design was validated by two industry experts and tried out in companies which were not participating in our study.

The interview questions were mapped on to the research questions to ensure that they were all covered and the interview guideline was organized in several subsections as shown in Appendix A.

The purpose of the Intro section is to warm up the participant and construct his or hers pathway to a testing career and see if the involvement in testing activities was the result of different factors and context or if it was a predetermined desire of the participant to actively pursue a testing career (Questions 1.1 – 1.3). The information retrieved at this stage is revisited again in Testing career and Closure sections where we asked the participants to express their views on a testing career (Question 6.1, 8.1).
The questions included in the *Testing organization* section allowed us to understand the working culture and methodology applied in the companies (Questions 2.1 – 2.3) which was a prerequisite for RO2.

Since *Technical issues* within the testing context was a recurring theme in our previous work we wanted to assess the existence and the weight of this problem in a company before going further into the communication and motivational factors (Question 3.7).

Since communication has a prominent role in a tester’s responsibilities, such as discussing defects with fellow developers and managers, we decided to explore this part of the tester’s work in the *Communication* section (Questions 4.1, 4.2, 4.2a). The quality of the communication can affect tester’s motivation and the sense of acknowledgement within the company, hence assessing this quality was mandatory for RQ1’s scope.

The questions belonging to the *Motivation* section are designed to elicit aspects of testing work which makes it unattractive, but also positive aspects. This entire section (Question 5.1 – 5.7) captures the information necessary for answering RQ1. Examples of daily work situations related to requirement handling and defects processing were included in the guide, as a basis for discussions in the interviews.

In order to retrieve the descriptive knowledge of the motivational strategies employed by the companies, we were interested to see what the companies were looking for in a tester in the first place when hiring for a testing position. We extracted this information through two questions in the *Tester profile* (Questions 7.1, 7.2) section and we compared the information provided with the employee’s perception (Question 5.7).

When interviewing managers we added some additional questions in the *Testing organization* section. These questions were related to the background and overall responsibilities of the employees who performed testing activities. In addition, when discussing the tester profile we enquired the managers of the criteria they used when hiring testers.

The guide was updated twice, first after a pilot interview and then after four initial interviews. The changes performed during these updates did not modify the general content of the guide, but the structure and order of the interview questions were modified and improved. The resulting interview guide is shown in Appendix A. In addition, we provided an email with general information to the interviewees before the interviews were done, to make them aware of the nature of the interviews, their rights to withdraw at any stage, and to ask permission for recording.

The aim of this study was to cover processes relevant to testing activities during the entire life cycle starting from the requirements stage through development to system testing and maintenance. For this reason, the population of our study contains software testing professionals as well as managers who are directly responsible for testing activities: testing, release and sections managers.

Under the software testers’ category we will include the software engineers who have software testing as their main job responsibility. In addition, we decided to include within the same category developers who were involved with testing for at least 50% of their person-hours. The participants were selected using a maximum variation sampling [53] and convenience sampling [53]. The participant companies were selected through the Software Testing group of The Norwegian Computer
Society. We invited 17 companies to take part in our study from which 12 accepted to participate in our research project. These above-mentioned sampling methods were combined with a snowball sampling [52] by asking the interviewees if they could recommend a person or a role in the company whom we could interview in order to get relevant information. This method proved an efficient way of getting new participants or companies involved. These suggestions were then matched against our selection criteria in order to obtain a wide coverage of the processes of interest. During the data collection process we contacted an internal representative for each company, who selected the most suitable persons in that company for the purpose of our study. Thus, the selection was not influenced by the researchers, nor were there any existent relationships between the interviewer and the interviewees. From an ideal scientific point of view, a complete random selection of interviewees might have been better, but this was infeasible due to the limited availability of candidate respondents.

A total of 36 participants from the 12 companies were interviewed. In many of the companies there were organizational units following different approaches with regard to development methodology. Within the same company we interviewed participants belonging to both types of groups: agile and traditional development teams. The testing manager from company L worked under both paradigms for long periods, and he was thus able to provide us with valuable inputs from both practices. Company J did not subscribe to any of the mentioned methodologies, the managers there chose to follow an internally developed procedure which had similarities with the Traditional style. A short description of the participant companies and interviewees is available in Table 1.

Data collection

The interview guideline included both closed questions, for which responses will be easier to analyze and compare, and open questions to allow the participants to point out issues that were not mentioned in the closed-form questions. The open questions concerned problems of testing, collaboration within the team and relationships with colleagues. In addition we enquired about their daily routines, working environment, schedules and the influence of the business domain. The interview guideline for the managers added questions concerning the strategies and the rules proposed or implemented for stimulating their employees.

The duration of the interviews varied from 30 to 90 minutes, and they were performed on the premises of each company, in quiet meeting rooms where each participant was interviewed individually. In the companies where several employees were interviewed in one visit, a 30 - 40 minutes time interval was allocated to each interview, while for the managers, 1 hour was planned from the beginning due to additional questions in the interview guide. During some of the interviews the participants expressed they had more time available, which allowed an extension of the discussion time up until 90 minutes.

During the interviews the respondents were encouraged to express their opinions freely, by guaranteeing them anonymity and assuring them that the records would be accessible only to the researchers involved in this study. As recommended by Myers and Newman [54], we used a mirroring technique in questions and answers in order to
encourage the respondents to share their stories. We used the interview guideline as a checklist to make sure that all the relevant topics were covered, and we asked the participants to talk about both current events and to reflect on previous scenarios.

Table 1: Companies and interviewees

<table>
<thead>
<tr>
<th>Company</th>
<th>Business</th>
<th>Size</th>
<th>Methodology</th>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Software producer &amp; service provider</td>
<td>medium international</td>
<td>Agile, TDD</td>
<td>Testing manager(1) Tester (2) Developer (1)</td>
</tr>
<tr>
<td>B</td>
<td>Software producer &amp; testing provider</td>
<td>medium international</td>
<td>Agile, Scrum</td>
<td>Testing manager(1) Tester (2)</td>
</tr>
<tr>
<td>C</td>
<td>Software producer</td>
<td>large national</td>
<td>Traditional</td>
<td>Section manager (1) Tester (1) Tester (2)</td>
</tr>
<tr>
<td>D</td>
<td>Software producer</td>
<td>large international</td>
<td>Traditional</td>
<td>Section manager (1) Tester (2) Tester (3)</td>
</tr>
<tr>
<td>E</td>
<td>Software producer</td>
<td>medium international</td>
<td>Agile</td>
<td>Testing manager (1) Tester (1) Developer (1)</td>
</tr>
<tr>
<td>F</td>
<td>Software/Hardware Producer</td>
<td>large international</td>
<td>Agile/Traditional</td>
<td>Testing manager(1) Tester (3) Developer (2)</td>
</tr>
<tr>
<td>G</td>
<td>Software producer</td>
<td>medium international</td>
<td>Agile/Traditional</td>
<td>Testing manager(1) Tester (1) Developer (2)</td>
</tr>
<tr>
<td>H</td>
<td>Software producer</td>
<td>large international</td>
<td>Kanban</td>
<td>Testing manager(1) Tester (1) Section manager (1)</td>
</tr>
<tr>
<td>I</td>
<td>Software/Hardware producer</td>
<td>large national</td>
<td>Agile/Traditional</td>
<td>Testing manager(1) Tester (1)</td>
</tr>
<tr>
<td>J</td>
<td>Software producer</td>
<td>large international</td>
<td>Internal developed process</td>
<td>Release manager (1)</td>
</tr>
<tr>
<td>K</td>
<td>Software producer &amp; service provider</td>
<td>medium international</td>
<td>Agile, Scrum</td>
<td>Section manager (1)</td>
</tr>
<tr>
<td>L</td>
<td>Software/Hardware producer</td>
<td>large international</td>
<td>Agile/Traditional</td>
<td>Section manager (1)</td>
</tr>
</tbody>
</table>
interviewee. Furthermore, an anonymous code was assigned to each interview transcript, in accordance with the regulations of the Norwegian Social Science Data Services (NSD).

Data Analysis

After all the interviews were transcribed, we proceeded with the data analysis stage. We first identified the segments of text relevant for the research questions and discarded the irrelevant passages. Then we proceeded with the Coding of the transcripts. Based on the research and interview questions a set of codes was produced. This set was then iteratively updated after the transcripts were revisited several times. The coding process was performed in several rounds and the results were reviewed and discussed with two other researchers. Each segment or sub-segment was labeled by means of one or more easily recognizable terms or categories, using a software tool designed for qualitative analysis (NVivo 10).

A couple of interview transcripts were selected and coded by all researchers. The differences in coding were then discussed, and the final set of codes was applied to all the transcripts.

During the Abstraction and grouping stage the collected data were organized into statements considered relevant to the goals and research questions. The process resulted in a manageable and indexed set of data that could more easily be navigated and analyzed, as done by Pettersson et al. [55] and Höst et al. [56]. Each statement received a unique identifier, title, and description, and the statement’s relationship to other statements was also abstracted by means of nodes.

A node is a collection of references to a specific theme, place, person or other area of interest. The references were obtained by 'coding' sources such as interviews. The nodes were organized hierarchically and aggregated into more general groups of nodes based on the research questions. Since the process of analysis tends to be a cumulative rather than a one-stage process, the meaning and structure of nodes change over time.

Interpretation of the collected data implies identifying the parts of the data relevant to a specific research question. The coded segments allowed the researchers to identify statements relevant to the research questions. Analysis of the graphical representation of the abstracted statements and the connections between them then facilitated the interpretation of the collected data, identifying nodes and groups of nodes related to the research questions.

Subsequently, the identified groups of nodes were aggregated into more general sets of groups for all the interviews taken together. This aggregation was done by Deak, and reviewed by Stålhane until consensus was reached. Some nodes were unassigned in the first round. These were reviewed by both researchers to reach consensus on one of three possible actions: (i) map the node to an existing group, (ii) create a new group for the node (no existing group appropriate) or (iii) consider the node out of scope for the research questions.
4 Results

This section presents the concepts for negative and positive factors affecting testers’ motivation and job satisfaction, as well as the motivational strategies applied by the company based on these factors.

4.1 Motivational and de-motivational factors

In Table 2 we list issues sorted by the number of respondents (rightmost column) who mentioned this issue during the interview. The table only includes concepts mentioned by at least five different interviewees, while those mentioned by fewer are omitted.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of influence and recognition</td>
<td>late involvement in the project, testing is underestimated in the company, afraid of opening defects, no control over the schedule</td>
</tr>
<tr>
<td>Unhappy with management</td>
<td>insufficient resources, unrelated tasks</td>
</tr>
<tr>
<td>Technical issues</td>
<td>versioning, insufficient number of test environments, poor quality, integration issues with simulators</td>
</tr>
<tr>
<td>Lack of organization</td>
<td>lack of clear processes, tasks, redundant meetings</td>
</tr>
<tr>
<td>Time pressure</td>
<td>squeeze, long days, short periods, overloaded schedule</td>
</tr>
<tr>
<td>Boredom</td>
<td>routine, repetitive tasks, unchallenging work</td>
</tr>
<tr>
<td>Poor relationships with developers</td>
<td>bugs related friction, stereotypic view of testing, slow defect fix rate, late changes to the code</td>
</tr>
<tr>
<td>Working environment issues</td>
<td>colleagues with no social antenna, open plan landscape related issues</td>
</tr>
</tbody>
</table>

Concept - Lack of influence and recognition

The concept stated by most respondents as a factor with negative impact was lack of influence and recognition. Under this concept we gathered the segments referring to the irregular working flow, and lack of control over an unstable schedule. Also frequently cited by respondents are testers’ late involvement in the development cycle, and the struggle for recognition. "When I as a tester or test manager enter a project too late in the process, is too late to make a reasonable contribution to the quality with the testing." (Tester, Company C). When the focus of testing activities is more on testing issues, like retesting defects, rather than testing the product or requirement, testers tend to miss a sense of accomplishment, rather feeling frustrated by not performing their real job. Under the same concept we aggregated the worries for a low likelihood of promotion, in comparison with other roles, such as the ones for developers. "The developing projects and the daily operations have to realize how important software testing is. The testing area has to be lifted up as an important part of the company’s work." (Testing Manager, Company C)
Opening a critical defect (i.e., registering a failed test as such in the test log, thus making it explicit as an issue to be dealt with) might delay the release of the product, especially if it is close to the deadline. Some respondents had experienced low influence for testers in such situations, reported defects talked down as uncritical by developers, so that the deadline could be met. Sometimes, testers had felt very unpopular when opening defects close to release time, thus becoming gradually more reluctant to do so, even if they felt it was a serious defect which should really be handled: "testers use a lot of time, they are afraid of opening defects". (Testing Manager, Company D).

**Concept - Unhappy with management**

The second most mentioned concept concerns the lack of support from management with regard to testing activities, such as unrealistic schedules and a scarcity of resources.

Due to the lack of resources allocated to testing, managers allocated testing tasks to developers. This frustrated the developers required to fulfill many roles in parallel, sometimes with conflicting responsibilities or assigned without warning or advance planning. "Since there are more developers than testers, from time to time, developers have to change their hat to meet the deadlines" (Developer, Company E).

Both testers and developers complained about lack of planning by managers resulting in unpredictable schedules and lost time: "the lack of planning, that we run a lot of tests, but we have no idea why we run and it might turn out that nobody really cares about this job" (Tester, Company F).

Some testers considered the time allocated to testing to be insufficient right from the planning phase, with no buffer periods allocated for unpredicted delays. As mentioned above, testers had felt low influence and loss of popularity when opening critical defects. They also reported dissatisfaction with the way some managers handled such situations. "I found bugs which stopped or held a release, which on one hand is a good thing, because if the bug will have go into production it will have created serious problems, but is also a little bit like putting your reputation in line. The release is stopped because of you." (Tester, Company A). If the tester were to raise a defect later shown to be invalid, there was some feeling of unfairness, i.e., such a mistake might be much more detrimental for a tester’s reputation than a design or coding mistake would be for a developer’s reputation. Some testers had felt pressure from managers to open fewer defects in the future.

**Concept - Technical issues**

**Technical issues** within the testing context covers problems with testing tools, development environments or a weak infrastructure. An insufficient number of test environments, poor quality or insufficient fidelity to the actual system being tested, together with integration with 3rd party tools or simulators were mentioned as technical obstacles. "It takes a lot of time to get the tests started, not everything works correctly, setting up an environment and also installing the software on our test servers." (Tester, Company B). In some companies, the participants complained about the weak infrastructure as the root cause of many false defects that wasted time and effort in investigations. "My main frustration is that we don’t have good enough tools..."
to do our work, and we have to use tools that make our work a lot more difficult than it should be.” (Developer, Company F)

**Concept – Lack of organization**
This concept refers to the inconstancy of testing organization and responsibilities. The interviewees were troubled with the continuously changing plans, change of resources or process and the volatile test planning. Participants considered the lack of organization or poor planning as a strong source for the recurrent time pressure problem. A high number of time-consuming meetings, often considered redundant or irrelevant for their work tasks, were also mentioned as a negative factor. In addition, some of the participants had an increasing number of assignments unrelated with testing or outside their focus area. “We fill a lot of time until we don’t have any left space [in the project schedule], but often we want to update the plan.” (Tester, Company F).

**Concept - Time pressure**
Another frequent concept was the time pressure associated with test execution. Traditional working teams often delay testing until the end of projects, squeezing the calendar time available. Unfortunately, projects often fall behind schedule, so the testing teams need to compress and sacrifice activities due to their shrinking time-frame. "I've been in this business for many years and testing is at the end of this lifecycle, and always pressed to so short periods, long days, and shortcuts. It's always like that.” (Testing Manager, Company D). Sacrificing time which was initially assigned for testing in order to recover delays in earlier tasks, mostly development related, often forces a compromise on the quality of the delivered product. "I don't like that we are the last link in the chain, and we don't always get the time that was promised in the beginning. Give us more time to finish our testing and do it properly.” (Tester, Company D).

The concept of time pressure also appears in the interviews with testers from agile teams, where the testing is occasionally facing similar time pressure. The companies have sprints with unbreakable deadlines, but since the first half is allocated to test case designs, issues are often discovered late in the sprint. This situation gives little time to fix the issues. "Sometimes it's difficult to plan because they [the developers] don't really know when they are ready. They want testing done immediately as they are ready, but they themselves don't really know when they are ready." (Tester, Company A)

**Concept - Boredom**
Some of the participants mentioned the routine of some testing activities and the feeling of boredom associated with maintenance testing. "Everything is routine, there are no surprises after the system is in production" (Testing Manager, Company D)

**Concept - Poor relationships with developers**
Another mentioned concept is the poor relationship between testers and developers, which can be problematic at times. Most of these frictions result from discussions related to bugs. "I do remember having discussions about bugs: Is it really a bug, or is it [the bug fix] really important enough to be included in the release.”
Another factor quoted by many participants was the stereotypic view of testing by the developers, "the classical view that they are developing and finally we are testing and then it's coming back with us saying <<that is not good, that is not good>>" (Testing Manager, Company A).

Two testers from different companies described their co-workers' view of testing as "a necessary evil". The slow defect fix rate and developers making unannounced late changes to code were also mentioned as sources of conflict between developers and testers. "It's a lot of things, challenges that take time, sometimes it can take time to get environments, sometimes you raise bugs and they don't take them quickly enough" (Testing Manager, Company D)

**Concept - Working environment issues**

Several participants complained about working in open space landscapes, noisy due to the nature of the office design and sometimes with colleagues with no social antenna. “When it comes to office conditions it can be quite noisy in this open landscape thing.” (Tester, Company F)

4.2 Concepts for positive factors

The relationship between codes and concepts for positive factors derived from the study are presented in Table 3. The positive factors are presented in descending order starting from the one mentioned by most interviewees.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Codes linked to concept</th>
<th>Positive factors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enjoy challenges</strong></td>
<td>Enjoy challenging yourself, every day you never know what's coming up, like the chaos, need challenges</td>
<td>11</td>
</tr>
<tr>
<td><strong>Focus on improving the quality</strong></td>
<td>finding bugs, to investigate, making things better, personal goal on improving the quality</td>
<td>11</td>
</tr>
<tr>
<td><strong>Variety of work</strong></td>
<td>work variation, combine testing and programming,</td>
<td>11</td>
</tr>
<tr>
<td><strong>Recognition</strong></td>
<td>ensure that testing tasks are important in the company, send testers to courses and conferences, get the support I need to do a good job</td>
<td>9</td>
</tr>
<tr>
<td><strong>Good management</strong></td>
<td>good communication in the team, with developers, enough resources</td>
<td>7</td>
</tr>
<tr>
<td><strong>Technically challenging work</strong></td>
<td>technically challenging work</td>
<td>6</td>
</tr>
</tbody>
</table>

**Concept - Enjoy challenges**

Most of the interviewed participants enjoyed challenges represented by the testing activities, challenging themselves or simply thriving on the chaos sometimes accompanying the daily activities of a tester. "When I perform my test and it works, I'm thinking: Am I doing something wrong? Is the test doing what it's supposed to? When it fails, I'm also thinking: Is it really doing things correctly?" (Tester, Company A, in an enthusiastic voice) and "There is always something new, new challenges towards different test scenarios." (Tester, Company D)
Concept - Focus on improving the quality
The second most occurring concept related to testers’ passion for improving the quality of the software, the pleasure of investigating and finding defects whose removal will lead to a better product. "I do have a passion for improving the quality and finding defects. And there I have learned that I have different focus than the developers, maybe the right focus for testing. I'm happy when I find bugs. Of course, I'm also happy when things are working." (Tester, Company A)

Concept - Variety of work
On several occasions the concept of variety of work was mentioned, referring to being included in the testing activities associated with the whole development cycle, not just a specific phase. Another contribution to the variety was to have a combination of programming and testing tasks as part of job responsibilities. "The biggest factor for me is that you do different things, it’s very varied and you get to see the whole picture. You can participate from the start of a project to the end doing various things, that’s the biggest thing for me." (Tester, Company B)

Concept – Recognition
The concept of recognition included the company’s awareness of the importance of testing, both among management and development teams, as well as positive feedback received from developers in relation to discovering and fixing bugs. “When we heard feedback from engineers, when we hear they say <<thank you, this test helped us to fix something that is wrong>>” (Tester, Company F). In the same category we included expressions of the pride participants experience by working in a company known for delivering high-end products. “I believe I work in a company that is delivering high end embedded software for the worldwide. I want to make sure that the software we deliver has high quality.” (Testing Manager, Company E)

Concept - Good management
Under the concept Good management we aggregated all the positive references to relations and communication with the managers, between the testers and with developers. "I think is important to be on good terms with the developers; if they are having some Agile approach, you as a tester or test manager will get invited to their daily Scrum, so you get a feel for the modules they are struggling with and so on. It can help you prioritize, when you start to test." (Testing Manager, Company D)

Concept - Technically challenging work
Another positive concept, Technically challenging work, was associated with the participants’ need to have allocated tasks reflecting their technical competencies. “The most interesting thing that you can have is interesting technology to work with.” (Developer, Company D).

Testers enjoy having variety in their work, the lack of influence and recognition is a major negative factor for most of the participants involved in this study. All these concepts are specific to the nature of testing activities with Technical issues within the testing context, involving large quantities of effort and time invested in items which should be readily available at the beginning of testing. The Time pressure concept is
referring to the tendency of testing time to shrink from the original estimate until the actual testing execution period takes place.

4.3 Motivational strategies for testing personnel

In this section we will present the strategies adopted by the companies for motivating their employees and stimulating their productivity. Several combinations or rotations of responsibilities and products are presented in the Combined responsibilities section. Under the Open office area segment we present the pros and cons of adopting an open office landscape. The Singular strategies portion presents policies which we found implemented in one or more companies, while the policies presented in the first two sections were applied in a generous percentage of the participant companies.

Combined responsibilities

The term combined responsibilities refers to the situation where a person, often a self-declared developer at heart, is in charge of several roles, one of which is the tester. This strategy is used as a method for personnel keen on development work, desiring a type of position which will imply more than just testing. Another reason provided for applying this strategy is to offer more variety for testing personnel, hence avoiding the sense of monotony and routine associated with testing activities, especially in the case of regression testing. The diversification of the roles had been proposed after discussions with the employees and established with common accord between testers, developers and managers.

One challenge which might arise for employees having "several hats" is the difference of perspective between a tester and a developer. This was mentioned by several participants. The testers tend to have a customer view when looking at the scenarios they are testing and are interested in how these scenarios integrate with the overall business concepts. On the other hand, developers tend to concentrate on building the product from a technical design perspective, or to the provided requirements, with less concern for the overall business concept [57], [58]. "For me personally it's a bit difficult to change hats. It is too easy for the testers to get involved with development, and that can be both good and bad... If you write system tests, it might be that you are developer involved in actually developing a feature and if you are also involved in writing tests for that feature you might not test enough or you might test things that...You may use the knowledge about code... maybe you don't test the code enough. It's easy to hang on the knowledge that you have and you do not see the defects that otherwise you might detect." (Developer, Company E)

In this category, we can also include situations where a group of developers are assigned for testing each other's code. The managers are aware of the possible lack of objectivity arising from testing one's own code, so this rotation is a risk reduction measure and fulfilling quality related requirements.
Open office area

Having an open landscape or collocating the testers in the same area or room was reported as an approach to encourage communication and build a sense of unity. The open office space is considered one of the most productive working environments for software development [59], and it is supposed to increase the amount of informal communication inside the team, stimulate discussion and improve the problem solving pace [59], [60].

However, there were also complaints about this type of landscape with several participants describing work in open landscapes as noisy, which concords with the results reported in [59] where the developers perceived open office spaces as distracting, especially when people doing programming were in the vicinity of other developers holding a meeting at the same time.

In one company, there were strong complaints about cases where collocated colleagues had no social antenna. "One of them has a cold and sneezes and coughs and infects the other three. And it's happened during a release time. He can bring down the whole team" (Testing Manager, Company F)

Another reason given for using an open landscape was that collocating testers and developers might lead to better communication and relationships between these two groups, who are prone to having a difficult relationship [61]. The tester-developer relationships can easily deteriorate due to defects related friction and discussion, late and un-reported changes to the code or due to the stereotypic view of testing by some developers. Communication among the stakeholders in a software development project is often of informal nature, mostly technology-based with daily meetings and discussions taking place over the phone, chat and video conference [62], [63]. Face to face communication is still considered the most optimal way for building trust among people belonging to the same group and implicitly ensure a higher productivity in comparison with distributed teams [62]. "In our team we communicate very well and a lot; mainly because we sit in the same room." (Tester, Company H)

Singular strategies

In this section we will present a collection of concrete changes which were applied in a single instance, in different companies. The trigger for these changes was a new management or technology which led to the implementation of concrete and effective changes for improving testing in those companies.

"(the changes started) with the change of project lead. A couple of new project managers came into the project couple of years ago "(Testing Manager, Company I)

One preconception related to testing is the view that it is an activity that starts only after the coding phase is complete, and that it has a limited purpose, mainly to detect failures. We found managers ready to challenge this prejudice by making substantial effort to have testing members involved from the software requirements process. Since time pressure in testing was a recurring issue in the testing activities, a buffer period was allocated for testing from the planning stage to ensure that delays in the development schedule would not reduce the focus on testing or give too little time for testing. "Also we have planned the testing period with a slacking period. If they take
from the time allocated for testing is not a crisis. Generally we have improved in that field.” (Testing Manager, Company I). The companies who wanted to improve testing decided they need to have people who are motivated by that type of work. In order to motivate, they decided to make a priority of communicating the importance of quality and to emphasize that testing is one of the ways to achieve it. Their effort focused on ensuring that testing was pervasive throughout the entire development and maintenance life cycle and that the tester personnel were included in the daily communication, and in the feedback from the customer. "I think we solved the motivation factor by having testers as part of the core team that delivers ... the tester receives the feedback from the customer now, bad or good." (Section Manager, Company K). "We have activities for testing on the sprint. When a module of testing is finished we don't put it to Done we put it To test. Testing is part of our daily tasks.” (Section Manager, Company H)

4.4 Contradictions between goals and implementations

Under the Contradictions between goals and implementations part we present the discrepancies found between the expressed goals and the concrete implementation and organization of testing activities. All the participant companies have strongly expressed an interest in ensuring and strengthening the quality of the product or service provided. They have developed internal rules and guidelines in order to ensure that the quality requirements are satisfied. However, the manner in which they choose to invest in the testing activities and implement these guidelines can lead to unsatisfactory results.

Not looking for testing skills as a mandatory or at least an essential requirement during the recruiting process might yield a team lacking the strength or motivation to perform efficient and thoughtful testing. "Usually we don't look for dedicated testers so testing is seen as an extra skill" (Section Manager, Company L)."Within R&D department we recruit new developers and if they are accustomed to testing or test procedure that will be a plus in the hiring process. It is more important that they fit in the organization and their personal skills rather than specific skills in testing.” (Section Manager, Company K).

Combined responsibilities were seen as a way to prevent a sense of monotony but it can also be perceived as a way of reducing testing costs by having a person fulfilling several roles. Having a 50% development task was considered a good way of keeping people assigned to test activities while providing them with development responsibilities. In case a person is mainly interested in development responsibilities he might execute his testing assignments without much enthusiasm or desire to be proactive and improve the testing process. This strategy can backfire on the quality of the product, when most of the people involved in the testing process are involved in it just as an additional part of the job or a temporary assignment.

When interviewing managers and testers from the same company, we observed discrepancies about the resources or effort required to run testing activities. Managers often mentioned that there was not enough testing work to justify hiring more fulltime
testers, while the testers and developers expressed a need for more testing resources to be assigned to their team or project.

Outsourcing of testing, entirely or partially, typically to lower cost countries, proved to be an established practice. The managers involved in this type of process mentioned problems with the quality of the service provided or the necessity of setting aside additional time and trips for improving the process or facilitating knowledge transfer and management.

5 Discussion

5.1 Motivational and de-motivational factors of software testing

Our first research question was about motivational and de-motivational factors for software testing personnel. In many ways, the study has confirmed the findings of previous studies (as well as the general image) that software testers struggle with many de-motivational factors. It has also found positive factors similar to those identified by Shah, Harrold and Sinha [14], with testers being enthusiastic about their jobs and the interesting challenges they are presented with. Reiterating the background theory, many of the negative factors found can be considered hygiene factors, for instance time pressure, low recognition, poor management, or poor relationships. For factors contributing positively to motivation, there are both extrinsic and intrinsic ones. For instance, recognition, good relationship with management and colleagues, and delivering high quality to the customer are examples of extrinsic motivation, while variation and challenge of the work tasks are examples of intrinsic motivation.

Of course, the number of factors, as well as the number of interviewees mentioning each factor, could have been different with another coding or other aggregation of interview phrases into factors. For instance, if making two factors “lack of influence” and “lack of recognition” instead of the one “lack of influence and recognition”, we would have had more factors in the table, but with a somewhat smaller number of interviewees per factor. However, influence and recognition were natural to group together with the given data material because both were often mentioned together. Although they are not synonymous, it is intuitively hard to imagine that a person would have high recognition in an organization yet low influence, or that a person would be able to maintain high influence over time in spite of low recognition. The exact borders between the various factors can also be a point of discussion. For instance, in connection with the problematic situation of discovering a critical defect near a release date, respondents both expressed a lack of influence and recognition (becoming unpopular) and dissatisfaction with the way managers handled such situations. However, there is a distinction in classification, as “lack of influence and recognition” would be the home for statements indicating that the tester does not have much say in plans, is not taken seriously enough, is not credited for making useful contributions to the projects. “Unhappy with management” would instead be the home for statements expressing dissatisfaction with managers’ attitudes, actions or
decisions. This again slightly different from “Lack of organization”, which would be a more general dissatisfaction with the ad hoc way of running projects in the company (or IT industry), not pertaining to decisions of particular managers. This again can of course be closely related to “time pressure”, but the latter applied to concrete testing tasks, typically on a daily or hourly basis, while lack of organization applied to the bigger picture of tasks, (re-)scheduling, and shifting roles.

Even taking into account the uncertainty of the borders between categories, it is evident from this study that a perceived lack of influence and recognition was a prominent negative factor brought up by the interviewees. Both testers and their direct managers saw this as an extensive problem. The age-old saying that nobody loves the bearer of bad news is perhaps the perfect illustration of the toll of testers in this respect: It is hard to become popular if one of your key tasks is to find and report defects in products that your colleagues have developed. During the interviews with managers, these signaled several problems with the testing position, indicating that not only individual testers but even whole testing teams sometimes face a struggle for recognition as a valuable part of the company. When repeated recommendations or requests related to their working activities are ignored (for instance related to scheduling and too short time for testing), this naturally leads to a feeling of lacking influence and recognition. We would like to underline this result since it corroborates with previous research [30], which has shown that primary motivators for the IT consultants are the variety in tasks and the opportunity to influence and/or manage an entire project.

One of the strongest motivational factors was the testers' focus on improving the quality of the software. There were several statements of pleasure of investigating and finding defects to help make a better product. By identifying themselves with the value of the activity, these testers are in concordance with the definition of autonomously motivated person based on SDT, [64]. Together with the satisfaction of basic psychological needs, autonomous work motivation represents the main variable mediating the relationship between job characteristics and the effort individuals put into their jobs, which are considered the behavioral manifestation of work motivation [65].

5.2 Motivational strategies for testing personnel

Our results included a section on the strategies adopted by the companies for motivating their employees and increasing their productivity. In several companies we encountered a combination or rotation of the employees’ responsibilities. Although providing variety of work should be seen as positive, it can also be an indicator of unwillingness to invest too much time in the testing, and of financial pressure. Hiring personnel who can switch several roles and fulfill different tasks is more cost efficient than having a dedicated person for each type of responsibility. From the employee perspective, a downside of this approach is the energy and time spent on changing the working context and hence the time restriction that must be applied for each task.

Although it has been proposed and studied for many years that one way of increasing the alignment between requirements and testing is to involve testers from the requirements phase [35], our findings suggests there is an established tendency to
perceive testing as an activity succeeding the completion of the coding with the exhaustive scope of detecting failures. This situation seems to emphasize the lack of influence and recognition of software testing and implicitly of testers, which was the most dominant negative factor mentioned in our interviews.

Another point that was touched during the interviews was the pros and cons of using open office landscapes. Although working in an open landscape does not apply exclusively to testers, the collocation of testers and developers in an open space was done with the purpose to improve communication and relationships between these two groups. Hence, we considered this issue to be of significant importance as a strategy directed at improving testers' motivation.

The results of this study provided us with some observations related to the software development methodology. Testers working in agile teams do not belong to a separate testing group, but work within the development team. They consider testing an ongoing process that happens throughout the development process, not just something that happens in a separate phase after development is done. Another point is that testing is done by the whole team, rather than just by testers, and the relationship between testers and non-testers tends to be collaborative rather than adversarial. Still, it was interesting to notice that more agile testers were unhappy about their relationship with the developers since testers get more respect on agile teams where they are seen as colleagues, and are involved much earlier in the process, making it easier to ensure a system is produced that's easy to test. The problem might be related to a situation where a company applies a customized version of agile methods “for good organizational reasons” [39]. Participants from both categories complained about the heavy load and unrealistic schedules which is in concordance with earlier research results [11].

5.3 Limitations and threats to validity

It is important to describe that human factors cannot be considered in isolation and on the individual level but are affected by sociological, organizational factors and other environmental factors [66]. The results of our study should be treated with some caution since there are other factors which may impact the motivation of a tester such as the organization structure, internal policies and processes. In addition, motivation can be influenced by human factors such as personality types [20], and individual characteristics such as age [67].

Construct validity

The construct validity is concerned with the relation between the theories behind the research and the outcome. In our research we used semi-structured interviews, including open-ended questions where the participants are encouraged to express their own opinions. The interview guideline was developed in several iterations, and based on previous literature [42] and by earlier work within the same group [51]. During the interviews the respondents were encouraged to express their opinions freely, by guaranteeing their anonymity and assuring them that the records will be accessible only to the researchers involved in this study, hence alleviating the potential problem of evaluation apprehension [68].
In order to mitigate the risk of identifying incorrect factors, presented by Robson [52] in this kind of research, we ensured observer triangulation by having the data analyzed by three researchers. In addition, the collected data and the results of this study were compared with our earlier quantitative study [69], which allowed us to apply both data and method triangulation.

**Conclusion validity**

Threats to conclusion validity arise from the ability to draw accurate conclusions. The interviews were conducted in different companies and in case there were a group of people to be interviewed all the interviews were performed in one session. Hence, we avoided that answers might be influenced by internal discussions. To ensure interviews of high quality, several pilot interviews were conducted at the study initiation, in order to avoid poor phrasing or structure for the interview guideline. A longitudinal study may provide further insights into the motivational and de-motivational factors of software testing personnel.

**Internal validity**

Internal validity threats are related to matters that may affect the causal relationship between treatment and outcome. Threats to internal validity include instrumentation, maturation, and selection threats. The potential problem of instrumentation threats was attenuated by developing a research instrument with close reference to the literature relating to quality requirements, influenced by a previously validated interview instrument and a previously piloted interview study. By collecting background information about the participants we were able to perform the interview session in approximately 60 minutes, which alleviates maturation threats. The threat of selection bias is always present when study subjects are not fully randomly sampled. One of limitation of our study is the small number of the participant companies and the selection criteria. In order to avoid the threats to internal validity we used data source triangulation by interviewing multiple roles at a company.

**External validity**

The external validity is concerned with the ability to generalize the results [70], which for this study translates with the applicability of the findings beyond the participant companies. Qualitative studies focus by nature in explaining and understanding the situation under study, rather than generalizing the results. The nature of qualitative design itself makes it impossible to replicate a study, due to the improbability to reproduce exactly the same context, but it can lead to a theory which can provide understanding for similar cases and situations. Since several of the participants from our study identified the same set of the discovered factors, it increases the possibility of transferring the results to other situations. To avoid the interaction of selection and treatment, interviewees were selected according to their roles within the company by an internal representative; the researchers did not select the subjects themselves. Moreover, the companies selected belonged to different geographical locations and different industry areas.

We are aware that the low number of participants is a limitation, and given the high number of variables playing an important role in the survey, the results of this study should be considered as preliminary. However, the focus was on depth instead of breadth, so we still think that the participants were a typical sample of Norwegian testing professionals, providing us with a lot of inputs and perspective. Generalizability can be difficult to achieve in software engineering due to the nature
of the context [71], however our qualitative analysis spanned across twelve companies using both traditional and agile methodologies, performing functional and non-functional testing, which could give better generalizability than performing interviews in just one company. However, the findings may be generalized to companies with similar characteristics as the participant companies by theoretical generalization [54].

6 Conclusion and further work

Our study aimed to cast light on how professional software testers can be motivated. Alongside, we explored the policies and rules conceptualized and implemented inside software development projects. The results of our study showed that systematic and sincere effort was invested in motivating the testers and establishing a respected place among the stakeholders involved in creating a software product or providing a software reliable service.

The findings of our work can be used as a basis for a broader questionnaire survey which could reach a high number of software testers in Norway or even internationally. It will be interesting to investigate whether companies have used any of the strategies for motivating testers or not, and whether testers in the companies that had used such strategies were then indeed better motivated than others. Another possibility will be to use action research, by trying to go into selected companies with a perceived need to improve testers’ job satisfaction, using some of the suggested strategies, and then evaluating whether they were successful.

Software testing is the main process for evaluating a tool or a technology [72], and despite the awareness of this process’ necessity and importance, more emphasis is laid on minimizing project costs and duration. Testing professionals are motivated by interesting, engaging and diverse work, by being treated with respect and regard and by building a high quality product that meets the user needs regardless of time and budget constraints. Our study shows that the perception of success and motivational factors varies between stakeholders; however, constructing a strong team requires members with diverse experience and complementary skills. The companies which chose to show reservation in recruiting experienced testing personnel might face a decrease of quality and implicitly of success, when facing a new or more complex product or technology.

References


71. Zimmermann, T., Nagappan, N., Gall, H., Giger, E., Murphy, B.: Cross-project defect prediction. Proceedings of the 7th joint meeting of the European software engineering


Appendix A

Interview guideline for testing professionals

<table>
<thead>
<tr>
<th>1. Intro</th>
</tr>
</thead>
</table>
| 1.1 Can you tell me a little about yourself and your career?  
How did you become involved with testing? |
| 1.2 What are you working on at the present? |
| 1.3 Can you tell me something about your role in the current project? |

<table>
<thead>
<tr>
<th>2. Testing organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 What is the software development process you are using?</td>
</tr>
<tr>
<td>2.2 How is testing organized in your company?</td>
</tr>
<tr>
<td>2.3 How are the testers integrated in the development team?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Technical issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Are you satisfied with the development and testing environments that you have?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 How would you describe your communication with managers and fellow testers and developers?</td>
</tr>
</tbody>
</table>
| 4.2 Have you observed any changes in the team dynamics during high pressure times?  
a) Have you faced a situation where you find yourself in a disagreement over a defect, over the validity of the defects, severity or the priority? |

<table>
<thead>
<tr>
<th>5. Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 On your recent work: what did you enjoy about it?</td>
</tr>
<tr>
<td>5.2 What aspects of a project make you remain in software testing?</td>
</tr>
<tr>
<td>5.3 What are the motivating aspects of software testing?</td>
</tr>
<tr>
<td>5.4 Is there a part of being a tester that you prefer over other aspects?</td>
</tr>
<tr>
<td>5.5</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>5.6</td>
</tr>
<tr>
<td>5.7</td>
</tr>
</tbody>
</table>

6. **Testing career**

6.1 What do you think about testing as a long term career?

7. **Tester profile**

7.1 In your opinion, what are the most important characteristics for a software tester?

7.2 In your opinion, how would you recognize a high performing tester?

8. **Closure**

8.1 Do you have any recommendations for the future testers?