A Comparative Study of Testers' Motivation in Traditional and Agile Software Development

Anca Deak

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

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

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

1 Introduction

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

Although there is extensive work on motivation in IT personnel [6] and on motivation agile teams [7] and [8], to our knowledge there is a lack of research focusing specifically on motivation in software testing. A tester's position can be similar to a software engineer, but there is a particularity of the testing jobs which is

A. Jedlitschka et al. (Eds.): PROFES 2014, LNCS 8892, pp. 1–16, 2014.

prone to situations where discussions or diplomacy might be required. The results of a survey looking at the human factors which have a negative influence on real practice of software testing in software companies in Spain identified the following factors: instability of testers positions (48 %), lack of attractiveness of testing (48 %) and poor career development for testers (41,7 %), [9]. The study advises us to seriously take into consideration these factors due to the high percentage of respondents. The human and social aspects of working in testing, or inside a testing team, as well as the attitude towards the testing team in a company, were studied from the testers' perspective in a case study by Shah and Harrold [10]. These lead us to believe that it is worthwhile to make specific investigations about motivation among software testers. The research will be guided by the investigation of these factors, their relationships and effects on job outcomes.

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

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

2 Research Context

2.1 Traditional and Agile Testing

Software testing is a process through which the functionality of a software program is assessed during the development process. The investigation conducted for the assessment will focus both on verification and validation [12]. During the verification, the testers ensure that the software correctly implements a specific function and otherwise formulated, that it answers the question: are we building the product right? Validation ensures that the software has been built in order to satisfy customer's requirements. Validation answers the question: are we building the right product? The software testing process is also used to evaluate the nonfunctional quality of a system, by assessing aspects such as performance, security or usability.

For the traditional methodology, the software development lifecycle is constructed from a sequential set of stages: starting with a feasibility study at the top level and finishing with the implementation of the product. One trait of this methodology is the placing of testing towards the end of the project life cycle, which leads to defects being discovered close to the production deployment stage. In the requirement stage, the software tester can check if the requirements are according to the client's wishes, while during the design phase, the tester can verify if the design document covers all the requirements and review the design document from the architecture perspective. In the coding phase the testing team can execute test cases, as well as generating testing data. In the testing stage running the system test cases can verify whether the system operates according to the stated requirements. Once the product reaches the maintenance stage, the tester can retest new fixes and patches and afterwards use regression testing to ensure that the new changes do not impact functionality in an unintended manner.

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

2.2 Studies on Testing Practice

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

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

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

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

2.3 Existing Motivation Models in Software Engineering

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

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

3 Research Method

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

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

3.1 Survey Design

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

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

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

Table 1. Companies and interviewees

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 producer	large international	Agile/Traditional	Testing manager(1) Tester (3) Developer (2)

The duration of the interviews varied between 30 minutes and 90 minutes, and they were performed on the premises of each company, in quiet meeting rooms where each participant was interviewed individually. During the interviews the respondents were encouraged to express their opinions freely, by guaranteeing their anonymity and assuring them that the records will be accessible only to the researchers involved in this study. As recommended by Myers and Newman [24], we used a mirroring technique in questions and answers in order to encourage the respondents to share their stories. During the interviews we asked the participants to talk about both current events and to reflect retrospectively on previous scenarios. All interviews were recorded and transcribed, and the transcription was sent to each participant for final checking and approval. Notes were also taken with the prominent issues for each interview. The transcribed interviews were coded in several rounds. All data has been anonymised, which included changing names and removing unnecessary details.

Starting the process of analyzing the research data available, we first identified the segments of text relevant to the research question and discarded those having no relation to it. Afterwards, we proceeded with the coding phase and labeled each segment (or sub-segment) by means of one or more easily recognizable terms or categories, using a software tool designed for qualitative analysis (NVivo 10). The codes were analyzed and similar codes were aggregated into more general codes in order to reduce the number of codes utilized and retrieve the emerging categories. The transcripts were revisited several times, and the coding process was performed in repeated rounds and the results were reviewed and discussed with my senior colleagues. Each category and code can be linked to quotations from the interviews and these are used to strengthen and enhance the results. The categories were derived based on the results provided in the studies by Beecham et al. [4] and [5] as a model for constructing a list of motivators and de-motivators for software testers. Two tables, one combining the de-motivators, and another one combining the motivators from both studies and the ones emerging from our study, are available in the Appendix.

4 Results and Discussion

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

4.1 Concepts for Negative Factors

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

Table 2. Relationships between codes and concepts for negative factors

Concepts	Codes linked to concept
Negative factors	
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
Unhappy with management	insufficient resources, unrelated tasks
Technical issues (NEW)	versioning, insufficient number of test environments, poor quality, integration issues with simulators
Lack of organization	lack of clear processes, tasks, redundant meetings
Time pressure (NEW)	squeeze, long days, short periods, overloaded schedule
Boredom	routine, repetitive tasks, unchallenging work
Poor relationships with developers	bugs related friction, stereotypic view of testing, slow defect fix rate, late changes to the code
Working environment issues	colleagues with no social antenna, open plan landscape related issues

Concept - Lack of influence and recognition

The concept which appeared most often as a factor with negative impact was the **lack of influence and recognition.** Under this concept we gathered the segments referring to the irregular working flow, and lack of control over an unstable schedule. Testers' late involvements in the development cycle, together with the struggle for recognition are also frequently cited by the participants *"When I as a tester or test manager enter a project too late in the process to get a reasonable contribution to the quality with the testing."* (Tester, Company C). When the focus of testing activities is more on testing issues, like retesting defects, rather than testing the product or requirement, testers are not provided with a sense of accomplishment, but rather with a frustration of not performing their real job. Under the same **no sense of recognition** concept we aggregated the worries for an unattractive career path development, with a low likelihood of promotion, in comparison with other roles, such as the ones for developers. *"The developing projects and the daily operations have to realize how important software testing is. The testing area has to be lifted up as an important part of the company's work."* (Testing Manager, Company C)

Concept - Unhappy with management

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

Concept - Technical issues

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

Concept – Lack of organization

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

Concept - Time pressure

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

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

Concept - Boredom

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

Concept - Poor relationships with developers

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

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

Concept - Working environment issues
Several participants complained about working in open space landscape which is considered noisy due to the nature of the office design but also due to colleagues with no social antenna. *“When it comes to office conditions it can be quite noisy in this open landscape thing.”* (Tester, Company F)

4.2 Concepts for Positive Factors

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

Table 3. Relationships between codes and concepts for positives factors

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

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

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

Concept - Variety of work

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

Concept – Recognition

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

Concept - Good management

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

Concept - Technically challenging work

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

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

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

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

Positives factors group by methodology	Agile	Traditional	Total
Enjoy challenges	3	8	11
Focus for improving the quality	7	4	11
Variety of work	6	5	11
Recognition	4	5	9
Good management	2	5	7
Technically challenging work	3	3	6

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

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

Testers working in Agile do not belong to a separate testing group, but work within the development team. They consider testing an ongoing process that happens throughout the development process, not just something that happens in a separate phase after development is done. Another point is that testing is done by the whole team, rather than just by testers and the relationship between testers and non-testers tends to be collaborative rather than adversarial. It was interesting to notice that more agile testers were unhappy about their relationship with developers since testers get more respect on agile teams where they are seen as colleagues, and are involved much earlier in the process, making it easier to ensure a system is produced that's easy to test. It might be related to a situation where a company applies customized version of agile methods “for good organizational reasons” [18]. Participants form both categories complained about the heavy load and unrealistic schedules which is in concordance with earlier research results [6].

Both categories of testers face the time pressure issue and although the initial model proposed by Beecham includes stress as a strong de-motivational factor, we feel that time pressure is such a specific and persistent problem during testing activities that we can assign it a separate category. A complete list of motivating and de-motivating factors for software testers, including the ones proposed during this study are available in the Appendix.

Limitations and Threats to Validity

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

5 Conclusion and Further Work

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

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

References

1. DeMarco, T., Lister, T.: Peopleware: Productive Projects and Teams. Dorset House (1999)
2. Herzberg, F.: One More Time: How Do You Motivate Employees. Harv. Bus. Rev. 46, 53–62 (1968)
3. Mayo, E.: The social problems of an industrial civilization. Routledge & Kegan Paul, London (1949)
4. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in Software Engineering: A systematic literature review. Inf. Softw. Technol. 50, 860–878 (2008)
5. Franca, A.C.C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., da Silva, F.Q.B.: Motivation in software engineering: A systematic review update. In: 15th Annual Conf. on Evaluation & Assessment in Softw. Eng. (EASE 2011), pp. 154–163. IET (2011)
6. Boehm, B.W.: Software Engineering Economics. Prentice Hall (1981)
7. Whitworth, E., Biddle, R.: Motivation and Cohesion in Agile Teams. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 62–69. Springer, Heidelberg (2007)
8. McHugh, O., Conboy, K., Lang, M.: Using Agile Practices to Influence Motivation within IT Project Teams. Scand. J. Inf. Syst. 23 (2011)
9. Fernández-sanz, L., Villalba, M.T., Hilera, J.R., Lacuesta, R.: Factors with Negative Influence on Software Testing Practice in Spain: A Survey 2 Analysis of Testing Practices in Organizations, pp. 1–12
10. Shah, H., Harrold, M.J.: Studying human and social aspects of testing in a service-based software company. In: Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2010, pp. 102–108. ACM Press, New York (2010)
11. Deak, A., Stålhane, T., Cruzes, D.: Factors Influencing the Choice of a Career in Software Testing among Norwegian Students. In: Software Engineering, p. 796. ACTA Press, Calgary (2013)
12. Boehm, B.W.: Verifying and validating software requirements and design specifications. IEEE Softw. 1, 75–88 (1984)
13. Grindal, M., Offutt, J., Mellin, J.: On the Testing Maturity of Software Producing Organizations. In: Test. Acad. Ind. Conf. Pract. Res. Tech. (TAIC PART 2006), pp. 171–180 (2006)
14. Runeson, P., Andersson, C., Host, M.: Test processes in software product evolution: a qualitative survey on the state of practice. J. Softw. Maint. Evol. Res. Pract. 15, 41–59 (2003)
15. Briand, L., Labiche, Y.: Empirical studies of software testing techniques. ACM SIGSOFT Softw. Eng. Notes. 29, 1 (2004)
16. Beer, A., Ramler, R.: The Role of Experience in Software Testing Practice. In: 2008 34th Euromicro Conf. Softw. Eng. and Advanced Applications, pp. 258–265. IEEE (2008)
17. Turley, R.T., Bieman, J.M.: Competencies of exceptional and nonexceptional software engineers. J. Syst. Softw. 28, 19–38 (1995)
18. Martin, D., Rooksby, J., Rouncefield, M., Sommerville, I.: "Good" Organisational Reasons for "Bad" Software Testing: An Ethnographic Study of Testing in a Small Software Company. In: 29th Int. Conf. Softw. Eng. (ICSE 2007), pp. 602–611. IEEE (2007)
19. Causevic, A., Sundmark, D., Punnekkat, S.: An Industrial Survey on Contemporary Aspects of Software Testing. In: 2010 Third International Conference on Software Testing, Verification and Validation, pp. 393–401. IEEE (2010)

20. Shah, H., Harrold, M.J., Sinha, S.: Global software testing under deadline pressure: Vendor-side experiences. Inf. Softw. Technol. 56, 6–19 (2014)
21. Taipale, O., Smolander, K.: Improving software testing by observing practice. In: Proc. 2006 ACM/IEEE Int. Symp. Int. Symp. Empir. Softw. Eng., ISESE 2006, p. 262 (2006)
22. Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H.: Models of motivation in software engineering. Inf. Softw. Technol. 51, 219–233 (2009)
23. De, O., Melo, C., Santana, C., Kon, F.: Developers Motivation in Agile Teams. In: 2012 38th Euromicro Conf. Softw. Eng. Adv. Appl., pp. 376–383 (2012)
24. Myers, M.D., Newman, M.: The qualitative interview in IS research: Examining the craft. Inf. Organ. 17, 2–26 (2007)
25. Kanij, T., Merkel, R., Grundy, J.: An empirical study of the effects of personality on software testing. In: 2013 26th Int. Conf. Softw. Eng. Educ. Train., pp. 239–248 (2013)
26. Boumans, N.P.G., de Jong, H.J., Janssen, S.M.: 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, 331–350 (2011)
27. Robson, C.: Real World Research, 2nd edn. Blackwell Publ., Malden (2002)
28. Deak, A., Stalhane, T.: Organization of Testing Activities in Norwegian Software Companies. In: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, pp. 102–107. IEEE (2013)

Appendix

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

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

Table 7. List of motivating factors

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