Information and Software Technology 73 (2016) 1–15

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

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

ARTICLE INFO

Article history:
Received 7 March 2015
Revised 5 January 2016
Accepted 6 January 2016
Available online 12 January 2016

Keywords:
Software development
Software testing
Motivation
Testers
Human factors
Management

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 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.

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 staff.

Conclusions: The study shows that combining testing responsibilities with development and ensuring a variety of engaging, 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.

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 labour, 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, while low motivation can lead to poor testing and overlooking of 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 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, rather than a solution for 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.

* Corresponding author. Tel.: +47 94123441.
E-mail addresses: leaf357@gmail.com, deak@idi.ntnu.no (A. Deak), stalhane@idi.ntnu.no (T. Stålhane), guttorm.sindre@idi.ntnu.no (G. Sindre).

[1] E.g., Britain: http://www.computerweekly.com/news/1280091893/UK-is-short-on-software-testing-skills, Australia: http://www.itwire.com/it-people-news/recruitment/45107-software-testing-skills-shortage, India: http://www.siliconindia.com/guestcontributor/guestarticle/354/Software_Testing_The_Next_Big_Employment_Wave_Pradeep_Chennavajhula.html

http://dx.doi.org/10.1016/j.infsof.2016.01.002
0950-5849/

The research on human factors in software testing is 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 has been reviewing. 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 the 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 to understand its negative issues, which must somehow be reduced, and its positive sides, which should be kept or strengthened during the 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, already received some attention. Testing education is addressed in ([15–18]) 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 the theory of work-related motivation which, unfortunately, is 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 have a broader look 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 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 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]. Csíkszentmihályi'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 influence 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 into 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], where processes were designed to answer a single, specific question, 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 appreciating of the 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 analysing 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 to 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 previously 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 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 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. This 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

Summing up what to distil from this section, a 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., [45–47], 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 strategies deployed by companies 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 easily collected from a broad range of respondents, and statistical analysis could discover 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 are possible, e.g., observation or interviews. Observation would be highly relevant if investigating industrial practice (e.g., what are employees doing, 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. 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

3.2.1. Planning and selection

We constructed a set of interview questions to cover both research questions, using a strategy with relatively few questions. 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 had questions related to the testers' 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 interview guide was constructed by the first author with inputs from the second author, and 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 guideline was organized into sections and subsections as shown in Appendix A. We used four sections: Introduction (1), Company strategies – RQ2 (2), Motivational factors – RQ1 (3) and a Closure (4). Since a company's organization influences how the employees experience their working situation, we needed to understand the organization of each company before we had a closer look at the motivation.

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, to make them aware of the nature of the interviews, their rights to withdraw at any stage, and to ask permission for recording.

1. *Introduction* – a general opening used to discover the interviewee's role in the current project.

   Section 1: The purpose of this section is to warm up the participant and understand his or hers pathway to a testing career and see if the involvement in testing activities was the result of several 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 answers collected here are used to help us to interpret the answers that come later in the interview. Thus, we ask about how the person got involved with testing, what he or she is working on at the present and the interviewee's role in the project he is currently working on. The information retrieved at this stage is revisited in part 3.

2. *Company strategies* – development process, testing organization and how these two activities are integrated. The questions included in this section help us to understand the working culture and methodology applied in the companies. This understanding was a prerequisite for RQ2.

   Section 2: Here we want to understand how testing and development are organized: development process, testing organization and how the two are cooperating (2.1–2.3).
   Section 3: Development process (technical issues). How happy are you with the organizations described above (literal a). Since Technical issues within the testing context was a recurring theme in our work we wanted to assess how well

this issue is handled in the company before going further into the communication part (3.1).

Section 4: Communication has a prominent role in a tester's responsibilities in activities such as discussing defects with developers and with managers. We decided to explore this part of the tester's work in the Communication section (Questions 4.1, 4.2). The quality of the communication can affect the tester's motivation and the sense of acknowledgment within the company, hence assessing this quality was important for RQ1. We want the general impression of communication in the company and whether this changes during stressful periods.

3. *Motivation* – the interviewee's own experience, plans for the future and what are the important abilities to possess for a tester. The questions in the *Motivation* section are designed to elicit positive and negative aspects of testing work. 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 wanted to see what the companies were looking for in a tester when hiring for a testing position. This information was compared the information was compared with the employee's perception – question 5.7.

Section 5: Reasons for interviewee's own motivation (5.1–5.7).

Section 6: Testing career (6.1). Are you considering a long term testing career?

Section 7: Important tester abilities (7.1, 7.2). What are the most important tester characteristics and how would you recognize a good one?

4. *Closure* – the interviewee's recommendations for future testers (8.1). The information retrieved in the introduction was revisited in the Closure sections where we asked the participants to express their views on a testing career.

When interviewing managers we added some questions in the Testing organization and Tester profile sections. These questions were related to the background and overall responsibilities of the employees who performed testing activities (2.4–2.6). In addition, when discussing the tester profile, we enquired the managers of the criteria they used when hiring testers (7.3, 7.4).

The aim of this study was to cover the processes which were relevant to testing activities during the entire life cycle, 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. The inclusion of the managers was found necessary to uncover the motivational strategies followed by the companies (RQ2), since such strategies might often be discussed and applied by the managers rather by the testers themselves. Being directly responsible for testers, the managers interviewed were also expected to have valuable insights related to RQ1 on motivational factors influencing testers «in their daily activities». I.e. it should be noted that all the managers included were in contact with testers on a daily basis, not higher/more remote managers with lesser operational involvement in the testing activities.

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. The 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 to be an efficient way of getting new participants or companies involved. These suggestions were 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 in the relevant populations 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 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 participating companies and interviewees is shown in Table 1.

3.2.2. 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 included additional questions concerning the strategies and the rules proposed or implemented for stimulating their employees.

The duration of the interviews varied from 30 to 90 min, 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 min time interval was pre-allocated to each interview, while for the managers one hour was planned from the beginning due to additional questions in the interview guide. During some of the interviews the participants expressed that they had additional time available, which allowed an extension of the discussion time up until 90 min.

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.

All interviews were recorded after the consent was given, and transcribed verbatim. The transcripts were sometimes sent to each

Table 1
Companies and interviewees.

Company	Business	Size	Methodology	**Interviewees**
A	Software producer & service provider	medium international	Agile, TDD	Testing manager(1) Tester (2) Developer (1)
B	Software producer & testing provider	Medium international	Agile, Scrum	Testing manager(1) Tester (2)
C	Software producer	Large national	Traditional	Section manager (1) Testing manager (1) Tester (2)
D	Software producer	Large international	Traditional	Section manager (1) Testing manager (2) Tester (3)
E	Software producer	Medium international	Agile	Testing manager (1) Tester (1) Developer (1)
F	Software/Hard-ware Producer	Large international	Agile/Traditional	Testing manager(1) Tester (3) Developer (2)
G	Software producer	Medium international	Agile/Traditional	Testing manager(1) Tester (1) Developer (2)
H	Software producer	Large international	Kanban	Testing manager(1) Tester (1) Section manager (1)
I	Software/Hard-ware producer	Large national	Agile/Traditional	Testing manager(1) Tester (1)
J	Software producer	Large international	Internal developed process	Release manager (1)
K	Software producer & service provider	Medium international	Agile, Scrum	Section manager (1)
L	Software/Hard-ware producer	Large international	Agile/Traditional	Section manager(1)

participant for validation with additional questions for clarifications. This process gave the participants the possibility to edit the transcripts or perform relevant updates. The final transcripts were used for the later data analysis.

The transcribed interviews were coded in several rounds. The transcripts were divided into sections of text consisting of a couple of sentences each, which allowed us to refer to specific parts of the interviews. All data were anonymized by removing names and details such as family issues, preference for certain job locations or any other type of data which might lead to the identification of the interviewee. Furthermore, an anonymous code was assigned to each interview transcript, in accordance with the regulations of the Norwegian Social Science Data Services (NSD).

3.2.3. 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 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 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 areas 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 changed 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 facilitated the interpretation of the collected data and the identification of the nodes and groups of nodes related to each research question.

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

Table 2
Relationships between codes and concepts for negative factors.

Concepts Negative factors	Codes linked to concept	
Lack of influence and recognition	Late involvement in the project, testing is underestimated in the company, afraid of opening defects, no control over the schedule	21
Unhappy with management	Insufficient resources, unrelated tasks	19
Technical issues	Versioning, insufficient number of test environments, poor quality, integration issues with simulators	16
Lack of organization	Lack of clear processes, tasks, redundant meetings	13
Time pressure	Squeeze, long days, short periods, overloaded schedule	13
Boredom	Routine, repetitive tasks, unchallenging work	12
Poor relationships with developers	Bugs related friction, stereotypic view of testing, slow defect fix rate, late changes to the code	7
Working environment issues	Colleagues with no social antenna, open plan landscape related issues	5

five different interviewees, while those mentioned by fewer are omitted.

4.1.1. 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).

4.1.2. 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, mangers allocated testing tasks to developers. This frustrated the developers required to fulfil 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.

4.1.3. Concept - technical issues

Technical issues within the testing context cover 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)

4.1.4. 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 volatile test planning. Participants considered the *lack of organization* or poor planning as an important source for their recurrent time pressure problems. 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 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, Company F).

4.1.5. 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, causing the testing teams to compress and sacrifice activities due to their shrinking time-frame. *"I've been in this business for many years and testing*

Table 3
Relationships between codes and concepts for positives factors.

Concepts Positive factors	Codes linked to concept	
Enjoy challenges	Enjoy challenging yourself, every day you never know what's coming up, like the chaos, need challenges	11
Focus on improving the quality	Finding bugs, to investigate, making things better, personal goal on improving the quality	11
Variety of work	Work variation, combine testing and programming,	11
Recognition	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	9
Good management	Good communication in the team, with developers, enough resources	7
Technically challenging work	Technically challenging work	6

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)

4.1.6. Concept - boredom

Some of the participants mentioned the routine of some of the 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).

4.1.7. Concept - poor relationships with developers

Another concept mentioned 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."* (Tester, Company A). 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 coworkers' 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).

4.1.8. Concept - working environment issues

Several participants complained about working in office 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.

4.2.1. Concept - enjoy challenges

Most of the interviewed participants *enjoyed the 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).

4.2.2. 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).

4.2.3. 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).

4.2.4. 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).

4.2.5. Concept - good management

Under the concept *Good management* we aggregated all the positive references to relations and communication with the managers, with other 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).

4.2.6. 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 greater percentage of the participant companies.

4.3.1. 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 will imply more than just testing. Another reason invoked 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 according 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 strategy, 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 help to fulfil quality related requirements.

4.3.2. 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 an 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 best way for building trust among people belonging to the same group and implicitly ensures 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).

4.3.3. 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 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 errors. We found mangers ready to challenge this idea 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 participating companies have expressed a strong 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 this 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. When 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 low 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 demotivational 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 demotivational factors. The study 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, and poor relationships. Table 4 shows how the motivational factors described in our study compare to the prior research.

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.

Our study revealed the *lack of influence and recognition* perceived by testing staff as the dominating negative factor, contrasting with Beecham et al. [42] work where the *Lack of influence* associated with *not involved in decision making* or *having no voice* came the last in the ranking. The importance of motivation and appreciation in ensuring testing of high-quality has already been acknowledged by Shah et al. [14], Villalba et al.[2], while our own investigations for the prospects of a testing career path among students [4] revealed a similar image of undesirable status for testing work, with most students perceiving a development position as the one which are more rewarding from a career and financial perspective.

The *Time pressure* factor, which has emerged as a new negative factor in our previous study, was reinforced during out interviews as an important factor which is constantly affecting the testing activities. The testers regularly face the challenge of finishing the assigned testing tasks on time by potentially compromising quality, or maintain the quality at the risk of maybe not finishing all the tasks, even in cases when working overtime is applied [64]. The testers considers their exposure to pressure higher than ones of the other teams involved in software development, mostly because they need to accommodate and adapt to delays in previous stages such as the design and development, which corroborates with previous work on factors influencing testing work performed by Shah et al. [14] and Villalba et al.[2]. We could establish a possible link with D9 *Unrealistic goals/phoney deadlines* factor from Beecham et al. [42] work (see Table 4), but since it is almost a preponderant challenge in the testing world we think it is justified to be nominated as challenge on its own. Rooksby et al. [65] reported time as a significant factor in determining the organization of testing-related activities, the most common among these factors was delay in code delivery from the developers. Villalba et al.[2], has emphasized the situation where the testing stage is located at the end of the project, and in consequence sufferers a shortened schedule due to delays in the development stage and the impossibility of postpone the delivery of the product.

The study by Franca et al. [43] shows that two of the motivators discovered in [45] were no longer present: appropriate working conditions and sufficient resources. However, the need for more *testing resources* was expressed the testers and developers we interviewed.

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, it was natural to group influence and recognition together with the given data material because both were often mentioned together. Although they are not synonymous, it is 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 borders between the various factors can also be a point of discussion. For instance, in connection with

Table 4
Motivational factors in literature.

Study	Purpose	Major outcomes in relation to motivational factors
Villalba et al. [2]	Human factors negatively affecting the practice of software testing	Testing is performed end of the project suffering shortened schedule due to delays of the previous development phases and the impossibility of postpone delivery to customer. When delays or finance problems appear, it is usual to shorten quality and testing effort. Instability of tester positions, Lack of attractiveness of testing, Poor career development for testers
Shah, Harrold and Sinha [14]	An empirical study of a vendor organization involved outsourced, offshored software testing.	The quality of testing is affected by testers' motivation and the importance of appreciating testers' efforts. Many of the participating testers were enthusiastic about their job Testers experience more pressure than the other teams because they must accommodate and adapt to delays in the design and development phases.
Beecham et al. [42]	SLR focusing on software engineers motivation	Motivators: M.1 Rewards and incentives, M.2 Development needs addressed, M.3 Variety of work, M.4 Career path, M.5 Empowerment, M.6 Good management, M.7 Sense of belonging, M.8 Work/life balance, M.9 Working in successful company, M.10 Employee participation, M.11 Feedback, M.12 Recognition, M.13 Equity, M.14 Trust/respect, M.15 Technically challenging work, M.16 Job security, M.17 Identify with the task, M.18 Autonomy, M.19 Appropriate working conditions, M.20 Making a contribution, M.21 Sufficient resources De-motivators: D.1 Risk, D.2 Stress, D.3 Inequity, D.4 Interesting work going to other parties, D.5 Unfair reward system, D.6 Lack of promotion opportunities/stagnation/boring work, D.7 Poor communication, D.8 Uncompetitive pay, D.9 Unrealistic goals/phoney deadlines, D.10 Bad relationship with users and colleagues, D.11 Poor working environment, D.12 Poor management, D.13 Producing poor quality software, D.14 Stereotyping/role ambiguity D.15 Lack of influence
Franca et al. [43]	SLR extending Beecham et al. [42] SLR	New motivators: M.1 team quality, M.2 creativity/innovation, M.3 fun, M.4 professionalism, M.5 having an ideology, M.6 non-financial benefits, M.7 penalty policies, M.6 good relationship with users/customer New de-motivator: task complexity.
Our previous work [51]	A comparison of motivational challenges for traditional and agile testers	New motivators: M.1 Enjoy challenges, M.2 Focus for improving quality New de-motivator: D.1 Time pressure, D.2 Technical issues
Our work	Motivational factors for software testers	Motivational factors: M1.Enjoy challenges, M2.Focus on improving the quality, M3.Variety of work, M4.Recognition, M5.Good management, M6.Technically challenging work De-motivational factors: D1.Lack of influence and recognition, D2.Unhappy with management, D3.Technical issues, D4.Lack of organization, D5.Time pressure, D6.Boredom, D7.Poor relationships with developers, D8.Working environment issues

the problematic situation of discovering a critical defect near a release date, respondents both expressed a lack of influence and a lack of 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 is 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 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 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 the 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, [66]. 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 [67].

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 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.

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 errors. 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 mentioned during the interviews was the pros and cons of using open office landscapes. Although working in an open office 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 notice that human factors cannot be considered in isolation and on the individual level but are affected by sociological, organizational factors and other environmental factors [68]. 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 [69].

5.3.1. 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]. However, the relationship between research questions, interview questions and the factors derived from the answers could have been clearer if the interview guide had been based on a thorough a priori theoretical instrument. Especially concerning the interviews with managers, it was convenient to use the same interview guide as for testers, just with some additional questions, but conceptually it might have been cleaner to use two clearly different interview guides for the two different respondent groups and thus keep the data more apart in the analysis.

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 [70].

In order to mitigate the risk of identifying incorrect factors – see 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 [71], which allowed us to apply both data and method triangulation.

5.3.2. 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 separate sessions. 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.

Related to challenge that there might have been a more detailed theoretical instrument as basis for the interview guide, as mentioned under Construct validity, there were observed some cases during data analysis where an interview response could be argued to fit into several factors. Hence, some caution is necessary when drawing conclusions about the relative importance of categories based on the number of interviewees who mentioned them.

5.3.3. 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 first collecting background information about the participants we were able to perform the interview session in approximately 60 min, which alleviates maturation threats. The threat of selection bias is always present when study subjects are not fully randomly sampled. One of the limitations 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.

5.3.4. External validity

The external validity is concerned with the ability to generalize the results [72], which for this study translates with the applicability of the findings beyond the participant companies. 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. Since several of the participants from our study identified the same set of 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 [73], 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 shed light on how professional software testers can be motivated. Alongside this, we explored the policies and rules conceptualized and implemented inside software development projects. The results of our study showed that a 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.

For industry, the findings from this paper might be a source of inspiration in several ways. Software companies that do not yet have any strategy for encouraging their testers could try out some of the strategies reported from our respondent companies, and those that already have something in place, could use findings about common motivational and de-motivational factors to have ideas for improving the situation. In any case, companies should let the employees participate in the discussions, as the importance of various motivational and de-motivational factors might vary over both companies and individuals. Even with such variation, a set of identified factors like those provided in this paper can give a valuable starting point for discussions in the company.

Software testing is the main process for evaluating a tool or a technology [74], 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.

Appendix A. Interview guideline for testing professionals.

1. Intro	
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?
2. Testing organization	
2.1	What is the software development process you are using?
2.2	How is testing organized in your company?
2.3	How are the testers integrated in the development team? Testing organization - managers section
2.4	Which are the employees who perform testing?
2.5	What other responsibilities do they have?
2.6	What are the backgrounds of their employee?
3. Technical issues	
3.1	Are you satisfied with the development and testing environments that you have?
4. Communication	
4.1	How would you describe your communication with managers and fellow testers and developers?
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?
5. Motivation	
5.1	On your recent work: what did you enjoy about it?
5.2	What aspects of a project make you remain in software testing?
5.3	What are the motivating aspects of software testing?
5.4	Is there a part of being a tester that you prefer over other aspects?

(*continued on next page*)

(*continued*)

5.5	Can you describe me a day or a project that you didn't enjoy so much?
5.6	What are the de-motivating aspects of software testing?
5.7	What your employer can do to improve your satisfaction in your role as a software tester?
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?
	Tester profile - managers section
7.3	What will be the mandatory criteria for hiring testes?
7.4	What will be a strong factor which would discourage you from hiring a tester candidate?
7. Closure	
8.1	Do you have any recommendations for the future testers?

References

[1] M.V. Mäntylä, J. Itkonen, J. Iivonen, Who tested my software? Testing as an organizationally cross-cutting activity, Softw. Qual. J. 20 (2011) 145–172.
[2] L. Fernández-sanz, M.T. Villalba, J.R. Hilera, R. Lacuesta, Factors with negative influence on software testing practice in spain: a survey, Softw. Process Improv. 42 (2009) 1–12.
[3] H. Shah, M.J. Harrold, Studying human and social aspects of testing in a service-based software company, in: Proceedings of the 2010 ICSE Workshop on CHASE '10, New York, New York, USA, ACM Press, 2010, pp. 102–108.
[4] A. Deak, T. Stålhane, D. Cruzes, Factors influencing the choice of a career in software testing among norwegian students, Software Engineering, ACTAPRESS, Calgary,AB,Canada, 2013.
[5] T.A. Majchrzak, Best practices for the organizational implementation of software testing, in: Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS), 2010, pp. 1–10.
[6] M. Grindal, J. Offutt, J. Mellin, On the testing maturity of software producing organizations, in: Proceedings of the Testing: Academic & Industrial Conference - Practice And Research Techniques, 2006, pp. 171–180.
[7] B. Hang, Software testing training in vocational technical education, Adv. Intell. Soft Comput. 108 (2011) 567–574.
[8] Z. Bin, Z. Shiming, Curriculum reform and practice of software testing, in: Proceedings of the International Conference on Education Technology and Information System (ICETIS 2013), Atlantis Press, 2013, pp. 841–844.
[9] V. Garousi, J. Zhi, A survey of software testing practices in Canada, J. Syst. Softw. 86 (2013) 1354–1376.
[10] A. Bertolino, I.A. Faedo, Software Testing Research: Achievements, Challenges, Dreams Software Testing Research: Achievements, Challenges, Dreams. (2007).
[11] B.W. Boehm, Software Engineering Economics, Prentice Hall, 1981.
[12] E. Whitworth, R. Biddle, Motivation and cohesion in agile teams, in: Proceedings of the 8th International Conference, XP 2007, Como, Italy, 2007, pp. 62–69.
[13] O. McHugh, K. Conboy, M. Lang, Using agile practices to influence motivation within it project teams, Scand. J. Inf. Syst. 23 (2011).
[14] H. Shah, M.J. Harrold, S. Sinha, Global software testing under deadline pressure: Vendor-side experiences, Inf. Softw. Technol. 56 (2014) 6–19.
[15] S.H. Edwards, Using software testing to move students from trial-and-error to reflection-in-action, in: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education - SIGCSE'04, New York, New York, USA, ACM Press, 2004, p. 26.
[16] S. Elbaum, S. Person, J. Dokulil, M. Jorde, Bug hunt: making early software testing lessons engaging and affordable, in: Proceedings of the 29th International Conference on Software Engineering (ICSE'07), IEEE, 2007, pp. 688–697.
[17] V. Garousi, A. Mathur, Current State of the Software Testing Education in North American Academia and Some Recommendations for the New Educators, pp. 1–8.
[18] J.C. Carver, N.A. Kraft, Evaluating the Testing Ability of Senior-level Computer Science Students 1 . Introduction and Background.
[19] L. Shoaib, A. Nadeem, A. Akbar, An empirical evaluation of the influence of human personality on exploratory software testing, in: Proceedings of the 2009 IEEE 13th International Multitopic Conference, IEEE, 2009, pp. 1–6.
[20] T. Kanij, R. Merkel, J. Grundy, An empirical study of the effects of personality on software testing, in: Proceedings of the 2013 26th International Conference on Software Engineering, Educa Train, 2013, pp. 239–248.
[21] R.M. Ryan, E.L. Deci, Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being, Am. Psychol. 55 (2000) 68–78.
[22] S.P. Robbins, Essentials of Organizational Behavior, Prentice-Hall, NJ, 1997.
[23] F.M. Toates, Motivational Systems, Cambrige University Press, Cambridge, 1986.
[24] E.L. Deci, R.M. Ryan, The "What" and "Why" of goal pursuits: human needs and the self-determination of behavior, Psychol. Inq. 11 (2000) 227–268.
[25] F. Herzberg, B. Mausner, B.B. Snyderman, The Motivation to Work, Rev. Française Sociol. 1 (1959) 180.
[26] F. Herzberg, One more time: how do you motivate employees, Harv. Bus. Rev. 46 (1968) 53–62.
[27] M. Csikszentmihalyi, Finding Flow: The Psychology of Engagement with Everyday Life, NY: BasicBooks, New York, New York, USA, 1997.
[28] J. de Jonge, H. Bosma, R. Peter, J. Siegrist, Job strain, effort-reward imbalance and employee well-being: a large-scale cross-sectional study, Soc. Sci. Med. 50 (2000) 1317–1327.
[29] L.D. Sargent, D.J. Terry, The moderating role of social support in Karasek's job strain model, Work Stress. 14 (2000) 245–261.
[30] L.G. Wallgren, J.J. Hanse, The motivation of information technology consultants: The struggle with social dimensions and identity, Hum. Factors Ergon. Manuf. Serv. Ind. 21 (2011) 555–570.
[31] R. Kalimo, M. Theo, Psychological and behavioural responses to stress at work. Psychosocial Factors at Work and Their Relation to Health. pp. 23–36 (1982).
[32] J.P. Connell, J.G. Wellborn, Competence, autonomy, and relatedness: A motivational analysis of self-system processes. In M. Gunnar & A. Sroufe (Eds.), Minnesota symposium on child psychology (Vol. 23, pp. 43-77). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
[33] J.A. Elliot, S.C. Dweck, Handbook of Competence and Motivation. Guilford Publications (2013).
[34] D.B. Smith, J. Shields, Administration in Social Work Factors Related to Social Service Workers' Job Satisfaction: Revisiting Herzberg's Motivation to Work Factors Related to Social Service Workers' Job Satisfaction: Revisiting Herzberg's Motivation to Work. 37–41.
[35] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, R. Feldt, Challenges and practices in aligning requirements with verification and validation: a case study of six companies, Empir. Softw. Eng. 19 (2013) 1809–1855.
[36] P. Runeson, C. Andersson, M. Höst, Test processes in software product evolution: a qualitative survey on the state of practice, J. Softw. Maint. Evol. Res. Pract. 15 (2003) 41–59.
[37] L. Briand, Y. Labiche, Empirical studies of software testing techniques, ACM SIGSOFT Softw. Eng. Notes, 29 (2004) 1.
[38] A. Beer, R. Ramler, The role of experience in software testing practice, in: Proceedings of the 2008 34th Euromicro Conference Software Engineering and Advanced Applications, IEEE, 2008, pp. 258–265.
[39] D. Martin, J. Rooksby, M. Rouncefield, I. Sommerville, "Good" organisational reasons for "bad" software testing: an ethnographic study of testing in a small software company, in: Proceedings of the 29th International Conference on Software Engineering (ICSE'07), IEEE, 2007, pp. 602–611.
[40] A. Causevic, D. Sundmark, S. Punnekkat, An industrial survey on contemporary aspects of software testing, in: Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation, IEEE, 2010, pp. 393–401.
[41] O. Taipale, K. Smolander, Improving software testing by observing practice, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ISESE '06, 262, 2006.
[42] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in software engineering: a systematic literature review, Inf. Softw. Technol. 50 (2008) 860–878.
[43] A.C.C. Franca, T.B. Gouveia, P.C.F. Santos, C.A. Santana, F.Q.B. da Silva, Motivation in software engineering: a systematic review update, in: Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011), IET, 2011, pp. 154–163.
[44] H. Sharp, N. Baddoo, S. Beecham, T. Hall, H. Robinson, Models of motivation in software engineering, Inf. Softw. Technol. 51 (2009) 219–233.
[45] T. Hall, N. Baddoo, S. Beecham, H. Robinson, H. Sharp, A systematic review of theory use in studies investigating the motivations of software engineers, ACM Trans. Softw. Eng. Methodol. 18 (2009) 1–29.
[46] P. Lenberg, R. Feldt, L.-G. Wallgren, Towards a behavioral software engineering, in: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014, New York, New York, USA, ACM Press, 2014, pp. 48–55.
[47] E. Whitworth, R. Biddle, The social nature of agile teams, AGILE 2007 (AGILE 2007), IEEE, 2007, pp. 26–36.
[48] S. Beecham, H. Sharp, N. Baddoo, T. Hall, H. Robinson, Does the XP environment meet the motivational needs of the software developer?, An Empirical Study. AGILE 2007 (AGILE 2007), IEEE, 2007, pp. 37–49.
[49] C. de O. Melo, C. Santana, F. Kon, Developers motivation in agile teams, in: Proceedings of the 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012, pp. 376–383.
[50] B. Tessem, F. Maurer, Job satisfaction and motivation in a large agile team, Agile Processes in Software Engineering and Extreme Programming, Springer, Berlin Heidelberg, 2007, pp. 54–61.
[51] Deak, A.: A comparative study of testers' motivation in traditional and agile software development. Product-Focused Software Process Improvement. pp. 1–16 (2014).
[52] C. Robson, Real World Research, 2nd. ed., Blackwell Publ, Malden, 2002.
[53] M.Q. Patton, Qualitative Evaluation and Research Methods (2nd ed.). SAGE Publications.
[54] M.D. Myers, M. Newman, The qualitative interview in IS research: Examining the craft, Inf. Organ. 17 (2007) 2–26.

[55] F. Pettersson, M. Ivarsson, T. Gorschek, P. Öhman, A practitioner's guide to light weight software process assessment and improvement planning, J. Syst. Softw. 81 (2008) 972–995.
[56] M. Höst, R. Feldt, F. Luders, Support for different roles in software engineering master's thesis projects, IEEE Trans. Educ. 53 (2010) 288–296.
[57] P. Kettunen, Bringing total quality in to software teams: a frame for higher performance, Lean Enterprise Software and Systems, Springer, Berlin Heidelberg, 2013, pp. 48–64.
[58] S. Alter, Defining information systems as work systems: implications for the IS field, Eur. J. Inf. Syst. 17 (2008) 448–469.
[59] J. Rasmusson, Introducing XP into greenfield projects: lessons learned, IEEE Softw. 20 (2003) 21–28.
[60] C. Mann, F. Maurer, A case study on the impact of scrum on overtime and customer satisfaction. Agil. Dev. Conf. 70–79.
[61] M. Lindvall, D. Muthig, A. Dagnino, C. Walling, M. D.K. Stupperich, Agile software development in large organizations, Computer (Long. Beach. Calif) 37 (2004) 26–34.
[62] K. Henttonen, K. Blomqvist, Managing distance in a global virtual team: the evolution of trust through technology-mediated relational communication, Strateg. Chang. 14 (2005) 107–119.
[63] D. Šmite, Global software development projects in one of the biggest companies in Latvia: is geographical distribution a problem? Softw. Process Improv. Pract. 11 (2006) 61–76.
[64] R.D. Austin, The effects of time pressure on quality in software development: an agency model, Inf. Syst. Res. 12 (2001) 195–207.
[65] J. Rooksby, M. Rouncefield, I. Sommerville, Testing in the wild: the social and organisational dimensions of real world practice, Comput. Support. Coop. Work. 18 (2009) 559–580.
[66] M. Gagne, E.L. Deci, Self-determination theory and work motivation, J. Organ. Behav. 26 (2005) 331–362.
[67] R. De Cooman, D. Stynen, A. Van Den Broeck, L. Sels, H. De Witte, How job characteristics relate to need satisfaction and autonomous motivation: implications for work effort, J. Appl. Soc. Psych. 43 (2013) 1342–1352.
[68] T.M. Amabile, Motivational synergy: Toward new conceptualizations of intrinsic and extrinsic motivation in the workplace, Hum. Resour. Manag. Rev. 3 (3) (1993) 185–201.
[69] N.P.G. Boumans, H.J. de Jong, S.M. Janssen, Age-differences in work motivation and job satisfaction. The influence of age on the relationships between work characteristics and workers' outcomes, Int. J. Aging Hum. Dev. 73 (2011) 331–350.
[70] C. Wohlin, P. Runeson, M. Hoest, C. Ohlson, B. Regnell, A. Wesslen, Experimentation in Software Engineering: An Introduction, Kluwer Academic, 2000.
[71] A. Deak, T. Stalhane, Organization of Testing Activities in Norwegian software companies, in: Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, IEEE, 2013, pp. 102–107.
[72] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2008) 131–164.
[73] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, B. Murphy, Cross-project defect prediction, in: Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the Acm Sigsoft Symposium on the Foundations of Software Engineering on European Software Engineering Conference and Foundations of Software Engineering Symposium - E, New York, New York, USA, ACM Press, 2009, pp. 91–100.
[74] L. Briand, J. Wüst, J. Daly, D. Porter, Exploring the relationships between design measures and software quality in object-oriented systems, J. Syst. Softw. 51 (2000) 245–273.