The Software Tester:
An Exploration of the Skills and Practice of the Role

Raluca Madalina Florea

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Sciences
University of Oslo

August 2022
© Raluca Madalina Florea, 2023

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2601

ISSN 1501-7710

All rights reserved. No part of this publication may be
reproduced or transmitted, in any form or by any means, without permission.

Cover: UiO.
Print production: Graphics Center, University of Oslo.
This thesis is dedicated to my family and all the software testers out there
Abstract

Given the rapid changes in how software is developed and the ever-increasing complexity of software systems, test tools, and test environments, it may be challenging for software testers to grasp the variety of skills required for their job. This thesis seeks to determine the skills that employers expect of software testers and evaluate these requirements in relation to testers' role performance.

Through quantitative and qualitative analysis of 500 job advertisements, employers' requirements for hard and soft testing skills were uncovered (400 for testers and 100 for developers). Based on 34 interviews with software professionals, a comparison between these requests and the tester's role was created.

Every company required hard skills, but only two-thirds of employers required soft skills. The majority of job postings expected applicants to possess a wide range of hard and soft abilities, according to the findings. Test planning and design, test automation, functional testing, performance testing, and progress reporting were the most commonly requested hard skills. Employers prioritized the hiring of communicative and collaborative testers. Compared to earlier studies, the skill requirements for autonomous work, work ethics, fulfilling employment commitments, stress tolerance, openness, and flexibility have increased.

The interviews have revealed five testing roles: domain-specific tester, test automation specialist, test infrastructure specialist, user experience tester, and test manager. Due to the efficient application of skills, the experience gained through job repetition, and the abundance of tasks, testers tended to assume single tester roles. All roles found in the interviews appeared in the hiring ads, and the majority of employers required testers to fill multiple roles simultaneously. The most in-demand positions were those of test automation specialist and domain-specific tester.

Employers may take the five specified software tester roles into consideration when hiring testers or assigning test-related duties. In addition, software testers can use the role-specific information to make decisions regarding career advancement and skill growth. Academic institutions and training providers are invited to include the software tester roles and required skills into their curricula.
Acknowledgments

My supervisors, Viktoria Stray and Dag I.K. Sjoberg, led and calibrated my academic discoveries, without which my academic endeavors would not have been successful.

I would like to express my appreciation to the University of Oslo's Department of Informatics for supporting me with my studies.

I am grateful to the specialists with whom I conducted interviews for their candor, cooperation, and enthusiasm.

My gratitude goes to my husband, Cristian, for supporting me in the long and sometimes painful process of completing my academic studies, and to my three wonderful, bright, and inspiring children: Alex, Daniel, and Livia: Mom loves you dearly.
List of Papers

The following papers are included in this thesis:

**Paper I: Software tester, we want to hire you!**

**Paper II: The skills that employers look for in software testers**

**Paper III: A global view on the hard skills and testing tools in software testing**
Raluca Florea and Viktoria Stray. In Proceedings of the 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada, 2019, ACM/IEEE, pp. 143-151, DOI: 10.1109/ICGSE.2019.00035

**Paper IV: A Qualitative Study of the Background, Skill Acquisition, and Learning Preferences of Software Testers**

**Paper V: Exploring Human Factors of the Agile Software Tester**

**Paper VI: On the Roles of Software Testers: An Exploratory Study**
Raluca Florea, Viktoria Stray and Dag Sjøberg. Journal article under submission to the Journal of Systems and Software, 2022

**My contributions**
I was the first author with overall responsibility for Papers I, II, III, IV, VI, and the second author of Paper V. In Paper I, I was responsible for the design of the study, data collecting, data curation, data analysis (quantification of information, mapping it on the skill taxonomy), presenting and discussing the results. Stray and I completed the writing jointly. In Paper II, I was responsible for data analysis, as well as presenting and discussing the results. Stray and I worked together during the
writing process. In Paper III, I was specialist for data collection and analysis. The results presentation and discussion were completed together with Stray. In Paper IV, I was responsible for data collection, translation, transcription, and analysis of the interviews. Stray contributed to the thematic analysis and discussion of the results. The writing of the paper was a joint effort with Stray. Paper V was the result of a collaboration with Stray and Paruch. I was responsible for interviewing seven subjects. Paruch was the responsible for interviewing 15 IT professionals, acquiring supplementary data, and doing a literature review. Together with Stray, I performed the thematic analysis. I collaborated with Viktoria Stray and Dag I.K. Sjøberg on Paper VI. As the same data was used for Papers I, II, and IV, I collected the data. All three writers collaborated on the analysis, presentation and discussion of the results, as well as the drafting of the article.
# Contents

1 Introduction ......................................................................................................................... 9  
  1.1 Motivation ...................................................................................................................... 9  
  1.2 The software tester in current software development .................................................... 10  
  1.3 Research questions ....................................................................................................... 11  
  1.4 Research setting .......................................................................................................... 11  
  1.5 Thesis statement ......................................................................................................... 12  
  1.6 Contributions .............................................................................................................. 12  
  1.7 Thesis structure .......................................................................................................... 13  

2 Terminology ....................................................................................................................... 19  

3 Background ......................................................................................................................... 20  
  3.1 Software testing as a profession .................................................................................. 20  
  3.2 The function of testers in software development .......................................................... 22  
  3.3 Software testing expertise ............................................................................................ 23  
  3.4 The skills of software testers ...................................................................................... 24  
  3.5 Motivation of software testers .................................................................................... 27  

4 Research method .............................................................................................................. 28  
  4.1 Data collection ............................................................................................................. 28  
  4.2 Data analysis ............................................................................................................... 33  

5 Results ................................................................................................................................. 37  
  5.1 RQ1: What are the essential soft skills of software testers, according to employers and practitioners? ................................................................. 37  
  5.2 RQ2: What are the essential hard skills of software testers, according to employers and practitioners? ................................................................. 40  
  5.3 RQ3: How is the software tester’s role portrayed and carried out? ............................ 45  

6 Discussion ............................................................................................................................ 52  
  6.1 Software testing, a social and tactful job ..................................................................... 52  
  6.2 Extensive hard skill requirements for the software testers ............................................ 54  
  6.3 Selective role practice of the software testers ............................................................... 56  
  6.4 Implications for practice ............................................................................................... 59  
  6.5 Implications for research and further work ................................................................. 60  

7 Limitations ........................................................................................................................... 61  

8 Concluding remarks ........................................................................................................ 63  

9 References .......................................................................................................................... 64  

Paper I: Software tester, we want to hire you! An analysis of the demand for soft skills.... 77  
  Abstract ................................................................................................................................. 77  

1 Introduction ......................................................................................................................... 77  
  2 Data collection and analysis ............................................................................................. 79  
  3 Results ............................................................................................................................... 82  
  4 Discussion ......................................................................................................................... 85  
  5 Limitations ....................................................................................................................... 86  
  6 Implications ....................................................................................................................... 87  
  7 Conclusion and future work .............................................................................................. 88  
  8 References ....................................................................................................................... 89  

Paper II: The Skills That Employers Look for in Software Testers .................................. 91  
  Abstract ................................................................................................................................. 91
Part 1: Summary

1 Introduction ........................................................................................................... 92
2 Research method .................................................................................................. 95
3 Data collection ..................................................................................................... 95
4 Data analysis ...................................................................................................... 96
5 Results ................................................................................................................ 100
6 Discussion .......................................................................................................... 112
7 Implications ........................................................................................................ 122
8 Limitations .......................................................................................................... 124
9 Conclusion .......................................................................................................... 125
10 References ......................................................................................................... 126

  Abstract ............................................................................................................... 131
1 Introduction ........................................................................................................ 131
2 Data collection and analysis ............................................................................. 134
3 Results and discussion ..................................................................................... 135
4 Implications for Practice .................................................................................. 144
5 Limitations .......................................................................................................... 146
6 Conclusion .......................................................................................................... 147
7 References ......................................................................................................... 147

Paper IV: A qualitative study of the background, skill acquisition and learning preferences of software testers. 151
  Abstract ............................................................................................................... 151
1 Introduction ........................................................................................................ 152
2 Methodology ...................................................................................................... 152
3 Results ................................................................................................................ 154
4 Discussion and implications ............................................................................. 161
5 Limitations .......................................................................................................... 163
6 Conclusions and future work ........................................................................... 164
7 References ......................................................................................................... 164

Paper V: Exploring Human Factors of the Agile Software Tester .............................. 166
  Abstract ............................................................................................................... 166
1 Introduction ........................................................................................................ 166
2 Background and related work .......................................................................... 168
3 Related work ...................................................................................................... 169
4 Case study methodology ................................................................................... 174
5 Data analysis ...................................................................................................... 178
6 Discussion .......................................................................................................... 187
7 Implications for practice .................................................................................... 190
8 Conclusion .......................................................................................................... 192
9 Acknowledgments .............................................................................................. 193
10 References ......................................................................................................... 193

Paper VI: On the Roles of Software Testers: An Exploratory Study ......................... 198
  Abstract ............................................................................................................... 198
1 Introduction ........................................................................................................ 198
2 Related Work .................................................................................................... 200
3 Research Method .............................................................................................. 201
4 Results ................................................................................................................ 204
5 Discussion ........................................................................................................................................... 212
6 Conclusion and future work .............................................................................................................. 215
7 References ......................................................................................................................................... 216
Appendix A ........................................................................................................................................ 221
Appendix B ......................................................................................................................................... 222
Part 1: Summary

1 Introduction

1.1 Motivation

Throughout my nearly 20-year career in the software sector, I held many testing positions, beginning as a functional tester in telecommunications, continuing as a technical tester in logistics, and working as a test manager prior to becoming a project manager. I am currently engaged in the banking industry as a senior test manager. After a few career changes, some of the testing jobs felt more natural to me than others. There was always a variety of responsibilities at each of my positions, but certain duties were unquestionably more in line with my testing abilities. Like many others, I was repeatedly asked to take on responsibilities that I was not comfortable with or prepared for. "Because it's testing" was the most prevalent rationale offered by management for assigning me such tasks.

After reading many job advertisements while searching for new employment prospects, I realized that software testers are expected to possess a wide range of skills. It looked like employer requirements for developers were more uniform, and it felt somewhat simpler for a developer to prepare for a job and advance in their career. Consequently, I wondered what businesses expect from software testers. What do jobs as a software tester entail? What varieties of software testers exist?

As with many others, I initially encountered software testing as a job by accident. Because the university I attended did not offer a course in software testing, I learnt the job from the ground up. The search for a theoretical basis for software testing was rigorous, and I soon realized that, in 2005, the specialist literature was scant. However, the International Software Testing Qualifications Board (ISTQB) curricula gave a valuable overview of software testing and the software tester role.

I became an active member of the Norwegian Testing Board (NTB) and the Advanced-Level Working Group for Test Automation in ISTQB, where I contributed for eight years. In addition, I developed a PhD-level introductory course on software testing at the University of Oslo in 2012, which was later expanded into a full-semester course for bachelor's and master's degrees. The training was well-received by participants, with high participation rates and excellent reviews. Each
year, over 200 students join in the software testing course; they are enthusiastic and eager to learn.

These occupations and personal experience provided the foundation for my academic research on software testers, to whom I devote my thesis.

1.2 The software tester in current software development

The way software is produced has changed dramatically over the last two decades (de Souza et al., 2015). The software development discipline has been significantly transformed, as a result of new methods and concepts widely adopted in software engineering, such as Agile and DevOps (Cunningham et al., 2019). Hence, the roles and responsibilities of those who create software have shifted.

Roles such as full-stack developer, back-end developer, front-end developer, and mobile developer are clearly described in the literature (ISO/IEC/IEEE 24765:2017). (Montandon et al. 2021). However, although being well-established, the "software tester" role is not precisely defined (Cunningham et al., 2019). Assavakamhaenghan et al. (2019) defined software testers as the specialists responsible for developing test plans, testing the generated software for flaws, and ensuring that it fits the specified requirements.

The challenge for professionals in software testing is to assume and exercise a broad set of skills, and this may be a difficult achievement (Kurokawa and Shinagawa, 2008; Rooksby et al., 2009) given the diversity of testing practices and skills required to carry out testing activities (Yilmaz et al., 2012). Several studies describe the skills that software testers need to master as abundant: software testers must be proficient in various methodologies, tools, practices, and techniques (do Carmo Machado et al. 2014; Cerioli et al., 2020).

Garousi and Mäntylä (2016) authored a tertiary study focused on software testing, containing 101 secondary studies published in the timespan of 1994 to 2015. The results revealed that, while the literature was rich in reviews on test types (Banerjee et al. 2013), methods (Yusifoğlu et al. 2015; Rafi et al. 2012), practices (Vanhanen and Mäntylä, 2013), systems under test (Garousi et al., 2013; Doğan et al., 2014) and testing phases (Mäntylä et al., 2013), there was a shortfall of secondary research on the software tester role, the software testers’ skills, and areas such as test management, test environments, and exploratory testing. Given the scant literature in this field, gaining insight into the tester role is therefore essential.

Itkonen et al. (2012) proposed a categorization of testers based on domain expertise, system knowledge, and generic software engineering knowledge, in an effort to bring structure to the tester position. Saldaña-Ramos et al. (2012) suggested a model of software testing roles based on role attributions, which included test contract
manager, test manager, and test engineer. This thesis extends the skill-related and role-related testing work and offers a contemporary perspective on the software tester's function.

1.3 Research questions

Given that the hard and soft key competencies of software professionals are the subject of current research (Assyne et al., 2022), the purpose of this thesis is to reveal the overall skills required for testers and to compare them to the role's actual practice. Consequently, the following themes were trailed: mapping and ranking employers' requirements for the soft skills and hard skills of software testers; revealing the practice of the tester role, essential skills, learning preferences, and barriers, as described by software testing practitioners; comparing testers' ability to assume the role with employers' demands. As a result, this thesis examines the skill need and job fulfillment in software testing with the following questions:

RQ1: What are the essential soft skills of software testers, according to employers and practitioners?

RQ2: What are the essential hard skills of software testers, according to employers and practitioners?

RQ3: How is the software tester's role portrayed and carried out?

This thesis consists of six papers that address the research questions. More specifically, RQ1 is addressed in Papers I and V, RQ2 in Papers II, III, IV and VI, and RQ3 in Papers IV and VI, see Fig. 1.

1.4 Research setting

Fig. 1 Mapping of the research questions to the papers included in the thesis
The overall research consisted of two major components: an employers’ perspective on the relevant skills of software testers, and an examination of the testers’ practice and description of their job. In the end, a contrasting of the employer’s and practitioners’ views of the role was performed, see Fig 2.

1.5 Thesis statement

This thesis statement is as follows:

*Working as a software tester necessitates specific soft and hard skills. A greater understanding of the skills, tasks and responsibilities of the software tester will facilitate the allocation and execution of testing tasks, hence contributing to the development of better software systems.*

The objective is to collect evidence regarding the requirements and behaviors of software testers and to identify ways to assign software-testing work that are mutually beneficial to employers and employees.

1.6 Contributions

![Fig. 2 Research setting and timeline](image)

This research is based on a qualitative and quantitative analysis of the needs of employers and the job performance of practitioners. Therefore, the thesis contributes to the research community's understanding of the software testing role. More precisely, the following are the principal contributions:
First, by introducing the roles in software testing, the thesis provides empirical evidence for software testing practice and expands the research community's understanding of how testers' activities might be divided.

Second, the usage of job advertisements as the data source to examine employer requirements for testers. This thesis contributes to the broader topic of skill research by analyzing the skills requested by employers for software testers. Although companies invest a substantial amount of time and resources in attracting and choosing suitable candidates, relatively few academics have examined the employers' skill requirements, as indicated through job advertisements.

Third, mapping the employers’ demands on a testing process taxonomy increases the research community’s understanding of the quantity and type of testing skills that are required of the software testers. The sought-after software tester profile is completed by the mapping of hard and domain-specific abilities required of software testers. The research of trends in the soft skill needs of employers reveals the ongoing changes in the profile of testers, as required by businesses to successfully build software.

Lastly, comparing the viewpoints of employers and practitioners through theme analysis of advertising and interviews gives researchers greater insight into the disparities between the perspectives of these two parties and suggests strategies to harmonize them.

1.7 Thesis structure

This thesis consists of two major sections, a summary and a collection of papers, grouped as follows:

Summary. The "Introduction" section describes the thesis's research topic and the included papers. The vocabulary used in the thesis is presented in the "Terminology" section. The "Background" section provides an overview of the existing literature on the role of testers in software development, testing as a career, and software testing skills. The section concludes with a discussion of the current testing skills challenges in the industry and the prospects for software testers. The section titled "Research Method" elaborates on the procedures used in this research, including context, data collecting, and data analysis. The "Results" section summarizes the findings and describes how they help to solving the thesis's overarching research concerns. In the "Discussion" section, the general findings are interpreted in light of existing academic studies, along with recommendations for future research and practice. The "Limitations" section describes the context in which the results were interpreted, as well as the data sample restrictions. Within "Concluding remarks", a
summary of the findings and an overview of the paper's relevance to academia and industry are provided.

**Papers:** Each of the six papers included in this thesis is briefly described below.

Paper I. "Software tester, we want to hire you! An analysis of soft skill demands", shows the soft skills most required by employers of testers, with a focus on trends in the requirements, as well as a separate analysis of the soft skills required for agile software testers. The examination was conducted by analyzing the content of 400 online job adverts collected from 33 countries. Approximately two-thirds of companies need a minimum of five soft skills. In addition to team-playing skills, autonomous working, adaptability to change, and rapid learning have been the most in-demand qualities. New requirements for testers' soft skills included work ethic, customer focus, proactivity, and the ability to operate under pressure. The abstract states:

"One important discussion in the software development field is related to the skills that people need to have to build successful software products. This debate is generated on one hand by a large number of failures and delays of software projects. On the other hand, the debate is triggered by the need to build even better-quality software in a rapidly changing world. We will examine to which extent soft skills are relevant when hiring software testers and if there are any specific skills required for agile testers.

We analyzed 400 job advertisements for testers from 33 countries, out of which 64% ask for soft skills. Of the advertisements asking for soft skills, there is, on average, a request for 5 soft skills, 11 testing skills, and 5 technical skills. Only 30% of the companies ask explicitly for agile testers. However, our analysis shows no notable differences in skill demands for agile testers and the rest.

Software companies want to hire testers who can communicate well and have analytical and problem-solving skills. There is a significant increase in the need for openness and adaptability, independent-working and team-playing since 2012. In addition, there are new categories of soft skills identified, such as having work ethics, customer focus and the ability to work under pressure".

Paper II. "The skills that employers look for in software testers", maps the hard skill requirements for testers on test-related and technical taxonomies, organizes the domain-specific requirements by industry, then quantifies and ranks the employers' expectations for hard skills. Businesses have high skill requirements for software testers, requiring an average of ten test-related skills and five technical skills, with one-third of employers requiring domain-specific knowledge. Employers were
interested in test planning and design, test automation, functional testing, performance testing, and reporting on test progress. The most in-demand test-related tools were Selenium, Jira, HP Quality Center, QTP, and JMeter, and testers were expected to be proficient with them. Three-quarters of businesses demanded technically competent testers with knowledge in SQL scripting and programming in Java, C#, JavaScript, or Python. In addition, several organizations inquired about the testers' project management skills, estimate ability, and risk management understanding. The results indicate a vast array of highly specialized skills. The following abstract summarizes the paper:

"Software testing is an integral part of software development that provides better quality products and user experiences and helps build the reputation of software companies. Though software testers perform a role that requires specific tasks and skills, in-depth studies of software testers lag. Research studies of other roles within software development teams. In this paper, we aim to create a profile of testers by presenting an empirical analysis of the skills the industry currently needs. We analyzed data from 400 job ads in 33 countries. We mapped the skills on a taxonomy comprising test-related, technical, and domain specific skills. In addition, we looked at the demand for educational attainment, relevant certifications, and previous experience requirements.

Our findings show that employers are mostly interested in skills related to test planning and design, test automation, functional testing, performance testing, and progress reporting. One third of the job advertisers were interested in people with the skills to operate test execution tools. Selenium was the testing tool most in demand. The testers must have strong technical abilities, including programming skills in Java, C#, and SQL. Also, they must handle project management tasks such as estimation, risk management, and quality assurance.

Employers do not emphasize domain specific knowledge, which indicates that they consider testing skills portable across industries. One in seven job ads asks for a software testing certification. Our study helps clarify the complexity of the testing job and outlines the capabilities one needs to fulfil a software tester’s responsibilities".

Paper III. The goal of the third paper, "A Global View on the Hard Skills and Testing Tools in Software Testing", is to identify disparities in employer requirements for testers based on the geographic distribution of companies. In addition, test-related developer needs were mapped and compared with test-related tester requirements. A total of 500 advertisements were reviewed, including 400 pre-collected job postings
for testers and an additional 100 job postings for developers. The testing requirements for developers were minimal and relatively vague: only one in eight listed developer roles required test automation, and one in ten required unit testing. The different formulations of test-related requirements for testers and developers indicate that companies primarily require testers to execute software-testing tasks. The abstract states:

"Developing software with high quality is challenging in distributed software development. The purpose of the current study is to investigate the testing skills and tools required in the ever-changing world of global software engineering, according to industrial needs. We analyzed 500 job ads from 33 countries. The results show that a quarter of the testers and a fifth of developers are asked to work in distributed projects. The testers are asked to be highly skilled in a variety of test activities and tools, while the testing-skills demand for developers is low and somewhat vague. The profile of testers has a strong technical component in addition to the managerial one. Our findings show that employers need most that testers are competent in automated testing. Furthermore, the industry does not cover all aspects of testing with the demand for testers and developers. Surprisingly, neither role is asked to test the implementation of the general data protection requirements. Our study bridges the industrial needs and the practitioners’ skill development process. Therefore, software testers can use our study as a reference point to enhance their skills. Employers should use our results to check their testing-skill coverage within the development teams. Tertiary education providers are encouraged to use our findings, to update the curriculum in the software development area.”

Paper IV. The focus of the paper "A Qualitative Study of the Background, Skill Acquisition, and Learning Preferences of Software Testers," switches to shedding light on the testers' role practice, background, enablers, and barriers. Through 19 in-depth, semi-structured interviews with experienced practitioners in testing, an overview of software testers is provided, including information on their educational background, contact with the role, means of learning, sources of stress, collaborations, preferences, and role performance. More than half of the interviewers had degrees in disciplines other than IT. The majority of testers described their first contact with the tester profession as an occurrence rather than a deliberate career decision. The interviewees viewed their jobs as a series of tasks, and their experience as an ever-increasing pile of information that was sharpened through task repetition. Most of their learning came from informal sources, and they learned significantly through teamwork on tangible job projects. The professionals interviewed stressed the contextual nature of their work experience and their propensity to assume specializations within the role. The abstract of the paper states:
"Context: There is an indisputable industrial need for highly skilled individuals in the role of software testers. However, little is known about the educational background of these professionals, their first contact with the role, their preferences in acquiring their skills, the impediments they face, and their perception of the software testing role. Objective: The current paper reports on the background, the skills, learning preferences and role profiles, as described by the professionals in software testing, spanning over a significant number of industries, countries, and software development models. Method: We conducted 19 in-depth, semi-structured interviews of software testing practitioners, across eight industries. We performed a content and thematic analysis of the collected data. Results: The practitioners in software testing had a diverse educational background, and their first contact with the testing role was accidental. Exploratory testing was the preferred testing technique, while curiosity was identified as the most important feature in their skill set. Our respondents collaborated extensively with the developers, which they perceived as a learning source and a symbiotic work partner. Conclusion: The professionals in software testing described their skills as a rather undefined heap of knowledge, increasing with each work-task. They used mainly informal and hands-on learning approaches. They found it necessary for education providers to present information on software testing. Generally, companies assisted them well in the skill-development, but need to allow sufficient time-allocation for the learning. We identified five specialties of the role: product owner in testing, UX tester, DevOps tester, test automation specialist, and test-process coordinator.

Paper V, "Exploring Human Factors of the Agile Software Tester" examines the essential characteristics of successful software testers. Interviews with 22 software professionals (developers, interaction designers, and testers) revealed the following traits: the capacity to understand the big picture, strong communication skills, attention to detail, organization skill, inventiveness, curiosity, and adaptability. A competent tester was required to communicate extensively, on time, to give constructive feedback, to mitigate potential negative team dynamics. With detail-oriented testers in the team, developers felt comfortable implementing code modifications. To perform well in their profession, testers were required to demonstrate creativity in testing from several user perspectives and to comprehend the tested features in the context of the entire project. Additionally, efficient testers were described as inquisitive and capable of performing non-standard testing. In addition, peers believed that good testers were those who were organized. Nonetheless, valued testers were versatile and able to rapidly adjust work situations. The abstract of the paper is given below:
"Although extensive research has been conducted on the characteristics of the agile developer, little attention has been given to the features of the software-testing role. This paper explores the human factors of the software testers working in agile projects through a qualitative study focusing on how these factors are perceived. We interviewed 22 agile software practitioners working in three international companies: 14 testers, five developers, and three designers. Additionally, we observed 11 meetings and daily work of 13 participants in one of the companies. Our findings show that the views on the human factors shaping the agile software tester’s role were crystallized into seven traits, which the agile team members saw as central for the software-testing role: the ability to see the whole picture, good communication skills, detailed-orientation, organization, creativeness, curiosity, and adaptability. The testers spent half their day communicating and learned how to mitigate the fact that they had to bring bad news to other project members. They also facilitated communication between the business side and development. Based on our results, we propose the seven traits as dimensions to consider for organizations recruiting agile software testers, as well as a reference for IT and non-IT professionals considering a software-testing career".

Paper VI, "On the Roles of Software Testers: An Exploratory Study", explores the roles that emerge from testers’ practice and compares them to employer expectations for testers, determined by a thematic analysis of the Paper IV data collection. Five roles have emerged: the domain-specific tester, the test automation specialist, the test infrastructure specialist, the user experience tester, and the test manager. The testers were inclined to act in one role, and they defended their decision by citing experience with task repetition, efficient use of time and effort, and quality control of the work outcomes. The acquired data for Papers 1-3 was then assessed from the perspective of the roles. In contrary to the testers’ desire, the majority of companies required them to serve in several roles. As a result, recommendations were made to the industry regarding defining expectations for testers, which would aid in the development of consistent skills. The following abstract summarizes the paper:

"Context: Software development organizations need testers with high skill levels in a broad range of technical areas and application domains. Accordingly, we need a better understanding of how testers meet such skill demands in the practice of their role. Objective: This work aims to deepen the understanding of the typical tester role. Method: We performed a thematic analysis of 19 in-depth, semi-structured interviews with software testers working in various industries. To investigate employers’ views on such roles, we conducted a thematic analysis of 400 job ads. Results: From the interviews, we identified five subroles of software testers: domain-specific tester, test automation specialist, test infrastructure specialist, user
experience tester, and test manager. Most of the practitioners preferred to develop skills and act in one subrole. In contrast, most of the job ads requested that testers act in multiple roles. **Conclusion**: Our findings provide a deeper understanding of the tester role which may guide testers in their acquisition of skills and employers in the recruiting of testers”.

### 2 Terminology

Software testing has relatively many test-related ontologies (Souza et al. 2013) and therefore different definitions of terms related to testing. As such, I provide below the definitions used throughout this thesis for the following terms: *software tester*, *software testing skill*, and the *roles of the software tester*.

*Software tester* is a frequently used term but falls short of a comprehensive definition. While in established standards (IEEE 610-1990, ISO/IEC/IEEE 24765:2017) *software developer* is well-defined, software tester is recognized as a specific role but lacking further details. The standard ISO-29119:1, dedicated to software testing, defines a tester as "one who develops and tests deliverables and completes the processes associated with the dynamic test process". In the Standard Glossary of Terms, the International Software Testing Qualifications Board (ISTQB) defines testers as "skilled professionals involved in the testing of a component or a system". None of the popular lexicons (Cambridge Dictionary, Merriam-Webster Dictionary, Collins Dictionary, Oxford English Dictionary) nor the World Standards Cooperation (IEEE, ISO, IEC, ITU) give definitions or supplementary information on software testers, software testing, testing requirements, testing tasks, testing skills, or testing responsibilities.

Mathur and Malik (2010) defined software testers as the individuals responsible for testing and developing test cases and test strategies. Davidov et al. (2010) defined testers as specialists who find new defects from failed test cases, assess problems, and report them in a bug-tracking system. In the context of this thesis, a software tester is a practitioner in software development whose primary tasks include testing software products and confirming the program to functional and nonfunctional requirements.

*Skill* is defined as a practitioner’s ability to do something well (Attewell, 1990). Hence, *software testing skill* is the ability to perform software testing well. This ability draws on test-related knowledge and experience and influences the quality of test cases and their execution (Fujiwara and Yamada, 2001).
Soft and hard skills are two broad categories of software skills (Pieterse and Eekelen, 2016, González-Morales, 2011). According to Lippman et al. (2015), soft skills are the competences, behaviors, attitudes, and personal traits that enable people to effectively navigate their environment, collaborate well with others, perform well, and achieve their goals. Hard skills for software testing can be roughly categorized as test-related, domain-related, and technical. According to the ISTQB Glossary of Testing Terms (accessed 2021) and ISO/IEC/IEEE 24765:2017, test-related skills include test planning; designing test cases, test case implementation, testing on different levels, executing different types of tests, development, and maintenance of testware, test-data generation and administration, test-related measurements and progress reporting, test-related risk analysis, testing tools, and test quality improvement.

Domain-related skills include the cognitive skills that are applicable to a specific specialized domain or a class of problems (Smith, 2002), such as assessing the software implementation of national accounting laws or having experience in the banking, financial services, or insurance domains.

Technical skills are the non-testing competencies generally associated with STEM-related jobs, including programming, engineering, IT networks, and IT operations (Litecky et al., 2004; Kurokawa and Shinagawa, 2008).

The concept of role is useful for illustrating the authority, responsibility, and function of group members; it also promotes the sharing of information efficiently (Zhu et al., 2006). Roles have been frequently utilized in sociology, medicine, and administration since ancient times. Zhu (2003, 2006) and Zhou (2006) advocated for the use of roles in software development and their definition and application in software engineering.

3 Background

The following section explores literature related to the software testers’ part in software production, career prospects in software testing, testing-relevant skills, testing techniques, and software testers’ enablers and barriers.

3.1 Software testing as a profession

It has been shown that workers’ performance suffered when their skills did not match the high job requirements (Dawis and Lofquist, 1984; Muchinsky and Monahan, 1987). Kalleberg (2008) argued that when employment match the talents, requirements, and preferences of workers, people are generally content with their work and life, and workplaces are more likely to operate smoothly and efficiently. Nonetheless, mismatches between workers and their employment were anticipated.
Part 1: Summary

to result in a variety of problems for workers, businesses, and society. In recent decades, such mismatches have become prevalent, mostly due to the increased skill requirements of firms, globalization, and the expansion of IT and technical innovation (Kalleberg, 2008).

Recruiting and retaining highly productive personnel is essential for software firms to remain competitive (Bartlett and Ghoshal, 2002). As a result, companies express required skills during the employment process in a detailed and consistent manner (Schmidt and Hunter, 1998). Many firms place a greater emphasis on hiring skills than developing them, which can result in more particular and varying job needs among employers, hence raising the complexity of hiring (Cappelli, 2012). In addition, there is an underlying drive for a shift of duties from employers to education providers in the creation of the workforce, as firms no longer feel accountable for developing individuals to match shifting skill requirements (Cappelli, 2012).

In spite of the growing demand for software testers, Deak et al. (2013) found in a study of IT students that they preferred to be hired for other software-related occupations, such as development, with their choice sometimes being impacted by the vagueness and breadth of the employers' requirements. In addition, empirical research indicate that companies have difficulty locating applicants with the abilities necessary to do the tasks they need to fill (Cappelli, 2012).

According to practitioners, variables that negatively impact the careers of software testers include the immaturity of the labor market for testers, career-path-related challenges, and the instability of testing jobs (Fernández-Sanz et al., 2009).

About half of software-producing businesses utilized a generic evaluation approach for testers, lacking skill-related or role-specific performance evaluation methodologies (Kanij et al., 2012). However, the biggest reason not to pursue a career in testing, if given the option, was the high level of detail-oriented abilities required of software testers (Lizama et al., 2020). Nonetheless, Lizama et al. (2020) discovered that the software tester was still an unpopular profession among IT practitioners, mostly due to engineers' lack of adoption and the negative connotations associated with delivering bad news.

Additionally, confused work settings that provided ambiguous or inconsistent information badly affected newcomers (Wanous et al., 1992). In the Computer and Technology Industry Association's 2012 research titled "State of the IT skills gap," 93% of the more than one thousand respondents claimed that they have a skills gap, primarily owing to the rapid changes in technology making it difficult for IT personnel to maintain their abilities. The majority of businesses did not have a
procedure or method to identify potential IT gaps among their staff, according to the report.

However, the future of the tester's role remains challenging. To keep up with the current trends in software development, it is likely that additional domain-specific strategies and tactics will be required (Cunningham et al., 2019).

A factor to consider when predicting the future of software testing is the elimination of the tester role in favor of developer-performed testing. Even though there are benefits such as code flexibility and improved code maintenance, implementing test-driven development (TDD) to the extent desired by the industry presented considerable challenges (Causević et al., 2011). Lack of developer skills, lack of upfront design, domain- and tool-specific challenges, legacy code, and a lack of industry discipline to adhere to the TDD protocol were the primary factors preventing widespread adoption of TDD. As a result, the industry lacks perspective regarding how to integrate development and testing efforts and how to employ TDD more effectively.

Research has consistently indicated a need for academic instruction on software testing (Carrington, 1997) using various methodologies, such as black-box testing (Chen and Poon, 2004), peer reviews (Smith et al., 2012), gamification of testing (Fraser, 2017), and automated interactive systems (Smith et al., 2017). Software testing has not received the same emphasis in the university's curriculum as software development has, despite the fact that it accounts for more than fifty percent of development costs (Astigarraga et al., 2010, Mahmood et al., 2021). Consequently, many IT grads possessed good programming skills but lacked testing expertise (Astigarraga et al., 2010). In recent years, software engineering degree programs have placed a greater emphasis on quality assurance, according to Trejos-Zelaya (2020). Among the potential areas of collaboration between industry and academics are the development of a shared understanding of capabilities and the exchange of information on the technical skills and knowledge required for excellent performance in testing roles.

3.2 The function of testers in software development

Even though incremental software development does not have particular focus on the testers, Sharp and Robinson (2004) identified that agile testers are pivotal in building mutual trust between developers and customers. Seth et al. (2012) highlighted that the approach of incorporating quality into the software development process differed significantly amongst firms, and that the level of customer interaction had a direct effect on software quality. In addition, it was discovered that the quality of a software product does not necessitate the best technical implementation, but rather the solution that meets the majority of customers' needs.
As a result, testers played a major role in the management of user requirements throughout the development project's life cycle.

Nevertheless, the engineers and project managers engaged in training and practice sessions for their new responsibilities, as they transitioned from sequential development, which is characterized by complex processes and well-defined roles, to agile development. Frequently, testers had to determine their own responsibilities and work models (Magalhes and Miglioli, 2011). Testers were necessary in agile, as a separate role, because their attitudes differed from that of developers; while a developer could want to demonstrate the accuracy of the product, a tester would likely be more concerned with locating bugs (O'Regan, 2019).

Laporte et al. (2007), in an effort to add structure to the tester profession, recognized two specializations: software test designer and software tester. Based on the authors' experiences, Saldaña-Ramos et al. (2012) developed a model of competence in software testing comprised of four roles: test contract manager, test manager, test engineer, and tester. Based on their responsibilities in testing and the software development model, Yilmaz et al. (2015) identified the following tester roles: software quality assurer in plan-driven development, validation and verification tester in system engineering, tester in extreme programming (XP), and tester in feature-driven development. Yilmaz et al. highlighted a lack of clarity in the execution of roles, particularly in modern development, which makes role allocations less predictable.

### 3.3 Software testing expertise

In a research examining 25 years of testing technique experiments, Juristo et al. (2004) noted that more than half of the knowledge on testing techniques lacked a formal basis, as they were primarily based on subjective sensations and perceptions. Testing software is a knowledge-intensive endeavor. Thus, knowledge management should be used to direct software testing skills. Despite the growing interest in the subject, few studies have addressed this research field (De Souza et al., 2015).

Testing and quality assurance is a significant expense in software development; nonetheless, apart from test automation and input minimization, few additional methods have been identified to reduce testing effort. Due to the continuous difficulty of reducing the testing effort, more research was needed in this area (Elberzhager et al., 2012). Allocation and usage of resources are among the primary methods utilized in software testing to increase its efficacy (Farooq et al., 2011). Through a variety of test cases, the tester's experience was portrayed as a crucial aspect in reducing testing expenses (Singh et al., 2012; Catal and Mishra, 2013).
Most testing approaches relied on the expertise, knowledge, and intuition of the tester (De Souza et al., 2015). In exploratory testing, a knowledge-intensive, creative, and skill-requiring testing technique, the tester rather than the documentation plays a prominent role, and exploration helps to improved software testing since it better answers the demands of the user (Itkonen et al., 2015). Itkonen et al. (2012) discovered that the tester's own experience aided significantly in identifying software faults without thorough case descriptions. Previously, Itkonen et al. (2011) discovered that exploratory testing was more effective than test-case-based testing for functional testing, because it surfaced failures that deviated from the predetermined path of the test cases.

Beer and Ramler (2008) focused on identifying the sources of testers’ experiences and found that, when working on previous versions of the software, involvement in the design, development, and bug-fixes of previous iterations were the main sources of experience. The experience was contextual and consisted of a sort of repetition of the test activities, with the alterations presumed by the new software versions.

When methodologies for test oracles like modeling, specifications, contract-driven development, and metamorphic testing failed or were insufficient, the final source of oracle information remained with humans who were familiar with domain-specifics, informal requirements, expectations, and norms (Barr et al., 2014).

In a poll of software testers addressing the elements that influence testing efficiency, experience and domain expertise were deemed to be essential tester abilities (Kanij et al., 2014). However, testing knowledge within businesses is predominantly tacit and acquired via experience, making it difficult to express and communicate (De Souza et al., 2015).

### 3.4 The skills of software testers

**Soft skills**

Most typically, the literature emphasizes testers' hard testing abilities, such as their responsibility for doing testing and creating test cases and test plans (Mathur & Malik, 2010). In another instance, testers were depicted as experts who found new defects from failed test cases, examined problems, and reported them in a bug-tracking system (Davidov et al., 2010).

However, the testing process requires not only technical skills but also specific socio-technical abilities (Sánchez-Gordón et al., 2020). Therefore, an efficient software tester must possess a wide range of skills and knowledge that frequently extends beyond testing. In this direction, progress has been made with the rise of agile. Saiedian and Dale (2000) characterized testers as crucial for bridging the gap between developers, stakeholders, and customers, as they understand and track...
business needs throughout the development process. However, more studies focused on the soft skills of testers are needed (Sánchez-Gordón et al., 2020).

In a survey study on four Norwegian companies, Deak (2014) identified a set of traits considered as defining a successful software tester, including communication, a strong IT background, and attention to detail. Indeed, the software testers were required to communicate with developers, stakeholders, and customers more than any other position (Ahmed et al., 2015). However, despite growing interest, the study of the testers’ soft skills still lags behind that of IT (Ahmed et al., 2015).

In prior research, intellect, intent, and emotion have been recognized as three differentiating characteristics of software testers (Ekwoge et al., 2017). Test-related intellect included problem-solving and decision-making. Intent encompassed motivation, call to action, and testers’ perceptions of the value of their contributions. Emotion consisted of the tester's feelings of self-respect and team loyalty.

The job requirements for various IT roles were mapped to five personality qualities, in order to discover the personality types suited for software testing (Rehman et al., 2012). The results indicated that the software tester role possessed a particular openness to experience and a notable sense of duty. In addition, the Myers-Briggs personality types disliked testing, with the one exception: the ISTJ (introvert-sensing-thinking-judging) personality type was predisposed toward it, indicating that focusing on soft skills of software-related positions could increase testing efficiency (Gulati et al., 2015). Indeed, testers displayed greater conscientiousness than any other group of development professionals, a feature deemed particularly important to their position (Kanij et al., 2015). However, testing was viewed as a burdensome task needing persistence, the ability to select from a large array of testing options, and a high level of attention to detail (Capretz et al., 2015).

**Hard skills**

Formerly, it was believed that software usability could be measured through the number of help screens and menu options (Bevan and Macleod, 1994; Benyon, 1993). Even so, a significant portion of this perspective has shifted, and as a result, testers now evaluate essential software-related characteristics such as the effort required to learn, operate, and comprehend the system (Rubin and Chisnell, 2008).

In a mapping study of testing techniques, it was concluded that the techniques-related knowledge was limited (Juristo et al., 2004). In a follow-up study on testing techniques, González et al. (2014) demonstrated that the situation has not altered considerably, since the same deficiencies in testing technique knowledge were identified.
In a systematic review of regression test selection strategies, Engstrøm et al. (2010) offered an exhaustive overview of testing techniques: 28 strategies were identified only for regression testing. In addition, no technique was deemed preferable, as their implementation and efficacy depended on numerous variables. Numerous static and dynamic testing methods are established, according to the evidence (Elberzhager et al., 2012). The application of these techniques frequently consumed more than fifty percent of the total software development effort; consequently, enhancing the effectiveness of testing is a crucial part of software development (Elberzhager et al., 2012).

The size and complexity of systems design, as well as the rigorous time-to-market constraints, place a great burden on the requirements and specifications bug-finding process (Bhadra et al., 2007). Multiple complimentary verification procedures and technologies were used by the testers to address this issue and raise the effectiveness of bug discovery to a satisfactory level. However, these instruments and techniques had to be utilized with care and precision in order to accomplish the intended results.

Despite the fact that businesses were increasingly interested in combining requirements and testing, this connection often did not exist in practice, resulting in gaps between the two processes, especially in the case of nonfunctional requirements (Barmi et al., 2011). A study of test execution practices revealed that testers utilized a variety of execution approaches and strategies during testing and did not rely on documentation. Nonetheless, when performing exploratory testing, testers mainly depended on experience-based testing (Itkonen et al., 2011).

A trend related to the technical aspect of a tester role is testing in DevOps teams, with the responsibility to continuously test, with test automation as part of the role, as well as responsibility for a suite of ever-changing tools in testing (Cunningham et al., 2019). Cruzes et al. (2019) found that testers working in a DevOps context needed to understand code, architecture, and test automation; continuous testing, which involved checking code quality, pipeline automation, application quality, and customer experience, was distinct from automated and manual testing. Continuous integration (CI) is frequently used to promote frequent and rapid releases, and test automation is one of CI's primary pillars (Sthål and Bosch, 2014). Important for the success of CI in contemporary software development is the maturation of test automation procedures (Shahin et al, 2017).

(Yusifoğlu et al., 2015) asserts that automated tests provide software testing with benefits such as repeatability, predictability, and efficiency. In projects that implement test automation, numerous testing scripts are typically developed. However, the creation of such scripts is complicated and prone to error. Therefore, it was necessary to establish strategies and consistently devote work in script maintenance. Co-maintenance and co-evolution with production code, the detection
of test smells, and the oracle assertion adequacy were a few of the most prevalent problems in maintaining automated test scripts (Yusifoğlu et al., 2015).

Cruzes et al. (2017) determined that more work should be devoted to security testing, given that many agile teams rely on incidental pen testers, lack experience in static testing, and take an unstructured approach to security testing despite the critical necessity of software security. The same absence of testing structure is observed for nonfunctional features of software systems, generally due to limited time and testing resources (Camacho et al., 2016).

The quality of software developed by geographically distributed teams can be affected by the team distribution (Jabangwe et al., 2016). At the same time, distributed software testing has witnessed increased adoption over the years (Shah et al., 2014). However collaborative software teams may be, distances negatively affect the quality of software products (Nguyen-Duc et al., 2015; Tell and Babar, 2012), as well as team performance (Anh et al., 2012). Therefore, the competencies necessary to overcome distances, inherent sourcing issues, technological challenges, and role-specific drawbacks emerged as decisive.

3.5 Motivation of software testers

De Souza Santos et al. (2017) identified five factors that influence the motivation of software testers: acquisition of useful knowledge during work, work variety, creativity in solving tasks, well-defined work with a precise sequence of steps, and recognition of work; they emphasized the last factor, which had a lasting effect on the testers' motivation and led to an increase in individual productivity and enhanced teamwork. The testers were demotivated by obsolete testing environments, demobilization, and the devaluation of testing professions, which were frequently viewed as merely an extension of development (Gonçalves et al., 2017).

While the testers were motivated by challenges and had a personal focus on quality improvement, they were negatively affected by the lack of recognition of their role, i.e. the undervaluing of testing in the organization, followed by issues with management such as inadequate testing resource allocation and the assignment of unrelated tasks (Deak, 2014). The overemphasis placed on project KPIs such as the amount of automated test cases, while the benefits of other types of testing were neglected, also impeded the testers' work (Shah et al., 2014).

Regarding ethical requirements in software testing, the primary responsibilities of testers were to test the system thoroughly, to be transparent about testing results, and to communicate them, while employers were required to provide the necessary testing resources and time (Nindel-Edwards and Steinke, 2008). Despite the efforts
of employers to acknowledge the significance of testing and reward software testers, there was still a substantial problem with corporations minimizing costs and the duration of projects, so compromising quality and reducing the testing effort (Deak et al., 2016).

Although testing is essential to the successful operation of software systems, it is the least understood and most challenging aspect of software engineering (Lizama et al., 2020). Moreover, its critical nature adds to its difficulty; hence, software testers viewed it in a very bad light, preferring other roles due to their generally higher acceptability (Lizama et al., 2020).

While the benefits of test automation are acknowledged, testers are demotivated by the inability of test scripts to replace certain manual testing jobs, the complexity of maintaining automated tests, and the selection of unsuitable test automation methodologies (Rafi et al., 2012). In addition, fifty percent of testers thought that available testing tools were either incompatible or inadequate for their needs (Rafi et al. 2012). Nonetheless, one of the primary deterrents for testers to pursue a career in testing was their job's intricacy (Capretz et al., 2021).

4 Research method

As the research objectives did not indicate a positivist or interpretivist method to skill study (Oates, 2003), I opted for a pragmatic strategy (Cherryholmes, 1992), while the software testing roles were constructed inductively (Thomas, 2006).

I collected and coded the complete text of job advertisements and performed a quantitative and qualitative analysis of the coded data in order to study employers' perceptions on the software tester role. Through theme analyses of interview data, I uncovered the role and essential characteristics of software testers, as described by practitioners. I used the roles and, thematically analyzing the job advertisements for software testers, I found number and combinations of roles that the employers' asked the testers to assume. Then, I compared the practitioners' and employers' perspectives of the testing role.

4.1 Data collection

To analyze the employers’ requirements for software testers, hiring advertisements were chosen, as the hiring process calls for a thorough and consistent evaluation of one’s skills (Floyd and Gordon, 1998), the employers explicitly state their needs and search for those who can satisfy these needs (Marlow and Dabbish, 2013). Moreover, the hiring advertisements were found to crystallize and express competency-related requirements (Galanaki, 2002).
SimilarWeb's Web traffic analytics were employed to determine the most popular job-search engines to collect the data. According to global user traffic, the top five engines were Monster.com, Indeed.com, Glassdoor.com, Careerbuilder.com, and Simplyhired.com. Consequently, these engines were utilized to acquire samples of software testing assignments. I relied on the built-in intelligence of the job engines, which gave a multiplicity of job postings for positions whose primary responsibility was software testing: software tester (77 ads), QA tester (41 ads), test analyst (11 ads), automation tester (9 ads), test engineer (7 ads), manual testers (4 ads), penetration testers (4 ads). Only three of the 400 job postings required test leads, and only three required test managers. Additionally, 213 adverts for the software tester profession included unique terms (only one instance was found in all the data sample collected). Examples include "MS dynamics CRM tester", "senior ETL tester", "testing specialist with HIX/HBX experience", "senior testing automation for network management", "linguistic tester", "algorithms for driver assistance tester", "content producer tester", and so forth.

Two-thirds of the gathered job postings were in English, while one-third were in multiple national languages. Duplicates were deleted, and every non-English material was translated into English using two automated translation tools: Google Translate and ETranslate. The parallel translations were cross-checked, and only advertising with similar translation outcomes were retained. In addition, more than 90 percent of the translated content was reviewed by an IT expert who was a native speaker of the language in which the original advertisement was written (Spanish, French, Italian, Norwegian, Swedish, Chinese, Vietnamese).

Ads that did not meet the quality standards (such as coherent content, correct domain/industry) were deleted after an initial quality assessment. We evaluated and quality-checked the job advertisements on the fly, and we did not collect duplicate advertisements, advertisements with low-quality text, or advertisements with varying translation outcomes. A set of 400 English-texts/translations was retained, all of which were unique and vetted for quality. This sample size was sufficient to produce an accuracy rate of 95% and a margin of error of +5% (sample size = 384 and does not change for a population greater than 20,000, and we know that the testers population is at least 500,000 based on the number of testing certifications awarded by ISTQB).

The obtained data set contained advertisements from 33 countries, including the United States (96), Canada (65), Norway (22), the United Kingdom (20), Argentina (17), France (17), Mexico (15), South Africa (14), China (14), Vietnam (13), India (12), Sweden (10), Australia (10), Spain (9), and Germany (8), among others (59). The adverts were posted by both private and public in-house software developers and staffing agencies. Microsoft, Tesla, IBM, Capgemini, Bank of America, Texas
Instruments, the New South Wales Government, and the National Bank of Canada, to name a few, were among the employers whose size, industry, and software specifications differed significantly.

I sought to create a more comprehensive view of testing requirements for a software development team. With this goal in mind, I made initial efforts to comprehend how testers and developers divided testing responsibilities. As a result, I collected a smaller sample of advertisements for developers and studied their test-related required skills. The sample size was lower (100 ads) because I wanted to complete the picture of testing tasks within a software team and determine which roles are responsible for various testing tasks. I used the same job-search engines for developers as I did for testers. A search for "developers" yielded advertisements for programmers (32), web-developers (19), full-stack developers (11), front-end (11) and back-end (8) developers, software engineers (11), software designers/architects (5), mobile developers (2), and devOps (1).

The goal of the interviews with IT professionals was to determine the essential skills of a tester as perceived by their colleagues. We conducted interviews with 34 IT professionals over the course of two sessions. The first round (Round 1) was conducted in October and November 2019 with the participation of 19 software testers (see Table 1). We used the same interviews, to investigate the practice of the tester position.

15 IT practitioners, including developers, testers, and software designers, participated in the second round of interviews in February 2020. Florea conducted the 19 interviews in Round 1, while Paruch conducted the 15 interviews in Round 2. Appendix A contains the interview guidance. Seven of the interview data included in Paper V also appeared in Papers IV and VI. The interview guide is available in Appendix B.

The data obtained for the studies comprised of interview transcripts and secondary data (CVs and public professional information from the interviewees). These data assisted in describing the academic background of the respondents in order to convey their professional trajectory in various workplaces, industries, roles, and job duties in a complete manner.

Interviews are a purposeful discussion between two individuals (Kahn and Cannell, 1957). One-on-one, face-to-face, and internet-mediated electronic interviews were conducted using a semi-structured interviewing method. Even though some interviewees may have been inhibited by an audio recorder, audio recordings were used to gather and save data because they allowed us to focus on listening to the interviewees, to minimize ambiguities when transcribing the interviews, and to re-listen as necessary. The inherent problem of the audio recordings was reduced by
establishing a pleasant environment and beginning with a few short questions to put the participants at ease. None of the participants withdrew from the study. The saturation concept was adhered to (Saunders et al., 2018), and the procedure was concluded when the data no longer provided new information. All the interviewers had extensive software testing experience (see Table 1). At the time of the interviews, the respondents had positions in test management, quality assurance/business analysts, test automation, software testing, domain expertise, and IT consulting.

The respondents volunteered to participate in the study, and their consent was freely given and based on complete information regarding participation rights and data use. All participants were informed of the goal of the research, their rights as interview participants, and the measures used to protect their data; all interviewees provided written consent for the interview, data collection, storage, and processing. The questions asked during the interviews were appropriate and pertinent, and they target specific scientific aims. All sensitive and personal data was anonymized, and the responses, their transcriptions, and their translations were stored securely.

At least once, each interviewee had changed professions or teams in software testing. Over half of the interviewees did not have a degree in computer science or information technology. During the course of their employment, the respondents served as testers in a variety of industries, including accounting and logistics, banking, telecommunications, automotive, audit, tourism, and health sector.
Table 1. Background of the interviewed software testers, Round 1

<table>
<thead>
<tr>
<th>Id.</th>
<th>Gen</th>
<th>Age</th>
<th>Test Exp.</th>
<th>Roles in Testing ***</th>
<th>Com. Size*</th>
<th>Educational Attainment *</th>
<th>Data used in papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>♀</td>
<td>46</td>
<td>16y</td>
<td>T, TM</td>
<td>L</td>
<td>Economics (BA)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>12</td>
<td>♂</td>
<td>44</td>
<td>13y</td>
<td>T, QA, BA, TM</td>
<td>L</td>
<td>Logistics (MSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>13</td>
<td>♀</td>
<td>29</td>
<td>7y</td>
<td>QA, T</td>
<td>S/M</td>
<td>Computer Science (MSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>14</td>
<td>♂</td>
<td>44</td>
<td>19y</td>
<td>T, QA, BA, TM</td>
<td>L</td>
<td>Economics (MA)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>15</td>
<td>♀</td>
<td>35</td>
<td>12y</td>
<td>T, QA</td>
<td>L</td>
<td>Mathematics (BSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>16</td>
<td>♂</td>
<td>39</td>
<td>9y</td>
<td>T, QA, BA, Tau, TAn</td>
<td>L</td>
<td>Accounting &amp; Finance (BSc)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>17</td>
<td>♂</td>
<td>51</td>
<td>19y</td>
<td>T, TM</td>
<td>L</td>
<td>Biochemistry (BSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>18</td>
<td>♀</td>
<td>55</td>
<td>19y</td>
<td>T, QA</td>
<td>S/M</td>
<td>Undergrad.</td>
<td>IV, VI</td>
</tr>
<tr>
<td>19</td>
<td>♀</td>
<td>40</td>
<td>8y</td>
<td>QA</td>
<td>S/M</td>
<td>Business &amp; Economics (MA)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>110</td>
<td>♀</td>
<td>36</td>
<td>8y</td>
<td>T, QA</td>
<td>L</td>
<td>Economics (MA)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>111</td>
<td>♀</td>
<td>40</td>
<td>8y</td>
<td>T, QA, BA</td>
<td>L</td>
<td>Linguistics (MA)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>112</td>
<td>♀</td>
<td>41</td>
<td>13y</td>
<td>T</td>
<td>L</td>
<td>Computer Science (BSc)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>113</td>
<td>♀</td>
<td>33</td>
<td>7y</td>
<td>T, QA, BA</td>
<td>L</td>
<td>Economics (MA)</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>114</td>
<td>♀</td>
<td>59</td>
<td>27y</td>
<td>T, TM, OT, UAT, QA, BA, TA</td>
<td>L</td>
<td>Computer Science (BSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>115</td>
<td>♂</td>
<td>49</td>
<td>14y</td>
<td>T, QA, DT</td>
<td>L</td>
<td>Undergrad.</td>
<td>IV, V, VI</td>
</tr>
<tr>
<td>116</td>
<td>♂</td>
<td>36</td>
<td>11y</td>
<td>T, QA</td>
<td>L</td>
<td>Computer Science (MSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>117</td>
<td>♂</td>
<td>50</td>
<td>10y</td>
<td>T, VE</td>
<td>L</td>
<td>Informatics (MSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>118</td>
<td>♂</td>
<td>41</td>
<td>11y</td>
<td>T, TM, C, TAn, QAM, DTM</td>
<td>L</td>
<td>Physics (MSc)</td>
<td>IV, VI</td>
</tr>
<tr>
<td>119</td>
<td>♀</td>
<td>43</td>
<td>9y</td>
<td>T, TM, TAU, ST, AUa</td>
<td>L</td>
<td>Banking (MA)</td>
<td>IV, VI</td>
</tr>
</tbody>
</table>

* BA = Bachelor of Arts; BSc = Bachelor of Science; MA = Master of Arts; MSc = Master of Science
** S/M = Small / Medium (<250 employees); L = Large (>250 employees)
*** AuA = Automation Architect; BA = Business Analyst; C = Consultant; D = developer; DT = data tester; DTM = Domain Test Manager; OT = Operations Tester; QA = Quality Assurer; QAM = Quality Assurance Manager; Sr. = Senior; ST = Security Tester; T = tester; TAU = Test Automator; TAn = Test Analyst; TM = Test Manager; UAT = User Acceptance Tester; VE = Validation Engineer
4.2 Data analysis

I manually collected and analyzed the advertisements, and they were all reviewed by another researcher. We chose to manually analyze them because the structure of the ads was not always the same, also the templates used by the job-search engines were different; some of the job-search engines even allowed job advertisers to post in their own format. Finally, not all job posters had the same understanding of how the information had to be filled in an ad. For instance, we found soft skills in the sections dedicated to job attributes / job responsibilities. We went manually through each of the job ads, looked for all the hard skills and soft skills requirements and grouped them accordingly.

For the thematic analysis of roles, both the advertisements and the interview-data were thematically coded by me and thoroughly reviewed by two researchers. This process was iterative as well, and it comprised two iterations. We discussed the coding and emerging themes. In the rare cases where we did not agree on a code, we did not code the text.

The content of each advertising was categorized into discrete units of information: data points holding a single piece of relevant information, such as a responsibility, a job attribute, or a required type of experience, and an example of which is shown in Fig. 3. The abilities were then divided into two categories: hard skills (testing, technical, and domain-specific) and soft skills.

On the dataset of job advertisements, both quantitative and qualitative analyses were conducted, as well as qualitative analysis of the interview data. The collected data from job advertisements consisted of descriptive text, including, for each advertisement, the role name, description of the vacant position, requirements for the role, responsibilities to assume, education and certifications required, minimum experience required, and other requests. This material was converted into categorical data using taxonomies of software tester-specific skills (see Figures 5, 6, and 8).

We used the ISTQB Foundation Level Syllabus (2011) to generate a taxonomy of hard skills, and the widely available taxonomy offered by Ahmed et al. (2012) to map soft skills. The hard skill taxonomy was evaluated using an initial sample of 20 job postings. In subsequent revisions, the taxonomy was examined and refined.

More than 8000 data points were collected. For numerical processing, the same data was grouped into a single category, and each required skill was allocated a unit that was tallied once. There was no weight assigned to skills or categories. As the relative difference between the skills was crucial to the investigation, the information gathered consisted of discrete ratio-data.
As a Software QA Tester, you will be joining a large, global team focused on helping our business partners consolidate their processes around the world. Our client’s program will touch many of our investments processes and supporting systems from the Front Office, the Middle Office, Back Office Operations as well as our Performance and Analytics teams. You will be working within a team of business experts, application experts, project managers, and testers co-located in one of our collaborative workspaces in North America. We are seeking an individual with 5 plus years of experience as a Quality Assurance Lead.

**Tasks:**
- Drive implementation of test process, and tools for a new Client Reporting and Client Lifecycle Management vendor software implementation including test strategies and test plans. Software QA Tester is specialist for planning, implementing, and reporting on testing aspects of software products/projects. Lead manual & automated testing activities supporting integrated systems development teams. Construct test environments for execution across multiple test instances and projects. Develop test plans in coordination with product development and create test cases based on these requirements. Implement testing plans and test cases and present testing results. Create long case, develop and maintain test specifications, test cases, test scripts and test data. Implement test practices, standards and processes. Troubleshoot test issues, record test results, track and prioritize defects, participate in release decisions and establish measures of test efficiency, effectiveness and product readiness. Lead defect analysis/reporting, prepare test scenarios and scripts, and defect tracking system. Provide timely and succinct project information to stakeholders to include:
  - Test planning and execution status, defect reporting, and ensure appropriate resolution of issues.
  - Provide input for systems release readiness reviews, change management/milestone reviews, and business Go/No-Go decision gates.

**Requirements**
- Experience with Waterfall, Agile, iterative development and testing methodologies, investments industry knowledge, particularly with Front, Middle and Back Office Self-starter with ability to work independently and work with remote teams in different locations. Demonstrated ability to coordinate cross-functional and cross-application work streams toward testing task completion and delivery.
- Test estimates and test deliverables. Review and analyze the business requirements and specifications to validate the testability and identify the testing requirements. Develop test strategies and plans for all projects and releases. Ensure completion of all required test documentation including test scripts/cases, traceability matrices, test schedule, test results and test status. Manage onshore and offshore test resources and multiple tasks and changing priorities. Support User Acceptance Testing. Experienced with Jira, SQL and Beyond Compare applications. Excellent management, written and verbal communication, and organizational skills. Experience with a test tracking software package is desired.
- 5-5 years of prior software testing experience is required.

**Fig. 3** Extract of the data analysis, job advertisements
### Part 1: Summary

#### Fig. 4 Extract of the thematic analysis, with examples from the interviews

<table>
<thead>
<tr>
<th>Tasks preferred</th>
<th>Tasks avoided</th>
<th>Issues in testing</th>
<th>Relation with others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informal specialisation</td>
<td>Lack of knowledge</td>
<td>Project issues</td>
<td>Closest collaborators</td>
</tr>
<tr>
<td>&quot;I think about everything that I have learned, and I learned it by doing&quot;</td>
<td>&quot;I didn’t actually have the right knowledge [...], because it turned out to be pretty technical. So I just had to ask people for help all the time&quot;</td>
<td>&quot;If you have people with the role testers, they need to include themselves from the beginning of the project, because they won’t be invited&quot;</td>
<td>&quot;It’s the developers that I collaborate most with&quot;, &quot;product owners&quot;, etc.</td>
</tr>
<tr>
<td>Picking responsibilities</td>
<td>Amount of work</td>
<td>Management issues</td>
<td>Collaboration extent</td>
</tr>
<tr>
<td>&quot;I like positive test scenarios, happy paths, not so much negative testing, I also like a lot test automation&quot;</td>
<td>&quot;Don’t let them get you to do manual tests, because you will never get rid of this&quot;</td>
<td>&quot;I guess that the person who hired me didn’t understand what my expertise was&quot;</td>
<td>&quot;It’s pretty much half of the time every day, because it’s important that we work together&quot;</td>
</tr>
<tr>
<td>Prioritizing own tasks</td>
<td>Conflict potential</td>
<td>Lack of time</td>
<td>Object of collaboration</td>
</tr>
</tbody>
</table>
| "I want to contribute with things I am good at" | "Without the help of a test script [...] it might not be pleasant to bring the bad news" | "If there is not time to sit down and learn, then I think the company will stagnate" | "It's a lot about new functionality [...] what we are about to implement."
| Keeping a career path | Resisting change | Lack of personnel | Need for collaboration |
| "I am more of a domain-expert...not with the technical testing" | "They kind of want me to do something about it, but I can’t" | "My impression is that they were not so much interested in my skill set" | "I think the testing part is important for the developers to do the testing before sending it out" |
| Investment in skill | Reason for changing jobs | Test-scripts issues | |
| "It’s more important for me to work on the hard skills" | "The second job was not right for me. I missed being more connected to the product" | "And one thing that it is not understood by those who don’t make automated testing: they think that everything is out of the box, and it just takes... they don’t realise that you have to go into the code actually" | |
In the thematic analysis of the interviews and annex data, we extracted qualitative information using explicit codes, which were defined as patterns of meaning (themes) across qualitative data sets (Braun and Clarke, 2006; Patton, 2014). Using the NVivo12 software suite, we analyzed the transcripts. Inductive analysis was conducted utilizing open and axial coding (Stol et al., 2016).

We began phase 1 by transcribing and reading the interview transcripts. Additionally, we collected and reviewed the interviewees’ resumes. We often documented our thoughts and ideas. In the second phase, open coding was applied to each transcript. This investigation yielded 388 codes. In step three, we utilized axial coding to group the codes into twenty subthemes, see Fig. 4 for examples.

In phase 4, the codes and subthemes were refined in two rounds. For instance, we eliminated candidate themes such as "perceived job safety" that were irrelevant to
Part 1: Summary

In this study, we identified and named four themes: "preferred tasks," "avoided tasks," "testing issues," and "collaborators".

We used these four interview-data themes to thematically code the text in order to identify different software tester profiles (see Fig. 5). This coding gave rise to software tester roles. In addition to conducting interviews, we combed through the job postings to identify roles that were not identified during the interviews. We discovered a few additional requirements, such as pen testing, but no new clear roles in the advertisements.

5 Results

This section describes the soft and hard skills required of software testers by employers, identifies the roles of software testers, and compares the requirements of employers with the roles assumed by testers.

5.1 RQ1: What are the essential soft skills of software testers, according to employers and practitioners?

Two-thirds of tester job postings included requests for soft skills, with an average of five soft skills listed. The ability to communicate effectively was requested the most frequently (60%), on a decreasing trend in a six-year timespan, see Ahmed et al. (2012). Employers subsequently requested that testers be analytical and problem-solvers (52%), team-players (41%), independent workers (40%), and have organizational skills (39%; see Fig. 6).

The majority (36%) of new trends in the demand for soft skills concerned work ethics. Employers specifically requested integrity and trustworthiness in interactions with customers and team members. Employers frequently requested that testers test from the customer's perspective and keep the customer's requirements in mind. In addition, testers were expected to demonstrate initiative in picking up and completing job duties. Multiple employers requested responsibility-related qualifications, emphasizing that the tester must be able to deliver completed work with well-grounded results.

The results of the thematic analysis of 22 practitioner interviews reveal that they share a number of perceptions regarding the qualities that make testers effective at their jobs. The following qualities of the testers emerged from the interviewees' accounts.
5 Results

Fig. 6 Demand and trends in soft skill requirements for software testers

*Ability to see the whole picture:* According to the practitioners interviewed, a tester must be able to comprehend the entire project in order to be effective. As a central tester trait, the ability to see the high-level aspects of project objectives, customer requirements, quality demands, domain knowledge, and technical knowledge was cited.

*Detail-orientation:* The interviewees characterized proficient testers as those who were able to keep track of a large number of details regarding the projects they worked on. One interviewee stated that the testers with whom they collaborated uncovered user experience issues that the designers missed.

*Good communication skills:* The interviewees reported that agile increased the frequency of communication, especially with customers. The testers described themselves as frequently serving as a buffer between the business and development. Therefore, the ability to provide information in a clear and coherent manner was crucial. The less experienced testers primarily communicated with the developers, whereas the more experienced testers also communicated with product owners and support. In any case, testers were required to provide feedback on the work and errors of others. Therefore, effective communication skills are required for a professional tester.

*Organization skills:* To succeed as a tester, according to the interviewees, one must be structured; additionally, testers contributed to the management of the requirements.

*Creativity:* The interviewees highlighted the significance of creativity in testing. They were able to locate uncommon bugs due to their creativity. One of the interviewees highlighted the importance of creativity to the profession.
**Part 1: Summary**

*Curiosity:* The interviewees viewed curiosity as a crucial aspect of the testing role, as a way to encourage collaboration with others. One interviewee described how asking open-ended questions can aid a tester’s investigations: "I'm not familiar with that; can you tell me more about it?" The testers believed that stakeholders and project customers interpreted their interest as commitment and were receptive to greater collaboration.

*Adaptability:* The testers were required to perform a variety of tasks, and their adaptability to ever-changing work conditions was valued.

---

**Table 2. Comparison of views (employer, practitioner) on soft skills of testers**

<table>
<thead>
<tr>
<th>Skill category</th>
<th>Employers’ demands</th>
<th>Practitioners’ description of the skill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>“Excellent communication skills with team members and business contacts.”</td>
<td>“Communication is so personal sometimes; it is quite difficult to give feedback on an employee’s work, for example, because it is so personal and sensitive.”</td>
</tr>
<tr>
<td>Interpersonal</td>
<td>“Is able to interact with system developers, business analysts and others.”</td>
<td>“Somehow, they will have to intermediate the customer’s wishes with what is technically possible, even though the developers say that everything is possible. [...] So the testers need to find that balance. So, somehow, they have to listen to everyone.”</td>
</tr>
<tr>
<td>Analytical and problem-solving</td>
<td>“Demonstrated ability to analyze and solve technical issues.”</td>
<td>“The testers worked really hard to systematically map, find, and clarify the requirement specifications for us, and even found things that we overlooked. And, as a result, we also adjusted and improved our work processes.”</td>
</tr>
<tr>
<td>Organizational</td>
<td>“You must be well-organized with the ability to work to deadlines.”</td>
<td>“Even if one performs exploratory testing [...] If you’re just exploring without being systematic, then I don’t think you can retrace your steps. So, in my opinion, all testers must be structured.”</td>
</tr>
<tr>
<td>Innovation</td>
<td>“Experience working on unusually complicated problems and providing solutions that are highly creative and ingenious.”</td>
<td>“I’ve managed to find weird bugs by being creative, such as mid-way forced shutdowns and performing unusual process sequences. One has to test like that because the users are always creative.”</td>
</tr>
<tr>
<td>Openness</td>
<td>“Ability to work in a rapidly changing environment.”</td>
<td>“It can sometimes be as early as, after looking over my tasks and feeling ready to start, someone would put me on the shoulder and say that I have to do something else.”</td>
</tr>
<tr>
<td>Others</td>
<td>“Ability to establish and maintain a high level of customer trust and confidence.”</td>
<td>“Testers need to have imagination, a passion for the end users, and a passion for quality.”</td>
</tr>
</tbody>
</table>
Curiosity was identified by practitioners as a crucial skill for testers, despite the fact that it did not appear to be the explicit focus of hiring managers. Employers mentioned three categories of soft skills in their advertisements that practitioners did not mention: teamwork, independent work, and quick learning.

Comparing the perspectives of employers and employees on the same category of skills, it appears that employers emphasized the qualities that would enable testers to deliver under time constraints and in a rapidly changing work context, while practitioners emphasized the characteristics that would enable testers to discover software defects most efficiently; see Table 2 for examples. Employers emphasized skills related to independent work, rapid learning, and customer relations, whereas practitioners highlighted imagination, a passion for quality, and an focus on the end user.

5.2 RQ2: What are the essential hard skills of software testers, according to employers and practitioners?

Nearly all of the 400 advertisements analyzed indicated that employers were seeking highly skilled testers, requiring an average of ten testing skills and five technical skills, and a third of the advertisements requested at least one domain-specific skill (Table 3).

Table 3. The employers’ skill demand for testers

<table>
<thead>
<tr>
<th>Type of skill</th>
<th>Percentage of ads</th>
<th>Avg. skill/ad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Testing skills</td>
<td>97.7%</td>
<td>10.2</td>
</tr>
<tr>
<td>Technical skills</td>
<td>82.5%</td>
<td>5.4</td>
</tr>
<tr>
<td>Soft skills</td>
<td>64.2%</td>
<td>4.8</td>
</tr>
<tr>
<td>Domain specific skills</td>
<td>34.7%</td>
<td>1.7</td>
</tr>
</tbody>
</table>
The requests covered extensive areas of testing, including test management, test types, and particular testing tools; see Fig. 7. Employers required, on average, four process-related skills, including the ability to plan testing activities, design tests, implement and execute tests (manual, automated), and conduct closure activities; see Fig. 8.
Employers also requested, on average, three test management-related skills, including defining and following up on test-related metrics, estimations, risk analysis, reporting on the test progress, bug follow-up to ensure they are fixed, testing strategies, and testing methodologies, as well as management of test environments and test data. We discovered an average of two test-specific requirements per posting, such as load and stress, security, regression, and white-box testing.

Employers also requested, on average, familiarity with two specific testing tools and software testing on at least one level (unit, integration, system, acceptance) and in one to two specific system categories (web, mobile, desktop, large/ultra-large systems).
The most in-demand testing tools were execution tools such as Selenium, QTP, Cucumber, and SoapUI (30%), followed by test management tools such as Jira and HP Quality Center (22%). Less frequently requested were test performance tools like JMeter and LoadRunner (7%) and unit-testing tools like Junit and TestNG (2%).

Employers had a strong preference for testers with technical expertise. The majority of testing positions required experience with multiple facets of software engineering, including programming and software project management (see Fig. 9).

On average, testers were asked about three types of technical skills related to software engineering, such as programming experience with various technologies and frameworks, two specific programming languages - primarily SQL combined with Java, JavaScript, C#, and Python - and two aspects of software management, such as quality assurance, project management, and business requirements handling (Fig. 10). Candidates for testing positions were occasionally asked to demonstrate proficiency with deployment tools (6%), such as Jenkins, build tools (4%), such as Maven or TeamCity, and source-control systems (5%), such as GIT or TFS.

Domain-specific expertise was less in demand than technical expertise. The majority (14%) of the one-third of job postings requesting domain-specific skills related to financial services, banking systems, and other financially related fields, such as payment, e-commerce, and accounting. In addition to the telecom industry (5%), logistics (3%), and gaming software (2%), we discovered a vast array of other software-specific inquiries.
Table 4. Testing requirements for developers in hiring advertisements

<table>
<thead>
<tr>
<th>ID.</th>
<th>Skills in testing</th>
<th>Dev. ads</th>
<th>Examples from job ads for developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Testing</td>
<td>30%</td>
<td>“Participation in testing efforts”</td>
</tr>
<tr>
<td>D2</td>
<td>Code reviews</td>
<td>23%</td>
<td>“Take responsibility for quality assurance by performing regular code reviews”</td>
</tr>
<tr>
<td>D3</td>
<td>Test automation</td>
<td>13%</td>
<td>“Experience of Continuous Integration and test automation”</td>
</tr>
<tr>
<td>D4</td>
<td>Unit testing</td>
<td>11%</td>
<td>“Manage quality assurance by maintaining high code standards, design principles and source code analysis tools through to the use of continuous integration, automated unit testing, deployment automation and code coverage tools.”</td>
</tr>
<tr>
<td>D5</td>
<td>Debugging</td>
<td>9%</td>
<td>“[...] you will plan, design, analyze, develop, code, test, debug and document programming to satisfy business requirements [...]”</td>
</tr>
<tr>
<td>D6</td>
<td>Application testing</td>
<td>9%</td>
<td>“Test and debug the application across supported environments, ensuring delivery of quality product”</td>
</tr>
<tr>
<td>D7</td>
<td>Design reviews</td>
<td>7%</td>
<td>“You will participate in design reviews, providing input to the design recommendations.”</td>
</tr>
<tr>
<td>D8</td>
<td>Test-driven dev.</td>
<td>7%</td>
<td>“Create automated unit tests using a Test-Driven Development approach”</td>
</tr>
<tr>
<td>D9</td>
<td>Security testing</td>
<td>5%</td>
<td>“Experience in security analysis of web applications ranging from threat modeling to penetration testing”</td>
</tr>
<tr>
<td>D10</td>
<td>Coding standards</td>
<td>5%</td>
<td>“You’ll do all this while maintaining high standards, adhering to the team’s conventions and constantly improving code, even if it wasn’t yours to start with”</td>
</tr>
<tr>
<td>D11</td>
<td>Performance testing</td>
<td>5%</td>
<td>“Development, unit, system, UAT, performance testing and test automation”</td>
</tr>
</tbody>
</table>

Table 4 compares the skill test-related requirements for testers and developers. Only fifty-two percent (52%) of the advertisements asked developers about testing, with the majority having either vague requirements (D1, D6) or developer-specific activities, such as code reviews (D2), debugging (D5), or coding standards (D7) (D10).

While testers were asked for an average of ten test-related skills with specific requirements, testing requirements for developers were minimal and rather general. On average, only two test-related competencies were requested in those job postings asking for testing expertise from developers. One in eight advertised developer positions required test automation, and one in ten required unit testing.

The correlations between skill requirements were analyzed in order to identify patterns in the demand for competencies. The results of the study are accessible online (Florea and Stray, 2019). "Spearman coefficients, skill analysis at the highest
Part 1: Summary

level”). Spearman correlation coefficient was utilized because it reveals correlations between variables linked by any monotonic function, and Dancey and Reidy's interpretation of Spearman correlation coefficient in psychology (2007). There was a moderate correlation between the requirement for technical skills and the requirement for project management and domain-specific skills, which denotes that it is fairly common for employers to request all three areas of competencies.

The interviewees found it difficult to describe both the concrete hard skills required for a tester's job and their own hard skills. It was common for testers to describe their abilities as an accumulation of knowledge that grew with each new assignment. In addition, "learning by doing" was identified as the preferred method for skill acquisition. One interviewee stated, "The thing I enjoy most is getting my hands on the product and getting things done". Rather than a one-time effort, those interviewed described their skill acquisition as continuous learning from multiple sources.

5.3 RQ3: How is the software tester's role portrayed and carried out?

We discovered a disparity between employers' efforts in recruiting testers and IT professionals' limited access to information about the tester role. The majority of interviewees did not study software testing during their academic careers and were unaware of the existence of testing positions. A respondent summed it up as follows: "We were never told about this job. We had no idea that this existed". Some respondents supported presenting testing-related topics to students in faculties other than IT: "That would be very important. Whether banking, finances, or any economic specific [...]. Because there are many firms developing accounting software. And nobody in economics faculties knows they can take this path. I find it a gain and really useful to be presented a semester-long course with this knowledge".

The interviewees stated that they stumbled upon their first testing jobs by chance. Those without IT training read the job postings superficially, as their lack of experience prevented them from comprehending the role-specific details. They applied for the position in response to the ad's main requirements. As one respondent explained their situation: "You didn’t realize from the job description what the job assumed [...]. But on the other hand, if they would give reference to these complicated standards, I would probably look it up and would have not applied to it, as I would think the job was too complicated and it was not for me. So there are advantages and disadvantages. Leaving this gate open, it made me apply for this job".
According to practitioners, the hiring process focuses primarily on the technical skills of the tester. Those testers without an IT background (2) reported that, during the job application process, they were tested solely on their programming skills, which left them confused and disheartened. A respondent recalled: "I was tested at this company, and I was under the impression that I received a wrong test. Even now, I have no idea what happened there. I was left with the impression that they were searching for developers, and I just went to the wrong testing room". Nonetheless, some testers with an IT background (2) were pleased to demonstrate their programming skills during the hiring process and viewed such tests as essential for those seeking a tester's position: "If you want to hire on test automation, even if you want to hire a junior, you are interested in them having some technical concepts, and you want them to know at least what a class is".

In approximately half of the advertisements (54%), employers requested prior testing experience, averaging 3.4 years. Eight out of ten employers required a completed bachelor's degree, while one out of ten required a master's degree or higher. The majority of employers did not require specific testing certifications. Of the 14% of employers who did request them, the ISTQB Foundations Level Tester certification was the most desired. Application Lifecycle Management (ALM), information systems security-related (CISSP), and network penetration testing certifications were infrequently requested (1%).

The interviewees agreed that certifications provide credibility for their testing experience, which is particularly important for those without a technical educational background or those in consulting positions. Fourteen of our respondents were accredited in the fundamentals of software testing (ISTQB). In addition, a few interviewees held scrum master certifications (4), project management certifications (4), or advanced test manager certifications (2).

Two interviewees preferred on-site formal learning to acquire skills: "Sometimes I read and something distracts me. But when I am in the classroom, I assimilate information much better". However, the majority of testers (14) favored asynchronous and informal sources of education. Online, they sought primarily product specifications, test-related examples (17), and test technique video tutorials (8). Moreover, some respondents utilized online training providers (3) and books (2). Regarding the video format of presentation, our subjects preferred generic YouTube (6), Udemy (1), and Coursera (1) training videos because they were simple to follow and navigate.

Eighteen interviewees estimated that they gained knowledge through collaborative work; they spent considerable time working with others: "I think that in general, I spend about half of my time working with other colleagues. On many things". Another respondent stated, "Half of the time I am in discussion with one or another
on a certain task, or asking for help, or giving help”. Those with less experience tended to work primarily with developers, whereas those with more experience worked with other stakeholders too, such as product owners and support-specialists, because they believed these stakeholders had goals that could be overlooked otherwise. The interviewees had to have a substantial amount of knowledge for each task; therefore, they required time to acquire this information. Respondents emphasized that this procedure was time-consuming but necessary for assignment completion.

If the developers were overworked, it was challenging to request their time. One of our respondents with a non-technical profile reported experiencing intense stress when they were unable to reach the developers to obtain technical testing-related information, such as test data: “I hate going to databases with these SQL queries. I feel it is out of my reach somehow. And I have little control and it truly stresses me [...] even if we test a lot, but it does stresses me a lot, especially with the product I am working now, which is money coming in and out, salaries, etc. So I get very stresses by the thought that maybe I missed something that was important and maybe I was supposed to see. Maybe someone is not going to get the salary next month because of me.”

A major source of stress was identified as the limited time allotted for testing with short and strict deadlines: "And they tell you: you have three days to get in place the system tests and prepare the testing task. And there is this big thing that you don’t know how to approach. And I find this extremely stressful. And I would lay awake at night, thinking: how am I going to do this?"

An additional stressor for our interviewees was having unsuitable personal performance metrics, such as a minimum number of bugs to be discovered, or a defect report rejection rate of zero. These metrics impeded the tester's exploration of the system being tested, and such cases were inevitable in testing. Aslo, imposing a minimum number of bugs puts software testers in the mode of bug hunting. One respondent remembered, "there are companies in which they give the people as performance target to find a certain number of bugs. And then the people torture themselves and pulling out of their magic hat bugs. A tester is there to test, not to produce bugs”.

In general, employee feedback indicated that their employers were supportive of skill development (14). One interviewee described, "I think if I would ask more, I would get more. Support was not an issue. I am not sure how it’s like in other places”. However, four respondents indicated that the pace and direction of skill development depended heavily on the team leader: "You depend a lot on the
manager. I had team managers that had follow-up with me once a year. I also had more informal discussions every three months. And I liked it better that way”.

A common theme raised by interviewees was the importance of mastering testing terminology and, consequently, being able to communicate with their colleagues using the same terminology: "If you have a certification, you are not necessarily a good tester, but at least you have been through the same training with everyone else with this certification and you at least have been taught the same terminology. You might not use it all in your daily life. But you do know what you’re talking about and using the terminology correctly. So that makes it for fewer misunderstandings underway”.

Five of the non-IT-educated testers viewed their transition into the testing industry as a shock: "I had some idea about test automation, but I had no knowledge about the people in the back, testing all these reports and functionalities of this financial software. So it was a bit shocking for me changing the domains, zero prior knowledge, and the language on top". Three of the IT-educated testers viewed career transitions as any new task allocated to them: "I like being hands-on and learning new things, and I approach different industries like anything else. I just learn what I have to learn to do my job".

Based on their stated preferences, testers tend to develop a limited set of skills rather than acquiring broad testing competencies. As these preferences were shared by multiple interviewees, they were thematically analyzed and categorized into roles. Respondents preferred to act in their jobs according to a single role, avoiding the responsibilities of other roles.

As depicted in Figure 11 and Table 5, five roles for software testers have emerged: domain-specific tester, test automation specialist, test infrastructure specialist, user experience tester, and test manager. The testers portrayed each of these roles with a unique perspective on its value, issues it addressed, beneficiaries, and skill requirements.
Table 5. Description of the testing roles assumed by the software testers

<table>
<thead>
<tr>
<th>roleID</th>
<th>Description of the role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role 1</td>
<td><strong>Domain-specific tester</strong></td>
</tr>
<tr>
<td>Tasks preferred</td>
<td>Transforming the business requirements into system specifications; validating that the implementation follows the specification.</td>
</tr>
<tr>
<td>Tasks avoided</td>
<td>Writing automated test scripts; tasks related to the test infrastructure.</td>
</tr>
<tr>
<td>Main collaborators</td>
<td>Project stakeholders to elicit business requirements; developers to check the implementation of the system requirements in short iterations.</td>
</tr>
<tr>
<td>Perceived issues in testing</td>
<td>Lack of enough product owners allocated to the teams leads to poor-quality implementation of the business requirements.</td>
</tr>
</tbody>
</table>

**Role 2 Test automation specialist**

**Tasks preferred**
Creating automated test scripts and test suites and running them on test configurations; investigating the issues found; optimizing the automated tests for reuse.

**Tasks avoided**
Working on tasks related to test infrastructure.

**Main collaborators**
Software developers (investigating the failures reported by the test scripts); Role 1 testers (converting the system specifications into step-by-step test scenarios suited for automation).

**Perceived issues in testing**
A lack of resources allocated to automated test scripts, resulting in test suites not groomed and execution failures not followed up.

**Role 3 Test infrastructure specialist**

**Tasks preferred**
Building and maintaining tools and infrastructure for testing; creating complex test data for the system under test; developing and pilot-testing new testing tools.

**Tasks avoided**
Writing automated test scripts. Domain specific testing.

**Main collaborators**
Software architects (creating test data); operations engineers (setting up test infrastructure solutions).

**Perceived issues in testing**
A lack of dedicated resources to create and maintain up-to-date test infrastructure; outdated test environments (resulting in lack of testing and misleading results).

**Role 4 User experience tester**

**Tasks preferred**
Testing the usability of the system from the end-user’s perspective.

**Tasks avoided**
Automating test scripts; maintaining test infrastructure.

**Main collaborators**
UX engineers (to obtain detailed information on the design specifications for the system under test); end-users (to better reproduce the context of use of the software under test).

**Perceived issues in testing**
A lack of resources to test UX specifications; insufficient attention to end-users’ needs; making decisions regarding software release based only on the success of executed automated test scripts, while disregarding usability problems that are not covered by the tests.

**Role 5 Test manager**

**Tasks preferred**
Creating test strategies and test plans; setting priorities to testing activities; monitoring that the priorities are followed; enabling collaborations between testers and developers.

**Tasks avoided**
Beyond-basic test scripting; working on domain specific or infrastructure-related tasks.

**Main collaborators**
Project stakeholders (to agree on priorities in testing and resource allocation); other testers (regarding competence development, resource allocation, and other needs); managers (related to resources and competence needs); software developers; customer support.

**Perceived issues in testing**
Lack of commonly agreed testing prioritizations (leading to insignificant testing on par with business-critical testing); lack of clear responsibility for prioritizing test activities and resource allocations.

The domain-specific testers (Role 1) had domain-specific knowledge and acted as a liaison between the product owner and developers. This position entailed transforming business requirements into system specifications and monitoring their implementation. They evaluated and tested the implemented solutions with the
The Role 1 testers were willing to manually write step-by-step scenarios for the Role 2 testers to automate, but they declined to automate the scripts themselves.

*The test automation specialists (Role 2)* converted manual tests into automated scripts. In addition, they organized scripts into execution suites and oversaw their repeated execution. They followed up with developers to determine the cause of failing scripts and report any issues discovered. The scripts were routinely optimized for modification and reuse.

*The test infrastructure specialists (Role 3)* built and maintained the testing tools and infrastructure: typically, they created the test environments and kept them current with the most recent version of the system under test (SUT), tools and libraries, and test execution suites. Additionally, they provided data for the SUT. They created, pilot-tested, installed, and integrated new testing tools on a regular basis.

---

**Fig. 11 Main collaborators of the roles in software testing**

Software tester role

In Development/Engineering area: Role 2 (test automation specialist), Role 3 (test infrastructure specialist), Role 5 (test manager). In both Development/Engineering and Business areas: Role 1 (domain-specific tester), Role 4 (user experience tester).

Roles that software testers interact with

In Development/Engineering area: developers, architects, user experience engineers, deployment engineers, project managers. In the Business area of software development (Business area): end-users, customers. Note: sometimes, the customers are the end-users. In both areas of Development/Engineering and Business: product owners, customer support.

Collaboration reported by both roles (base and pointer of the arrow)

Collaboration reported by one role (at the base of the arrow)
Additionally, they could manage virtual testing machines and cloud-based testing services.

The user experience testers (Role 4) evaluated the UX guidelines' implementation and the user's interaction with the software under test. Those in this role focused on end-to-end scenarios that required the system to be utilized for several hours. They targeted issues that could result in a user's subpar performance on the job due to the use of inappropriate software.

Test managers (Role 5) created or assisted in the creation of test plans and test strategies, followed up on them, prioritized tests, verified testing coverage, and tracked testing-related metrics. Those in the role routinely facilitated the testers' work by assuming responsibilities for onboarding, training, and other resources, as

<table>
<thead>
<tr>
<th>Roles found in ad (count)</th>
<th>Ads asking for role count</th>
<th>Role 1</th>
<th>Role 2</th>
<th>Role 3</th>
<th>Role 4</th>
<th>Role 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>93 (23%)</td>
<td>45</td>
<td></td>
<td>33</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>111 (28%)</td>
<td>37</td>
<td>34</td>
<td>15</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>96 (24%)</td>
<td>48</td>
<td>13</td>
<td>11</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>62 (15%)</td>
<td>30</td>
<td>15</td>
<td>14</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>15 (4%)</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>400 (100%)</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>
well as facilitating collaboration and facilitation.

**Comparing the employer's requirements with the testers' job practice**

While testers preferred to perform a single role and justified their preference with the skill development and experience gained through task repetition, employers routinely demanded performance in multiple roles that frequently combined domain-specific and technically demanding competencies.

We conducted a qualitative analysis of the advertisements using the five roles as themes (see Fig. 4) to identify employers' role-related hiring requests. Approximately three-quarters of employers (71%) sought testers with a combination of domain-specific and technical responsibilities; see Table 6. Nearly half of employers requested that candidates play three or more roles.

The most requested combinations were domain-specific testers who would automate test scripts and work on the testing infrastructure (Role 1 & Role 2 & Role 3) and domain-specific testers who would automate test scripts, work on the testing infrastructure, and assume responsibilities in test management (Role 1 & Role 2 & Role 3 & Role 5). A quarter of the job postings required applicants to fulfill two roles, primarily requiring test automation specialists who would also build and maintain test tools and test environments (Role 2 & Role 3) or domain-savvy testers who would also automate business requirements into testing scripts (Role 2 & Role 3). (Role 1 & Role 2).

Among the one-fourth of employers requiring testers to perform multiple roles, the majority sought a test automation specialist (Role 2), followed by a domain-specific tester (Role 1). Much less demand existed for test infrastructure specialists, test managers, and user experience testers. The few employers (6%) who did not specify a subrole advertised either entry-level or manual testing positions. Small companies (50 employees), medium companies (50–249 employees), and large companies (>250 employees) requested an average of 1.9, 2.4, and 2.6 roles, respectively.

**6 Discussion**

This section summarizes the answers to the research questions and discusses research and practice implications.

**6.1 Software testing, a social and tactful job**

The answer to the research question RQ1, "What are the essential soft skills of software testers, according to employers and practitioners? " is presented below.
By analyzing the interviews, we discovered that the most frequently mentioned characteristics of a good tester were in the area of soft skills, indicating that social factors are of paramount importance to practitioners and play a central role in successful software testing. Despite the fact that soft skills contribute significantly to the success of software projects (John et al., 2005; Sánchez-Gordón et al., 2020), one-third of employers did not request them. This result indicates that some of the testers' employers should increase their awareness of the social skills required for the job and express their specific hiring needs.

The most notable of the new categories of in-demand soft skills is work ethics. Employers asked testers to demonstrate honesty and reliability in their interactions with customers and team members, as well as professionalism in their deliveries and a commitment to the customer. They were instructed to be proactive by accepting and completing tasks rather than waiting to be assigned work.

In addition, employers increased their expectations for responsibility-related skills, requiring testers to produce definitive work and grounded results. Nindel-Edwards and Steinke (2008) have previously emphasized that testers must complete testing with a user-centric mindset and must consider the consequences of defect release. This finding indicates that companies tend to prioritize ethical considerations in software development.

Other notable new requirements include the capacity to work under pressure, to be motivated, committed, detail-oriented, and focused on quality. This finding indicates pressure on the job market for software testers to perform more broadly and more quickly, a point raised by van Veenendaal as well (2020).

A surprising result was the disparity between employers and practitioners regarding certain soft skills for the job. It appears that employers and employees perceive soft skills differently. For instance, under the category of skills “communication”, employers required the aspect of the skill that would enhance a candidate's employability and positively impact the business. At the same time, the practitioners described the characteristics that enabled the testers to perform well. Moreover, they viewed communication as something personal, such as discussing someone's error in their work (see Table II). Therefore, when hiring testers, employers are encouraged to investigate the qualities described by practitioners as well.

Employers appeared to be more concerned with the soft skills that would enable testers to deliver work under pressure, learn quickly, and adapt, whereas practitioners appeared to be more concerned with the soft skills that enabled testers to find defects: being detail-oriented, inquisitive, and structured. The difference in perception is consistent with the findings of Lizama et al. (2019) on the employer
6 Discussion

side and the learning opportunities and significance of the work listed by the practitioners (Capretz., 2019).

Testers must communicate frequently with a variety of other roles, making testing a social occupation. Therefore, effective communication skills continue to be the trait most desired by employers. However, new trends and changes in the required level of soft skills were also discovered. According to an analysis of employer requirements, there appears to be a growing demand for teamwork, rapid learning, independence, openness, and adaptability. Due to the widespread adoption of Agile, which places teamwork at the center of software development (Lindsjørn et al., 2018), the level of demand for these skills is expected to remain high.

Universities of Computer Science and Information Systems can significantly contribute to the development and practice of students’ soft skills. We observe that regardless of the setting in which software testers operate, the demand for and value of soft skills remain constant. Universities should therefore train the skill types highlighted in our study. Categories such as analytical and problem-solving, organizational, individual-working, and work under pressure can be trained in disciplines requiring individual project delivery.

Universities could include specific group-delivery tasks that train students’ innovative, quick-learning, and collaborative skills. In addition, universities and colleges can incorporate into their courses real-world examples or even partnership with IT firms that develop soft skills such as work ethics, client focus, and grounded work results. Students would benefit from targeted exercises, which would allow them to practice their attention to detail, quality consciousness, and critical thinking. Nonetheless, institutions may encourage initiative and commitment among students.

6.2 Extensive hard skill requirements for the software testers

The answer to the research question RQ2, "What are the essential hard skills of software testers, according to employers and practitioners?" is presented below.

Employers required testers to be highly skilled and proficient in a wide range of test-related and technical competencies, as indicated by the findings. Due to the breadth of the demand, testers were not required to be experts in just a single testing activity, such as performance testing, test automation, or test management. Rather, they were expected to be well-versed in numerous testing facets and play an active role throughout the testing process.

Simultaneously, testers viewed their hard skills as a vast repository of information that grew with each task and was rooted in their explorations of the system under test. In accordance with Zhu and Zhang, the practitioners viewed exploratory testing as the most effective method for learning, testing, and discovering critical bugs.
Part 1: Summary

(2013). Several orders of magnitude more productive than scripted testing, exploratory tests were deemed to be a potent and efficient method of learning and assessment. Therefore, businesses are advised to allow the time and facilitate its implementation.

The most frequently requested hard skill by employers was the capacity to design tests. Stevenson and Wood (2017), who emphasized the significance of design for maintainability, provide additional evidence for the significance of the skill. The ability to execute tests automatically and manually ranked second in terms of the demand for hard skills. Both were in high demand, and the levels of demand were comparable. The essence of a testing job is test execution; however, the widespread demand for testing automation may be explained by the extensive use of development models that require regression testing. Therefore, a substantial portion of regression tests are automated (Collins et al., 2012).

Surprisingly few employers required testers to perform test-closure tasks, such as ensuring there were no ongoing bug fixes at the time of release, reviewing the deliverables list, archiving information and documentation, and transferring test-related data to other roles, such as customer support. As pointed out by Kassab et al. (2017), this low demand may be due to the fact that such activities are not regarded as a standard testing practice. However, employers are advised to prioritize these activities because they safeguard delivery quality.

One out of every ten employers requested that testers create and maintain test documentation and test data. Team members view these artifacts as necessary because they view documentation as a useful and important communication tool (Dias-Neto et al., 2017), even if outdated (Forward and Lethbridge, 2002), whereas the absence of test documentation creates problems regarding the comprehension of the testing process or the reproducibility of tests. Moreover, improper test data may be unsafe because they may violate regulations such as the general data protection regulations (GDPR) (Tankard, 2016), whereas oversimplified test data may conceal serious performance-related product issues. Therefore, the absence of such requests indicates a possible deficiency in the employer's requirements for testers.

By comparing the requirements for software testers to those for developers, we discovered that only fifty percent of employers require developers to possess any testing skills (slightly over two, on average, in the job adverts asking the developers for testing competencies). This indicates that employers expect testers to assume the majority of testing responsibilities and tasks. Causevic et al. (2001) confirmed the lack of testing expertise among developers. The result may be explained by the perception that a developer's time spent testing is not optimally utilized (Beller et al., 2015). As Baldassarre et al. (2022) noted, the reason for low unit testing requests
may be that the more experienced developers are with unit testing, the more negative affective reactions they will have toward TDD.

### 6.3 Selective role practice of the software testers

The answer to the research question RQ3, "How is the software tester's role portrayed and carried out?" is presented below.

Three out of four employers required a bachelor’s degree in IT from testers, indicating a preference for IT-trained testers. Kettunen et al. (2010) supported the conclusion that technical expertise is more important than domain-specific experience, given that the majority of employers requested a wide range of technical skills, such as programming and software deployment, from job candidates. However, half of the testers interviewed lacked an IT academic background, indicating a potential mismatch between employer requirements and the testers’ market. However, the fact that software testing professionals hail from a variety of educational backgrounds indicates diversity in the landscape of software testing talent, which is a crucial aspect of creating high-quality software (Kaner et al., 1999).

The finding that hiring ads with too many hard skill requirements were intimidating, particularly for domain experts, suggests that hiring companies should, whenever possible, broaden their horizons in the advertisements so that a larger applicant pool is encouraged to apply for software testing jobs, a finding that Kaner (1999) also deemed advantageous. Despite the tendency of employers to post advertisements for testers with a technical focus, many accomplished testers have backgrounds and domain experience that are vastly diverse. They gained the most knowledge through informal training, work assignments, and teamwork.

Working with other roles, particularly developers, was crucial for our respondents to mitigate some of the issues resulting from the gaps in requirements (Bjarnason et al., 2011) and their interpretation, such as different approaches to problem-solving or the prioritization of testing tasks. In addition, the fact that software testers rarely collaborate with other software testers suggests that test-skill acquisition is typically an individual effort as opposed to an intra-role group effort. Collaboration, whether online or in-person, is advantageous (Lindsjørn et al., 2021); consequently, the findings are applicable to both co-located and distributed teams. Nevertheless, practitioners identified exploratory testing as the most effective method of learning—the preferred testing approach for large complex systems (Eidenbenz, 2021).

Certifications were a popular way for testers to supplement their formal education. However, certification was less in demand among employers than academic degrees. A small number of companies required certification in testing, security, penetration
Part 1: Summary

testing, or application lifecycle management. Garousi and Zhi's (2013) study indicates that there will likely be an increase in the demand for certifications related to training and testing. Given that many employers did not request specific seniority, seniority-related requests are most likely not a priority. The result validated the findings of Poon et al. (2011), which concluded that experienced testers were not always superior to inexperienced testers.

Currently, non-academic certification tracks such as usability, automation, product ownership, and test management provide a portion of the theoretical background for certain software testing roles. However, neither academic education providers nor certification agencies provide complete support for these career paths (Crespo et al., 2010). The majority of testers interviewed felt that an academic focus on software testing would be beneficial, even in non-IT academic institutions, as this type of information could introduce domain experts to the software testing role.

The result that testers received adequate support at the company level to gain or grow skills indicated that employers had a general interest in the skill development of software testing professionals and allocated budgets for it, and that the majority of employers allowed testers to choose their educational paths rather than influencing the training plan. However, the majority of the interviewed testers identified the time allocated for learning as a problem. Unappreciated by certain respondents was the use of bug-related metrics in performance evaluations.

The testers selectively developed their skills in one of the five roles in order to achieve and maintain proficiency in it, despite the fact that the employers required a wide range of abilities. The testers described the software-testing role in terms of distinct skill ranges based on their stated preferences in acquiring skills and their choice of work tasks: domain-specific (Role 1), test scripting (Role 2), test infrastructure activities (Role 3), user experience (Role 4) and test management (Role 5). Each role required a unique set of complexities and distinct traits in order to acquire proficiency in it. In addition, the testers highlighted the day-to-day responsibilities that each role entailed.

Researchers de Ste Croix and Easton (2008) and Paasivaara et al. (2009) identified a deficiency in the availability of POs on development teams. This deficiency was compensated for by domain-savvy testers (2012). Even though they contributed significantly to software engineering, many domain-savvy testers felt neglected during the hiring process and in their daily work. Role 1 testers were hesitant to apply for the position because many ads requiring domain expertise also included extensive technical requirements. They felt intimidated by the technical requests' specifics and were reluctant to apply. The testers with a non-IT academic background were pleased to contribute to the development of the project in the areas
of requirements definition, implementation, and verification; however, they were extremely uneasy with responsibilities requiring technical expertise, as their lack of skills could jeopardize the project's quality or prevent its completion.

The test automation specialists viewed manual and automated testing as two distinct processes, as opposed to two distinct ways to accomplish the same task, and they were inclined to acquire test scripting skills. Ramler and Wolfmaier (2008) and Karhu et al. (2009) support the need for specific skills to automate test scripts, namely the ability to write test scenarios that can be reused. In addition, evidence suggests that constant human intervention is necessary for executing and maintaining a test suite, at the very least to validate execution results and repair broken test scripts (Bach, 1999) or to investigate test execution failures (Kaner et al., 2008). The determination of some testers to perform automated testing, coupled with their frustration at being unable to do so, highlighted the problems with the architecture of automated testware in many organizations. This issue was also identified as a central concern in automation (Graham and Fewster, 2012); companies that conduct automated testing must address it.

According to Rafi et al. (2012), the test infrastructure was the most common obstacle in the automation process, and testers with a preference for test infrastructure were willing to overcome it. Despite the fact that executing automated test scripts reduced test execution costs, Karhu et al. (2009) noted that new costs arose from implementing and maintaining the infrastructure for automated tests, which had to be accounted for separately. This finding validates the scope of infrastructure-related work duties. Chen et al. (2001) also discovered that test infrastructure maintenance was expensive: in small projects, because it was frequently constructed and maintained manually, and in large projects, because of the development and execution of environment-management scripts.

The user experience testers made up for the lack of resources that could have verified the UX guidelines' implementation. This gap was identified by Larusdottir et al. (2010) when they demonstrated that UX practitioners desired to focus on usability testing but were generally unable to do so due to the short iterations and small changes implied by iterative-incremental development.

Ramler et al. (2006) signaled the need for dedicated testers to coordinate the test activities that the interviewees argued for, noting that different tests tended to treat code with equal importance when, in practice, decisions had to be made to maximize the return on investment in testing. Although this responsibility belonged to the project manager at the project level, it was unclear who was responsible for it during testing.
Part 1: Summary

The size of the software to be tested and the number and complexity of integrations present testing challenges. Determining test environments, performing end-to-end testing as opposed to using stubs, and continuously implementing and executing automated regression tests necessitates a significant investment of effort. Experts in the business domain are required for testing, with a focus on enabling value-chains testing and identifying the potential consequences of discovered issues (Bjerke-Gulstuen & Cruze, 2020).

6.4 Implications for practice

The purpose of this thesis is to provide support to employers, testers, and tertiary education in order to assist IT professionals in pursuing careers as testers.

Given that many testers had a domain-specific background, when hiring new test personnel, employers may consider the number of technical details in advertisements targeting domain experts so that these professionals can respond. It is advisable to check the number and breadth of requests for the advertised positions, as well as whether the advertised role requires simultaneously in-depth IT and non-IT skills, given that the majority of employers advertise for test positions with multiple responsibilities.

Moreover, given that fifty percent of employers required testers to perform three or more roles on a regular basis, employers are encouraged to state explicitly which qualifications are primarily needed for the advertised testing position. Given that skill mismatch was one of the barriers for testers and that candidates were hesitant to apply for positions with numerous broad requirements, employers could use the information in the thesis to evaluate the workload for the job to be performed and formulate role requirements by incorporating – whenever possible – role-based testing responsibilities.

The generic test-related requirements for developers require a more precise specification of the developer's test-related tasks and a clear division of testing responsibilities among software roles.

The mapping of the skill requirements may be of interest to testers because it can provide an overview of the current requirements for the role and serve as a guide when they seek to update their skill portfolio. The testers could use the findings to familiarize themselves with in-demand skills outside of their toolbox and practice them via training or individual practice. In addition, testers can use the five roles in testing to support the development of specific skills. Nonetheless, the findings could benefit software testing professionals, whether or not they have a technical background, by establishing a framework for the role's soft skill requirements.
6.5 Implications for research and further work

This thesis contributes to the broader field of skills research with the profile of a software tester that is currently sought by employers. The requirements for soft skills and the demand trends were presented. Also provided are the type and level of demand for test-related, technical, and domain-specific skills. To obtain this data, two taxonomies were developed and implemented to classify testing skills and technical skills requirements.

The findings can be used to enhance comprehension of the evolution and fluctuation of skill demand. In addition, the results can be used to increase knowledge regarding the five specializations of the software tester role. Further study could investigate the impact of assigning testing tasks taking the practitioner's role preferences into account.

Recent studies have found that soft skills are essential for developing sustainable software solutions to address complex issues (Caeiro-Rodriguez et al., 2021). Therefore, future research could investigate how testers function in agile projects and the various tasks and sub-roles they assume.

Which soft skills will be crucial for testers in the future? would be an interesting topic for research. How will requirements for soft skills change? Do the new soft skills required of developers and software testers correspond? Future research could also investigate why some companies do not list soft skills in software tester job postings. It is necessary to conduct research to better comprehend how the competencies of software testers change over time, as well as whether and how testers change their preferences for roles and sub-roles.

In addition, research should examine whether additional test-related requirements are imposed during the hiring process during a subsequent recruiting phase, and whether testing skills are required as part of developers' work after hiring. Given the current hiring practices for testers, one could also investigate whether employers should take a different approach to evaluating the testing skills of software testers. The influence of company size, team size, and industry sector on the role of software tester would also be an intriguing area of study.

Future research could also investigate how other facets of the software tester role affect testers' motivation and preferred tasks. Examples of such factors may include gender, work experience, testing experience (methodology, automation), software and organizational settings (product firms, service firms, testing-only firms), and geographic locations (Scandinavia versus other more hierarchical countries). One could also consider testers working on globally distributed projects. In addition, it
might be interesting to investigate sub-roles, career changes in testing, and motivation through a longitudinal study.

In recent years, there has been an increase in interest in software test automation, and test automation has been identified as a promising improvement area for testing activities (World Quality Report, Capgemini Consulting, HP & Sogeti, 2015; ISTQB, 2016). Therefore, future research should investigate the effects of software test automation and related tools on software testing as a profession and testing employees' quality standards. Regarding software testing tools, however, additional research is required on the evaluation and selection of such tools.

Since software testing has been found to be an unattractive career path (Waychal & Capretz, 2016; Weyuker et al., 2000), it is essential to further investigate how to attract and retain software testing professionals.

Asplund (2019) found that contextual knowledge and technical ability (e.g., programming skills) affect the efficiency of exploratory testing. The findings of this thesis highlight the importance of contextual knowledge and technical ability (e.g., programming skills) in software test automation.

According to the results of this study, it would be beneficial to examine the perspectives of practitioners in greater depth. Given the diverse academic backgrounds of software testers, and the fact that many do not possess STEM degrees, research may shed light on career paths and task assignment for non-technical testers.

Lastly, the research community could investigate the effect of various role-assignment models and examine how they can be implemented in the industry, taking into account various team sizes, project durations, and other resource allocation constraints.

7 Limitations

As the requirements for testers in this dissertation were collected from job advertisements, it would be difficult to generalize the findings to any industrial setting, such as safety-critical or industry-regulated software. In addition, the requirements for testers should not be equated with the entire testing process; other testing skills and competencies may fall under the purview of other roles, such as operations personnel or customer support.

Given the availability of prior studies, we could analyze trends in the demand for soft skills. However, it was not possible to conduct a similar analysis of trends in
employers' demand for hard skills due to the paucity of prior research in this area. Consequently, this thesis reflects the current industrial demand for hard skills.

In comparing and contrasting the test-related requirements for developers and those for testers, the sample size for developers (100 ads) was smaller than the sample size for testers (400 ads). This difference is due to the fact that we wanted to make a rough comparison between the test requirements for testers and the test-related work distribution between these two roles. We utilized weighted averages to solve this problem.

The job postings were researched online, and the search engines' inherent logic was utilized to select test-related positions. This strategy may have prevented search engine filters from identifying certain test-related job postings. Because the majority of job openings in the software industry are published online, we decided not to look at newspaper job ads. This option may have overlooked a market segment that could only find candidates through printed job advertisements. However, we made this decision based on the assumption that electronic job postings provide employers with a greater opportunity to detail their requirements than printed ads.

The results of a qualitative analysis of testers from a variety of industries and companies provide a comprehensive overview of the practice of the role. However, the role-related results may not apply to testers required to adhere to strict procedures when testing mission-critical systems or other specialized software.

The interviews did not cover as many industries and geographical locations as the job advertisements. The interviewees were located in four countries but had eleven-country work experience. The fact that the majority of employees were hired in Norway may have had an effect on the outcomes, given that Scandinavian countries possess unique characteristics such as a flat organizational structure.

The job advertisements we analyzed spanned a broader range of industries and locations than the practitioners we interviewed. However, by conducting a qualitative analysis of testers from a variety of industries, employers, and countries, we obtained a comprehensive understanding of the role's practice. Still, our findings may not be applicable to practitioners who are required to adhere to strict procedures when testing specialized software, such as mission-critical systems.

We relied on human judgment to evaluate the interview data and content of the advertisements. In order to reduce the possibility of inconsistent or overly subjective analysis, we achieved uniformity in the coding and annotations by having one researcher conduct the entire coding and theme construction while a second researcher verified the results. To mitigate this limitation, we shared preliminary results with all of our participants. We requested feedback on any incorrect or possibly missing information, as well as consideration of any additional input. Ten
of the nineteen interviewees who were contacted responded to our request and reviewed the preliminary results. We received no objections to the analysis's findings.

8 Concluding remarks

This thesis examined the hard and soft skills that employers seek in software testers and compared them to the role's practice in order to gain a better understanding of how the software tester role is portrayed and executed. As a result, the results of the study revealed the magnitude and breadth of skills required of testers, as well as testers' perspectives on the job, skill development, learning practices, and primary collaborators.

Two-thirds of employers requested candidates with soft skills. Communication skills, analytical reasoning, problem-solving, teamwork, and the ability to work independently were requested most frequently. Peers viewed proficient testers as those who had a comprehensive understanding of the project, could communicate effectively, paid close attention to details, were organized, creative, inquisitive, and flexible. Although employers focused primarily on the soft skills that would enable testers to deliver under pressure, learn quickly, and adapt quickly, practitioners described the soft skills that enabled testers to find bugs: attention to detail, curiosity, and organization.

Employers requested numerous hard skills, with an average of ten test-related skills, five technical skills, and one-third requiring domain-specific experience. The testers were specifically tasked with test planning and design, automated testing, non-functional testing, managing test environments, programming, and mastering a variety of development and testing-related tools. In contrast, employers rarely required developers to participate in testing, and testing requirements were generic.

Those with a college degree in information technology were the most in-demand testers. However, about half of the testers we interviewed have diverse educational backgrounds outside of IT. As a result, while performing well in functional and domain-specific tasks, many felt stressed and out of control when assigned technical tasks, such as those involving scripting and test infrastructure. Similarly, technically inclined testers preferred not to interact with the business or management side of testing and were to concentrate on one set of tasks, either test scripting or test infrastructure-related. All of the testers favored task-based and hands-on methods of skill acquisition. The majority of the testers' education came from non-formal sources, and they viewed their skills as a pile of knowledge that grew with each new assignment. They relied on close collaboration with other roles, such as developers and product owners, to fulfill their responsibilities.
Five roles in software testing have emerged from a thematic analysis of job practice: domain-specific tester, test automation specialist, test infrastructure specialist, user experience tester, and test manager. The testers exhibited a propensity to play a single role and sought to maintain proficiency in it. They supported their decision with experience gained through task repetition, efficient use of time and effort, and quality control over the work results. Employers frequently combined technical and domain-specific testing when requesting testers to perform multiple of these five roles concurrently, as determined by a second analysis of job advertisements. About half of the employers required testers to perform three or more roles on a regular basis. The demands did not specify employer-defined testing roles.

By exposing the software testers' jobs and contrasting the employers' demands with the testers' ability to perform them, the primary contribution of this thesis is to provide insight into the software tester role that may aid in assigning test-related work in a manner that benefits both employees and employers. Future research should investigate the effect of allocating role-aware testing work to various team sizes and industry sectors to expand upon these findings.

9 References


Creating Interdisciplinary Skills for Complex Global Environments, IEEE. DOI: 10.1109/TEE.2010.5508833


Clarke, M. (2016), "Addressing the soft skills crisis", Strategic HR Review, Vol. 15 No. 3, pp. 137-139. DOI: 10.1109/SHR-03-2016-0026


Part 1: Summary


ISTQB https://www.istqb.org/, Retrieved 2022


Nindel-Edwards, J. and Steinke, G. (2008). "Ethical issues in the software quality assurance function." Communications of the IIMA: Vol. 8 : Iss. 1, Article 6


75
Part 2: Collection of papers

Paper I: Software tester, we want to hire you! An analysis of the demand for soft skills

Raluca Florea\(^1\) and Viktoria Stray\(^1\)
1 University of Oslo, Gaustadalléen 23 B, 0373 Oslo, Norway
{ralucamf, stray}@ifi.uio.no

Abstract

One important discussion in the software development field is related to the skills that people need to have to build successful software products. This debate is generated on one hand by a large number of failures and delays of software projects. On the other hand, the debate is triggered by the need to build even better-quality software in a rapidly changing world. We will examine to which extent soft skills are relevant when hiring software testers and if there are any specific skills required for agile testers.

We analyzed 400 job advertisements for testers from 33 countries, out of which 64% ask for soft skills. Of the advertisements asking for soft skills, there is, on average, a request for 5 soft skills, 11 testing skills, and 5 technical skills. Only 30% of the companies ask explicitly for agile testers. However, our analysis shows no notable differences in skill demands for agile testers and the rest.

Software companies want to hire testers who can communicate well and have analytical and problem-solving skills. There is a significant increase in the need for openness and adaptability, independent-working and team-playing since 2012. In addition, there are new categories of soft skills identified, such as having work ethics, customer-focus and the ability to work under pressure.

Keywords: soft skills, competency requirements, software tester competence, software testing, agile software development, industrial needs

1 Introduction

Software testing is a complex activity that implies mastering both technical and soft skills. To be a productive software tester, one needs to understand business requirements from customers and to communicate them to the developers. Testers need to be organized, efficient and able to prioritize their work. Furthermore, they have to bear the pressure of finishing their job as soon as possible, so that the product can be released [1]. It is essential that they learn fast and master many kinds
of responsibilities [2]. Testers need to be especially flexible because they face stress [3] and changes [4] throughout the development process.

It may also be that testers need soft skills additional to the ones required for developers or managers [5], [6]. Because of the nature of their job combining different domains and perspectives, it may be that user focus [7] and critical thinking [8] have to be traits of efficient software testers. Moreover, the user-centered design and agile development are already common practices in companies, and the results look promising. But there are hinder to these practices such communication breakdowns or the lack of acknowledgement of user involvement [9], issues deeply connected to soft skills.

Rivera-Ibarra et al. [10] found that the quality and innovation of software products strongly depends on the knowledge, abilities, and talent of all the people developing software. Technical and hard skills have been a long-time focus point for research in the software development field [11]. Soft skills, human factors, and intrinsic motivation have recently begun to gain attention, but with a focus on the software developer role [12]. However, the role of software tester has not been given the same attention. It can constitute a drawback since building software is a teamwork and essentially a human activity [13] shaped by the human skills of all contributors into bringing software to live. Systematic literature reviews in software testing show relevant testing areas or methods of testing [14], but we did not find information about the soft skills in the testing world. Other research has looked at soft skills depending on various phases of development, from requirements engineering to design, implementation and testing [15]. But this approach does not look at testing as an on-going activity, involved in all phases of developing software [16]. Nor does it comprise the role of tester as a sum of all these activities [17].

Soft skills are defined by Lippman et al.[18] as: “The competencies, behaviors, attitudes, and personal qualities that enable people to effectively navigate their environment, work well with others, perform well, and achieve their goals.” It refers to a combination of people skills, social skills, character traits and attitudes and complement other skills such as technical and academic skills [18]. We wanted to investigate: what do companies look for in software testers? What are the soft skills they ask for? How do these needs evolve? What is specifically required of agile testers? In this study, we aim to answer the following research questions:

- **RQ1**: What is the trend for soft skills requirements for testers?
- **RQ2**: Are there specific soft skill requirements for testers in agile projects?

To answer the first research question, we use a categorization of soft skills proposed in a study by Ahmed et al. [19], where the authors analyzed 500 job advertisements in IT positions (developers, designers, system analysts and software testers). By comparing with the result of the analysis specifically for software testers from 2012[19], we were able to look at the skills requirement trend in the last 5 years. Moreover, to answer the second research questions, we analyze specifically
the ads mentioning agile methods. Additionally, we make a preliminary analysis of job advertisements not asking for any soft skills.

This paper is structured as follows: Section 2 discusses the way we have collected and analyzed data. Section 3 presents the results of our findings. We discuss and interpret the results in Section 4 and present the limitations of our study in Section 5. In Section 6 we present implications and in Section 7 we draw the conclusion and discuss future work.

2 Data collection and analysis

We collected job advertisements from 33 countries on five continents. The majority of the ads were collected from the USA, Canada and Norway, see Table 1 for the details. We chose to use online job-search engines to collect the raw job advertisements. We decided to use such tools instead of going to specific hiring companies because we consider the search engines to be an efficient way of including in our analysis a large number of hiring companies, a great diversity of companies and large visibility to job-seekers. We investigated which were the most significant job-search engines by two dimensions: the number of users and the number of jobs posted. According to commercial web traffic data and analytics services provided by Alexa(Amazon)\(^1\) and SymilarWeb\(^2\), we chose the five most popular job-search engines, namely Indeed.com, Monster.com, GlassDoor.com, CareerBuilder.com, and SimplyHired.com.

To obtain a 95% confidence level with a confidence interval of +5% we needed a minimum of 384 job ads [20]. We thus decided to study 400 job ads. We only selected the jobs that referred to the role of software testers. We have included therefore jobs such as testers, QAs, technical testers, usability testers, performance testers, game testers and financial-system testers. We have not considered jobs referring to other roles within a software development team, such as developers, architects, technical writers, or UX designers.

We collected job ads posted in multiple national languages because we consider it to be relevant to include countries that are important actors in the software development industry, whose language is not English. We gathered job ads posted in 20 different languages. We collected 226 job ads that were posted directly in English and 174 job ads that we translated into English. To make sure we translated the advertisements correctly, we used two independent online translation tools from Google\(^3\) and DeepL Translator\(^4\), respectively Etranslator\(^5\), to translate and

\(1\) https://www.alexa.com/
\(2\) https://www.similarweb.com/
\(3\) https://translate.google.com/
\(4\) https://www.deepl.com/translator/
to cross-check the coherence of the translations. We included only the job ads where the results of translations were the same. Using in parallel translation tools and comparing the results worked well because most of the job advertisements were posted in plain language, using standard terms. However, we still triple-checked with a fluent software professional or native speaker the translations to English from French, Italian, Spanish, German, Hindi, Vietnamese and all Scandinavian languages and the translation results provided by the tools were confirmed.

As a last point, it is worth mentioning that the job advertisements were collected from both in-house software developers, as well as consultancy companies. Both the public sector and private sectors were represented. For example, Amazon, Norges Bank, Expedia, Nasdaq, Texas Instruments, Verizon, Motorola Solutions, Fujitsu, VISA, IBM, Nokia, New South Wales Government, National Bank of Canada, Accenture, Sogeti and Sopra Steria.

Table 1. Job advertisements collected from each country

<table>
<thead>
<tr>
<th>Country</th>
<th>No. of ads</th>
<th>% of the total ads</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>96</td>
<td>24,0 %</td>
</tr>
<tr>
<td>Canada</td>
<td>65</td>
<td>16,3 %</td>
</tr>
<tr>
<td>Norway</td>
<td>22</td>
<td>5,5 %</td>
</tr>
<tr>
<td>UK</td>
<td>20</td>
<td>5,0 %</td>
</tr>
<tr>
<td>Argentina</td>
<td>17</td>
<td>4,3 %</td>
</tr>
<tr>
<td>France</td>
<td>17</td>
<td>4,3 %</td>
</tr>
<tr>
<td>Mexico</td>
<td>15</td>
<td>3,8 %</td>
</tr>
<tr>
<td>South Africa</td>
<td>14</td>
<td>3,5 %</td>
</tr>
<tr>
<td>China</td>
<td>14</td>
<td>3,5 %</td>
</tr>
<tr>
<td>Vietnam</td>
<td>13</td>
<td>3,3 %</td>
</tr>
<tr>
<td>Greece</td>
<td>13</td>
<td>3,3 %</td>
</tr>
<tr>
<td>India</td>
<td>12</td>
<td>3,0 %</td>
</tr>
<tr>
<td>Sweden</td>
<td>10</td>
<td>2,5 %</td>
</tr>
<tr>
<td>Portugal</td>
<td>10</td>
<td>2,5 %</td>
</tr>
<tr>
<td>Australia</td>
<td>10</td>
<td>2,5 %</td>
</tr>
<tr>
<td>Spain</td>
<td>9</td>
<td>2,3 %</td>
</tr>
<tr>
<td>Italy</td>
<td>8</td>
<td>2,0 %</td>
</tr>
<tr>
<td>Germany</td>
<td>8</td>
<td>2,0 %</td>
</tr>
<tr>
<td>Other countries</td>
<td>27</td>
<td>6,8 %</td>
</tr>
<tr>
<td><strong>400</strong></td>
<td><strong>100,0 %</strong></td>
<td></td>
</tr>
</tbody>
</table>

Coding of soft skills

In this paper, we examine which categories of soft skills are in most demand from employers. We determine which categories of soft skills are most popular for tester roles, compare them with the existing studies and interpret the findings.

5 http://www.etranslator.ro/
We chose to manually analyze the data because the ads have different structures: some of the job-search engines allow job advertisers to post in their specific format, but also various job-search engines use different formats for job ads. Last but not least, not all advertisers have the same understanding of the information that has to be filled in an ad. Therefore, we found soft skills in the sections dedicated to requirements, job attributes, duties. We went manually through each of the job ads and looked for soft skills requirements. We copied the content of the ads into the following categories:

- Country
- Job title
- Job description
- Responsibilities and tasks
- Job requirements
- Education needed
- Other certification needed
- Minimum prior experience required
- Nice to have

### Table 2. Definition of the soft skills categories

<table>
<thead>
<tr>
<th>Skill category</th>
<th>Definition based on [19]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication skills</td>
<td>The ability to convey information so that it is well received and understood.</td>
</tr>
<tr>
<td>Interpersonal skills</td>
<td>The ability to deal with other people through social communication and interactions.</td>
</tr>
<tr>
<td>Analytical and problem-solving skills</td>
<td>The ability to understand, articulate and solve complex problems and make sensible decisions based on available information.</td>
</tr>
<tr>
<td>Team player</td>
<td>The ability to work effectively in a team environment and contribute toward the desired goal.</td>
</tr>
<tr>
<td>Organizational skills</td>
<td>The ability to efficiently manage various tasks and to remain on schedule without wasting resources.</td>
</tr>
<tr>
<td>Fast learner</td>
<td>The ability to learn new concepts, methodologies, and technologies in a comparatively short timeframe.</td>
</tr>
<tr>
<td>Ability to work independently</td>
<td>The ability to carry out tasks with minimal supervision.</td>
</tr>
<tr>
<td>Innovative</td>
<td>The ability to come up with new and creative solutions.</td>
</tr>
<tr>
<td>Open and adaptable to changes</td>
<td>The ability to accept and adapt to changes when carrying out a task without showing resistance.</td>
</tr>
<tr>
<td>Others</td>
<td>Soft skills that do not fit any of the above categories.</td>
</tr>
</tbody>
</table>

To map the soft skills, we used the categories defined in an earlier study of job advertisements in software development [19], as our coding scheme, see Table 2. Moreover, we added the tenth category: “Other soft skills”, where we coded soft
skills that did not fit any of the other categories, for example, “good sense of humor”, “multitasking”, “ability to work under pressure” and “customer focus”. We considered both the soft skills that were required imperatively (mandatory requirements) and the soft skills that were considered a plus for getting hired (nice to have). We did not distinguish between different strengths of the same skills. For instance, “strong communication skills” and “excellent communication skills”.

3 Results

Of the 400 ads, 257 ask for soft skills (64,2%). We identified in all ads a total of 1,218 of soft skills, which leads us to an average of 4,73 soft skills per advertisement that demands soft skills. In comparison, the same ads ask for an average of 11,3 testing skills, 5,27 technical skills, and 0,7 domain specific skills. Table 3 shows examples of soft skills requirements asked for in the ads.

Table 3. Examples of categorized soft skills

<table>
<thead>
<tr>
<th>Category name</th>
<th>Extracts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication skills</td>
<td>“Excellent communication skills with team members and business contacts”.</td>
</tr>
<tr>
<td>Interpersonal skills</td>
<td>“Is able to interact with system developers, business analysts and others”.</td>
</tr>
<tr>
<td>Analytical and problem-solving skills</td>
<td>“Demonstrated ability to analyze and solve technical issues”.</td>
</tr>
<tr>
<td>Team player</td>
<td>“Values teamwork”.</td>
</tr>
<tr>
<td>Organizational skills</td>
<td>“You must be well-organized with the ability to work to deadlines”.</td>
</tr>
<tr>
<td>Fast learner</td>
<td>“A passion for learning and testing”.</td>
</tr>
<tr>
<td>Ability to work independently</td>
<td>“Must be able to work with minimal or no supervision on extra-large and multiple concurrent projects and coordinate the work of others in this environment”.</td>
</tr>
<tr>
<td>Innovative</td>
<td>“An ability to think creatively”.</td>
</tr>
<tr>
<td>Open and adaptable to changes</td>
<td>“Ability to work in a rapidly changing environment”.</td>
</tr>
<tr>
<td>Others</td>
<td>“Customer-service orientation”.</td>
</tr>
</tbody>
</table>
In order to analyse the trend, we use [19] as a proxy for the skills demands in 2012. Fig. 1 shows the results of comparing the soft skills demands in [19] and all the ads in our study. Focusing on the ranking, in both studies the most important category of skills is communication. Second, comes analytical and problem-solving skills. We see similar results for four types of skills: interpersonal, analytical and problem-solving, organizational and innovation: in [19] and in our research, they are demanded in approximately the same measure. However, the demand for communication skills has decreased by 19%. The need for team-playing skills have increased by 14%, fast-learning skills have increased by 18%, independent-working skills by 23% and openness and adaptability skills have increased by a spectacular 25%.

The category named “Others” contains 121 ads and 161 skills that did not fit into the predefined categories in [19], see Table 4. When we analyzed these skills, we found that the first and most important was having work ethics. Here we included skills such as integrity, trustworthiness, ethical behavior, honesty, respectability, credibility. The other large categories of other skills were customer focus, proactivity and responsibility.

Out of the 400 analyzed jobs, 120 specify that the role is for an agile working environment. That is, there is mentioned “agile, Scrum, Kanban or Lean” in the advertisement. To see if there were particular requirements for agile software testers, we analyzed the 120 ads in more detail. We found that 91 ads ask for a total
of 429 soft skills, which leads to an average of 4.71 soft skills per job ad. There are 64.2% of all ads asking soft skills and 75.8% agile ads asking for soft skills. The average number of skills is, however, similar. Referring to the “Others” category, most ads ask for work ethics (8%), customer focus (7%) and be proactive (5%).

The analysis shows that 257 ads out of 400 are asking for soft skills. The percentage is good (64.2%), in the sense that more than half of the job advertisers recognize the need for soft skills from their testers and verbalize the need in their job advertisements. However, 144 job ads do not ask for soft skills. Job ads coming

Table 4. Analysis of the “Others” soft skills category, all job ads

<table>
<thead>
<tr>
<th>Other skills</th>
<th>No</th>
<th>%</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work ethic</td>
<td>24</td>
<td>9%</td>
<td>One that demonstrates professionalism in their work. A person that one can trust and work results that one can rely on.</td>
<td>“Perform all work with the highest industry and ethical standards”</td>
</tr>
<tr>
<td>Customer focus</td>
<td>20</td>
<td>7%</td>
<td>The ability to understand and to act towards serving the clients’ needs.</td>
<td>“Customer-service orientation”</td>
</tr>
<tr>
<td>Proactive</td>
<td>19</td>
<td>7%</td>
<td>The ability to identify work to be done and start doing it, rather than just respond to management’s requirements to complete work tasks.</td>
<td>“Pro-activeness, accountability and results orientation”</td>
</tr>
<tr>
<td>Specialist</td>
<td>12</td>
<td>4%</td>
<td>The ability to be trusted; delivering valid, definitive and well-grounded work results.</td>
<td>“Ownership and responsibility of work”</td>
</tr>
<tr>
<td>Works under pressure</td>
<td>12</td>
<td>4%</td>
<td>The ability to perform job duties under the mental distress of multiple matters requiring immediate attention.</td>
<td>“Ability to multi-task under tight deadlines and report status”</td>
</tr>
<tr>
<td>Critical-thinking</td>
<td>8</td>
<td>3%</td>
<td>The ability to understand information in order to identify, construct and evaluate arguments.</td>
<td>“Critical thinking: use logic and reasoning to identify strengths and weaknesses of alternative solutions”</td>
</tr>
<tr>
<td>Motivated</td>
<td>8</td>
<td>3%</td>
<td>One that has a reason for doing something.</td>
<td>“Thrive in self-motivated innovation-driven environments”</td>
</tr>
<tr>
<td>Detail-oriented</td>
<td>6</td>
<td>2%</td>
<td>One that pays attention to details and makes a conscious effort to understand causes instead of just effects.</td>
<td>“Detail-oriented and analytical individual”</td>
</tr>
<tr>
<td>Quality-oriented</td>
<td>6</td>
<td>2%</td>
<td>One that understands the product’s ability to satisfy the customers’ needs.</td>
<td>“With a focus on improving development team productivity and enhancing product quality.”</td>
</tr>
<tr>
<td>Committed</td>
<td>6</td>
<td>2%</td>
<td>The ability to give time and energy to something one believes in.</td>
<td>“The successful candidate will be a self-motivated and committed individual”</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>36%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part 2: Collection of papers

from Canada and the USA ask in the biggest proportion of soft skills: Canada with 80% and the USA with 74%. China, Vietnam and Portugal have a moderate number of job ads not asking for soft skills (around 30% each). The rest of the studied countries have less than half of their ads mentioning soft skills.

4 Discussion

Our findings show that while more than half of the advertisers ask for soft skills, the percentage is smaller than the ones asking for testing skills or technical skills. Therefore, even though the human factors contribute the most to the success of software projects [21], the demand for soft skills still lags significantly behind the demand for testing and technical skills. The results are however in line with the findings in [22], which underlines this paradox when hiring in the software industry.

To answer our first research question: What is the trend for soft skills requirements for testers? we compared the result of our analysis with a similar study from 2012 [19]. In the set of findings [19], by analysing the trends in requirements for soft skills, we observe that there is a stronger need for team-playing skills, fast-learning skills, independent-work skills and openness and adaptability skills. Agile software development puts an emphasis on teamwork having a central role in software development, therefore, being a team-player is essential. There is little literature on fast-learning skills. However, we can assume that, given the rapidly changing tasks and requirements, testers have to learn fast concepts, tools or even whole new domains, such as accounting or statistics to perform their work.

We know that independent-working skills are directly related to how much activity is required of the learner. If one can work independently, then the burden of supervision, control, and feedback for the individual learner or the team is decreased [23]. Therefore, independent-learning is a desirable skill, in order to decrease the burden on the team.

Both openness and adaptability are traits that need to be fostered within development teams, and the explanation is it makes it easier for team members to admit their own mistakes, to negotiate, listen and facilitate [6]. Therefore, it is explainable that the request for these skills has increased, as a consequence of the desire of software companies to go overpass mistakes sooner and with fewer consequences. It is unclear why the demand for communication skills has such a significant fall. Even the agile manifesto makes communication a central part of software development. Therefore, an explanation for a down-trend in the demand for communication is that has become such a common trait that the job advertisers do not specifically mention it.

Concerning new categories of skills required of testers, the most important is work ethics. Job advertisers specifically ask for integrity and trustworthiness, in both relations with customers and team members. They ask for a work done with
professionalism first and foremost. Customer focus is a close second. There are specific requirements for testers to test through the customers’ perspective and to keep into account customers’ needs. An interpretation of the agile manifesto [24] sets the customer as an integral part of the development team; therefore more customer-focus is not surprising. Testers are specifically asked to be proactive, in the sense of picking-up and fulfilling job tasks, rather than wait and be told what to do. It can be a sign that there is a tendency in the software development to move away from micro-management to autonomous teams [25].

Job advertisers seem to exhibit a trend now in asking for responsibility-related skills; they emphasize the candidate to be able to deliver definitive work and well-grounded work results. We can translate this requirement by that work half-done is not accepted. One has to finish their job, and one has to prove that their work results are correct. Other notable new demands are the ability to work under pressure, the ability of critical-thinking, be motivated, committed, detail-oriented and quality-oriented. We consider it to be a positive signal the small number of requests for multitasking (4%), as this practice, especially inter-projects, is often associated with a loss of effectiveness [26].

Our second research question was “Are there specific soft skill requirements for testers in agile projects?”. Only 30% of the companies mentioned agile in the advertisement. Since a recent study found that at least 70% of developers work in agile projects [27], we would expect the number of agile testing ads to be higher. One explanation for the low number is that nowadays so many projects work with agile methods, that employers do not specify it anymore in the job ads.

The agile ads are slightly more oriented towards interpersonal skills, organizational skills and openness and adaptability skills. It is explicable from the perspective of agile development, where the focus is on people talking to each other, self-organizing teams and facing changes at a faster pace. Regarding particular requests for agile testers, it seems that more ads are asking for soft skills. It can be that in an agile environment the need for soft skills is more visible, therefore the demand for soft skills is more often expressed. The results show that there are no significantly different requirements for the agile testers: the soft skills requirements are relatively similar.

5 Limitations

The time span of the data collected is 2 months, December 2016 – January 2017. We have made a choice not to look at job ads in printed newspapers because the majority of job-ads in the software industry are published online. We have decided to choose a confidence level of 95% with a confidence interval of +5%, which we consider to be satisfactory enough. If we were to have selected a confidence level of 99% with an interval of +1%, we would have had to analyze over 16.600 jobs.
A limitation of the study can be the fact that two categories of skills, as grouped by Ahmed et al. [19], contain a combination of two skills. These categories are “analytical and problem-solving” and “openness and adaptability”. We have adopted that way of grouping the soft skills because we aimed at a comparative analysis with [19].

Last, a limitation is that the number of the job ads we have collected per country does not reflect directly the importance the country has in software development. This limitation comes from the fact that we do not have figures for how many jobs in software development or testing exist for the majority of countries.

6 Implications

Practitioners can use our results to identify which soft skills they should practice or enhance. Based on our findings, companies can identify their gaps in soft skills requirements. An analysis from 2009 shows that IT executives prefer soft skills and competencies for their employees [28] and that companies seek for more rounded practitioners with well-developed soft skills [29]. Soft skills, an essential factor in the way people collaborate, play a major part in team norms, in the sense of shared expectations of how to behave and interact with team members. Team norms have an important role in affecting team performance in software teams [30]. Therefore, teams having more insight regarding their team members’ soft skills can have a positive influence on the team performance.

Soft skills have a strong influence on certain parts of the test activities. While soft skills might not affect unit or integration testing, it is likely that the soft skills shape the way testers conceive and perform the kinds of tests having the user at their center, i.e. user experience and accessibility testing [31].

The Computer Science and Information Systems universities can have a significant contribution to practicing and shaping students’ soft skills. We notice that indifferent of the environment in which software testers work, the demand for and the importance of soft skills stay the same. Therefore, universities should aim to train all categories of skills identified in our study. While categories such as communication, analytical and problem-solving, organizational, individual-working, and work under pressure skills are partly addressed through the way students are trained and evaluated in the courses they attend, it is important for universities to include in the tasks for the students to train their interpersonal, innovation, fast-learning or team-playing skills. Also, universities and colleges can include in their teaching examples regarding work ethics, customer-focus, well-grounded work results. It would also be a valuable addition for to students to be able to practice, through exercises, their attention to detail, mindfulness towards quality, critical thinking. The universities could encourage the students to be pro-active and
committed. The reason is that soft skills are often seen by practitioners more important than hard skills in new graduates, therefore the need to introduce soft skills in the curriculum [32].

7 Conclusion and future work

Given the fact that more than half of the job advertisements ask explicitly for soft skills, we can conclude that employers consider them as qualities that affect the job performance of software testers. The most important soft skill for software testers is having good communication skills. The demand for soft skills seems to be stable over the years. However, we found that the trends in these demands are changing. There is an increase in the skills requirements regarding being a team-player, fast-learner, independent-working and having openness and adaptability skills. The trend could point to an increase in the number of responsibilities for testers. It may be that software testers now have to be involved in more aspects of the development pipeline: from creating software to managing it, quality-check it, building it and releasing it. It could also mean that projects themselves have changed the structure, becoming smaller and more open to changes, and people working on them must adapt to this new way of developing software.

Additionally, we identified new skills that employers want software testers to have: work ethics, customer focus, pro-activeness, responsibility, ability to work under pressure, focus on details, focus on quality and commitment. Exhibiting professionalism in one’s work, delivering finished work and trustable results get more into the focus of employers.

We expected that testers working in agile environments are requested for significantly more team-playing, communication, interpersonal and fast learning skills. But the results show that these requirements are the same for all kinds of job ads. A possible explanation is that nowadays so many projects work in an agile manner, that they do not specify it anymore in the job ads. But we can also say that, indifferent of the development model adopted, the relevant soft skills for testers remain unchanged.

Future work should get a better insight into the soft skills required in agile projects by interviewing agile project members. Other possible questions are: how will the trends in soft skills requirements evolve in the next years? Will the agile tester be required for more specialized soft skills? Will the new categories of soft skills gain even more importance? What are the new trends in soft skills requested from developers: do they correspond to the new categories of soft skills asked from software testers? Future work should also investigate why some employers do not ask for any soft skills in their advertisements for software testers.
8 References

Paper II: The Skills That Employers Look for in Software Testers

Raluca Florea✉1, Viktoria Stray1

1 University of Oslo, Gaustadalléen 23 B, 0373 Oslo, Norway
ralucamf, stray@ifi.uio.no

Abstract

Software testing is an integral part of software development that provides better quality products and user experiences and helps build the reputation of software companies. Though software testers perform a role that requires specific tasks and skills, in-depth studies of software testers lag behind research studies of other roles within software development teams. In this paper, we aim to create a profile of testers by presenting an empirical analysis of the skills the industry currently needs. We analysed data from 400 job ads in 33 countries. We mapped the skills on a taxonomy comprising test-related, technical, and domain specific skills. In addition, we looked at the demand for educational attainment, relevant certifications, and previous experience requirements.

Our findings show that employers are mostly interested in skills related to test planning and design, test automation, functional testing, performance testing, and progress reporting. One third of the job advertisers were interested in people with the skills to operate test execution tools. Selenium was the testing tool most in demand. The testers must have strong technical abilities, including programming skills in Java, C#, and SQL. Also, they must handle project management tasks such as estimation, risk management, and quality assurance.

Employers do not emphasise domain specific knowledge, which indicates that they consider testing skills portable across industries. One in seven job ads asks for a software testing certification. Our study helps clarify the complexity of the testing job and outlines the capabilities one needs to fulfil a software tester’s responsibilities.

Keywords: software-testing skills, software-testing skill requirements, software industry needs, empirical software engineering, human aspects of software engineering
1 Introduction

The skills of the members of software development teams are crucial to the success of software projects because they directly affect central aspects of software attributes such as performance, reliability, and simplicity [1]. Numerous studies have evaluated the impact of team members’ skills on the quality of software produced, the performance of the team, and the competitive market advantages those skills bring (see [2], [3], [4]). Alternatively, as many researchers have shown, having a lack of skilled employees on software development teams directly affects the software projects’ costs, their delivery times, and even their completion [5]. The consequences of badly done testing can creep into software development, which often leads to low-quality software. As the tragic series of events involving Boeing 737 Max airplanes demonstrated recently, insufficient software testing can damage products and companies [6], leading to the loss of billions of dollars, logistical nightmares [7] rippling through dozens of large companies, the disruption of business [8], and the loss of people’s jobs or even lives.

Analyzing the skills of employees, particularly those trained in a specific work environment, is important because the application of those skills enables managers to control and adjust the work process [9]. Skilled team members can also help create and maintain an enriched professional environment that fosters responsibility, the identification of problems and mitigating actions, the spectrum of task assignment, and participative decision-making [10]. Studying the skills expectations for testers is relevant for understanding the nature of the tasks they need to carry out, the importance and the extent of the technical tasks they have to accomplish, and the amount of specialized domain knowledge they require to fulfil their jobs.

Over the years, software developers and their skills have gained attention from both an industry perspective [2], [3], [4] and an educational perspective [11], [12]. However, the same cannot be said about software testers: No earlier than ten years ago, testing was seen as a side-activity, with testers sometimes regarded as second-class citizens whose role was considered a junior- or entry-level position [13]. A recent study by Capretz et al. [14] showed an even more startling picture: Among the people working in software development, tester and maintainer are the least popular roles. They found that the vast majority of engineers do not like testing and would not choose the role of tester [14]. Another study reported that testers in agile projects received less attention than developers in daily meetings, mainly because the Scrum Master viewed the information from the developers as more important [15]. In consequence, for us to look into the job, tasks, responsibilities, and demand for software testers becomes of clear importance.

Currently, “software tester” is a frequently used term but lacks a concrete definition. While “software developer” is well-defined [16, 17], tester is recognized as a
specific role without further details. The software testing standard ISO-29119:1 [18] defines tester as “one who develops and tests deliverables and completes the processes associated with the dynamic test process”. In the Standard Glossary of Terms, the International Software Testing Qualifications Board (ISTQB) defines testers as “skilled professionals involved in the testing of a component or a system” [19]. None of the most used lexicons [20-23] or the World Standards Cooperation [16, 18, 24, 25] provides definitions or side information on software testers, software testing, testing requirements, testing tasks, testing skills, or testing responsibilities. However, by searching for the same term not related to software, we derive the following: A tester is a person or machine who assesses the quality or state of a thing by testing it [19, 23].

Mathur et al. [26] described testers as the ones responsible for carrying out testing and building up test cases and test plans. Davidov et al. [27] described testers as professionals who identify new defects from failed test cases, analyze defects, and report them in a bug-tracking system. Some literature portrays testers as the development team’s worst enemies because they check everything and bring news of defects [28]. However, in teams with high psychological safety, testers are not afraid that developers are offended by the bugs discovered or mistakes they point out that developers have done, but rather that developers are actually happy when the testers report bugs [29]. Even though the descriptions we found are useful, practice shows that testers do much more than the literature describes.

Currently, testers are much better integrated into development teams than they used to be. Software companies recognize them as an important part of the team, especially in Agile environments [30], and the consolidation of Agile teams merits careful attention regarding learning, work, and performance [31].

Another reason to study testers is that we understand intuitively that they do more than testing, and we aim to unveil what else is under testers’ responsibilities and how important those responsibilities are for employers. As Dromey demonstrated 15 years ago [32], a preventative approach to software quality is more efficient than a curative one. But this approach entails a number of complex tasks, including design, process management, and adherence to standards. This complexity requires skills to manage it and, in this context, analyzing the skills testers need to succeed in their job becomes a necessity.

To increase the understanding of the software tester’s role, we extracted its comprehensive profile from the needs the industry currently expresses. In our analysis, we focus on the software tester role instead of on the overall process of software testing. We chose this approach because there is a significant difference between the skill requirements at the whole team level and the skill requirements for
the software testers, and we aim to determine traits, discover gaps, follow trends, and make recommendations for the software testers.

Regarding skills, understanding the state of practice and checking whether the testing practice meets the industry’s needs is a difficult process [33] that requires scrutinizing many aspects. The current state of the industry requires a thorough and consistent evaluation of one’s skills at hiring, including inspection of the education and experience listed on the CV and follow-up with interviews and tests [34]. We chose this approach because in the hiring phase, employers need to formulate clear and direct requests for employees’ skills that will allow the company to succeed in delivering completed software projects.

We researched two subsets of skills in two previous studies [35], [36]: soft skills demands for testers and the overall testing and tool requirements for testers. The previous studies show that testers need to have good communication, analysis, and problem-solving skills. At the same time, the need for adaptability, openness, and independent-working and team-playing skills has increased significantly. New demands have emerged in the areas of work ethics, customer focus, and the ability to work under pressure. Our current study extends to a large degree the context of the analysis beyond these two skill categories by focusing on an in-depth look at test-related and technical skills and requirements for educational attainment and previous experience, for which we created taxonomies to categorize and analyze the demand.

In our current study, we make a thorough analysis of three kinds of skill requirements for testers: First, we study the testing-related skills demand; then, we study the technical skills that testers need to have; and last, we analyze the domain specific knowledge that testers need to possess for their skills to be portable when they switch jobs. We also examine testers’ required level of experience and educational attainment and other training and certifications necessary to prove their skills.

In this light, we aim to answer three research questions:

RQ1: What skills do software testers need to have according to industrial demand?

RQ2: Are there any correlated skills shaping the profile of the software tester?

RQ3: What educational attainment, certified qualifications, and previous experience do hiring companies ask of software testers?

The remainder of the paper is organized as follows: Section 2 describes the approach we undertook to obtain the skills requirements for a software tester. Section 3 reports the results of our analysis. Section 4 discusses the findings and provides our interpretation of the results. Section 5 presents the implications of the analysis for
different categories of audience. Section 6 approaches the limitations of our study. Section 7 draws the conclusions of our findings and suggests future development of the current research.

2 Research method

The current study is an empirical analysis of industrial needs for various skills for software testers from the employers’ perspective. To analyze the software tester role, we created a taxonomy for the software testers’ requested skills. We chose to create a taxonomy to map the skills required for software testers, as it is a common way to organize knowledge systematically in a domain of interest in which there is little knowledge on a topic [37] and one needs to identify categories, topics, and constructs.

Our analysis is based on a collection of job ads that we processed to allow us to perform both quantitative and qualitative data analyses. The number of job ads we studied gives us a 95% confidence level and a 5% confidence interval for the quantitative analysis. The approach is relevant in that it complements the theory of skill with a hands-on analysis of concrete skill requirements. Additionally, the quantitative and qualitative analysis combination enables us, on one hand, to discover which kinds of skills are in most demand, the level of the demand, and the average number of relevant skills employers are seeking and, on the other hand, to substantiate our findings, mining information to reveal specific traits of the skills as demanded by employers. As we found no significant differences in the number or kind of demands between the ads for the different names of the role, we analyzed the entire set of job ads as one.

The structure we used for the testing-related skill analysis is based on the structure of the ISTQB Foundation Level Syllabus [38], and we created a structure for analyzing the technical skills. We understand technical skills to refer to any programming, software engineering, project management, or other skills needed in the area of software development that are not typically the responsibility of software testers: the skills that the programmers, software designers, software architects, team leaders, auditors, etc., are more likely to need.

3 Data collection

To get an overview of the skill requirements for testers, we collected and analyzed 400 job ads published online for 33 countries between December 2016 and February 2017. We chose to use online job-search engines to collect the raw job
advertisements instead of going to specific hiring companies. This approach is an efficient way of including in our analysis a large number of employers, a good diversity of job profiles, and large visibility to job seekers.

We investigated the most significant job-search engines through two dimensions: the number of users and the number of jobs posted. According to commercial Web traffic data and analytics services provided by Alexa (Amazon) and SimilarWeb, we chose the five most popular job-search engines [39-42].

We collected the job ads from the following countries: USA (96), Canada (65), Norway (22), UK (20), Argentina (17), France (17), Mexico (15), South Africa (14), China (14), Vietnam (13), Greece (13), India (12), Sweden (10), Portugal (10), Australia (10), and others (52).

One of the amenities job search engines offer that we used is that they interpret the search criteria and give relevant results for a range of jobs related to the search term. Therefore, we input the search string “software tester”, and we obtained as results job ads for software testers, technical testers, functional testers, QAs, usability testers, performance testers, mobile testers, Web testers, security testers, automated software testers, financial-systems testers, game testers, and so on. Using this tool helps avoid data-collection bias related to the name of the jobs advertised for software testers.

4 Data analysis

We collected all the information we found on the job descriptions, responsibilities, requirements for all the analyzed ads. We chose this approach, as not all advertisers use the same format for the ads: We found skill requirements for the advertised position in almost all the fields we studied. Next, we cleaned, categorized, and coded the collected information to group it and then analyzed it quantitatively and qualitatively. We created the structure for the demands as shown in Fig. 1 and used it for grouping the information we collected.
To examine the software testing-related skills, we created a separate taxonomy using the structure of the information in the ISTQB Foundation Level Syllabus 2011 [38] as a basis. We chose this syllabus as a reference for our research since it is a representative tool for structuring software testing knowledge. The ISTQB uses it as the theoretical foundation of examining and providing a standardized qualification for software testers. Currently, the organization has issued over 640,000 certifications in 120 countries [43]. By following its table of contents and selecting from it the parts related to skills, we obtained a subset of the original table of contents that forms the taxonomy of test-related skills, as shown in Fig. 2.

We calculated the percentage of ads and the mean number of skills demanded for each category and subcategory in our testing-skills taxonomy, and where it was useful, we performed a free-text analysis of the frequency of terms to obtain information of interest regarding specific demands. For the free-text analysis, we used an online tool that analyses the frequency of words in large strings and offers options such as inspecting the roots of words, grouping variations together, and excluding insignificant strings, which we used on the collection of coded information that we obtained as a result of the job ad processing. By looking at the categories of software testing-skills demands, we got an overall picture of the skills the industry currently seeks. By looking at the ranked list of subcategories, we determined the popularity and the spread of the demands.
Fig. 2 Taxonomy of the software-testing skills for testers

Through a qualitative analysis, we created a separate taxonomy for mapping the technical skills by labelling and bracketing all the demands related to the technical aspects of development (see Fig. 3, depicting the areas of demand for testers' technical skills).
We did not create a taxonomy related to domain specific skills. Instead, created a ranking of these skills, in which we used domain specificity to perform a free-text analysis of the information we collected, and we extracted the most frequent terms. We manually went through the results and sorted them again by grouping demands logically belonging together. We proceeded in the same manner for the educational attainment, relevant certification, and seniority requirements. We also performed a quantitative analysis for the minimum number of years of prior experience that employers requested.

For the quantitative assessment, we used SPSS to perform a statistical analysis of the numerical parameters of our data. We obtained the percentage of ads asking for each category of skills, the mean number of skills per category, the maximum number of skills per category, and the number of skills demanded in the lower and the upper quartiles of the ads. The data file available online for access and download [44] provides the details regarding the techniques we used to analyse each category and subcategory of skills.

Moreover, we aimed to discover any patterns in the subcategories of skills being asked for together in the same ad. A strong correlation of skills would have pointed to the existence of various tester profiles that employers are looking for. Inversely, a lack of patterns would have indicated that there no clear software tester profiles emerged from the employers’ demands at hiring. To study this aspect of our research, we took into account all of the subcategories of skills in the advertisements and checked how often they appear together in the same ad, two by two. Therefore, we used SPSS to generate a correlation matrix for the identified skills. We looked at the Spearman correlation coefficient, which indicates if there are correlations between pairs of variables. The coefficient’s value indicates the nature (direct or inverse) and strength of the correlation.
We searched for significant differences in the requirements for various categories of testers, such as testers and test analysts, and we searched for particular requirements for test managers.

5 Results

The profiles of software testing jobs

A first finding in the landscape of software testing regards the diversity of names used for the software testing role. Out of all the ads, the top ranked requests were for software testers (77 ads), QA testers (41 ads), automation testers (9 ads), test engineers (7 ads), test analysts (7 ads), manual testers (4 ads), penetration testers (4 ads), and business analysts (4 ads). Out of the 400 job ads, only three asked for test leaders and three asked for test managers. From the remaining ads, more than half (213) used unique names for the software tester role. Examples include “IT tester”, “cybersecurity testing”, “MS dynamics CRM tester”, “senior ETL tester”, “testing specialist with HIX/HBX experience”, “senior testing automation for network management”, “digital validation engineer”, “linguistic tester”, “QC staff”, “algorithms for driver assistance tester”, “content producer tester”, “technical junior”, etc.

Focusing on the skills demand, Table 1 displays the industry’s level of demand for all software testers, divided into the four skill categories (Fig. 1). With more than 97% of advertisers and an average of more than 10 skills per ad, the overwhelming majority of employers asked for testing-related competence from testers, followed closely by technical skills. Two thirds of the ads asked for soft skills, and only one third of the ads sought domain specific skills.

<table>
<thead>
<tr>
<th>Type of skill</th>
<th>Percentage of ads</th>
<th>Avg. skill/ad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Testing skills</td>
<td>97.7%</td>
<td>10.2</td>
</tr>
<tr>
<td>Technical skills</td>
<td>82.5%</td>
<td>5.4</td>
</tr>
<tr>
<td>Soft skills</td>
<td>64.2%</td>
<td>4.8</td>
</tr>
<tr>
<td>Domain specific skills</td>
<td>34.7%</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Software-testing skills demand for testers

Table 2 and Fig. 4 describe the industry’s need for software-testing skills, from complementary viewpoints. Table 2 features a ranking of the categories of software-testing skills and the percentage of the ads asking for them, while Fig. 4 shows the interquartile ranges for the number of software-testing skills in each category.
Most of the hiring ads asked for skills related to the testing process (see Table 2) and, moreover, at least four skills of this kind (see Fig. 4). Three out of four ads asked for test management skills, and testers had to have mastered at least three aspects of it. The ability to perform at least two different types of testing, such as functional, performance, was asked for by more than half of the advertisers. The test tools were another sought-after category of skills, with more than half of the ads asking for testing-tool skills. Most of the ads sought software testers who had mastered about two tools and a little below half of the ads asked for specific test tools. Only one fourth of the ads asking for testing tools asked for competency in five or more tools.

We observed that the request for specific levels of testing was on the lower side, with less than one in three advertisements asking for it and with less than two skills to master.

<table>
<thead>
<tr>
<th>Category code</th>
<th>Skill category</th>
<th>Ad (percentage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Test process</td>
<td>87.2%</td>
</tr>
<tr>
<td>T2</td>
<td>Test management</td>
<td>73.2%</td>
</tr>
<tr>
<td>T3</td>
<td>Test types</td>
<td>65.7%</td>
</tr>
<tr>
<td>T4</td>
<td>Test tools*</td>
<td>59.5%</td>
</tr>
<tr>
<td>T5</td>
<td>Test levels</td>
<td>28.2%</td>
</tr>
<tr>
<td>T6</td>
<td>Software categories</td>
<td>25.5%</td>
</tr>
<tr>
<td>T7</td>
<td>Reviews and static analysis</td>
<td>12.2%</td>
</tr>
</tbody>
</table>

* 42.0% of the ads asked for specific test tools.

* Extremes
° Outliers
Going into the details of the skill request, our findings show that most of the advertisers asked for specific skills belonging to the category of test process (T1): planning, design, and execution of tests, whether assisted or not by tools (see Table 3). Next in the ranking is the request that the testers track their own testing through progress reports, bug reports, and follow-up of the bug fixes. The industry is interested in testers with skills in test strategies and methodologies to a smaller extent. Similarly, marginal requests were for the software testers to manage their test environments, write test-related documentation, manage testing risks, handle test data, and handle exit criteria and their corresponding test-related metrics.
Table 3. The hierarchy of subcategories of testing skills demand for testers

<table>
<thead>
<tr>
<th>Subcategory of skill requested</th>
<th>Percentage of ads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test design ((T1))</td>
<td>52.7%</td>
</tr>
<tr>
<td>Automated test execution ((T1))</td>
<td>43.5%</td>
</tr>
<tr>
<td>Manual test execution ((T1))</td>
<td>42.0%</td>
</tr>
<tr>
<td>Test planning ((T1))</td>
<td>40.0%</td>
</tr>
<tr>
<td>Automated test execution tools ((T4))</td>
<td>30.7%</td>
</tr>
<tr>
<td>Test progress reporting ((T2))</td>
<td>29.5%</td>
</tr>
<tr>
<td>Follow-up of bug fixes ((T2))</td>
<td>28.2%</td>
</tr>
<tr>
<td>Bug reporting ((T2))</td>
<td>27.7%</td>
</tr>
<tr>
<td>Functional testing ((T3))</td>
<td>27.2%</td>
</tr>
<tr>
<td>Performance testing ((T3))</td>
<td>23.7%</td>
</tr>
<tr>
<td>Test management tools ((T4))</td>
<td>22.5%</td>
</tr>
<tr>
<td>Web applications testing ((T6))</td>
<td>18.5%</td>
</tr>
<tr>
<td>Regression testing ((T3))</td>
<td>16.0%</td>
</tr>
<tr>
<td>Test implementation ((T1))</td>
<td>14.7%</td>
</tr>
<tr>
<td>Bug identification ((T1))</td>
<td>14.5%</td>
</tr>
<tr>
<td>Mobile applications testing ((T6))</td>
<td>13.5%</td>
</tr>
<tr>
<td>Testing methodologies ((T2))</td>
<td>13.2%</td>
</tr>
<tr>
<td>Test strategies ((T2))</td>
<td>13.2%</td>
</tr>
<tr>
<td>Acceptance-level testing ((T5))</td>
<td>12.7%</td>
</tr>
<tr>
<td>Test techniques ((T3))</td>
<td>12.0%</td>
</tr>
<tr>
<td>System-level testing ((T5))</td>
<td>11.5%</td>
</tr>
<tr>
<td>Test environments-related activities ((T2))</td>
<td>10.2%</td>
</tr>
<tr>
<td>Integration-level testing ((T5))</td>
<td>10.0%</td>
</tr>
<tr>
<td>Documentation writing ((T2))</td>
<td>9.7%</td>
</tr>
<tr>
<td>Risk management in testing ((T2))</td>
<td>9.2%</td>
</tr>
<tr>
<td>Interface (API) testing ((T3))</td>
<td>9.0%</td>
</tr>
<tr>
<td>Bug-tracking tools ((T4))</td>
<td>8.2%</td>
</tr>
<tr>
<td>Test data management ((T2))</td>
<td>7.7%</td>
</tr>
<tr>
<td>Unit-level testing ((T5))</td>
<td>7.5%</td>
</tr>
<tr>
<td>Test-related metrics ((T2))</td>
<td>7.0%</td>
</tr>
<tr>
<td>Performance-testing tools ((T4))</td>
<td>7.0%</td>
</tr>
<tr>
<td>Black-box/grey-box testing ((T3))</td>
<td>4.7%</td>
</tr>
<tr>
<td>Exit-related criteria ((T2))</td>
<td>4.5%</td>
</tr>
<tr>
<td>White-box testing ((T3))</td>
<td>4.0%</td>
</tr>
<tr>
<td>Maintenance testing ((T3))</td>
<td>2.0%</td>
</tr>
<tr>
<td>Test effort estimation ((T2))</td>
<td>1.7%</td>
</tr>
<tr>
<td>Desktop applications testing ((T6))</td>
<td>1.7%</td>
</tr>
<tr>
<td>Large-scale applications testing ((T6))</td>
<td>1.5%</td>
</tr>
<tr>
<td>Client-server architecture testing ((T6))</td>
<td>1.2%</td>
</tr>
<tr>
<td>Test closure activities ((T1))</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

The results show that testing the functionality of software products was the most sought-after type of testing skill \((T3)\), followed by the requirement for performance
testing skills. A relatively small number of advertisers from the industry asked explicitly for regression testing or maintenance testing.

Regarding software testing on different levels, the industry seems to focus on high-level testing, with acceptance testing and system testing occupying the first two positions in the ranking of test-levels skills. The employers asked one in ten testers to perform integration testing. Unit testing was not in high demand for testers.

Most of the ads did not ask for skills in testing specific structures of software, such as distributed applications. Less than one in five ads asked for skills in testing Web systems, and one in seven ads asked for skills in testing mobile systems. About half of these ads asked for both mobile and Web competency. Besides these requests, other demands were rare, with very few advertisers needing skills for testing large-scale systems, client-server architecture, multi-interface systems, or database systems.

The job advertisers were interested in hiring people with the skills to operate tools in the category of automated test execution. The tools in the category of test management tools were demanded to a slightly smaller extent than the test execution tools, with one in four employers looking specifically for them. One in twelve advertisers asked for competencies in using tools in the category of performance-testing.

<table>
<thead>
<tr>
<th>Testing tools</th>
<th>Tool category</th>
<th>% of ads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selenium</td>
<td>Automated test execution</td>
<td>18.0%</td>
</tr>
<tr>
<td>Jira</td>
<td>Test management, bug tracking</td>
<td>10.2%</td>
</tr>
<tr>
<td>HP Quality Center</td>
<td>Test management, bug tracking</td>
<td>8.7%</td>
</tr>
<tr>
<td>QTP</td>
<td>Automated test execution</td>
<td>5.7%</td>
</tr>
<tr>
<td>JMeter</td>
<td>Performance</td>
<td>5.2%</td>
</tr>
<tr>
<td>LoadRunner</td>
<td>Performance</td>
<td>4.0%</td>
</tr>
<tr>
<td>Cucumber</td>
<td>Automated test execution</td>
<td>3.7%</td>
</tr>
<tr>
<td>SoapUI</td>
<td>Automated test execution, API testing</td>
<td>3.2%</td>
</tr>
<tr>
<td>Confluence</td>
<td>Test management</td>
<td>2.7%</td>
</tr>
<tr>
<td>TestNG</td>
<td>Automated test execution, unit testing</td>
<td>2.5%</td>
</tr>
<tr>
<td>JUnit</td>
<td>Unit testing</td>
<td>2.2%</td>
</tr>
<tr>
<td>HP ALM</td>
<td>Test management</td>
<td>1.7%</td>
</tr>
<tr>
<td>Microsoft Test Manager</td>
<td>Automated test execution, Test management</td>
<td>1.5%</td>
</tr>
<tr>
<td>Fitnesse</td>
<td>Automated test execution</td>
<td>1.2%</td>
</tr>
</tbody>
</table>

Looking into the details of the demand for tools, the most asked for was Selenium, a tool in the category of automated test execution. The second most requested tool, Jira, is one of the most commonly used tools for test management [45] and comes with comprehensive bug-tracking systems and a myriad of add-ons for project management.
management, time tracking, and integration. HP Quality Center, the third of the ranked tools requested, is a test execution tool that is part of the Application Lifecycle Management solution offered by Hewlett-Packard. QTP, the fourth most requested tool, is also a popular automated test execution tool for functional testing [46]. JMeter and LoadRunner are both performance-measurement tools that used to be the exclusive responsibility of developers [47] but are currently widely used by testers as well [48].

**Technical skills demand for testers**

Table 5 and Fig. 5 describe the industry’s need for technical skills from complementary perspectives. Table 5 ranks the categories of technical skills demand for testers and the percentage of the ads asking for them, while Fig. 5 shows the interquartile ranges for the number of technical skills in each category.

Two thirds of the ads want testers with programming and software engineering skills, while almost half of the ads need the testers to master specific programming or scripting languages (Table 5). However, our results show that testers are asked to be proficient in no more than two programming languages (see Fig. 5). One third of the ads asks for something related to the management of software development, and another third of the ads asks for a specific development life cycle to be followed.

One in seven employers is interested in testers with specialised knowledge on a particular operating system (OS; Table 5). Requirements to master more than one OS are rare (Fig. 5). One in four employers ask the testers to use specific development environments or tools. When the testers are asked to use these, in most cases, they have to use just one development tool or development environment.

**Table 5. Categories of technical skills demand for testers**

<table>
<thead>
<tr>
<th>Skill category</th>
<th>Percentage of ads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming &amp; Engineering</td>
<td>69.0%</td>
</tr>
<tr>
<td>Programming or Technologies Languages</td>
<td>42.2%</td>
</tr>
<tr>
<td>Software Management</td>
<td>36.2%</td>
</tr>
<tr>
<td>SDLC-related</td>
<td>33.2%</td>
</tr>
<tr>
<td>Development Tools and Systems</td>
<td>25.0%</td>
</tr>
<tr>
<td>Operating systems-related</td>
<td>14.5%</td>
</tr>
</tbody>
</table>
As Table 6 shows, most of the demand is for programming languages or frameworks, but a significant number of ads ask for the following software engineering skills as well: continuous integration, deployment, web services, or networks. The only development technique that is specifically asked for is test-driven development (TDD), but this request is made in rather low proportions (3%).

From the programming languages in demand, SQL takes first place with one in five ads asking for it. The second most demanded language is Java. Third place is taken by C#, which has half of the demand that Java does. We observe, therefore, that the testers are currently being asked mostly for compiled languages. The next two languages in demand are dynamic: JavaScript closely follows the proportion of demand for C#, and Python falls only slightly behind JavaScript. Next in demand is C++. HTML, XML, Ruby, C, CSS, and Perl are close and rather low in demand.

In software management, the competency with the most demand is the quality assurance (QA) of software systems and the processes implementing it. Next in demand is the request for skills in project management and requirements management. While there is some demand, testers are not necessarily asked to be business analysts at the same time, with only over 1% of ads demanding such skills.

Regarding various software development life cycles that software companies adopt, the most demand is by far for Agile. Only a fraction of employers asks for their testers to master Waterfall, Kanban, or V-model.
With focus on the specific development tools or environments, most of the demand is around deployment tools, such as Jenkins or TeamCity. Build tools, such as Maven, are also demanded, as are classic source-control systems, such as GIT or TFS. SharePoint was one of the few online collaboration frameworks specifically asked for and had rather low but notable demand. Only a limited number of jobs require testers to have skills in the use of specific tools or systems with regards to software development. The top request, with one in twenty ads asking, is for Jenkins, a tool used for continuous integration and continuous delivery.

Related to operating systems, most in demand seems to be desktop OSs and Unix/Linux OS. The third in the ranked requests is Windows. Last are mobile OSs, Android being most demanded and iOS following it. No other mobile OS was specifically asked for.

Table 6. Sub-categories of technical skills demand for testers

<table>
<thead>
<tr>
<th>Technical skills requested</th>
<th>Percentage of ads</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming &amp; Engineering</strong></td>
<td></td>
</tr>
<tr>
<td>Programming tasks</td>
<td>42.2%</td>
</tr>
<tr>
<td>Various technologies</td>
<td>13.0%</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>6.7%</td>
</tr>
<tr>
<td>Various frameworks</td>
<td>6.2%</td>
</tr>
<tr>
<td>TDD</td>
<td>3.0%</td>
</tr>
<tr>
<td>Web Services</td>
<td>2.5%</td>
</tr>
<tr>
<td>Networks</td>
<td>2.0%</td>
</tr>
<tr>
<td>Continuous delivery</td>
<td>1.7%</td>
</tr>
<tr>
<td><strong>Programming Languages or Technologies</strong></td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td>20.2%</td>
</tr>
<tr>
<td>Java</td>
<td>14.5%</td>
</tr>
<tr>
<td>C#</td>
<td>7.7%</td>
</tr>
<tr>
<td>JavaScript</td>
<td>7.2%</td>
</tr>
<tr>
<td>Python</td>
<td>6.7%</td>
</tr>
<tr>
<td>C++</td>
<td>6.0%</td>
</tr>
<tr>
<td>HTML</td>
<td>3.5%</td>
</tr>
<tr>
<td>XML</td>
<td>3.2%</td>
</tr>
<tr>
<td>Ruby</td>
<td>3.0%</td>
</tr>
<tr>
<td>C</td>
<td>2.7%</td>
</tr>
<tr>
<td>CSS</td>
<td>2.2%</td>
</tr>
<tr>
<td>Perl</td>
<td>2.2%</td>
</tr>
<tr>
<td><strong>Software Management</strong></td>
<td></td>
</tr>
<tr>
<td>QA processes</td>
<td>12.5%</td>
</tr>
<tr>
<td>Project management</td>
<td>7.5%</td>
</tr>
<tr>
<td>Requirements management</td>
<td>7.2%</td>
</tr>
<tr>
<td>Business analysis</td>
<td>1.5%</td>
</tr>
<tr>
<td><strong>SDLC-related</strong></td>
<td></td>
</tr>
<tr>
<td>Agile</td>
<td>31.2%</td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Waterfall</th>
<th>3.2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kanban</td>
<td>1.2%</td>
</tr>
<tr>
<td>V-model</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

**Development Tools and Systems**

<table>
<thead>
<tr>
<th>Jenkins</th>
<th>4.2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIT</td>
<td>2.2%</td>
</tr>
<tr>
<td>TFS</td>
<td>1.7%</td>
</tr>
<tr>
<td>Maven</td>
<td>1.2%</td>
</tr>
<tr>
<td>TeamCity</td>
<td>1.2%</td>
</tr>
<tr>
<td>SharePoint</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

**Operating Systems-related**

<table>
<thead>
<tr>
<th>Linux</th>
<th>9.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix</td>
<td>5.0%</td>
</tr>
<tr>
<td>Windows</td>
<td>3.7%</td>
</tr>
<tr>
<td>Android</td>
<td>3.5%</td>
</tr>
<tr>
<td>iOS</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

### Domain specific skills demand for testers

The demand for domain specific knowledge covers a wide variety of industries. Out of the one third of the ads that ask for domain specific competencies, most of the demand is around financial services (4.7%) and banking systems (3.2%) (see Table 7). Most of the other notable requests are in financially connected areas, such as payment, e-commerce, and accounting. Next are requests for the telecom industry (4.5%) and logistics (1.2%). The request for skills in gaming software is at around 2% among all job ads, while the other ads (30.7%) cover a large spectrum of other kinds of software.

**Table 7. Domain specific skills demand for testers**

<table>
<thead>
<tr>
<th>Domain specific skills requested</th>
<th>Percentage of ads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial services systems</td>
<td>4.7%</td>
</tr>
<tr>
<td>Telecom</td>
<td>4.5%</td>
</tr>
<tr>
<td>Banking systems</td>
<td>3.2%</td>
</tr>
<tr>
<td>Gaming software</td>
<td>2.2%</td>
</tr>
<tr>
<td>Insurance industry</td>
<td>2.0%</td>
</tr>
<tr>
<td>Payment systems</td>
<td>2.0%</td>
</tr>
<tr>
<td>E-commerce</td>
<td>1.7%</td>
</tr>
<tr>
<td>CRM</td>
<td>1.5%</td>
</tr>
<tr>
<td>Automotive</td>
<td>1.5%</td>
</tr>
<tr>
<td>Networks</td>
<td>1.5%</td>
</tr>
<tr>
<td>ERP</td>
<td>1.2%</td>
</tr>
<tr>
<td>Accounting</td>
<td>1.2%</td>
</tr>
<tr>
<td>Healthcare</td>
<td>1.0%</td>
</tr>
<tr>
<td>Others</td>
<td>30.7%</td>
</tr>
</tbody>
</table>
Correlated skills

To identify patterns in the profiles of software testers that emerge from the employers’ ads, we checked for correlations between the different skills asked for in the advertisements. Our sample size of \( n = 400 \), where we included all the testing-related, technical, and soft skill requirements for testers, was grouped by using the taxonomy in Fig. 2, Fig. 3, and study [35]. We split the data in two sets to look for correlations among the high level of skills (the first level of nodes in our taxonomies) and to look for correlations in the details of the skills (the leaf nodes of the taxonomies).

We used the Spearman correlation coefficient (\( \rho \)) for each pair of skills, as it shows correlations between variables linked by any monotonic function, not just by a linear relationship. At the same time, the Spearman coefficient is appropriate for ordinal variables, the form in which our skills-related data are stored. The resulting Spearman coefficients can be downloaded on the Web: [49] (high-level skill analysis) and [50] (low-level skill analysis). We used the interpretation of the Spearman correlation coefficient in psychology given by Dancey & Reidy [51]

Correlation analysis on the high level of skills

In the analysis of skills at a high level, we found a set of relevant correlations, which are presented in Table 8.

<table>
<thead>
<tr>
<th>Skills</th>
<th>( \rho ) values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strong Correlations</strong></td>
<td></td>
</tr>
<tr>
<td>Technical &amp; programming</td>
<td>0.79</td>
</tr>
<tr>
<td>Testing &amp; testing process</td>
<td>0.77</td>
</tr>
<tr>
<td>Testing &amp; test management</td>
<td>0.74</td>
</tr>
<tr>
<td><strong>Moderate Correlations</strong></td>
<td></td>
</tr>
<tr>
<td>Test management &amp; test process</td>
<td>0.57</td>
</tr>
<tr>
<td>Testing &amp; specific types of testing</td>
<td>0.49</td>
</tr>
<tr>
<td>Technical &amp; software management</td>
<td>0.49</td>
</tr>
<tr>
<td>Testing &amp; test tools</td>
<td>0.48</td>
</tr>
<tr>
<td>Min. years of experience &amp; other test-management activities</td>
<td>0.44</td>
</tr>
<tr>
<td>Technical &amp; operating system knowledge</td>
<td>0.42</td>
</tr>
<tr>
<td>Technical &amp; domain specific</td>
<td>0.39</td>
</tr>
<tr>
<td>Testing &amp; test levels</td>
<td>0.38</td>
</tr>
<tr>
<td>Testing &amp; software management</td>
<td>0.38</td>
</tr>
<tr>
<td>No. of responsibilities &amp; min. years of experience</td>
<td>0.33</td>
</tr>
<tr>
<td>Testing &amp; soft skills</td>
<td>0.30</td>
</tr>
<tr>
<td>Testing tools &amp; specific software development life cycle</td>
<td>0.30</td>
</tr>
</tbody>
</table>
When the employers have technical demands for testers, they will most likely be related to programming and then related to using various OSs. But there is also a significant connection between the technical skills that the employers look for in testers and software management skills and domain specific skills. Part of the technical testers are likely needed to take responsibilities from the project management area and to hold domain specific knowledge. However, it is likely that they will not be involved in requirement analysis and manual testing.

A second result is the strong connection between the testing skills and testing progress and test management skills. The data show us that employers strongly opt for testers with a grip on the entire testing process, and the ability to manage testing. Many employers seem to ask together for bug-tracking skills and testing related to changes.

Part of the domain-specific testers are asked to have the ability to operate testing tools, but they are not likely to be asked to design automated tests, to do white-box testing, or to have performance-testing skills.

The more experienced the testers are, the more likely they will be asked to perform other miscellaneous activities related to test management. In addition, the more experience is required from the testers, the more responsibilities and tasks it is likely that they will get. Our results show that experienced testers are not likely to be asked for education attainment nor for specific certifications. A surprising result shows that it is not likely that experienced testers are asked for soft skills.

**Correlation analysis on the low level of skills**

In the analysis of skills at a low level, we found a relevant set of correlations, which is presented in Table 9.

**Table 9. The correlation of skills analysis, low level**

<table>
<thead>
<tr>
<th>Skills</th>
<th>p values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strong Correlations</strong></td>
<td></td>
</tr>
<tr>
<td>Automated test design &amp; automated test execution</td>
<td>0.71</td>
</tr>
<tr>
<td><strong>Moderate Correlations</strong></td>
<td></td>
</tr>
<tr>
<td>Black-box testing skills &amp; white-box testing</td>
<td>0.61</td>
</tr>
<tr>
<td>Specific testing tools &amp; automated testing tool</td>
<td>0.58</td>
</tr>
<tr>
<td>Specific testing tools &amp; test management tools</td>
<td>0.51</td>
</tr>
<tr>
<td>Manual test design &amp; manual test execution</td>
<td>0.51</td>
</tr>
</tbody>
</table>
The correlation analysis results show the following: the employers ask for testers who can perform both the design and execution of tests. Part of the employers ask the testers who perform white-box testing to perform black-box testing too. In addition, the correlation analysis confirms the quantitative analysis that when the employers ask for testing tools, they will most likely ask for automated test execution tools, test management tools, and some other specific testing tools. They are less likely to ask for specific bug-tracking tools.

The domain-experts are not likely to be asked for performance testing, design of automated tests or white-box testing. The more technical skills the employers look for in software testers, the more likely they will not ask for abilities with manual testing.

Regarding soft skills, we observe a tendency among the employers to ask testers to have the ability to work independently while having problem-solving and organization skills. These employers associate fast-learning skills with interpersonal skills and the openness and adaptability skills in their demands. Employers also seem to associate fast-learning skills with mastering various test techniques, such as exploratory testing, use-case testing, or boundary-values testing.

Another notable result is the correlation that the employers make between the team-playing skills and the ability to define and use test metrics.


**Discussion**

**Education attainment and certifications**

The greatest number of ads (85%) ask for completed bachelor studies. Only one-tenth of the ads ask for more, such as completion of master-level studies. While there is a request, the number of ads asking for PhD studies is negligible (0.5%). Most employers do not require specific certifications related to testing. From the 14% who ask for such certification, the majority of the requests are for ISTQB Foundations-level Tester (9%) and ISTQB Advanced-level Tester certifications (1%). A negligible number of ads (<1%) ask for Application Life Cycle Management (ALM) certifications, information systems security-related certifications (CISSP), and network penetration testing certifications (GPEN).

**Previous experience**

About half of the ads (54%) ask for previous experience. The calculated average of years required from these ads is 3.4. With less than 1%, only a few exceptional ads ask for 10+ years of prior experience. After removing the exceptional cases, the majority of ads ask for two to three years of previous relevant experience. Most of the job ads (83%) do not ask for specific seniority; only 10% of the ads ask for senior testers, and 5% of the ads ask for juniors.

We found that test managers are asked for nine skills on average per advertisement. Looking into the details of the demand, the first request is for test managers to perform software testing. Second comes the demand for test management skills, test planning skills, and skills with automated testing tools. Only two out of the six ads ask the test leaders to manage testers, to perform test estimations, and to monitor the test’s progress. The specific requests in the area of tools is varied, and there is no strong request for a particular tool.

**6 Discussion**

Our results show that software testers need to be highly skilled in their job and that they need to master a wide range of skills. By looking at the spread of the demand, testers are not asked to be specialised in one certain activity, such as performance, automation, or test management; instead, they are asked to master many and various aspects of testing. We infer that most testers are asked to play an active role throughout the testing process. By analysing the landscape of job names used for the role of software testers, we conclude that the employers place the skills essential to them in the title of the job they advertise for, such as the knowledge of a tool (Jira, Selenium, CRM) or a class of testing skills (functional, network, penetration).

Aside from this, most employers ask testers for a wide span of technical skills. In regard to testing, the technical requests are not specialized on narrow aspects; they
vary from programming skills to development environments and tools, as well as integration skills, deployment, project management, development models, and many others. This finding is in line with an observation from 2010 [52], which found that companies see the technical knowledge of testers as important and more significant than domain knowledge. The finding is also supported by [32], stating that the best approach in software development would be extensive consultation with domain experts.

In a previous study in which we included developers [36], we found that only half of developers need to master anything related to testing and that they are asked to have an average of 2.5 testing skills; the result denotes that testers need to master development better than developers need to master testing. Therefore, developers remain highly focused on programming and software building, while testers are asked to acquire more technical skills and fulfil more programming, software engineering-related, or software management tasks.

**Testing skills, technical skills, and domain specific skills**

In this section, we will discuss our first research question: “What skills do software testers need to have according to industrial demand?”

**Designing tests is the main demand**

The most important software-testing skills for the industry seem to include the ability to design tests. The result reveals that employers want testers to be able to extract the testable parts of software and design corresponding high-level and low-level test cases from the product documentation or other available sources. The requirement to design tests offers a side benefit: a way to evaluate the testability of software, in which the goal is to have a more testable software. Also, through designing tests, testers give flexibility and structure to the testing initiatives: tests are grouped in families, are given ranks and criticality levels, and are categorised as main-stream, edge-scenario, or failure cases. This makes the testing process capable of changing with regards to deadlines, prioritisation, and stakeholders’ feedback. The survey [53] finds that design quality is vital if software is to be maintainable and finds a relationship between design quality and testing practices. Therefore, our results support this survey through the employer’s demand foremost for design skills, which also encompasses the need for maintainable software, good design, and good testing practices.

**Automated testing is more asked for than manual testing**

The advertisers put more focus on automated tests than on manual tests, and we assume that the higher level of request for automated execution of tests involves
more complexity [54] and therefore requires more skills than the manual execution of tests. Moe et al. [55] report that automated testing, an important enabler for Agile development, heavily depends on the knowledge of the testers. They found that most testers still do manual testing, and they consider that it takes too long to update the automated tests. Our finding confirms the continuing need for skills in writing automated tests, reflecting the persistence of this industry gap.

The high request level for test automation tools can be explained by the popularity of Agile development, where working in iterations and regression testing is essential; hence, a significant part of regression tests is automatized. Automating regression tests is the leading approach for maintaining and consolidating the quality of software systems. It is an ongoing process that requires knowledge on automated testing and corresponding tools [56]; consequently, we expect to see a constant rise in this type of demand in the future.

We also expect a clearer allocation for various test automation tasks, such as a behavioral consequence of not having it be shift blame between developers and testers, as reported by Wiklund et al. in their systematic literature review [57]. We find it surprising that only one in five job advertisers asks for test management tools. This misbalance may indicate a tacit requirement for test management tools skills.

The high request for Selenium does not come as a surprise, given that it does not require very technical skills to operate it, it is open-source, and it is portable through all major OSs and web browsers [58]. The high request for other specific test tools confirms the employers’ need for testers to perform tool-assisted work and that there is already a tool structure in place and waiting to be used, while the performance-testing tool demand also confirms the technical component sought in the profile of software testers.

**A common demand is test planning and implementation**

The requirement for skills in test planning signals that a significant number of testers are needed to actively contribute to high-level testing. Test planning means determining the scope and the objectives of testing; establishing the test approach; determining the test resources needs (human and logistic); determining the test schedule; establishing test metrics; identifying the potential risk; and mitigating actions, start/stop, and exit criteria. We infer that the testers are needed to manage this aspect of testing as well, as should do so above the typical testing tasks of creating and executing tests.

Test implementation assumes the ability to group the test cases into execution suites and sequence their execution for both manual and automated tests. Babinet and Ramanathan identify three major challenges in large-scale development: the unpredictability in feature development, conflicting priorities, and overlapping
release cycles [59]. In this light, we consider that our findings confirm and show the industrial need for a clear prioritisation of tests

**It is not common to ask for bug identification, test documentation, test data, or test closure activities**

We obtained two surprising results for the two least-demanded skills: bug identification and test closure. Even though one of the most important aspects of a tester’s job is to find bugs, few advertisers explicitly ask for it. An explanation is that it is a tacit requirement and is therefore not explicitly set in job advertisements. The other surprising result is that only 1% of the ads ask testers for test-closure activities. This activity assumes checking what deliverables have been delivered, archiving the test ware and documentation, checking that all the project issues and bugs are either closed or deferred, ensuring test-related information is handed over to other departments, and analyzing lessons learned for the future. A possible explanation for the lack of demand is that hiring companies do not focus on the activity itself. This result is also confirmed by Kassab et al. in a report on the state of practice in software testing, where closing the test process is not listed as a method or practice in testing [60]. It is also likely that employers assume that it is only the responsibility of project managers, as assumed in a study from 2014 [61].

We expected the industry to give higher importance to test documentation, test data, and test metrics. These three aspects of testing need to be mastered by all the testers within a team. It is not sufficient that only one tester in ten writes test documentation and one in twelve creates and maintains test data. We state this because team members see documentation as a useful and important communication tool [62], even if it is outdated [63], while the lack of test documentation can generate issues around the understanding of undergoing testing or the reproducibility of tests.

Improper test data can be costly because it can either break laws and regulations, such as GDPR [64], while oversimplified test data can hide serious product issues, especially in the performance area. The test metrics are closely related to test management and the predictability of testing; therefore, the lack of requests for it can flag a potential shortcoming in the employer’s skills demand.

Even though there is a low demand for risk management and exit criteria, these may be aspects that fall mostly under either the test manager or the project manager’s responsibility. Hence, there is no explicit need for testers to master this area.

**The software testers need to follow-up on bugs to reach a resolution, but the tools to use in bug tracking are unimportant**
The high request for following-up on bug reports is likely to be generated by consequences to the testing budget in case the follow-up is not done. The explicit request for bug reports can be an efficient way to formalize the testing process so that all the work around bugs is visible and budgeted. A reason for the request could also be that most software development implies managing a large number of bugs on a daily basis and fixing them is one of the most common and time-consuming activities for developers [65]; therefore, this activity needs to be measured and planned in detail.

Another surprising result is the low request for bug-tracking tools. The fact that only one in twelve job ads ask for it does not mean that one in twelve testers should use bug-tracking tools: it most likely means that testers use these tools and workflows with standard users’ rights, much like they use their email client. The result can be explained by the fact that a significant number of companies have bug-tracking tools that are set-up and maintained by a separate team-handling infrastructure. It is also likely that team members use the same bug-tracking tools as end users who report bugs, which indicates that learning how to operate these tools should be straightforward so that no or very few special skills are required.

**There is a high interest in acceptance testing and a lower interest in integration testing**

We expected that employers would ask the testers to have more skills in API testing, especially since our previous study [36] shows that 9% of the developers are asked to have this skill. We expected that transitioning software towards micro-services and cloud-architectures and the emergence of software supporting the Internet of Things (IoT) would have a greater footprint regarding the demand for all three kinds of testing, especially since the IoT exhibits a large number of resources, connectivity, and interoperability issues [66].

The most significant industrial demand seems to be placed on high-level testing, with acceptance testing and system testing occupying the first two positions in the ranking of skills for test levels. The result concerning acceptance testing is in line with our previous finding on soft skills [35], where there is a tendency towards asking testers for more customer focus. It is also in line with the fact that the lack of user acceptance has long been a major impediment for the success of new software systems [67]; hence, employers currently put more focus on it. Both these levels of testing assume that testers have a good grasp on the overall purpose and functionality of the software being tested.

We expected, however, to see more requirements for distributed software because testing web systems assumes testing different characteristics and doing so with
different tools than mobile systems or desktop systems [68]. Therefore, the lack of specificity in this area is most likely a lack in the industry’s demand.

Given that review skills rank the last in the testing-related demand formulated by the industry, we consider that testers are still not involved enough in the review process. A significant part of the cost issues and defect sources are found in the design phase. Our result confirms Lee’s finding that, of all testing activities, designing tests is practiced by the fewest companies [69]. The fact that only one in eight testers is asked to perform reviews is concerning.

It is natural for testers not to be asked to have static analysis skills. Even though it is a form of testing, static analysis is performed with dedicated tools directly on code by developers and architects; it is done mainly to treat symptoms in the code that would facilitate defects and failures.

The software testers have to undertake software development and performance testing

An interesting result is the demand for testers to have performance test tools. Performance checking, which used to pertain exclusively to developers, is now asked of testers, too. If we compare this with the same requirement for developers, it appears that the demand for these skills is slightly higher for testers than for developers [36]. The finding indicates a shift in responsibilities for performance testing towards testers, divergent from the common practice two decades ago [47].

Moving our focus towards technical skills demand, the advertisers’ need software testers to write their own SQL queries when performing testing, therefore, the highest request for this skill. The result is can be explained by two essential aspects of testing: the test data and the verification of the results for the execution of tests. It is a common practice to store data in databases [70]; therefore, the test oracles and the test execution results are to be found in such structures. Nonetheless, databases are heavily used in automated testing, reflecting the need for testers to have query-language-related skills.

Not surprisingly, the demand for testers to manage parts of project development comes as an extension for the test-management request. We observe that, apart from the industrial request for a technical profile, testers also have to possess strong managerial skills. Project management is a step up from test management, but it has many similar traits with [71], such as prioritization, estimation, resource allocation, risk management, defining and following-up metrics, starting, stopping, and resumption criteria. Therefore, it is natural to ask testers to take responsibility for aspects of managing the project, given that their skills in test management are transposable to project management.
The software testers need to be quality assurers and project managers and act as customer support

In addition to the classic testing-related requirements corresponding to their role, a significant proportion of employers ask testers to master knowledge in either the QA field; business requirements elicitation and handling; customer support; or project management, such as risk, release, and resource management. Most of the demand is towards QA, and it is not coincidental that even though the nature of the two job profiles is different [72], many companies merge the roles of testers and quality assurers into one profile. While testers need to check the conformity of the developed software with the software’s requirements, determine fit for use, and aim to discover bugs, the main responsibility for QAs is to establish, control, and maintain quality criteria for the software under development and to make sure that the internal and external quality standards are being followed and implemented.

Also, it is natural to involve software testers in the incipient phase of requirements management. As Nuseibeh and Easterbrook report [73], better modelling of problem domains, as opposed to the behavior of software, is a key factor for the development of successful software systems. Testers play an important role in understanding the business side of the software that is to be built and following it up through the software-development process. Requirements management is also an extension of the tester’s job to analyze the testable parts of system requirements, to group them, and to priorities them. Therefore, asking testers to handle the elicitation and design of business requirements, transform them into system requirements, and follow them up with relevant stakeholders is another case of early defect prevention in which the testers’ skills are reused at the project level.

However, testers are not asked for too much on the business side of the companies. They are not asked to assess or handle the business model of the hiring companies nor to recommend software solutions to meet the business’s needs. The lack of requirements in this area is understandable in that no test-related skills can be reused in the area of business processes.

Regarding development models for software, a recent study [74] states that 70% of all development projects follow the Agile development model. Combining it with our results, it seems that Agile has become such a widespread practice that the need to specifically ask for it is no longer relevant. Therefore, employers do not frequently ask for Agile-related skills; instead, they just assume testers have them. At the same time, the low request for sequential development models signals that Agile has become the principal state of practice across industries.
There is a strong preference for technical testers

With two out of three ads asking for technical skills, we can state that there is a clear preference for testers with technical abilities. Programming and scripting languages are a very common request, along with the request that the testers be familiar with the various approaches in the development and deployment of software. These kinds of requests add a significant dimension of complexity to the profile of software tester.

Moving our attention to software development tools, we note that Jenkins and similar tools for continuous integration are commonly asked for. The most likely reason for this is that these tools are not tailored solely for developers but are part of operating a continuous integration model [75], common in Agile and used across multiple roles. We observe, therefore, an industrial preference for testers with competencies in the technical areas of integration, deployment, source-control management, and build management. These requests for skills are currently a rather common demand within development teams [76]. On top of this, a common practice is that testers use the same tools as developers to trigger new builds or to deploy builds in the testing environment. Ideally, all these activities need to be done without developer assistance, resulting in the demand for such skills. Still, the level of the demand is not high, and an explanation for it is that these tools are easy to operate for standard usage; therefore, specific skills are not necessary.

It is natural to ask more competencies for Unix/Linux-based OSs than for other OSs, as they are highly configurable, exposing several thousand configurable features on over two dozen architectures [77]. It is unclear why the low demand for mobile OSs exists. It would have been reasonable to see more demand in this area, as testing on mobile devices with different user profiles, performance measurements, access rights, and security settings are all common. A possible explanation is that testing is done on preinstalled OSs with standard settings, with the possibly of only minor changes being made to it. It is unclear whether the lack of testing of software with various configurations of OSs is not being done because of a lack of time, lack of resources, or lack of specifications or needs in customer contracts or user agreements.

Domain specific knowledge is a less important request

With one in three ads needing testers who have knowledge in domains other than software, this request is not negligible. However, reading the result in reverse, we notice that two thirds of employers do not ask for domain specific skills. We can thus state that the employers search for testers with testing skills more than personnel who are specialised in their industrial profile. Most ads asking for domain
specific knowledge pertain to the area of financial services software (online transactions, banking systems, payments). The request can be explained by the high number of rules and regulations that financial services [78], including software, have to be compliant with, therefore assuming testers need the skills related to testing it. Seldom do the telecommunication, software, gaming, and other industries ask for domain specific skills. There were no significant requests from other branches of these industries, such as infrastructure, governance, IT, health care, tourism, agriculture, and defence. From this point of view, we attain the result that software testing is a role considered to be portable across industries by employers.

There is no set profile of the software tester, but there are correlated skills in the employers’ demands

In this section, we discuss our second research question: “Are there any correlated skills shaping the profile of the software tester?” The two correlation matrixes show that there are no set and clear patterns in the software tester profile emerging from employers’ hiring ads. However, we discovered four interesting correlations. Looking into the details of the request for skills, the result implies that employers prefer software testers who master the testing process in its entirety from conception to completion, not fragments of it. The connection between bug-tracking skills and testing related to changes is most likely due to the following: the more software changes, the more likely is to find a more bugs and, therefore, to have a formal way to follow-up on them from find to fix.

The correlation between technical competencies and software management tasks is important, as it signals that even though employers prefer technical testers, they do not ask for purely technical testers, for which it would be sufficient to have the ability to program and manage tools.

Regarding the domain expertise, our findings point to a profile of a domain specialist who is in touch with the technical aspects of development, but will most likely not work with programming, debugging, or carry load and stress tests.

Regarding the accomplished testers, experience plays an important role during hiring, especially as a replacement of formal education and other certifications. However, it remains unclear why employers tend to ask for fewer soft skills from more experienced people.

Focusing on the soft skills, we observe that the employers who need their software to be tested in multiple ways ask for fast learners. This suggests that the various techniques of testing, such as code analysis, use-case testing, boundary values, dry runs, and so on, are time consuming and require significant amounts of effort to master.
An interesting result is the link that the employers make between team-playing skills and the ability to define and use test metrics. More precisely, by looking into the ads, we found that employers need the testers to define metrics, to collect them, and to follow-up on the metrics and the test coverage. Our finding confirms the point in paper [79], which depicts communication as a mandatory part of the metrics life cycle. Employers need to educate the testing team about which information has to be captured to process the metric. Therefore, this is a likely explanation for the correlation that emerged from the employers’ hiring advertisements.

The values of the Spearman correlation index signify that there are no clear predetermined career paths that software testers can take. Moreover, there is no predominant or strongly correlated skills demand that depends on the job title being advertised (tester, test analyst, test manager), nor on the seniority of the position (junior, middle, senior). We can infer that the job description and, consequently, the skills demand is determined by the everyday needs of a company or the immediate needs of a project rather than a standard distribution of responsibilities in software testing. As paper [33] shows, there is no one practice in software testing, and the same seems to be true when hiring software testers.

**Seniority, previous experience, educational attainment, and certifications**

Regarding our third research question, “What education attainment, certified qualifications, and previous experience do hiring companies ask from software testers?”, formal education does not seem to be a focus of employers when they hire software testers. We notice only a moderate demand for testers to have graduated from a certain education program; only half of the ads explicitly ask for a degree from this program. A negligible number of ads ask for the completion of high school studies. The vast majority of the ads asking for education attainment request a bachelor’s degree in computer science or a related domain. Very few of these ads ask for a completed master’s, and even fewer ask for PhD studies. By comparing the level of request, we can affirm that skills are in more demand than educational attainment.

Certifications are a common way to complement formal education. However, the industrial demand for such documents is still on the low side compared to classic university diplomas; one in seven hiring ads asks for certain certifications within software testing. By one order of magnitude, the most popular demand is for the ISTQB Foundation Level Tester certification, followed by the ISTQB Advanced Level Tester certification [80]. Very few ads ask for certified knowledge in security, penetration testing, or application life cycle management. It seems that the employers usually want the testers to have general knowledge of the testing field. A small number of specialized jobs ask for other software testing certificates. The
certifications are demanded in addition to the requirements for completed studies. However, we expect an increase in the demand for training and testing-related certifications. The study [81] performed by Garousi and Junji in 2016 points in the same direction by showing an increasing attention given in Canada to certifications and training on software testing.

Seniority-related requests do not seem to be a focus of employers. This finding supports a study from 2010 [82], which finds that the software development experience cannot replace the demand for a systemic identification methodology and that experienced testers were not necessarily better than inexperienced testers in every aspect.

7 Implications

This paper contributes to the broader field of skills study with the profile of the software tester as currently asked for by the industry. For this specific profile, we distinctly show the type and level of demand for test-related, technical, and domain-related skills. In addition, we show the principal requirements that the employers have for test managers in terms of skills. We also provide insight regarding the type of education required, the certifications needed, and the level of experience sought after by the employers. To provide this information, we have developed and used two taxonomies to group the testing skills and to group the technical skills demanded of testers.

In line with the study by Capretz et al. [14], which notes that software testers need to choose from an immense range of possibilities and that they need to maintain a high level of attention to detail, our results show that the testers need both broad views and attention to detail. Therefore, it is likely that this multitude of requests is vast in nature and requires the expertise of many different kinds, which can be a reason that there are few people, indifferent of the personality type, who prefer to work as software testers [14].

Our study can be used by employers to calibrate their requests for testers when hiring. In case the job advertisement does not attract testers with exactly the set of skills requested, the employers could either request less or split the responsibilities in other ways. Additionally, our study can serve as a mapping tool for employers to identify skill gaps in the development team. Dependency on key people is reported by Wiklund et al. as the most dominating issue in Agile organizations in need of improvement [83]. Therefore, employers can use our results to examine the spectrum of skills in their development teams and identify how those skills are spread among the team members, compare them with the skills currently in demand.
in the industry, and either train the existing personnel to gain more skills or hire more skilled testers to the teams.

Our study shows that some of the most in-demand skills focus on the process of test automation. Automation does not mean that the testing happens automatically without the testers having to put work into it; automation is used to improve the effectiveness of testing [84], reduce human errors [85], and improve the reliability of testing, but it is not a human labor replacement. To write, execute, and follow-up on automated tests, a tester must have time and invest a good amount of effort into the process. Therefore, employers need to check whether the testers have the time and skills to develop and maintain a good automation process. This aspect is important, as Karlström et al. report that the main impediments to adopting software test automation are of a managerial and not a technical nature [86]. Also, employers can give more time and the effort needed to maintain the testing infrastructure, as poor-quality infrastructure can be a hindrance for testing and a significant source of loss in software projects [87].

Because there are no clear career paths to follow, software testers can at least gain a better understanding of the skills that are currently important for employers during the hiring process. They can work towards increasing their own skills by taking on tertiary education or certifications or by practicing their skills, especially in the area of test design [88] and test planning [89], test management, performance testing, or acceptance testing. These areas have been confirmed by our study to be important and in-demand in the industry. They can also catch up with their technical side by practicing their programming skills with SQL queries, Java, or C#. They can reflect on how their skills can be transferred to the project management level, especially in the areas of estimation, risk management, planning, reporting, and control.

Our findings show that test managers have to have about as many test-related skills as the testers, and they are asked to have a strong technical profile. The fact that there is a much stronger request for managers to master testing – particularly automated testing – than to manage people, shows the substantial need for qualified leaders who can drive the test automation process. Our previous finding [35] on the demand for independent testers leads to the conclusion that, for the employers in software testing, people management less important compared to the technical qualification of test managers.

The reflection of industrial needs in academic courses and practice has been a long-standing issue. As Garousi et al. state in a recent report [90], even though the software development industry and computer science tertiary education providers are two large communities, the number of joint activities is low. Therefore, an important improvement step towards for the current situation would be for the
tertiary education providers to update and expand their curriculum to include the industrial needs, as the results of our current study show. We find the demand for a strong technical profile of the tester particularly interesting. Our results are in line with report [90], highlighting that, with a large margin, the two biggest issues in the industry that are reflected directly in requests for academia are test management and test automation. Therefore, education providers can check whether they offer enough on the side of test automation: from writing to executing tests and setting up automated testing frameworks, preparing data and test oracles, reading logs, following-up on execution results, and so on. It is also important to give the students the chance to practice the continuous integration and deployment of code. In case there are course offerings in software testing, it is important to provide the students the chance to practice high-level test approaches in addition to writing tests. The results of our study show that planning tests, allocating resources, prioritizing tests, and designing tests from requirements are highly desirable skills which require practice.

8 Limitations

Obtaining the software tester profile with an industry perspective at hiring is a useful finding in determining what the employers need for the tester role. We made the research data, the protocol we used to create the skill taxonomy and the analysis results publicly available for access and download [44, 49, 50, 91], so that other researchers use it and assert it.

On the other hand, establishing the desired profile of the role at hiring does not mean we can equate it with all skills testers are being asked for throughout their careers. Nor we can equate the testers with the entire testing process; other abilities and critical competencies are the responsibility of other roles, such as developers, managers, customer support, contractors, stakeholders, and the like. Listed below, we have identified other potential threats to the validity of our research.

Generalization bias: Our analysis stands on observations that were collected from job ads produced by the industry. We cannot generalize our findings to any demand other than industrial or to the demand for the whole job-hiring process. We can also not infer conclusions on the demand for existing employees. We mitigated the limitation using a clear definition of the scope of the research.

Comparison bias: There is no previous study focused on the same parameters as the ones we analyzed to make it possible for us to look at trends in the demand. Our study is, therefore, a snapshot of current industrial needs. We mitigated the limitation by finding comparison points with other studies and research and
conducted comparisons where possible. We have also used those studies as a base for discussion and for drawing the implications of our study.

Sampling bias: The data were collected at the end 2016 and the beginning 2017, which means that our current analysis results have limited validity, given that it is most likely that industry needs will change over time. An additional limitation for the sample we used in our analysis is that we relied on the search engines’ intrinsic logic when we searched for the testing jobs. It could have been that there were additional testing jobs that these engines did not display for the search phrase “software tester”. This limitation is palliated by the fact that the search results we obtained are the same as the search results of the job seekers.

9 Conclusion

In this paper, we performed an empirical analysis of the software tester role in which we shaped its profile, as requested by the current needs of the industry. The focus was on making a structured inspection of the skills needed for the role, for which we grouped the demand into corresponding categories in a taxonomy.

The result of our analysis, performed on 400 job ads, shows that software testing is a distinct role that involves a high number of specific competencies that have a strong technical component and a managerial component. Designing of tests is the main industrial demand of software testing skills. The automated testing skill is more asked for than the manual testing skill. It is common among employers to ask the software testers the ability to conduct test planning and implementation. However, it is not common among the employers to ask for skills in bug identification, writing test documentation, generating test data or managing the test closure activities. The software testers are needed to follow-up the reported bugs to resolution, but the employers do not ask the testers for previous experience with specific bug tracking tools. Among the employers, there is a high interest in acceptance testing and a lower interest in asking for integration testing. The software testers are expected to undertake software development tasks and the responsibility for performance testing. Moreover, they are needed to act as quality assurers and to undertake parts of project management and customer support. None the less, there is a strong preference for software testers with a broad technical expertise.

Employers are not likely to be asking for domain specific knowledge during the hiring process, which denotes the good portability of software testing knowledge across industries. Formal education attainment is moderately important, with relevant bachelor’s degrees in computer science or related fields being mostly asked for. One in seven employees is asked to hold a software testing certification. The
industry is split in half regarding the demand for previous experience; the half that requires it usually does not demand more than three years of experience.

By exposing the industrial need for various abilities of the software testers, the main contribution of the study is to advance the understanding of the division of labour in software development teams. Empirical studies can be carried-out to analyse the skill demand during the hiring process. Furthermore, the profile of software testers can be broadened with an empirical investigation of skills demand for testers who have already been hired. The current study can be repeated, to observe how the industry shifts its preferences in skills demand.

10 References


43. ISTQB. ISTQB. Available from: https://www.istqb.org/.
54. Penov, F.P., et al., Test execution framework for automated software testing. 2005, Google Patents
55. Moe, N.B., et al. Continuous software testing in a globally distributed project. in Global Software Engineering (ICGSE), 2015 IEEE 10th International Conference on. 2015. IEEE.
90. Raluca Florea, Viktoria Stray. Processed data, software testing jobs. 2019; Available from: https://doi.org/10.6084/m9.figshare.8294054.v1

Ralucă Florea  
Department of Informatics  
University of Oslo  
Oslo, Norway  
ralucamf@ifi.uio.no

Viktoria Stray  
Department of Informatics  
University of Oslo  
Oslo, Norway  
stray@ifi.uio.no

Abstract

Developing software with high quality is challenging in distributed software development. The purpose of the current study is to investigate the testing skills and tools required in the ever-changing world of global software engineering, according to industrial needs. We analysed 500 job ads from 33 countries. The results show that a quarter of the testers and a fifth of developers are asked to work in distributed projects. The testers are asked to be highly skilled in a variety of test activities and tools, while the testing-skills demand for developers is low and somewhat vague. The profile of testers has a strong technical component in addition to the managerial one. Our findings show that employers need most that testers are competent in automated testing. Furthermore, the industry does not cover all aspects of testing with the demand for testers and developers. Surprisingly, neither role is asked to test the implementation of the general data protection requirements. Our study bridges the industrial needs and the practitioners’ skill development process.

Therefore, software testers can use our study as a reference point to enhance their skills. Employers should use our results to check their testing-skill coverage within the development teams. Tertiary education providers are encouraged to use our findings, to update the curriculum in the software development area.

Keywords— Software testing skills, Software testing tools, Automated testing, Manual testing, Unit testing, Global Software Engineering, GDPR

1 Introduction

Software engineering is a complex process by nature [1], whether the software size is small or big, whether built by few or many. This complexity is mitigated by the skills of the team members, who need to master, besides communication and teamwork, the technical aspects and the tools to develop, test, deploy and maintain.
software systems [2]. In the context of global software development, it is even more relevant to reveal the industrial demand for software testing skills, because software testing plays an essential role in development and, moreover, global software testing has witnessed an increased adoption over the years [1]. Additionally, distributed software development affects the quality of the delivered software [2].

Team members in distributed projects, whether developers or testers, are asked to master a large specter of technical skills. Moreover, as they tend to be agile, these teams need to have all the expertise necessary for every phase of developing software [3]. Additionally, there is a lack of studies regarding skills and tools for software testing, and quality in distributed software development [2]. In this light, we want to investigate which technical skills and tools testers and developers are asked to master with regards to software testing.

Skilled individuals are needed to implement the development and testing of software. Just as with development, both soft and hard skills are necessary to perform software testing, a process which involves the coordination of different roles, skills, and expertise [4].

Our aim is to provide a reference point to testers surviving and navigating the world of global development and engineering, who plan to enhance their testing skills, according to the current industrial needs. Furthermore, our study provides relevant information for employers operating in the global software engineering market as well. Even though they might have a good control on the skill demanded for testers, employers in the global software market might lose the overview of the testing skills needed in the whole development team, therefore not being aware of uncovered gaps between developer and testers skills in software testing. This aspect is relevant for employers approaching the organization of human resources at a global and at a local level, given that the way the software teams are grouped and organized has a high impact on the outcomes of the global software engineering [5].

The study also reveals the testing tools most in demand. This information is relevant for the employers and helps them in their choice of testing tools or tools-related testing skills at hiring. As mentioned by Tell and Babar in their mapping study of tools in distributed teams, distances negatively impact the quality of software products, therefore the use of suitable software tools becomes of major importance [6].

Monasor et al. [7] identified in a systematic literature review on teaching global software development that students need to gain knowledge about tools (e.g., control version tools), methods, data and how to manage uncertainty. Due to this, our study can be used by tertiary education providers to update their curriculum and education
offerings in the software development area, to reflect the current industrial requirements.

In an industrial study from 2018 [8], we outlined the soft skills that testers are required to possess, to best perform their job. The results showed that the levels and kinds of soft skill requirements changed significantly over just five years, with new trends emerging. In our current study, we advance these findings by examining the demand and the trends in the request for hard skills on software testing, for testers and developers, with a focus on software testers. Further, we investigate the industrial demand for testing tools and testing environments. Such an investigation exists for development tools used in agile teams [9]. There are studies on the communication skills between developers and testers in agile distributed projects [10]. But the lack of research on the skill demands and coverage needs focused on the technical side of software testing motivates us to find more on the industrial trends in software testing. Because of its importance and proximity, we also focus on the measures taken by the industry at hiring, regarding the testing of the implementation of the general data protection regulations (GDPR), mandatory from May 2018 [11] in the European Union. Even though originated and applied in Europe, GDPR has worldwide implications since business models and software developed in the United States, Canada, Australia run, store and process data on the EU territories, along with European software developers outsourcing testing to countries such as India, China, Vietnam, Mexico, South Africa and many more.

In a previous publication, Burnstein [12] approaches and enumerates eleven testing skills. We found that the skills listed are rather general, such as “understanding testing principles and practices”. Therefore, aiming to go more in-depth on what is actually required from testers, we targeted a more concrete set of test skills, we separated the testing skills from the rest of the hard skills such as programming, and we focused on the demand levels for each of the testing skills. Furthermore, Burnstein [12] approaches the testing tools with a general perspective and discusses the need for all kinds of tools in function of the test maturity model (TMM) employed. As opposed to [12], we excluded from our analysis project management tools, development and deployment tools. In our study, we focused specifically on the skills requirements for testing tools, and we performed our tool-related analysis on the testing activities the tools serve.

We used the Foundation Level Syllabus [13] to classify the required testing skills and to group the requirements for testing tools. In order to identify trends and possible gaps, we compared our results with the study of the industrial need for testing skills performed by Scott et al. in 2004 [14].
To define the object of our research, we approached the testing skills as seen by ISTQB, a combination of testing-related theoretical knowledge and practical skills derived from training and experience, and the ability to apply them in real-life situations [15]. The tools we analyzed in the current study are grouped into test automation, performance testing, static analysis, bug reporting and test management tools [15]. We understand hard skills as they are approached in [16], a person’s ability to perform a certain type of task. In our current study, the tasks refer to software testing, whether an independent task, or as a part of the testing process.

Our focus is to empirically investigate and bring into view the profile of testers and developers with regards to software testing, as currently demanded by the industry. Consequently, we divided this larger topic into the following research questions:

*RQ1:* What kind of software testing skills are most sought after by employers, when they hire software testers?

*RQ2:* Which qualifications are developers asked by the industry to hold regarding software testing?

*RQ3:* Which testing tools and which kind of competence on the GDPR compliance are required of both testers and developers?

## 2 Data collection and analysis

Analyzing the contents of job advertisements is an established method to observe trends in professions pertaining to academia [17]. We collected 500 job advertisements from 33 countries, using online job-search engines. We chose the most significant job-search engines by the number of users and the number of jobs posted. A total of 65% of the advertisements we collected were in English and 35% were in multiple national languages, which we had translated into English. The translation of all ads was cross-checked by at least two translation tools and 90% of the translated ads were also checked by a native speaker or a fluent software professional. We read each job advertisement and looked for requirements for testing skills and those related to testing tools.

Four hundred of the advertisements we collected were for testers and one hundred for developers. We, therefore, obtained a 95% confidence level and a ±5% confidence interval for testers and a 95% confidence level and a ±9.8% confidence interval for developers [18]. The advertisements for testers were collected between December 2016 and February 2017, the ads for developers in February 2018. The advertisements were collected from USA (119), Canada (68), UK (40), Norway (32), Australia (32), India (29), Argentina (17), France (17), Mexico (15), South
Africa (14), China (14), Vietnam (13), Sweden (10), Spain (9), Germany (8), and other countries (63).

In our search of job advertisements for software testers, we looked at posts for job titles that included testers, QAs, technical testers, usability testers, performance testers, game testers, financial-system testers, and so on. For developers, we analyzed advertisements for programmers, web-developers, mobile developers, full-stack developers, front-end and back-end developers, software engineers, software architects, and devOps. Of all ads, 73% mentioned explicitly that the hire is for agile teams. We assume in our investigation that there are no significant differences in the demands for skills from agile team-members and non-agile, as the results in our previous study [8] show.

We gathered job advertisements posted by both software development and consulting companies. We included job advertisements from the public and the private sectors. The companies varied substantially in size and profile of software being developed and included, among others, Amazon, Tesla, Microsoft, Capgemini, Bank of America, NBC, VISA, Expedia, Texas Instruments, IBM, Nokia, New South Wales Government and National Bank of Canada. Many of them practice distributed software development.

3 Results and discussion

The employers ask explicitly in the job advertisements 24.5% of the testers and 19% of the developers to work in distributed projects or teams (see Table I). We believe the actual number of testers and developers that has to work in distributed software development might be higher since not all employers will explicitly state it in the ad. In an earlier study, we found that 37% of developers work in distributed teams [19]. Furthermore, as shown in Table I, the testers need to master ten testing skills, while developers need to master two testing skills.

<table>
<thead>
<tr>
<th>Table I. Demand for testing skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ads analyzed</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Percentage of ads specifying</td>
</tr>
<tr>
<td>distributed software development</td>
</tr>
<tr>
<td>Percentage of ads asking for</td>
</tr>
<tr>
<td>testing skills</td>
</tr>
<tr>
<td>Total no. of testing skills</td>
</tr>
<tr>
<td>Avg. no. of testing skills per ad</td>
</tr>
</tbody>
</table>
Demand for testing skills of testers

We ranked and listed the testing skills according to the demand for them (the number of ads asking for a certain skill). From the ads for testers, we analyzed the 15 most sought-after testing skills (see Table II; we omitted the demand for testing tools from the table as they are presented separately in Section III.C).

Table II. Testing skill demand for testers

<table>
<thead>
<tr>
<th>Testing skills</th>
<th>% ads</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1. Write automated</td>
<td>46</td>
<td>“You have a minimum of 2 years’ experience of programming and/or experience to write automated tests”</td>
</tr>
<tr>
<td>T2. Execute automated</td>
<td>40</td>
<td>“Run automated tests for every compilation”</td>
</tr>
<tr>
<td>T3. Execute manual</td>
<td>38</td>
<td>“Perform manual testing where needed to support development of test automation, or when automated testing is not appropriate or cost effective”</td>
</tr>
<tr>
<td>T4. Write manual tests</td>
<td>38</td>
<td>“Experience writing, editing and executing manual test suites is required”</td>
</tr>
<tr>
<td>T5. Test planning</td>
<td>34</td>
<td>“As a Test Leader, you will be responsible for ensuring good test planning together with the supplier and other 3rd party suppliers involved in the solution.”</td>
</tr>
<tr>
<td>T6. Requirements</td>
<td>26</td>
<td>“Expected to organize requirement gathering sessions, eliciting requirements, and model the processes […], requirements management and communication, planning and monitoring the requirements, requirement analysis”</td>
</tr>
<tr>
<td>T7. Functional testing</td>
<td>21</td>
<td>“Creation and maintenance of critical (and complex) Functional Test Cases.”</td>
</tr>
<tr>
<td>T8. Performance</td>
<td>13</td>
<td>“Architect and develop your own tools to facilitate and enhance performance testing”</td>
</tr>
<tr>
<td>T10. Integration</td>
<td>11</td>
<td>“The position requires […] integration test expertise, […] attention to detail and significant experience working with Databases and SQL to verify data accuracy.”</td>
</tr>
<tr>
<td>T11. System testing</td>
<td>11</td>
<td>“Experience in system testing, data warehouse migration testing, data integrity testing and data transformation related testing.”</td>
</tr>
<tr>
<td>T12. Web-systems</td>
<td>10</td>
<td>“4+ years of testing experience with emphasis on mobile and Web applications using automated tools”</td>
</tr>
<tr>
<td>T13. Validation</td>
<td>10</td>
<td>“Conduct risk analyses, verification, validation activities and formal design reviews as applicable”</td>
</tr>
<tr>
<td>T14. Regression</td>
<td>10</td>
<td>“Experience creating regression test suites”</td>
</tr>
<tr>
<td>T15. Acceptance</td>
<td>9</td>
<td>“Support the various testing phases, which include API integration, End to End testing, and User Acceptance”</td>
</tr>
</tbody>
</table>
If we look at the industry needs for testing skills as measured by the study performed by Scott et al. in 2004 [14], we observe that the most important was integration testing, followed by unit testing, security, and validation. Next were performance testing, acceptance, followed by usability, beta-testing, compatibility, white-box testing and black-box testing. Last were recovery and regression testing. From the start, we see important new trends emerging. Requests that were not in demand ten years ago are now among the most requested testing skills, such as automated testing (T1, T2), test planning (T5) and requirements analysis (T6). Inversely, the demand for specific types and levels of tests such as integration testing (T10) seems to be decreasing, as are unit and security testing. Lastly, we notice a similarity between the study [14] and ours on the need for testing related to changes, both placed at the bottom of the ranking.

The ranking we obtained shows a clear shift of the tester towards a more technical profile. In addition, we observe a strong request for managerial competencies, based on the level of demand for high-level testing knowledge and testing organization. Moreover, testers are asked for skills in testing software performance and the ability to manage their testing tools and testing environments.

Our results show that 97% of job advertisements require testers to have testing skills, which means that almost all employers regard testing as qualified work and in consequence look for skilled employees to fill the role. Not only this but with an average of ten skills required per advertisement, we note that testers are asked to performed highly-skilled testing activities. However, only half of the advertisements for developers ask for any kind of skills in testing. This result does not automatically imply that developers do not need to be skilled to perform testing; instead, the low requirement for developers with testing skills most likely points towards a low industrial focus on testing skills for developers.

As a response to the first research question “What kind of software testing skills are most sought after by employers, when they hire testers?”, we observe that testers are most asked to write and execute automated and manual tests (T1, T1, T3, T4). While we see the same level of demand for writing manual tests and executing them, we notice a higher demand for writing automated tests (46%) than executing them (40%). The difference in the two latter can be explained by that, on the one hand, writing automated tests is a more skilled job than executing automated tests, and therefore employers ask specifically for it. On the other hand, it is likely that a significant number job advertisers not asking for test execution skills already have frameworks in place for the pre-scheduled execution of automated test scripts [20], and therefore such skill is not needed from testers.
Our finding supports the report [21] on the importance of automated testing in global software engineering and distributed agile teams. The results are also in line with the importance of automated testing recognized in research and industry [22]. Many organizations have extensive automation infrastructure, but they lack the competent employees to make full use of it [23], which explains the high industrial demand regarding test automation. It looks like developing the test automation competencies of employees lags behind the development of test automation infrastructure; our result points that it is likely that the strategy employed by the industry is to draw level by bringing such competencies into the teams through new employees, therefore the highest need for it at hiring.

<table>
<thead>
<tr>
<th>Testing skills</th>
<th>% ads</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1. Testing</td>
<td>30</td>
<td>“Participation in testing efforts”</td>
</tr>
<tr>
<td>D2. Code reviews</td>
<td>23</td>
<td>“Take responsibility for quality assurance by performing regular code reviews”</td>
</tr>
<tr>
<td>D3. Test automation</td>
<td>13</td>
<td>“Experience of Continuous Integration and test automation”</td>
</tr>
<tr>
<td>D4. Unit testing</td>
<td>11</td>
<td>“Manage quality assurance by maintaining high code standards, design principles and source code analysis tools through to the use of continuous integration, automated unit testing, deployment automation and code coverage tools.”</td>
</tr>
<tr>
<td>D5. Debugging</td>
<td>9</td>
<td>“[…] you will plan, design, analyze, develop, code, test, debug and document programming to satisfy business requirements […].”</td>
</tr>
<tr>
<td>D6. Application testing</td>
<td>9</td>
<td>“Test and debug the application across supported environments, ensuring delivery of quality product”</td>
</tr>
<tr>
<td>D7. Design reviews</td>
<td>7</td>
<td>“You will participate in design reviews, providing input to the design recommendations.”</td>
</tr>
<tr>
<td>D8. TDD</td>
<td>7</td>
<td>“Create automated unit tests using a Test-Driven Development approach”</td>
</tr>
<tr>
<td>D9. Security testing</td>
<td>5</td>
<td>“Experience in security analysis of web applications ranging from threat modeling to penetration testing”</td>
</tr>
<tr>
<td>D10. Coding standards</td>
<td>5</td>
<td>“You’ll do all this while maintaining high standards, adhering to the team’s conventions and constantly improving code, even if it wasn’t yours to start with”</td>
</tr>
<tr>
<td>D11. Performance testing</td>
<td>5</td>
<td>“Development, unit, system, UAT, performance testing and test automation”</td>
</tr>
</tbody>
</table>

Test planning and requirements analysis (T5, T6) are tightly linked and both are highly ranked in the demand for testing skills.

Test planning refers to the tester’s ability to provide information about the specific parts of software that will be tested and the pieces of software that will be left outside of the scope of testing; the ability to specify the extent to which testing will be carried out, how, and in what order; what equipment is necessary; and who will
be responsible for specific activities within the process of testing [24]. Requirements analysis skills involve the ability to extract from the business requirements the parts that can be implemented through software and to specify the testable parts of the software that can be validated against those requirements [25]. Both these activities have in common a high-level approach to testing [26], and assume special abilities to implement them: the high ranking of these two requirements comes from a need for testers with a broad view of the testing process and with the ability to trace their testing actions. A likely explanation for such a high level of demand can be found in a study from 2004 [27], which noted that around 85% of the companies use test management documentation, but they lack people with the right competence to make use of it. Therefore, the need for people with skills in the test management area has been observed since 15 years ago and has remained in the industry’s focus throughout time.

We consider the demand for specific test types or levels to be moderate, in the middle and towards the end of the ranking. At the leading edge, there seems to be a demand for functional testing (T7), followed by performance testing (T8), system testing (T11), integration testing (T10) and acceptance testing (T15).

The skills required for testing web applications (T11) are the most sought after when it comes to particular software architectures to be tested. This may be because web systems have characteristics whose testing assume mastering a large spectrum of testing skills: testing across devices, on multiple networks, with different loads and at different capacities, balancing requests, handling user traffic, security aspects, web APIs, and many others [28]. Even though testing such systems assume specialized skills too, the demand for skills in testing mobile systems, desktops systems, and embedded systems is low and does not appear among the 15 ranked test skills.

The design of tests (T9) is ranked in the lower-middle part of the requests for testing skills. Our result is in line with Lee et al. [29], whose findings show that of all testing activities, the design of tests had the lowest percentage of companies practicing it, indicating that hiring companies most likely do not acknowledge the importance of this activity and therefore prioritize other requests.

Testing related to changes (T14) does not seem to be in high demand, as the request for it occupies the bottom of our ranking. Validation testing (T13) also ranks towards the bottom of the demand for test skills. Even though these activities should occur throughout the process of testing, the low demand may be explained by that testers do not need special skills to essentially perform checks of expected results against obtained results of testing.
Demand for testing skills from developers

While the broadness of test-skills demand for testers is explainable through the large spectrum of activities that testers need to cover, the request is surprisingly low for developers: we only identified 11 varieties of test-skills requirements for developers (see Table III). Given the many testing activities that should fall under the developer’s responsibility, there is no direct justification for the result. Also, if we look at the level of industrial demand in the study [14], it seems that employers do need developers to have testing skills in order to perform their daily activities more than they ask for in job advertisements.

Our second research question was “Which qualifications are developers asked by the industry to hold regarding software testing?” The most sought-after testing skill for developers is expressed simply as “testing” (D1), a term that indicates the scope of the desired competence but lacks precision. This finding indicates that there is some confusion in the job market about what testing skills developers should possess. Only one in three developers are explicitly asked to have this general “testing” skill, and this level of demand may be insufficient. While the employers are concrete in the requests for testing skills among testers, they are not as precise in requests for developers. A recent survey [30] of 100,000 developers reports that about 6% of developers are allocated full-time to testing; the task of testing for developers is most likely shared by the whole development team and is performed as an integral part of the developers’ job. Therefore, only 30% of companies with a demand for testing for developers is not enough to fulfill the testing tasks falling under the developers’ responsibility.

Our results show that the second most demanded test-related skill for developers is code reviewing (D2). While it is true that static analysis together with code reviews are effective means to improve the quality of software [31], the approach is not essentially testing; it is a way to complement the testing activities. In many teams, the code review is part of the standard work, not intended to find bugs, but rather to check for omissions, incomplete implementation of requirements or code-styling discrepancies. Code reviews are also used as a learning tool, to disseminate knowledge in the team.

The request to follow coding standards is rather low (D10). The explanation is that the coding environments and the deployment tools used by developers have built-in instruments to enforce coding standards. The standards are most often set in place by architects or technical leaders, in accordance with a company’s existing guidelines, audit compliance requirements and so forth. Most of these tools do not allow checking-in code unless it first passes the established coding standards.
In the advertisements, 13% of developers are asked to write automated tests (D3). This indicates that in around one in eight companies, developers, in addition to testers, have to contribute to the process of automating tests.

An unexpected result is both the low percentage of requests and the low ranking for unit tests (D4) and TDD (D8): only one in ten of all hiring advertisements for developers ask for unit testing and one in twelve for TDD. This result is surprising, especially since a survey on the state of agile development for 2017 [32] shows that 74% team members state that their team does unit testing and 40% say that their team practices TDD. The discrepancy may indicate that unit testing and TDD is performed only by a fraction of the team, but the activity is accounted as a practice for all the team.

While an adequate proportion of tests to cover code can be determined [33], it is surprising to find that, as the ads show, the employers asked one developer out of ten to work with unit tests. The advantages of writing unit tests and maintaining this practice have been thoroughly studied and are clear. Still more, there is a strong current advocating for TDD [34]. The implications of not having unit tests can have a great impact on the costs of changing and maintaining software, can put a much higher burden on testers and entail an overall greater risk in the development and deployment of software. Therefore, the low number of requests for these particular skills is concerning.

**Demand for testing tools and GDPR competence**

In answering the third research question: “Which testing tools and which kind of competence on the GDPR compliance are required of both testers and developers?“, we observe that more than half of testers need to master testing tools, in contrast to about one in six developers. From the advertisements that ask for this skill, on average, testers need to use around two testing tools, while developers are asked for skills on one testing tool (see Table IV).

In Table V, we list the demand for both developers and testers to have mastered testing tools, then we present the percentage of advertisements that ask for these tools and take an example from the ads. The test tool categories are defined in [13]. Besides the demands listed in Table V, 39% of advertisers ask testers to perform other activities related to frameworks (design and set-up) or testing tools (implement, develop and maintain).

<table>
<thead>
<tr>
<th>Table IV. Statistics on the demand for testing tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ads analyzed</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Ads analyzed</td>
</tr>
</tbody>
</table>
### Table V. Demand for testing tools

<table>
<thead>
<tr>
<th>Tool category</th>
<th>% ads</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test execution</td>
<td>52</td>
<td>“Will develop expertise in 2-3 programs, each to provide execution coverage on multiple areas”</td>
</tr>
<tr>
<td>Test management</td>
<td>37</td>
<td>“Experience in test plan management tools: Spirateam, Testlink and bug tracking systems: Spirateam; BugZilla is a plus”</td>
</tr>
<tr>
<td>Other tools</td>
<td>29</td>
<td>“Excellent knowledge of penetration testing and vulnerability management tools”</td>
</tr>
<tr>
<td>Bug tracking</td>
<td>13</td>
<td>“Extensive experience working with common bug tracking tools like JIRA, Bugzilla, Version One, Mantis, etc.”</td>
</tr>
<tr>
<td>Performance testing</td>
<td>12</td>
<td>“Good experience with performance test; Load Runner”</td>
</tr>
<tr>
<td>Developers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other testing tools</td>
<td>23</td>
<td>“Manage quality assurance [...] through to the use of continuous integration, automated unit testing, deployment automation and code coverage tools.”</td>
</tr>
<tr>
<td>Test execution</td>
<td>17</td>
<td>“Understanding of automated testing, deployment process and tools”</td>
</tr>
<tr>
<td>Performance testing</td>
<td>5</td>
<td>“Implementation of various navigation algorithms &amp; testing tuning of system performance”</td>
</tr>
<tr>
<td>Test management</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Bug tracking</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

As expected, the tools that testers are most required to have mastered are automated test execution tools. Over half of testers being asked for competencies on test execution tools means that half of the testers have a mandatory technical component in their role profile. As Karlsson et al. reports [35] and Paasivaara and Lassenius prompt too [36], automated testing is especially difficult to implement in distributed projects. Therefore we can confirm the need and report the highest request for skills in this area. We also report the ranked request for test management tools, bug tracking tools, and performance testing tools.

An in-depth observation from analyzing the data is that more than half of the ads do not specify which test tools do testers need to operate, while a little below half ask for specific test tools. The most asked for is Selenium, a tool in the category of automated test execution. The second tool, Jira, one of the most commonly used tools for test management [37] equipped with comprehensive bug tracking systems as well as a myriad of add-ons for project management, time-tracking, and integration facilities. HP Quality Center, the third in the ranked requests, is a test
execution tool that is part of the application lifecycle management solution offered by Hewlett-Packard. QTP, the fourth in the top demand, is also a popular automated test execution tool for functional testing [38]. JMeter and LoadRunner are both performance-measurement tools that used to be the exclusive responsibility of developers [39], currently widely used by testers as well [40].

We interpret the request for performance tools as rather high: the fact that one in twelve testers have to handle performance tools means also that at least one in twelve testers need to have very high technical skills, adding an even stronger technical component to the job profile.

We noticed a high demand for other types of tools as well, such as generic testing tools, automated testing frameworks, unit testing tools, security testing tools, monitoring tools, configuration management tools, CI tools, web-testing tools, accessibility testing tools, and mocking tools. This means that testers not only use these tools, but they are requested to be responsible for the infrastructure around them: integrating tools, setting them up and maintaining their running environment, while at the same time offering support to the users of the ecosystem of testing tools. Our result support the affirmation from paper [41], which signals a need for infrastructure in distributed teams which allows practitioners to make use of their skills, as well as a organized approach to build and maintain these tool infrastructure.

We conclude that the high request for mastering automated tools, performance tools and entire testing frameworks points to a need for highly qualified software testers with strong or very strong technical skills. Also, the finding confirms that global software engineering with its distributed working environment benefits significantly from the use of testing frameworks in terms of collaboration and control of the testing process.

Developers are not asked to have the skills to operate bug-tracking and test management tools. Our results are in line with a recent survey of 50 000 developers [30], where these types of tools are not mentioned in the ranked list of the twelve most used tools by developers in their testing activities. This number is unexpected, given that, in the case of bug-tracking systems, the developers submit bugs along with testers, product owners, or customers [42], besides using these tools in the process of fixing the bugs.

In addition to the data presented in Table IV, 11% of ads for developers ask them to perform other activities around testing tools and testing frameworks, such as: implementing the test automation process, deploying tests, creating mockups, monitoring automated tests.
Implications for Practice

Few developers need to handle test automation and performance-testing tools. This indicates that the responsibility for automation and performance falls into the hands of testers. This result differs from a study conducted two decades ago [39] that suggested that performance testing was exclusively the development team’s responsibility.

Surprisingly, none of the advertisements, either the 150 from the EU or the 350 outside of it, mention GDPR. Therefore, we decided to look in the area of test data requirements, because extracting and anonymizing test data is an important part of testing, and the GDPR affects the way this process is carried out. We wanted to know if there is at least some demand for skills in this area. By interpreting the difference in the number of requests, we can state that the testers are likely the ones responsible for test data management. However, we observed that there is no strong demand for either testers or developers regarding test data. While only 1% of the advertisements ask developers to build data warehouses, 6% of the advertisements for testers ask to “create test data”, “create performance test data”, “design test data”, “gather test data”, “generate test data”, “prepare test data”, “maintain test data” “review test data”, “create test oracles”, “define test oracles” or “test data automation”. We conclude that while there is an interest around data collection, generation, design and use, it is not a widespread demand, even though having test data that satisfies testing requirements is a labor-intensive and expensive part of software development [43]. Our findings support a recent study [44] that suggests that practices related to test data are still heavily predisposed towards non-documented, non-standardized and non-anonymized data.

An interesting finding is that about one in two testers and one in four developers need to maintain test environments. Testers are asked to create, design, build, maintain, automate and improve test environments, while few advertisements ask developers for maintenance activities around testing tools or frameworks. Developers are asked instead to automate tests, deploy tests, create mockups, monitor automated tests and understand testing frameworks. This indicates that the testers are considered in charge of the tools and their environments. Therefore, testers need not only to know how to use tools but actually to develop and maintain the testing frameworks. We believe this is interesting because the most literature on this topics assumes the developers, not the testers attributed with these tasks [45].

4 Implications for Practice

Our findings are of particular interest for practitioners working in globally distributed projects, whether as testers, developers, or project managers. Since obtaining high quality of software is especially difficult in distributed projects [46] [47], our results help the software testers understand what is required by employers
in the global software engineering context and provide a reference to follow when seeking to update their skill portfolio, according to employers’ needs. This is especially important since even more testers will need to perform in a global context, as a result of the global projects gaining more terrain [48]. Currently, manual testing still plays an important role, but it comes in second after automated testing, the most persistently sought-after test-related skill, according to employers. Therefore, most of the testers will need to have the necessary technical skills to write, execute manage and handle tools for automated testing. Moreover, the testers will need to manage the testing process, from an overview of the tests, design and planning to defining and reporting test and testing-related metrics. The lack of demand for regression testing can be explained as this is an activity that is currently done mostly automatically.

The results we obtained on the lacking and vague demand for testing skills for developers should foremost alert the employers operating in the global software development. Equally precise requirements have to be asked from both roles in regards to software testing. Moreover, the employers need to recognize the importance of testing in developers’ jobs and need to ask for testing skills from developers when posting hiring advertisements, as they do for other kinds of competencies, whether technical or soft skills. Given that the terms used and the demands posted are rather general and vague, the hiring companies need to see if the informality of asking developers for testing at hiring isn’t translated in a too relaxed and too vague testing practiced by the developers. This trend can be corrected by employers by putting focus more on testing done by developers and by being more precise in their demand for testing skills from developers, both at hiring and also during the development.

Universities and other tertiary education providers are encouraged to use our findings to shape the future agile testers by updating their curriculum topics on the hard skills, to satisfy the industry needs. Essential skills to include are those related to automated testing, followed by performance testing and test management and design. Universities should also give their students the chance to practice code reviews and unit testing, as this is especially important for distributed projects [49].

Additionally, the managers of agile teams can use our results to check with their teams if the demand expressed in job ads for unit testing, testing tools and testing framework is actually reflecting the needs of the team. A common view on which testing skills are important and which ones are needed can guide the team and the actions, in a similar way that the study [50] reports on common values. This aspect is especially important in agile teams, where the team members have a larger degree of freedom in the way they work together. They can check if the activities around testing tools and frameworks are entirely undertaken by the teams, or if there are any
gaps; gaps in the use of tools can mean either unaccounted work or the unverified assumptions on the employees’ skills. Additionally, gaps between requests for maintaining tool environments and the actual needs could mean that the corresponding work is unmanaged, and this could generate a myriad of problems, from a waste of time to unreliable testing results [51]. If necessary, the gaps we discovered in the request for technical skill and tools – especially on the developer [52] side, can be assigned for follow-up to a special role, such as test analyst, as study shows or software architect.

Given the lack of requirements in advertisements for skill related to regulations around GDPR, the senior managers need to determine if this skill is being asked for later during the hiring process or if it is being implemented in another way that does not involve testers nor developers. GDPR is a compulsory requirement on the protection of all private data transacted through the European Union. This regulation is the most far-reaching technical demand on data, given that the software used in the EU certainly not only a European matter, as it is produced everywhere in the world and most likely the result of global software engineering. GDPR-related requirements should have been significantly more visible from the employers’ perspective, especially since testing of data protection implementation mechanisms is an ongoing process, as new versions of software most likely require new user data or need to process user data differently.

5 Limitations

The number of job advertisements for testers and developers that we have analyzed is clearly inclined towards testers. Only a fifth of the ads specifically targeted developers. The reason for this approach is that we wanted to study testers and contrast that study with an overview of developers. This explains the second limitation, the different times at which data was collected for testers and for developers: the ads for testers were collected in 2016-2017, while the ones for developers in 2018. Another limitation could come from a large number of advertisements originating from the USA, Canada, and the UK.

We checked if there is a bias in that the data we collected was published eighteen months before the deadline for implementing the GDPR regulations. Therefore, we made additional searches in April-May 2018 in the USA, Canada, and UK sites. While the industry searches to hire a number of GDPR consultants (550 vacant jobs out of over 69,619 in the software industry at the time of the search on https://www.monster.com), none of the testers’ job ads included any demand related to GDPR or data protection. Consequently, we maintain our findings and interpretation.
6 Conclusion

Motivated by the globalization that software testing has seen lately, we wanted to investigate which technical skills and tools is the industry is looking for when it comes to software testing. Our study serves the global software engineering with the overview of the tools and frameworks currently favored in software testing, given that this way of organizing the work relies heavily on collaboration tools and frameworks, and puts significant effort into building, using and maintaining them.

Currently, the software engineering industry throughout the world considers software testing not only a skilled job but a highly skilled one. The broad majority of testers were asked to possess an average of ten testing skills and to master a large spectrum of testing tools. Moreover, the profile of a tester sought by the hiring companies inclines towards a technical one. Many employers have the infrastructure to support the testing process, but in areas like automation and design they lack the qualified personnel, therefore the high level of demand. While performance testing was once the responsibility of developers, now it is handled to a large extent by software testers.

The proportion of demand for test-related skills is different when formulated for developers, and the demand is less precise. The finding points to a lack in the industry’s perception of significance concerning developers’ roles in testing.

Future work should be conducted to further explore our findings, in order to observe changes in the demand and in order to verify the causes for the drop in the demand for certain test skills such as integration or unit testing. Further studies should also look whether more test-related demands are being solicited from developers during another phase of the hiring, as whether demands for certain skills are formulated post-hiring, as part of developers’ jobs. It should be also checked whether the hiring companies need to revise and improve their approach on how they determine the test skills they require for both software testers and developers who will work in distributed projects.

7 References

Paper IV: A qualitative study of the background, skill acquisition and learning preferences of software testers

Raluca Florea
University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway
ralucamf@ifi.uio.no

Viktoria Stray
University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway
stray@ifi.uio.no

Abstract

Context: There is an indisputable industrial need for highly skilled individuals in the role of software testers. However, little is known about the educational background of these professionals, their first contact with the role, their preferences in acquiring their skills, the impediments they face, and their perception of the software testing role. Objective: The current paper reports on the background, the skills, learning preferences and role profiles, as described by the professionals in software testing, spanning over a significant number of industries, countries and software development models. Method: We conducted 19 in-depth, semi-structured interviews of software testing practitioners, across eight industries. We performed a content and thematic analysis of the collected data. Results: The practitioners in software testing had a diverse educational background, and their first contact with the testing role was accidental. Exploratory testing was the preferred testing technique, while curiosity was identified as the most important feature in their skill-set. Our respondents collaborated extensively with the developers, which they perceived as a learning source and a symbiotic work partner. Conclusion: The professionals in software testing described their skills as a rather undefined heap of knowledge, increasing with each work-task. They used mainly informal and hands-on learning approaches. They found it necessary for education providers to present information on software testing. Generally, companies assisted them well in the skill-development, but need to allow sufficient time-allocation for the learning. We identified five specialties of the role: product owner in testing, UX tester, DevOps tester, test automation specialist, and test-process coordinator.

CCS CONCEPTS

Software and its engineering → Software verification and validation; Social and professional topics → Professional topics → Computing profession → Testing, certification and licensing

KEYWORDS Software Testing, Skill Acquisition, Software Tester, Hiring Software Testers, Software Testing Profile
1 Introduction

Everywhere in the world, across industries, highly innovative companies post hiring advertisements for software testers, asking for a wide range of skills and competencies [1]. A previous study shows that the industry needs the software testers to be highly-skilled individuals, with an average of ten test-related skills, five development-related skills and two soft skills [2].

While there are consistent reports on the experience that tops the characteristics of high-performing testers [3], there are few studies focused on how the software testing practitioners actually become experienced. To fill this gap, our current study dives into the skills acquisition and learning preferences of the professionals in software testing, in order to provide recommendations to the industry and the academia on how to better assist these professionals in the acquiring of skills relevant to their role. We aimed, therefore, at answering the following key-questions:

RQ1: What is the background of the professionals in software testing, and what was their first contact with the software testing role?

RQ2: How do the software testers acquire their skills and which learning preferences do they have?

The remainder of this paper is structured as follows: Section 2 discusses the methodology for data selection and analysis undertaken in our study. Section 3 summarizes and structures our findings. We discuss and interpret the results in Section 4. We present the limitations of the study in Section 5. Section 6 wraps-up the research with the conclusions and discusses future work.

2 Methodology
In our research, we used the purposeful sampling technique, by targeting senior professionals with a minimum of five years’ testing experience, who have changed roles or companies at least once. We interviewed 19 professionals (10 females, 9 males) currently working for 13 companies. At the time of the interviews, sixteen respondents worked in large–sized companies (>250 staff), while three worked in small or medium-sized businesses (<=250 staff). For about half of our respondents, we were set in contact by the companies they currently work for, while we contacted directly the other half of our respondents.

We followed the saturation principle, conducting interviews until we reached the point of diminishing return for our qualitative sample. As we practiced semi-structured interviewing, we created an interview guide with points of interest to be followed, of which we present an excerpt in Fig. 1. For every question where the respondents provided brief answers, we followed-up with additional questions, in order to obtain more information on the respective topic. Before we interviewed our participants, we tested the guide on a trial respondent with more than ten years’ experience in the software testing field.

We stirred the direction of the interviews by having a clear understanding of the objectives we pursued, by asking targeted questions, and by giving appropriate feedback to the respondents: encouraging them to talk, reflecting on their remarks, probing on the remarks. At the same time, the interviews were to a great extent

---

**Fig 1. Excerpt of the interview guide**

- GENERAL INFORMATION
  - How long have you worked as a software tester?
  - In how many companies have you worked so far?
  - What were those software testing jobs about?
  - Why did you switch jobs?
  - Give me some examples of your responsibilities.

- EDUCATIONAL BACKGROUND
  - Do you hold an academic degree? From which institution?
  - Do you feel you should have studied more on software testing? How would that have helped?

- SKILL FOLLOW-UP
  - Are your skills measured and followed-up at work? How?
  - Do you wish for more support from your employer in the development of your skills? What kind of support?

- INSIGHTS ON SKILLS
  - What kind of testing do you like best?
  - How did you discover a significant bug?
  - Which skills are most valuable for a tester to have?
  - Are some testing skills more complicated to acquire?
  - What are the biggest challenges to your learning?

- LEARNING SOURCES
  - What sources of learning do you use?
  - Which learning sources do you like best? Why?
  - What determines your choice of a learning source?
  - Did you have to take any extra certifications? On what topic? Who had the initiative? Who paid for it?

- WORKING WITH OTHER ROLES
  - Who do you collaborate with the most (which roles)?
  - How much time do you work in a week with other roles?
protected from interruptions from the outside, from competing distractions and by not asking inhibiting questions. The atmosphere of the interview was relaxed and inviting to communication.

We conducted the interviews in the period October-November 2019, scheduling one up to two interviews in a day, allowing us time to prepare for it, to take notes, to write the transcripts and to identify new tracks to follow. We conducted the interviews in two ways: face-to-face where it was possible, and Skype video calls. We held 12 Skype video sessions, and 7 face-to-face interviews. The shortest interview lasted 40 minutes, while the longest lasted 1h:30m. All the respondents consented the recording of the dialogue and the publication of the results. During the interviews, we used the language in which the respondents felt most comfortable with: English, Romanian, parts in Norwegian. The first author of the paper transcribed and translated the material to English, therefore, all the data was processed in English.

After the completion of the interviews, we presented all the participants to our study the preliminary results, asked to provide us with feedback on erroneous or possibly missing information, and to reflect on additional inputs. More than half of our respondents provided the required feedback. There were no disagreements with the results of the study.

We chose thematic coding as our analysis method, for the reason that it fit our purpose of identifying information linked by a common theme or idea [4] [5]. We coded the data we gathered in the interviews, grouped the codes into themes and higher-order themes, see Fig. 2. For this purpose, we used NVivo, as we could easily organize the codes and explore the relation between them. We obtained 388 codes, combined in 44 themes, further aggregated in nine higher-order themes.

3 Results

Background of the interviewees

As understanding the context is pivotal in any empirical study [6], we provide a description of the background and work experience of our respondents. All of the interviewees had substantial practice in the software testing role, two participants having 20-30 years’ testing experience, eleven participants having 15-20 years’ experience, four participants 10-15 years’ experience, and two participants 8-10 years’ experience. At the time of the interviews, our interviewees had roles in test management, quality assurance / business analysts, test automation, software testing, domain-expertise, and IT consultancy. Eleven of our respondents have worked for more than three companies in the software testing field. All our participants changed
roles in software testing area at least once. The interviewees had a diverse tertiary educational background, ranging from economy (7), informatics (4), military (2), mathematics (2), engineering (1), physics (1), biology (1), linguistics (1). They worked in a wide range of industries, such as accounting (3), banking (3), logistics (3), telecommunications (2), automotive (2), IT services (2), appraisal (2), tourism (1), health (1), and exerted their roles in the UK, Ireland, France, Romania, Norway, Sweden, Denmark, Pakistan, Poland and Ukraine. Nine respondents were working in advanced economies, and eight in emerging economies.

Fig. 2 Example of the thematic coding of the collected data

Slightly over half of those interviewed did not have educational background in IT. When asked about the usefulness of introducing software testing education in non-IT faculties, one respondent answered: “That would be very important. Whether banking, finances, or any economic specific, it would be good to have this course. Because there are many firms developing accounting software. And nobody in Economics faculties knows they can take this path. I find it a gain and really useful to be presented a semester-long course with this knowledge.”

None of our respondents had learned, in their college years, about the software tester role. One professional with IT tertiary education recalls: “We were never mentioned about this job. We had no idea that this exists.”

First contact with the software testing role
The interviewees reported securing their first testing job as a result of a guess, or a chance. In general, those applying for the first time on a software testing role, seemed to read rather superficially the hiring advertisements, as their lack of experience made them unable to pay attention to all the role-specific details. Part of the professionals with non-IT expertise were intimidated by the ads containing numerous technical demands, to the point of abandoning applying for the position. The rather vague and high-level hiring advertisements attracted them to apply for the job. As one respondent mentioned: “You didn’t realize from the job description what the job assumed [...]. But on the other hand, if they would give reference to these complicated standards, I would probably look it up and would have not applied to it, as I would think the job was too complicated and it was not for me. So, there are advantages and disadvantages. Leaving this gate open, it made me apply for this job.”

Our respondents got discouraged and felt frustrated by the programming tests in hiring for software tester positions, focused on algorithm implementations. As one respondent remembered: “I was tested at this company and I was under the impression that I received a wrong test. Even now, I have no idea what happened there. I was left with the impression that they were searching for developers and I just went to the wrong testing room.”

Depending on their educational background and experience, changing industries when shifting to a testing role was perceived either as another work assignment, or as a shock: “I had some idea about test automation, but I had no knowledge about the people in the back, testing all these reports and functionalities of this financial software. So, it was a bit shocking for me changing the domains, zero prior knowledge, and the language on top.”

The six respondents who had onboarding sessions at the hiring companies found them to be useful from a networking point of view, but they considered them too general, or with little concrete benefit to their day-to-day jobs. However, two of our subjects started their jobs in software testing though apprenticeship programs of three to six months, with their time split roughly in half between various theoretical trainings and actual work. They found it to be an excellent way to be on-boarded on the job: “It was an entire learning process. I liked it a lot and it was very useful, as I did not know before about testing, bugs and so on. And this apprenticeship was really beneficial.”

Learning preferences

The respondents identified, as the most efficient source of learning, as well as testing, exploring the software through the concrete work-tasks assigned to them. One respondent mentioned: “The thing I like most is to get my hands on the product
and get things done. I mean the testing part.” Our respondents described their skill acquisition as a continuous learning from multiple sources, rather than a one-time effort. One respondent stated: “It’s always like that, and you always lack skills to do your job. I think people should always learn and learning for every task is normal.”

Two of our subjects preferred formal learning: “Sometimes I read and something distracts me. But when I am in the classroom, I assimilate information much better.” However, most interviewees (14) preferred informal learning sources. On the internet, they mainly searched for product specifications, test-related samples (17), as well as video tutorials on test techniques (8). Additionally, part of the respondents used online training providers (3) and books (2). From the video format of presentation, our subjects preferred generic YouTube training videos (6), Udemy (1), or Coursera (1), as they were easy to follow.

Thirteen of the professionals we interviewed liked conferences, whether internal or external to the company, as they had the chance to expand their contacts network: “Conferences are really nice. You get ideas, you meet new people that are working on the same stuff that you’re working, you get the temperature on how things are in the industry.” They identified this practice as important, because they saw the tester’s job as making heavy use of networking. As well, the conferences proved to be useful to synthetize knowledge that otherwise the respondents were not aware that they had.

Certifications

The respondents agreed that the certifications, in general, provide credibility for their experience in testing, a particularly important aspect for those without a technical educational background, or for the ones working in consulting jobs. Fourteen of our respondents had a certification related to the foundations of software testing (ISTQB [7]). Additionally, part of our subjects held scrum master certifications (4), project management certifications (4), or advanced test manager certifications (2). For most of the certifications, it was the employer offering the possibility to attend the courses, to take the examination, and also paying for it.

A frequent theme brought-up by our interviewees was the importance of mastering the testing terms, and hence the ability to use the same vocabulary in communicating with their colleagues: “If you have a certification, you are not necessarily a good tester, but at least you have been through the same training with everyone else with this certification and you at least have been taught the same terminology. You might not use it all in your daily life. But you do know what you’re talking about and using the terminology correctly. So that makes it for less misunderstandings underway.”
The foundation level of the software testing certification was described as relying extensively on remembering and reproducing concepts. Those professionals holding the advanced-level certification in testing affirmed that an advantage of the advanced-level training was to pick concrete examples and discuss them.

**Working with other roles**

Eighteen of our respondents estimated to spend around 50% or more of their time working together with others: “I think that in general, I spend about half of my time working with other colleagues. On many things.” Another respondent stated: “Half of the time I am in discussion with one or another on a certain task, or asking for help, or giving help.”

The ones with less work-experience tended to work mostly with the developers, while the ones with more experience worked with, besides developers, other stakeholders, such as product owners and support, as they considered that these stakeholders had goals that otherwise might be overlooked.

**Support from the employers**

In general, the feedback was that their employers provided them good support in the skill acquisition (14). One interviewee described: “I think if I would ask more, I would get more. Support was not an issue. I am not sure how it’s like in other places”. However, four respondents mentioned that the development of skills depended in a great deal of the team manager: “you depend a lot on the manager. I had team managers that had follow-up with me once a year. I also had more informal discussions every three months. And I liked it better that way.”

It emerged from the interviews that for most of our subjects, the company’s follow-up and appreciation for their role was important: “Honestly, I would like to have a follow-up with my manager. I miss it for some years now. Somehow, it’s an opinion that you don’t get to hear from someplace else. And then you know if you do well a certain thing, if some learning is relevant or not, etc. I think it’s a good thing to hear that.”

**Stress sources**

The most frequently mentioned stress sources were the lack of time (16), lack of collaboration with the developers (4), lack of management support (4), potential conflicts (4), unfit personal performance measurements (3), and unfit test environments (3).

The interviewees learned significantly for each work task, therefore they needed to be allowed the time to gain knowledge. The respondents emphasized that this
process was time-consuming, but necessary to complete their assignments. The little
time allowed for testing, with hard deadlines, was identified as a major source of
stress: “And they tell you: you have three days to get in place the system tests and
prepare the testing task. And there is this big thing that you don’t know how to
approach. And I find this extremely stressful. And I would lay awake at night,
thinking: how am I going to do this?”

If the developers were too busy, then it was difficult to ask for their time. Lack of
collaboration sometimes led to uncertainty related to test coverage. One of our
respondents described feeling stressed when she could not check with the developers
if her testing covered all relevant new-developed code: “Even if I test a lot, it does
stress me a lot, especially with the product I am working on now [...]. So, I get very
scared by the thought that maybe I missed something that was important and I was
supposed to see. Maybe someone is not going to get the salary next month because
of me.”

Another stress factor for our interviewees was to have unfit personal performance
metrics, such as a minimum number of bugs submitted, or having zero bugs
rejected. These metrics inhibited the exploration of the system under test, which
came inherently with tester’s mistakes. Reversely, imposing a minimum number of
bugs set the software testers in a bug-hunting mode, as one respondent recalled:
“there are companies in which they give the people as performance target to find a
certain number of bugs. And then the people torture themselves and others pulling
out of their magic hat bugs. A tester is there to test, not to produce bugs.”

A significant reason for our respondents to choose test automation was to relief
some of the daily stress implied by bringing-up issues: “It might be something not
so pleasant to always bring the bad news.” When those with no prior experience in
test automation tried to develop scripts for the first times, a major impediment
was setting the infrastructure just to write a first test. As one respondent recalled: “And
it’s like that now: I need Visual Studio to code, I need to go onboard with Python, I
need to get packages, I need to find my way through firewalls. All these things are to
actually get started. [...] And it takes a really long time. The problem is that the
company doesn’t even think to check that all the users are actually onboard so that
they contribute to test automation.”

Additionally, the inexistence, bad management or poor functional state of the test
environments was identified as a significant source of stress, insufficient testing,
 misunderstandings, and errors: “The biggest source of issues, challenges, problems,
 misunderstandings, quarrels, not testing good enough, lies actually, as far as my
experience tells, in bad test environments. As they are not properly maintained,
topped with bad management of test data, lots of defects arise from test
environments. And the thing is that this is a cost. Test environments are a huge cost, and nobody wants to pay it.”

Specialties within the role

During the data analysis, we detected that our subjects strongly preferred to be allowed to assume specialties within software testing, as opposed to broad software testing skills. One respondent described: “I have seen that sometimes companies push people in an area they don’t like: either technical or domain specific. And people don’t really like that”. A reason for specializing was the complexity which the software testing job has evolved: “It’s quite a change. What you do as a tester today is not the same as it was in 2007.” Five specialties emerged from the preferences and the activities assumed by those we interviewed: the product owner (PO) in testing (4), the user experience (UX) tester (3), the test automation specialist (4), the DevOps in testing (3), and the test-process coordinator (3).

Those with a preference for the PO role in testing acted as a bridge between the product owners and the developers, at a refined level. Many in this category did not feel comfortable working in the position of a business analyst by checking the viability, legality, opportunity and costs of a requirement. But they assessed to be doing a good job in transforming the business-requirements into fine-grained system-requirements. One interviewee recalled: “At the first job, I was really useful. The implementation they had was technically correct, but did not give correct economic results. So, they needed someone to know how it needs to work, and I knew how it was supposed to work - and I was straight out of the university.”

The UX testers focused extensively on the usability aspects, verifying their implementation on levels of detail that the UX designers could not, due to limited time availability. They were skeptical towards the extensive use of the automated tests, as they perceived a product release based only on their as a narrow quality indicator. One respondent summarized these concerns: “And it’s a lot of important testing, like automation, but there is so much more than that: it’s about having good error messages, it’s about creating interfaces [...] And for a tester it is important to give feedback on these aspects, the emotional part of the product. I think there is a lot of focus on the technical part, but the emotional part is also very important.”

The test automation specialists liked transforming manual test scenarios into automated test scripts, while they strongly felt they should not take more complex responsibilities, due to their limited technical experience. Our subjects in this category were previously affected by the stress of bringing bad news to the team. They explained that it was important for them to have the support of a failing script, when they reported issues: “It’s totally different! And you know what I like most? I have a different view of the world since then. Even in the day to day live it affected
me. *In automated testing, you are building something, not destroying it. And you know how good it feels? I feel happy about this!*”

The DevOps in testing were attracted to the set-up and maintenance of the automated testing platform, as a part of the deployment pipeline. An interviewee exemplified the need for this archetype: “*We started with this automation tool, and two years after, we realized that we have a lot of problems with maintenance, and our luck, new technologies appear. It’s really hard to find the spot-on right tool.*”

Those preferring the test-process coordinator specialty emphasized the importance of teams working together towards a common goal: “*Some companies say that the developers can do the job. But they forget that this is the role that keeps everything, the focus and the overall quality. Sometimes you need someone to see the bigger picture.*”

### 4 Discussion and implications

In the following section, we discuss the results of our research objectives, synthetized in the two research questions.

- RQ1: What is the background of the professionals in software testing and what was their first contact with the software testing role?

The fact that the professionals in software testing came from many education backgrounds, points to a diversity in the landscape of software testing, an essential factor in software development of high quality [8]. They followed both IT and non-IT education in rather similar proportions, however none of our respondents had academic contact with testing prior to getting hired in a software tester role. Our respondents found that an academic focus on software testing to be valuable within IT, as well as in other faculties, as such information was an opportunity to introduce to the domain experts the software testing role. Therefore, we report an ongoing need for the universities to include information or courses on software testing.

The finding that the hiring ads with too many requirements was intimidating, especially for the domain-experts, points that the hiring companies need, where possible, to broaden their horizons in the advertisements, so that a larger group of applicants are encouraged to apply for software testing jobs, fact pointed as beneficial also in paper [8].

1. RQ2: How do the software testers acquire their skills and which learning predilections do they have?
Discussion and implications

The fact that exploratory testing was perceived by the practitioners as the best way to learn, to test and to discover significant bugs, is in line with the position of the researchers in the domain [9], assessing such tests as a powerful and effective approach to leaning and testing, reported as orders of magnitude more productive than scripted testing. We recommend therefore the companies to allow the time and facilitate the practice of it.

Those respondents benefiting of trainee programs advocated the usefulness of extensive apprenticeships on the job. We recommend the companies to make use of apprenticeship practices for all newly-hired personnel, in addition to the junior testing positions. While for other professionals the online conference might be a viable option [10], the practitioners in software testing need the physical settings, to be able to create their contacts network.

We found that working together with other roles, in particular with developers, was essential for our respondents, to mitigate some of the problems arising from the gaps in requirements [11], their interpretation, such as different approaches taken in problem-solving, or to prioritize testing tasks. Additionally, the fact the practitioners in software testing seldom work together with other software testers indicates that the test-skill acquisition is often an individual effort, rather than an intra-role group endeavor.

We found that the ISTQB foundation certification in software testing reaches its stated purpose of introducing the practitioners to the field of software testing and presenting the key concepts in software testing [12]. Our certified respondents perceived its value, but as there was a strong request to have more concrete examples, we recommend the training providers to include more samples in the teaching, and at the same time to focus the training less on answering sample-exam questions.

The result that the most important traits of a good tester, as perceived by the practitioners, were coming from the area of soft skills, complements the findings on the focus on the human dimensions in testing [13], and indicates that they are primary in the successful software testing.

The determination manifested by a part of our respondents on getting started with automated testing, together with the frustration generated by the inability to do it, points to the lack of a technical architecture for the automated testware as a main issue in automation in [14], which needs to be addressed by the companies implementing test automation solutions. The practitioners’ preference for mainstream tools is in line with the findings in study [15].

The outcome that the respondents received at company-level good enough support into acquiring new skills indicates that the employers generally have a focus on the
professionals in software testing, and allocate budgets for it. However, at team-level, we recommend accounting in the task planning the time needed by the software testers to acquire the necessary skills, to successfully deliver their assignments.

Even though there are strong advocates against using bug-metrics in the personal performance appraisal [16], our respondents reported that some managers still use it. As this practice might have an additional impact on the product quality itself, we recommend the employers to work further to eliminate these metrics from the performance appraisals.

Our finding that the practitioners in software testing see in their role a number of distinct specialties, together with their wish for not being imposed to undertake multiple specialties, point to the need that the employers adapt their practices of roles in testing, particularly as guidelines on various aspects of testing are available, such as [17]. Currently, some of these testing specialties have a part of the theoretical background covered by certifications tracks, on the areas of usability, automation, product ownership and test management [7]. However, as neither tertiary education providers nor other certification organizations offer full support for these specialized career paths in software testing [18], it would be beneficial that both establishments consider to shape their tuitions on these five tracks.

5 Limitations

As in any empirical study, we identified a set of limitations, that we present and discuss in this section. The researchers were external to all the companies from which we interviewed the software testing professionals, having no agenda other than outlining the specifics of the software testing role, as seen by the ones practicing it.

We interviewed a number of 19 professionals with significant experience in software testing, to limit the bias of a singular job or hiring company. This number of interviews allowed us to develop themes on learning preferences and profiles of software testers. To avoid ambiguity, we presented the respondents the interview guide beforehand, to allow time for reflection on the questions. We were open to all input and insight, and we kept contact with our respondents until the finalization of the study. We routinely checked the consistency of the data in the transcripts, the codes and the themes we used. Our findings were presented to all the interviewees and lead to feedback, which we included in the study.
6 Conclusions and future work

Though 19 in-depth interviews, we conducted a qualitative analysis of the background, skills, learning preferences, and profiles of the practitioners in software testing. We obtained that the software testers have a varied educational background, they learned mostly from informal sources, they saw their own skills as a rather undefined heap of knowledge, increasing with each work-task. They preferred the hands-on and task-based approaches to acquire testing skills. Generally, employers provided good skill-development support, however the practitioners needed sufficient time-allocation for learning. We identified five specialties in software testing, emerging from the data analysis: the PO in testing, the UX tester, the DevOps in testing, the test automation specialist, and the test-process coordinator.

Future work should build on refining our findings in a field study of the software testing archetypes, and study their effect, by observing the patterns in function of the company size, team size or industry sector.

7 References

7. ISTQB. Available: https://www.istqb.org/
Abstract

Although extensive research has been conducted on the characteristics of the agile developer, little attention has been given to the features of the software-testing role. This paper explores the human factors of the software testers working in agile projects through a qualitative study focusing on how these factors are perceived. We interviewed 22 agile software practitioners working in three international companies: 14 testers, five developers, and three designers. Additionally, we observed 11 meetings and daily work of 13 participants in one of the companies. Our findings show that the views on the human factors shaping the agile software tester’s role were crystallized into seven traits, which the agile team members saw as central for the software-testing role: the ability to see the whole picture, good communication skills, detailed-orientation, structuredness, creativeness, curiosity, and adaptability. The testers spent half their day communicating and learned how to mitigate the fact that they had to bring bad news to other project members. They also facilitated communication between the business side and development. Based on our results, we propose the seven traits as dimensions to consider for organizations recruiting agile software testers, as well as a reference for IT and non-IT professionals considering a software-testing career.

Keywords: Software testing, Human traits, Soft skills, Agile software development, Agile tester
Part 2: Collection of papers

communication barriers, and lack of alignment (Ghobadi and Mathiassen, 2015; Dingsøyr et al. 2019). Software testing is a pivotal activity in agile software projects, to ensure the quality of the software product throughout the iterative development process, with frequent releases (Korhonen 2013).

In agile, a noticeable divergence from traditional plan-driven development methods entails involving testers from the beginning of each development increment. Therefore, dedicated testers can plan and complete various aspects of the test strategy, such as exploratory testing, usability testing, and improving test coverage with the developers (Bai et al. 2017; Santos et al. 2011). In such work environments, knowledge transfer also occurs more frequently and naturally between developers and testers, as both testing and development happen concurrently during each iteration. Even though testers had a difficult time finding their place in teams during the early phases of agile adoption, they quickly became integrated and recognized as an important part of agile software development (Cohn and Ford 2003).

The skills within development teams are crucial to the success of software projects because they directly affect central aspects of software attributes such as performance, reliability, and simplicity (Byrd and Turner 2001). There have been numerous studies on how team members’ skills affect the qualities of software as well as teamwork performance and the competitive market advantages those skills bring (Capretz and Ahmed 2010; Ebert and De Neve 2001; Faraj and Sproull 2000). Alternatively, as many researchers have shown, a lack of skills within development teams directly affects the costs of software projects, their delivery times, and even their completion (Jiang and Klein 2000).

In general, the literature presents testers with an emphasis on their hard skills—for instance, as the ones responsible for carrying out testing and building up test cases and test plans (Mathur and Malik 2010). Davidov et al. (2010) described testers as professionals who identify new defects from failed test cases, analyze defects, and report them in a bug-tracking system. However, the testing process requires not only technical skills but also specific socio-technical abilities (Florea and Stray 2018; Sánchez-Gordón et al. 2020); therefore, an efficient software tester is needed to cover a broad area of abilities and expertise, frequently extending beyond testing. For instance, in agile, software testers play a key role in connecting developers with development stakeholders and customers by intimately understanding and tracing business requirements throughout development (Saiedian & Dale 2000). Nevertheless, there is a need for further studies focusing on agile testers’ soft skills (Sánchez-Gordón et al. 2020).

In agile, attention has been given to developers’ skills from both industry (Capretz and Ahmed 2010; Ebert and De Neve 2001; Faraj and Sproull 2000) and educational
perspectives (Lethbridge 2000; Lindstrom and Jeffries 2004). However, the same cannot be said about the testers’ skills, as the role was largely considered a junior or entry position, and testing seen as a side activity (Juristo et al. 2006; Deak et al. 2013). This paper extends our preliminary findings on the human factors of the agile software tester from a single case study involving 13 participants (Paruch et al. 2020a Paruch et al. 2020b).

To better understand the agile software tester, we conducted an exploratory case study in three companies. The aim of this paper is to fill a research gap within the area of software testing, focusing on the human dimension of the software-testing role. Although there has been extensive research on the technological aspects of software testing—such as tool usage, test automation, and test processes—little research has been conducted on the human factors central to the role of the software tester (Kanij et al. 2015; Deak et al. 2016).

We aimed to answer the following research question: Which human factors are essential for the agile software tester?

The remainder of this paper is structured as follows. First, Section 2 presents the background and related work on the agile software tester. Section 3 reports on our case study methodology. Section 4 presents the results of the study. Section 5 discusses the results and implications for practice as well as the threats to the study’s validity. Finally, Section 6 concludes this paper and discusses future work.

2 Background and related work

In this section, we first present the background of the agile software tester. Then, we present related work on human aspects of software testers through a literature review.

Software testing in agile

In agile, a noticeable difference from sequential development methods entails involving testers at the beginning of each development iteration, instead of end-of-phase testing. One of the seven testing principles states that quality-assurance activities should be started as early as possible in the life cycle to avoid additional cost and time (ISTQB 2019). In agile, early involvement of testers at the start of the iteration enables an emphasis on the user stories and the system architecture, with which primary testing personnel will acquire a better understanding of the testing scope.

Two studies focused on the differences between the test activities conducted in an agile project compared to a plan-driven project (Dhir and Kumar 2019; Kettunen et
al. 2010). Kettunen et al. (2010) performed a study on agile projects and identified the absence of guidance on how testing should be arranged at the same time as development. Dhir and Kumar (2019) compared the testing of a Web application in a plan-driven project versus in an agile project. Their results showed an improvement in the agile project compared to the traditional one: the agile project had increased test coverage, reduced costs, and improved testing productivity.

Deak (2014a) identified and ranked testers’ motivation factors. The results showed that the testers working in plan-driven projects experienced more stress but manifested a more positive attitude for tackling challenges than their agile counterparts did. The agile testers were better integrated into their teams but expressed distress in their relationship with the developers, with whom they found it demanding to communicate.

The delegation of testing tasks is also an essential feature of agile, as developers are expected to perform unit tests independently. As such, testers can focus on test techniques such as exploratory testing and usability testing and improving the test coverage jointly with the developers (Santos et al. 2011).

In agile, knowledge transfer also happens more frequently and naturally among developers and testers, as testing and development occur concurrently in each iteration (Li et al. 2010). While new team members benefit from the ongoing feedback in their onboarding, this competence sharing also contributes to less misunderstandings or confusion during development. The tight coupling among testers, developers, and stakeholders enables fast learning and greater mutual understanding.

3 Related work

As we found limited research on the human factors of the agile software tester, we aimed to provide an overview of the human factors in software testing, regardless of development methodology. The literature review was performed from September 2019 to December 2019, with the main goal being to map the existing studies on the human factors in software testing and the secondary goal of finding research gaps and needs for further studies. We looked to the literature to answer the same research question as we had for our case study: Which human factors are essential for the agile software tester.

Review method

The search string presented in Table 1 was modified to fit the syntax of the different databases; however, the semantics were ensured to be consistent. We included the
search term “quality assurance” because some of the research papers use “quality assurance” instead of “software testing,” and the two terms generally are interchangeable in the agile terminology. We followed Kitchenham and Charters’ (2007) suggestion to select search criteria meant to identify the primary studies that provide direct evidence about our research question. In addition, we decided on the criteria before the study selection to reduce the likelihood of bias.

**Table 1 Search string for the literature review**

<table>
<thead>
<tr>
<th>OR</th>
<th>OR</th>
<th>OR</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(“human factors” OR “soft skills”) AND (“software testing” OR “quality assurance” OR “QA” OR “software quality assurance” OR “SQA”)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We devised the following protocols, which included both inclusion and exclusion criteria. In the inclusion criteria, we considered the relevant academic and industrial studies, both qualitative and quantitative, as well as relevant conference papers, journal papers, workshop papers, review articles, research articles, book chapters, and conference papers published in the timeframe 2009–2019. We excluded from the analysis any non-English contributions, studies not related to software engineering, encyclopedias, prefaces, book reviews, case reports, correspondences, tutorials, editorials, and news reports.

We obtained papers through four stages. The reviewing process was initiated by applying the search string to four scientific databases: Scopus, Science Direct, IEEE Xplore and ACM (Stage 1). The application of the inclusion criteria and exclusion criteria and the removal of duplicate studies resulted in 1160 papers, with most stemming from Scopus (956). All of the articles underwent a scrutinization phase where we inspected the title, abstract, and keywords for relevancy to our goal. We selected the relevant literature by removing studies without references to human factors and software testing. Most of the initial search results were excluded from further analysis, as they did not fit the general objectives of our research (Stage 2), and belonged primarily to the medical and psychology fields. After skimming through the 1160 papers, a total of 22 studies were included and read carefully. Seven of these papers did not focus on the human factors within software testing and were excluded from further analysis (Stage 3). In Stage 4, we used the snowballing technique, and found two additional articles on the human factors in software testing. As such, at the end of the selection process, we had obtained 17 studies relevant to the research question.
For the 17 selected papers, we extracted data manually and organized the information in an Excel spreadsheet. For each paper, we extracted the title, names of authors, publication year, source title, number of citations and whether it was a conference or journal paper. When reading the papers in detail, we highlighted passages and put relevant themes and comments into the Excel sheet. By analyzing the objective of the papers, their results, and the discussion of the findings, we found three themes emerging from the papers, and we grouped the 17 papers by the theme they shared: (1) software testing as a profession, (2) motivational factors for software testers, and (3) the personal characteristics of software testers. The grouping of the papers is shown in Figure 1, and the findings in the three categories are presented next.

![Fig. 1 Themes approached in the related literature](image)

**Software testing as a profession**

Research has been conducted before on software testing as a profession. Capretz et al. (2019) conducted a quantitative survey among professionals in four geographic regions to determine the profession’s degree of attraction. Their results indicate that testing was not a popular career option among software professionals. Among the enumerated reasons were the role’s perception as a lower-competence one and the
testing job’s complexities resulting in stressful and frustrating situations. Shah and Harrold (2010) mentioned that senior testers in a service-based software company located in India voiced similar opinions. Most of the seniors had a negative attitude toward testing and considered it as something that just needed to be done. The junior testers had a positive attitude toward the job, stating that testing helped them to learn the system better so they could become skilled programmers in the future. The study found that all but one participant did not want to work permanently as a tester.

Salman et al. (2019) conducted a controlled experiment to find out if testers exhibit confirmatory behavior - also known as positive testing - when designing functional test cases, and whether such behavior increased under time pressure. The findings resulted in the conclusion that confirmatory test cases were present regardless of time pressure; it is therefore necessary to make testers aware of the danger of confirmation bias (the tendency to look for evidence that strengthens his/her prior beliefs) and learn how to design test cases with a disconfirmatory attitude (Salman et al. 2019).

Ekwoge et al. (2017) identified three main categories affecting testers: cognition, conation, and affection. Cognition refers to memory, problem-solving, and decision-making and how testers perceive the testing infrastructure. Conation includes impulse and desire and how testers see the value of their contribution. Affection involves elicited feelings and emotions, and influences testers’ respect, team belonging, and social factors. Deak (2014a) found that agile testers were more unhappy about their relationship with developers than testers working in traditional plan-driven development. However, those testers in traditional development reported a higher degree of stress.

Motivational factors for the software testers

Santos et al. (2017) point out the importance of highlighting the testing activities as "a set of human dependent tasks," therefore emphasizing the need for research within motivation. They argue that five factors influence the motivation of software testers: acquisition of useful knowledge during work, work variety, creativity in solving tasks, well-defined work with the precise sequence of steps, and recognition of work. The authors highlight the last factor, which has a lasting impact on the testers’ motivation as well as an increase in individual productivity and teamwork enhancement.

Deak et al. (2016) also report similar motivational factors, such as enjoying challenges, variety of work, and recognition such as positive feedback received from both management and developers. The authors also identified de-motivational factors, exemplified by time pressure, poor relationships with developers, lack of clear processes, redundant meetings, or lack of influence and recognition (Deak et
Moreover, some of the participants in this study mentioned the tedious routine of some testing activities and the "feeling of boredom", which would furthermore increase the assumption that the profession is unattractive.

Hernández and Marsden (2014) investigated the challenges software testers face and how they collaborate with other teams. Their findings showed that the testers were highly motivated. They were motivated by the broad variety of topics they could work on and that they were able to have a complete view of the software. They were also intrinsically motivated when they experienced autonomy.

Gonçalves et al. (2017) identified three main types of factors influencing the software profession; cognitive, operational, and organizational. The cognitive aspects included stress, psychological pressure, and retention of information under mental workload. The operational aspects included conflict, receptiveness, and monotony, whereas the organizational aspects included a lack of training, participation, and division of activities. The study showed that professional testers face many demotivational factors such as outdated testing environments, demobilization, and the devaluation of testing careers—which were often seen as a mere extension of development.

**Personal characteristics of the software testers**

Kanij et al (2014) found that human factors are crucial in software testing and that important traits for testers are being open-minded and having curiosity and “attention to details”. Kanij et al. (2015) argued that the effectiveness of a tester role was related to their personality and that testers had significantly higher levels of conscientiousness compared to those in other software engineering roles. In the study, conscientiousness was related to being disciplined, hardworking, and dedicated. Although highly conscientious individuals were important in any profession, Kanij et al. (2015) suggested that this quality might be particularly important for testers. Deak (2014b) studied personal characteristics among software testing professionals and found that the most valuable characteristics to possess included communication skills, need for variety and being detail oriented and curious.

Livonen et al. (2010) explored the characteristics of high-performing testers, identified as such by a high defect-detection rate or by possessing traits that managers and other testers see as important. The authors found four themes: experience, ability to reflect, motivation, and personality. Within those themes, the top characteristics of the high-performing testers were thoroughness, carefulness, patience, and conscientiousness (Livonen et al. 2010). Itkonen et al. (2013) investigated the knowledge types used by testers in exploratory testing and obtained
three categories: domain knowledge, system knowledge (the act of knowing the system’s mechanisms, logic, and interactions), and generic software engineering knowledge (knowledge of the system’s usability and the ability to interpret error messages).

Two studies focused on the soft skills required by software testers. Matturro (2013) analyzed 43 recruitment advertisements to investigate the frequency of demand for soft skills. Among the most common soft skills for software testers were teamwork, proactive, analytical/problem-solving skills and being methodic. Similarly, Florea and Stray (2018) analyzed 400 recruitment advertisements for software testers across 33 countries, following a pre-existing skill taxonomy. The most frequently solicited traits were the ability to communicate both verbally and in writing, analytical and problem-solving skills, team spirit, and independent-working skills.

Cavin (2015) assessed the viability of military veterans in becoming software testers by initiating a coursework program. Findings show that most veterans possess human factors that were aligned with the characteristics of a tester - such as communication, team player and flexibility.

4 Case study methodology

To answer our research question, we conducted an exploratory multiple case study (Yin, 2018) following the guidelines proposed by Runeson et al. (2012). The qualitative research process included collecting data from a context-specific environment, analyzing the data, and interpreting the data.

We collected the data through interviews and supplemented them with observations. We interviewed 22 practitioners in three companies. In addition, we observed 11 meetings and 13 of our interviews in their daily work. We attended eight daily stand-up meetings, two test status meetings, and one domain-expert workshop.

Case contexts

Company A is a medium-sized software-service-providing company with over 500 employees in Norway, Denmark, Ukraine, and Slovakia. The company offers expertise in project management, software testing, software development, interaction design, maintenance, and security. The organization consists of over 1,000 employees in Scandinavia and has focused, among other things, on automotive financing, sales financing, and loans. The interviewed professionals used agile methods and mainly Kanban development, combined with Scrum ceremonies. They used product backlog items extensively—usually in the form of a user story. Some of the interviewees had no time-boxed sprints, instead working with a continuous stream of tasks, as they appeared in the backlog. They were involved in
prioritizing tasks and maintaining the product backlog. The testing professionals also participated in the test-automation processes. The team used mainly Slack, an instant communication tool that has 12 million active users daily (Novet, 2021). The tool allows written, verbal and video communication and supports coordination and problem solving in teams (Stray, 2021).

Company B is a large-size supplier of software and services, mainly in the business, accounting, resource-planning, procurement, and retail areas, with over 9,000 employees in Scandinavia, Europe, and South America. All of the interviewed software practitioners were members of agile teams, working distributed in two countries. These practitioners were parts of different teams, yet their agile practices were rather similar. The teams had 6–10 members and a team leader who acted as Scrum master. The product owners were external to the team, which used two-week sprints, with the product backlog as a central element in prioritizing and assigning tasks.

Company C is a large financial services group provider whose main offices are located in Norway but with branches throughout the world and a medium-size software-development department. The interviewed practitioners were parts of different teams, working in modified agile processes. The sprints spanned from weekly to monthly and could vary in length, depending on the potentially shippable product’s readiness.

**Interviews**

We conducted semi-structured interviews of 22 software professionals: 14 testers, five developers, and three interaction designers (see Table 2). We used purposeful sampling (Patton 2014) to provide rich data on the factors perceived as central to the software-tester role. The interviews were conducted between October 2019 and February 2020. A preliminary analysis of 13 of the interviews is reported in Paruch et al. (2020a, 2020b), and seven of the interviews are partly analyzed in Florea and Stray (2020).

**Table 2 Overview of the interviews**

<table>
<thead>
<tr>
<th>Informant no.</th>
<th>Current Role</th>
<th>Work experience</th>
<th>Interview method</th>
<th>Interview length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Software Tester</td>
<td>7 months</td>
<td>Face-to-face</td>
<td>53 min</td>
</tr>
<tr>
<td>2</td>
<td>Software Tester</td>
<td>1 year</td>
<td>Face-to-face</td>
<td>1 hr 3 min</td>
</tr>
<tr>
<td>3</td>
<td>Software Tester</td>
<td>4 years</td>
<td>Face-to-face</td>
<td>51 min</td>
</tr>
<tr>
<td>4</td>
<td>Software Tester</td>
<td>6 years</td>
<td>Face-to-face</td>
<td>1 hr 6 min</td>
</tr>
<tr>
<td>5</td>
<td>Software Tester</td>
<td>3 years</td>
<td>Face-to-face</td>
<td>41 min</td>
</tr>
<tr>
<td>6</td>
<td>Software Tester</td>
<td>11 years</td>
<td>Videocall</td>
<td>30 min</td>
</tr>
<tr>
<td>7</td>
<td>Software Developer</td>
<td>1 year</td>
<td>Face-to-face</td>
<td>24 min</td>
</tr>
</tbody>
</table>
All of the respondents (12 women, 10 men) consented to the recording of the interviews and the printing of the results. Where necessary, we furthered our understanding of our participants’ responses with follow-up questions and confirmed that we had captured their responses in entirety. We asked targeted open-ended questions, and we gave appropriate feedback by encouraging our respondents to talk and by reflecting on their remarks. Simultaneously, the interviews were, to a great extent, protected from outside interruptions and from competing distractions.

We employed semi-structured individual interviews because they allowed us to gain insight into each participant’s views on the human software-tester factors, based on their work experience. An extract of the questions we asked the interviewees, separated into testing and non-testing positions, is presented in Table 3. The interview guide was slightly modified as our research progressed. Appendix A shows how the full interview guide for the software testers looked in our main data collection period. We held 15 face-to-face interviews and seven Skype video sessions. The shortest interview lasted 23 minutes, while the longest lasted 1 hour and 10 minutes, the average length was 52 minutes.

Table 3 Extract of the interview guide used for the testing and non-testing positions

<table>
<thead>
<tr>
<th>Testing role</th>
<th>Non-testing role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe your role in the team, with typical tasks and assignments. Give me some examples of your responsibilities.</td>
<td>Tell me about your role and assignments. Give me some examples of test-related responsibilities you had.</td>
</tr>
<tr>
<td>What kind of testing do you like best? Why?</td>
<td>How do you collaborate with software testers? In which contexts?</td>
</tr>
<tr>
<td>When practicing your job, describe your</td>
<td>In what ways do you appreciate your</td>
</tr>
</tbody>
</table>
Observations

To collect additional information for the study, such as the relative position of testers with reference to the other team members, as well as the attitude toward testers, we conducted observations in one project taking place in a customer organization of Company A. We observed the project from November 2019 to January 2020. The meeting observations included daily stand-up meetings, status meetings, and a workshop (see details in Table 4). During these meetings, we gathered information on the participants’ identities and their role within the team and company, the duration of the meeting, the topics discussed, and who facilitated the meeting. Additionally, we took notes on the testers’ behavior during the meetings and our personal impressions on what was happening.

During the workday, we recorded the timestamp and description of the events relevant to our research, such as conversations between the testers and other team members, interactions between the testers, or calls for help involving testers. We also recorded information on each team member’s current tasks. When observing casual and formal interactions during the workday, we followed the same procedure as above, noting the timestamp and the participants during the interaction. We also documented the topic discussed, as we understood it.

Additionally, we had access to all the participants in Company A on Slack, and could observe the participants communicating in an open Slack channel for the testers. The observation notes, together with Slack logs, were used to extract additional support for the data collected through observations and interviews. For example, we observed what the participants discussed in the channel that could shed light on our findings and used Slack for follow-up questions to the interviewees.
Table 4 Overview of meetings observed

<table>
<thead>
<tr>
<th>Meeting</th>
<th>Number of participants</th>
<th>Length of meeting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily stand-up</td>
<td>7</td>
<td>8 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>7</td>
<td>12 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>8</td>
<td>12 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>6</td>
<td>8 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>8</td>
<td>11 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>8</td>
<td>14 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>8</td>
<td>15 min</td>
</tr>
<tr>
<td>Daily stand-up</td>
<td>8</td>
<td>17 min</td>
</tr>
<tr>
<td>Test status meeting</td>
<td>9</td>
<td>27 min</td>
</tr>
<tr>
<td>Test status meeting</td>
<td>8</td>
<td>38 min</td>
</tr>
<tr>
<td>Domain expert workshop</td>
<td>6</td>
<td>2 hours 40 min</td>
</tr>
</tbody>
</table>

5 Data analysis

We chose thematic analysis (Braun and Clarke 2006) for the data analysis, which we found to be a flexible approach for identifying, analyzing, and reporting patterns, or themes, within qualitative datasets (Braun et al. 2019). We approached the analysis inductively, using the data as a starting point to generate themes.

Through familiarization and coding, we identified patterns of meaning, with which we generated themes, which we reviewed in a constant, iterative process. We used meeting records and observations to check how well the themes worked, both individually and overall.

We coded the collected data and grouped the codes into themes (see Fig. 2). For this purpose, we used NVivo because it enabled us to organize the codes and explore the relationships between them easily. During our analysis, more than a hundred codes emerged, which we grouped into seven traits characteristic of the agile software tester role, as described by both the testers themselves and the other team members. Figure 2 shows an example of how the raw data were processed into codes and then allocated to a theme.
Fig. 2 Example of the thematic coding of the collected data

Results

Regarding the fundamental question on the need for software testers, we found the agile software-development projects needed testers. As Informant 17 reflected, “I have been on several agile projects where they did not want a tester or a test manager because they had a product owner who tested, they had the developers who tested. But then they pushed that button, and kaboom, the system went down. And they said, ‘Oops.’ They didn’t see that. Now, it’s the tester experience that sees that.” Informant 17 further reflected on how agile ways of working is changing testing: “So I think it will go a number of years now where you have a shift over to developers doing testing, and then you have a shift back and adjust somewhere in the middle. That’s my guess.”
Regarding the identified human factors characteristic of the software-testing role, our analysis of the data revealed seven traits of the agile software tester, as presented next and depicted in Fig. 3.

**Fig. 3** The seven traits of highly effective testers

**Ability to see the whole picture**

All of the professionals stated that having to understand the project as a whole is essential for an agile tester. The ability to see the high-level aspects of the project objectives, customer requirements, quality demands, and both domain knowledge and technical knowledge was voiced as a central tester trait.

Informant 15 described this factor: “Somehow, they will have to intermediate the customer’s wishes with what is technically possible, even though the developers say that everything is possible. It’s in fact the architecture, as it is at that point [when it] is too far off to be able to do something. So, the testers need to find that balance. So, somehow, they have to listen to everyone.”

Informant 3 indicated that the trait was significant with the following example: “It’s not about finding the most bugs or running the most test cases. A tester who completed one test case but has found that the system does not work how it is supposed to is often more effective than a tester who ran a hundred test cases and has not found anything. It’s about being able to see the total picture.”

An interviewed interaction designer recalled that the testers had impressive domain and technical skills as well as the ability to use both in dealing with high-level project aspects: “He is the only one in the team who has a good overview of how things are connected, a helicopter perspective, how insurance works, the scope of different insurance types, [and] how things are connected [on the] back-end.” (Informant 12)
A developer, Informant 1, justified this factor as rather specific to the tester role, as follows: “Often, as a developer, you receive a task and you do it. Developers are not very involved in the entire process. I think it’s more exciting to be involved in both the business and technical aspects, not just super-technical.” The need for this first trait was summed up by Informant 17: “Some companies say that the developers can do the job. But they forget that the tester is the role that keeps everything, the focus and the overall quality. Sometimes, you need someone to see the bigger picture.”

**Detail-orientation**

A good tester needed to be a person who was “paying attention to details” (Informant 16). The testers stated that they had to manage a considerable amount of details on things they worked on. One interviewee in an interaction designer role (Informant 12) mentioned that the testers with whom they worked helped to identify user experience issues that the designers themselves had overlooked: “I remember he went through an old application, he began to carefully read the text that was there. I thought we were pretty attentive on death insurances - the fact that someone needed to report it when a person dies. He tested the whole process and made me aware of the text; it wasn’t pleasant in a highly sensitive situation. So yeah, he is very aware of details”

In addition, the interviewed developers mentioned that good software testers needed to have attention to detail: “This is why I love working with them, because of exactly that—they pick up things that we don’t see and things we haven’t even thought of. They inform us that ‘This is wrong’ and ‘This wouldn’t work because of this method and this module’—very logical individuals.” (Informant 10). Moreover, one developer (Informant 7) remembered that it gave them a feeling of safety if a tester was available during development: “If it’s a complex product, then it is nice to have a tester next to you. It gives me safety in the form of, ‘Okay, he confirms that everything I do is correct.’”

Informant 15, who was in a testing role, described attention to detail with a technical perspective: “One thing that it is not understood by those who don’t do automated testing is that [...] they think that everything is out of the box. They don’t realize that you have to go into the code and actually have to check which data to take, which data to send, how to execute tests. And the automatic tests have to be maintained all the time.”

**Good communication skills**

According to our interviewees, and based on the results of our observations, agile software testers spend as much as 50% of their time communicating with others. Agile also increased the frequency of communication, particularly “the
communication and the contact with the customers. Because if you release two times a year, you contact people two times a year. But here, you can be in contact with the customers on a daily basis.” (Informant 19)

The testers explained that they were often in the “buffer zone” between business and development. Informant 15 said that they often had to facilitate communication between people in order to make them talk to each other. “It’s very important, that part with communication, when you must talk to people and you need to make people talk together—to organize them a bit, because some might not even talk with each other otherwise”. Informant 3 emphasized how working with people was one of the motivational factors for her: “What motivates me in my job is demanding tasks that require that I use my head and working with a lot of people, that is always fun”. Informant 2 stated how he was motivated by working with people in a team.

One developer, Informant 8, described the business-oriented aspect of the testers’ communication: “They are very active in meetings with product-owners and ask questions about functionalities and what they need to find out in order to set up test-scenarios.” Informant 13, who works in a testing role, confirmed this point: “What I found is that my accountant background is really useful because I know the accountant language, and the developer doesn’t know that. So, I sit in the middle and translate between the customer and the developer—and that is helpful. You need to know a bit about how the customers work and what info the developers need in order to do a good job. You basically need communication skills.”

Most of the testers mentioned that they communicate mostly—and rigorously—with the developers. Two testers mentioned that they talk mostly with developers because they are the ones requiring additional information while fixing the bugs. The ability to provide more information in an understandable, coherent way was therefore essential. One tester noted that the bugs described by business staff were not always detailed: “...we have many testers from the business aspect, and they vary in how good they are at describing. I often need to add additional information so that the developers don’t have to engage in dialogue with them.” (Informant 2)

The agile testers with less work experience communicated mostly with the developers, while the experienced ones communicated with other stakeholders besides developers, such as product owners and support, as they considered them to have goals that otherwise might be overlooked. A software tester (Informant 3), described the communications network: “If the specification is too vague and we find that there can be many different ways to interpret it, then often I’m the one who has to go ask the ones who wrote the specification and find out exactly what they mean because I’m the one who specifies concrete requirements to the developers.”
When communicating, the testers needed to provide constructive feedback regarding the issues discovered and to mitigate potential negative team dynamics. For example, one software tester (Informant 3) mentioned, “I was on a project a few years back where I sat next to the developers. When I found a bug, I stood up and walked towards them with a friendly smile.” Informant 5 mentioned “Whenever I find a bug, I go to the developer and ask him if it’s supposed to be like that. I try not to point any fingers because that’s never pleasant for anyone, and it’s not appreciated.” One interviewee stated, “You will always make mistakes, and it’s important not to be afraid of mistakes, but it’s important to learn from them. So, if I see a bug and don’t discuss the bug with people for a while, then it would be a problem if I discuss the bug at a later point. Then, knowledge gets lost” (Informant 17).

In communicating with the developers, the testers were often bearers of bad news. One tester described his task in these cases: “You have to talk to people, and not everyone is super-pleased to hear what you have to say. People are stressed and not really glad to hear about problems” (Informant 15). Another interviewee stated (Informant 22): I have been in an environment with the pressure of delivering. Everyone is yelling at each other and hating each other. I do not want to be that person who says, ‘We are not ready. I found another bug.’ But, it is my responsibility, so I let them know.”

**Structuredness**

During the interviews, the software testers emphasized that they were organized and structured and that they enjoyed creating and using checklists; they mentioned that they usually noted everything in lists, in notes, or in their calendars. One tester (Informant 2) stated that in order to succeed as a tester, one had to be structured: “Even if one performs exploratory testing—through just playing around—if you’re not structured, then you might not be able to describe the steps you performed when you found a bug. If you’re just exploring without being systematic, then I don’t think you can retrace your steps. So, in my opinion, all testers must be structured.”

The testers not only appreciated their own use of structure; they also thrived in structured job environments: “I applied for a job and got it. And what do I encounter? No structure. [...] I can’t work in these conditions, so I applied for another job.” (Informant 18)

A developer (Informant 8) explained how the project became significantly more organized when the testers joined the team. He mentioned that the project lacked a concrete work process and had vague requirement specifications, in addition to a general lack of documentation: “The testers worked really hard to systematically
map, find, and clarify the requirement specifications for us, and even found things that we overlooked. And, as a result, we also adjusted and improved our work processes.”

Creativity

The interviewees described the importance of being creative in testing. Creativity allowed them to find abnormal bugs. One participant (Informant 3) described the role of creativity in this way: “I’ve managed to find weird bugs by being creative, such as mid-way force shutdowns and performing unusual process-sequences. One has to test like that because the users are always creative.”

The interviewees emphasized that creativity was necessary for the profession. Informant 17 said, “Testers need to have imagination, a passion for the end-users and a passion for quality.” However, one participant (Informant 2) who worked in a testing role explained that the team could choose not to fix all the limit-scenarios: “I daresay I use a lot of creativity to the point that I was told, ‘the bug you’ve reported is so specific that it only affects one specific customer during a leap year, so we’re not going to fix it.’”

One of the interviewed designers (Informant 11) mentioned that they found the collaboration with the testers to be valuable, as the testers were creative in solving issues such as suggesting several alternative ways to resolve obstacles: “One finds strange things and loopholes by being creative. This is very much appreciated from a tester because we (non-testers) have “tunneled” ways of testing, [...] meaning we would only test the system’s behavior when we do things right.”

One interviewee working in a developer role (Informant 10) mentioned that creativity was important for a tester to possess, mainly because of the differences in the focus of the role: “As a developer, you receive a business requirement. Your job is only to fulfill those requirements. You can have a creative process where you construct the architecture, choose frameworks, etc. But in the end, you’re fulfilling a requirement. Testers are supposed to test a system that’s going to work 100%, and there could be many anomalies. So a creative tester is most likely much more important to have on a team than a creative developer.”

One developer (Informant 7) referred to both developers and testers as creative, but in different ways: “On the developer side, I reckon it’s more how one constructs things or puts them together, while for the tester, it’s more like, ‘How can I find ways to destroy the system?’” The developers also depicted the testers as creative when verifying the non-functional attributes of the system in development, such as system flow, memory leaks, and security checks.
Several testers appreciated working with agile because it gave them and their teams considerable freedom in their projects. One tester (Informant 4) remarked that projects with a high level of autonomy allowed creativity to flourish and encouraged new ways of testing. However, she also stated that to become creative, one needs to have enough experience in both the business and technical areas of knowledge, “[...] because if you don’t know it, then it’ll be hard for you to be creative. You need to understand the domain, know enough about domain knowledge and technical knowledge in order to open the doors to creativity.” They also needed enough time, one of the testers mentioned that they did not always have the opportunity to be creative, mainly due to time pressure. The participant added that some of the bugs that were found post-testing could have been found earlier if the testers had the time to play more with the system undergoing tests.

Curiosity

Curiosity was perceived as a key aspect of the testing role, as Informant 21 illustrated: “You need to be curious, you need to have a desire to break the system. Although you don’t necessarily need to break the system, you need to have such an urge.” Most of the testers brought up that they were curious and eager to learn when it came to meeting the unknown. One (Informant 2) stated that continually learning new things was their passion: “I think it is important as a tester to be eager and possess a wish to learn. I think acquiring domain knowledge quickly and using it [...] It goes without saying that the more you know, the better you can conduct testing.”

Furthermore, informant 2 explained that his curiosity helped him to ask basic but useful questions such as, “I’m not familiar with that, can you tell me more about it?” They stated that the stakeholders and the customers of the project perceived this interest as commitment, and opened up towards the testers.

One tester (Informant 5) mentioned that their curiosity in learning new things made them a central figure on the team: “I focus a lot on teaching others my domain knowledge and testing, and it gets noticed. It’s not that they panic when I’m not at work, but rather everyone works much better when I’m here. You contribute in a way to build everyone else up and not make yourself a bottleneck.”

One tester stated that she learned the most through observation and practical experience. She felt that although the literature gave her an idea of how to do things, her knowledge became more concrete as she received hands-on experience during work. Another tester revealed during the interview that he wanted to begin conducting security testing but was unsure about where to start. His curiosity and persuasion led him back to becoming a student and taking a practical course in
ethical hacking. The interviewee stated that the course helped greatly with deepening his technological knowledge.

The testers who were coaching juniors on the role emphasized that beginning testers needed to be eager to learn new things through practical experience if possible. They explained that they would show the juniors how to perform a test and then encourage them to perform their own testing tasks through trial-and-error. For example, one tester (Informant 2) working in a mentor role explained: “I usually sit together with them and say, ‘Okay, here are some test cases that I have written. I can walk you through the first one, feel free to ask any questions.’ Then I’ll ask them to write the next test case. Finally, I’ll tell them to write some without my supervision.” He also stated that this was a way to challenge themselves: “I’d ask them, ‘Is there something you think could go wrong? Are there any other test cases you can think of? Should we talk to a domain expert if you’re unsure?’ It’s a way to challenge them to a certain degree so that they are used to this way of working.”

One developer (Informant 7) stated that an excellent tester should always be curious: “For example, if I say, ‘We also have to test the APIs,’ a good tester would admit that API testing isn’t something familiar and would request half an hour to get more insight in it.” The developer mentioned that it was always better for a tester to spend some time and come prepared to meetings instead of not knowing how to approach the information received from the developers.

**Adaptability**

The interviewees working in the testing role stated that their responsibilities and tasks changed rather frequently, and therefore, they needed to be flexible to respond quickly to change and adapt to changing work conditions. According to informant 1: “I think this is sort of the highlight of someone who is working with agile, the fact that one can quickly switch between things and the fact that one has sufficient control so that it does not take long to adapt to something different.” When asked, those who had previously worked in sequential projects stated their preference for agile. One senior tester emphasized that working with agile improved the team’s effectiveness, but they had to learn to switch between assignments quickly: “I work here because I like the way we work, and I want to continue doing that, so I need to adapt my skill set to the company and see what is needed.”

Three of the interviewees working in a testing role mentioned that they had to adapt to follow a quickly developing work situation. Informant 1 explained: “It can sometimes be as early as after looking over my tasks and feeling ready to start, someone would pat me on the shoulder and say that I have to do something else.” Another tester (Informant 5) stated: “We work with prioritization, when an item of high priority is incomplete from the developers’ side, I’ll start to work on something
with medium priority. However, often, the developers finish before I get to complete the testing on that item, so I have to drop it and start testing on the one with higher priority.”

One junior tester stated that since she was newly qualified, the constant context switching proved to be challenging to keep up with, along with keeping the same pace as the rest of the team. During our observation, she mentioned working extensively with a more experienced tester, and she stated this work approach had significantly improved her ability to switch contexts.

6 Discussion

In this section, we will first relate our findings to the results of the literature review and then we will discuss our research question. We will end the discussion by giving some implications for practice and discussing the threats to the study’s validity.

The software tester role and motivational factors

Our literature review discussed three themes: 1) software testing as a profession, 2) motivational factors for the software testers, and 3) the personal characteristics of software testers. The literature review results regarding software testing as a profession pointed to a rather negative perspective of the software tester role. For example, the findings of Capretz et al. (2019) indicated that among the people working in software development, tester and maintainer were the least popular roles. The authors found that most engineers in Canada, China, Cuba, and India perceive testing jobs as unattractive and would not choose the role of a tester. However, our study indicates that this discouraging perception might not be the case in agile projects. The interviewed practitioners working in software testing or in close collaboration with testers expressed full recognition of the merits of the software tester role. Both the testing and non-testing interviewees stated that testers were viewed on an equal level and with the same status as the others. This outlook improved the testers’ motivation to perform in their work—an exciting find, considering the previous research, such as that of Shah and Harrold (2010), who reported that seniors considered testing to be boring, a steppingstone toward a career as a developer, and that testers did not get the recognition they deserved. Our results suggest that the agile software testers’ colleagues showed deep respect for the role of the software testers; moreover, they identified traits by which the testers uniquely added value to software development.

Furthermore, our literature review showed that there are several factors that motivate software testers, such as having variety in their work, using creativity in solving
tasks, and having well-defined work with a precise sequence of steps (Santos et al. 2017). Similarly, the factors that de-motivate software testers are time pressure, lack of clear processes, and working on test activities that are tedious, boring, or monotonous (Deak et al. 2016; Gonçalves et al. 2017). Our respondents were also motivated by being able to use creativity in their work, and they enjoyed structure and using checklists. Furthermore, the participants were also motivated by complex tasks and the fact that their job required communicating and working with many people in different roles.

The third group of papers in our literature review discussed personal characteristics of the software tester, which is related to our research question and will be discussed next.

**Human factors of the agile software tester**

There is a need to improve understanding of the part that software testers play in agile projects, and especially the human aspects (Sánchez-Gordón et al. 2020). Through the literature review, we found several common characteristics of software testers, such as being open-minded, curious, disciplined, and paying attention to details (Kanij 2015; Livonen et al. 2010; Matturo 2013; Deak 2014b). Livonen et al. (2010) found that high-performing testers were thorough and patient. Studies of job advertisements have found that testers are required to be team players, proactive, and good communicators (Matturo 2013; Florea and Stray 2018). Our empirical investigation of agile software tester also found many of these characteristics. We will now discuss our empirical results in light of our research question: *Which human factors are essential for the agile software tester?*

Our findings suggest that testers need to see the whole picture. In other words, they need to know what the business side requires, whether the user journey is plausible, and whether the technical modules work with each other as intended. Our findings support the work of Hernández and Marsden (2014), who pointed out that software testers work on a wide variety of topics because their role “requires a complete view of the software,” as well as communication skills for effective collaboration with other departments. Testers who are able to take a wider view can bridge the gap between the domain and the technical part of the product, while also commenting for each respective part on what to keep in mind for the other parts during implementation. This has the potential to increase team transparency, making teamwork more fruitful while minimizing misunderstandings.

Most of our interviewees emphasized attention to detail as imperative for a tester to possess because it is useful in user experience, technical solutions, and enhancing domain knowledge and technical knowledge. This finding is in line with the survey research conducted by Kanij et al. (2014) in which most respondents agreed that this
trait was something a good software tester should have. However, Capretz et al. (2019) found a set of demotivating factors in which the requirement of detail-oriented skill demotivated Cuban software testers. Our findings suggested that being attentive allows testers to catch mistakes earlier, with an increase in teamwork, effectiveness, and motivation.

It emerged from the interviews that fellow team members often called on agile software testers to provide support by clarifying requirements or providing additional information. This often leads to an increase in teamwork through transparency, efficiency, and constructiveness, suggesting that testers also must perform much social navigation to mitigate potential negative team dynamics. The agile manifesto makes communication a central part of software development, and it therefore was not surprising that the interviewees stated that they spent half of their time communicating and that they viewed good communication skills as crucial. Earlier studies also highlighted this trait. (Deak 2014b; Florea & Stray 2018). Ahmed et al. (2012) described software testers as “the software development team’s worst enemy.” Therefore, they need good relational skills. The testers in our study confirmed that they sometimes felt that way but that they had found ways to behave that would avoid any animosity, and they highlighted the value of being able to communicate in a diplomatic and friendly manner. As the software-testing role implies bringing unwanted news to the team, our findings confirm the need for good communication skills to avoid provoking conflicts. A recent study found that one of the key reasons why software practitioners do not want to take up a testing career is the worry that other project members will be upset when the testers report the results from assessing their work (Lizama et al. 2020).

Most testers stated that writing checklists and creating notes allowed them to free up their minds to focus on other tasks. Some of our interviewees indicated that working in an agile environment could be somewhat distracting. Kanij et al. (2015) utilized the Big Five taxonomy from psychology to highlight testers’ personalities. Their findings show that testers generally have a higher level of conscientiousness than other roles—conscientiousness including orderliness, self-discipline, diligence, and dedication. Our results show that testers tend to be more organized, systematic, and structured—all of which can be subsumed under conscientiousness.

Although creativity is generally useful in software development, the degree of importance varies greatly among roles (Li et al. 2020). Regarding software testers, our study points to that creativity was perceived as necessary in two courses of action: to conduct software testing through various user personas in different scenarios, and to come up with creative ways of testing the technicalities of the system. Our study confirms the results obtained by Itkonen et al. (2013), who found that exploratory testing nurtured diverse, creative opportunities for testing.
Curiosity was an important trait of the agile software testers as it gave them a desire to investigate and an ability to gain both domain knowledge and technical knowledge, often at a fast pace. The testers were eager to learn new things actively and investigate on their own. Kanij et al. (Kanij et al. 2014) found that 70% of the 104 respondents ultimately agreed that intellectual curiosity was an important characteristic, implying that the trait was useful in enhancing the destructive mindset of a tester and help in deliberately trying to break a system. In a similar sense, Deak (2014b) found that participants considered being curious an “incentive for continuously improving the understanding of the product” as well as coming up with unusual testing scenarios. Li et al. (2020) argue that curiosity is an important trait to have in the field of software engineering because the field is constantly changing, and that curiosity is a motivating factor behind learning. Our findings support these findings, as the interviewees perceived curiosity to be an essential trait for software testers.

We found that in an agile environment, good software testers were those who could switch context fast. Having good knowledge of the domain, as well as technical competencies greatly enhanced the ability to respond quickly to changes, such as changing, adding, or improving test cases. This ability can also foster effective communication with domain experts, product owners, or technical staff in order to shape the product to its highest quality. Ekwoge et al. (2017) mentioned that adaptability was needed for the use of new tools and techniques of testing. Our results suggest that although adaptability indeed benefits technical proficiency, the trait is more holistic in the sense that it also concerns domain knowledge and the ability to switch quickly between different contexts and different mindsets.

7 Implications for practice

Our findings on the important human factors of agile software testers can have several implications. Given the effort that software companies invest in recruiting and retaining the best-suited individuals for the roles, the first implication for the practice of our study concerns the employers in the software development industry. This paper’s findings could benefit the industry by providing a set of relevant traits that software testers should possess, useful as a checklist to those in charge of hiring new testing personnel.

Additionally, our findings could benefit all professionals, whether with a technical background or not, who are considering careers as software testers by setting a frame of expectations for the role in soft factors. Garousi et al. (2020) recently reported that one of the top-ranked knowledge gaps for software engineers was software testing and argue that it is vital to teach more about testing. The second implication of our findings relates to the ISTQB syllabus (ISTQB 2019) for the agile
tester. The listed skills of an agile software tester should be updated with, for example, being able to view the whole picture and being adaptable.

The results also show a change in the perception of the role in the last decade, when Capretz et al. (2019), and Shah and Harrold (2010) found that people were choosing roles in software testing when they were technically proficient.

Another notable result is the usage of digital communication tools in agile teams. We noted that the use of Slack was perceived as positive by the testers and increased team transparency—supporting recent studies stating that the use of Slack makes communication more transparent in agile projects (Calefato et al. 2020).

Universities could include more human factors in courses on software testing. For example, instructors might teach techniques for viewing the whole picture while at the same time focusing on details. As the testers in our study also acted as a bridge between business and development, courses might include some curriculum based on agile coaching. It would also be a valuable addition for students to be able to practice through exercises how to communicate poor results. Soft skills are often seen by practitioners as more important than hard skills in new graduates, and there is therefore a need to introduce soft skills in the curriculum.

Future work could investigate the traits that are important when specializing in test automation. Automation is used to improve the effectiveness of testing (Mahmud et al. 2014), reduce human errors (Rafi et al. 2012), and improve the reliability of testing, but it is not a replacement for human labor. To write, execute, and follow up on automated tests, one must have time and invest significant effort. Therefore, employers need to check whether the testers have the time and skills to develop and maintain a good automation process. This aspect is important, as Karlström et al. (2005) report that the main impediments to adopting software-test automation are of a managerial and not a technical nature.

**Threats to validity**

There are multiple threats when conducting qualitative research. For example, construct validity refers to how well a study measures its claimed construct and concerns whether the study has been affected by the researcher’s subjective judgment (Yin 2018). According to Yin, the use of multiple sources of evidence and “letting key informants review draft case study reports” are tactics that can increase the construct validity of a study. Since what people report is not always consistent with reality, triangulation may yield more reliable data than the use of only one data source (Robson, 2011). We complemented the interviews with observations and analysis of Slack discussions, and we presented and discussed our results with the interviewees in the study.
Observing the employees working for several months might pose a risk that the interviewees in Company A was influenced by the researchers. However, we aimed to be neutral and believe that because the interviewees became familiar with the interviewer and the aim of the research it made them share information more openly because they felt safe in the interview situation. Also, as suggested by Becker (2008), to reduce observer bias we noted down as much information as possible when observing the project members interacting to increase the ability to remember details.

External validity, also known as generalizability, refers to whether the study’s findings are generalizable beyond the contextual setting of the case study (Yin 2018). For case studies, special attention is recommended for analytical generalization—that is, striving for generalizable findings and “going beyond the setting for the specific case or experiment that had been studied” (Yin 2018). We interviewed people employed in three different companies. We also aimed to increase the external validity by interviewing three roles within the software development team and asking them to describe the relevant human factors affecting the role of the software tester. To avoid ambiguity, we allowed time for reflection on the questions. We were open to all input and insight, and we maintained contact with our respondents until the finalization of the study. We routinely checked the consistency of the data in the transcripts, the codes, and the themes we used.

8 Conclusion

Through a qualitative study of 22 software practitioners performed to capture their view of the relevant human factors for the testing role, we brought to light that the software testers in agile settings were perceived as professionals with seven distinct traits: the ability to see the whole picture, an orientation toward detail, the possession of good communication skills, a strong sense of structure, creativity, curiosity, and adaptability.

Being able to see the whole picture is a valuable trait for agile testers; they must think of the quality of all aspects of the product encompassing technical and domain-specific abilities in order to conduct testing effectively. The agile testers spend much time communicating and need to do this in a timely and friendly manner. They had to give feedback in a constructive way to mitigate potential negative team dynamics. The use of agile methods increased the frequency of communication, particularly with customer representatives. An essential trait to possess is being detail oriented. The testers enjoyed being attentive to details, catching things others did not see. The developers expressed a feeling of safety to have the detail-oriented testers on the team. Agile software testers have an advantage if they are creative. The testers in our study were creative in testing user
perspectives such as personas, domain coverage, and nonstandard scenarios, which made them discover abnormal bugs. Furthermore, they were seen as structured in testing and enjoyed creating and using checklists, and they were viewed as curious in ensuring the best agile product. In the final traits that we found, agile testers should be adaptable to changes and adept in switching contexts.

The findings from our study can be used in the industry, in particular by those with responsibility for recruiting software testers, helping to ensure that the candidates for the role exhibit these traits. Moreover, those considering a career in software testing should use our results to scrutinize and assess their fitness for a role as a tester in an agile environment.

9 Acknowledgments

We would like to thank all the participants for their generous and thoughtful collaboration on this study and for allowing us to observe and conduct interviews. We extend a special thanks to the companies supporting our research and for making the collaboration setup possible.

10 References


Software Testing and Analysis (pp. 231–240). New York, NY, USA: ACM. https://doi.org/10.1145/1831708.1831737


## Appendix A

### Interview guide (Paper V)

<table>
<thead>
<tr>
<th>Part</th>
<th>Question</th>
</tr>
</thead>
</table>
| **1. Introduction** | Present ourselves.  

Thank the participant for taking his/her time to be interviewed.  

Inform the participant about the privacy policy and voluntary participation.  

- Data gathered from the interviews will be used to answer the research question  
- Any personal data will be treated confidentially  
- All participants will be anonymized  
- It is voluntary to participate in the interview, and the participant can at any time withdraw his/her consent  

Ask permission to record.  

Estimate the length of the interview (45-60 minutes). |
| **2. Background and warm-up** | What is your current role within the company?  

How long have you been working as a software tester?  

How long have you worked at this company? |
| **3. Role and tasks** | Can you shortly explain the project you work in? How long have you worked on it?  

What kind of software development process do you use for the project?  

Describe your role in the team, with typical tasks and assignments. Give me some examples of your responsibilities.  

What kind of testing do you like best? Why?  

When practicing your job, describe your knowledge in testing and beyond testing.  

How do you discover a significant bug?  

Are your skills measured and followed up at work? How?  

Are some testing skills more complicated to acquire? |
| **4. Team context** | What can you tell me about your connection with the team (roles, team members)?  

Who do you collaborate with the most? For how long?  

Do team members show interest in other individuals’ tasks? How? How do you feel that software testing is integrated into the team’s practices? |
| **5. View on testers** | What traits should a good software tester have? Why?  

What about a suboptimal tester?  

Are any traits or skills more important for a tester to have? |
| **6. Barriers / negative** | Have there been challenges in your role related to testing? |
Appendix B

Interview guide (Papers IV, V, VI)

GENERAL INFORMATION
Walk me through your experience as a software tester: in how many companies have you worked so far? What were those software testing jobs about?
Why did you switch jobs?
What is your current job title?
Give me some examples of tasks and responsibilities that you have.

EDUCATIONAL BACKGROUND
Tell me about your educational background.
Describe courses that were useful to you in your current job.
Should you have studied more in the university about software testing?

THE HIRING PROCESS
How did you get hired throughout your testing career?
How were your skills verified and by whom?
Did you have extra skills worth showing at hiring? Were there some unimportant skills evaluated at hiring?

SKILL FOLLOW-UP AND APPRAISAL
Are your skills measured and followed up at work, throughout the year?
Do you wish for more support from the company in the development of your skills?

SELF-EVALUATION OF SKILLS
What kind of testing do you like best?
Which of your skills are most important in your job?
What skills do you lack to do your job? Would you like to strengthen some of your skills? Which ones?
Which skills are most valuable for a tester to have, in general?
Can you name some easy skills to acquire and practice, as a tester?
Are some testing skills more complicated to acquire? Which ones?
What are the biggest challenges to your learning?

LEARNING SOURCES
Do you prefer formal education (finalized with a certification) or informal education?
What were the sources of learning that you used? Which learning sources do you like best? Why?
What determines your choice of a learning source?
Is it important for you to obtain certification after you finish a course or training?

LEARNING PATTERNS
Do you have to switch often from one project to another? How do you feel about it?
Do you have to acquire new skills when switching responsibilities/projects/teams? Give me some examples.
Do you receive training when you rotate responsibilities?
Can you tell me something about the speed at which you had to acquire skills?
How do you prepare yourself for the future, in terms of skills?
LEARNING DRIVERS
Who should be the driver of your skill development: you or the company?

WORKING WITH OTHER ROLES
Who do you collaborate with the most (which roles)?
What are typical things that you learn from your colleagues?
How much time do you work in a week with other roles?

RETROSPECTIVE
Would you like to continue working in your role as a software tester, or do you consider switching roles in the future?
If you would have a magic wand to fix one thing in testing, what would that be?
Is software testing a respectable job?