The Journal of Systems & Software 204 (2023) 111742

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

On the roles of software testers: An exploratory study☆

Raluca Florea, Viktoria Stray *, Dag I.K. Sjøberg

University of Oslo, Oslo, Norway

ARTICLE INFO

Article history:
Received 1 September 2022
Received in revised form 19 April 2023
Accepted 2 May 2023
Available online 31 May 2023

Keywords:
Software testing skills
Software testing job advertisements
Software testing career
Agile testing
Empirical software engineering

ABSTRACT

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 subroles.
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.

1. Introduction

Despite its importance, the role of the software tester is vaguely defined (Cunningham et al., 2019) and is not as well-investigated as the role of the software developer, which includes subroles such as full stack developer, back-end developer, front-end developer, and mobile developer (IEEE Standards Association, 1990; International Electrotechnical Commision, 2010; Montandon et al., 2021). A survey conducted by Capretz et al. (2015) revealed that a software tester career was less attractive than a career as a system analyst, software designer, or software developer. Among the factors negatively influencing the choice of a career in software testing, Fernández-Sanz et al. (2009) identified a lack of maturity in the job market for testers and an unpredictable career path. Deak and Stålhane (2013) found that one reason for a (perceived) lower status of software testing was a lack of understanding of responsibilities and tasks, for example, the perception that testing does not require creativity or coding. In a study of 220 software professionals in four countries, Waychal et al. (2021) identified the lack of management support through unrealistically loaded work schedules, a scarcity of test resources, and not being involved in decision-making as barriers to a career in testing.

Cunningham et al. (2019) call for more in-depth analyses of the role of a software tester and a roadmap for the evolution of the role. With agile software development, there is a trend to focus less on specific testing roles (Bath, 2020). A challenge is to ensure that the range of testing responsibilities and associated required skills are covered. Defining a role means focusing on specific responsibilities and skills (Assyne et al., 2021).

Previously, we studied the employers' requirements for testers and proposed a taxonomy of software testing skills demanded by employers, including test-related, technical, soft, and domain-specific skills (Florea and Stray, 2019b). We observed that there is a need for better specification and a more refined division of the various roles to help clarify demands from employers and expectations of applicants for tester positions. Consequently, we formulated two research questions:

- RQ1: What are the different software tester roles?
- RQ2: How are the software tester roles reflected in job advertisements?

We interviewed 19 software testers in 13 companies who were asked about their job skills, tasks, and responsibilities. The analysis of the interviews identified different software tester subroles. We then investigated how these roles were reflected in 400 software tester job ads.

Our investigation provides the following major contributions. We identify five roles of the software tester: *domain-specific tester*, *test automation specialist*, *test infrastructure specialist*, *user experience tester*, and *test manager*. Then we describe the tasks

☆ Editor: A. Bertolino.

* Corresponding author.
E-mail addresses: ralcamf@ifi.uio.no (R. Florea), stray@ifi.uio.no (V. Stray), dagsj@ifi.uio.no (D.I.K. Sjøberg).

https://doi.org/10.1016/j.jss.2023.111742

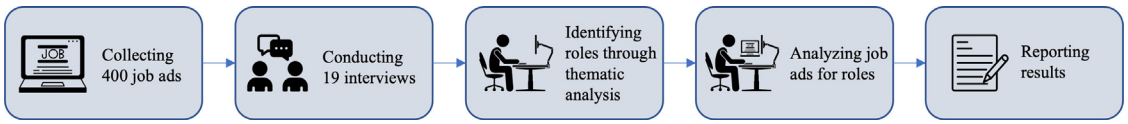

Fig. 1. Workflow of the steps of the study.

preferred and avoided, main collaborators, and what is perceived as issues by the different roles. The analysis of job ads reveals that most employers ask in their ads that candidates fill multiple testing roles. Test automation specialist is the most demanded tester role. Finally, we provide a checklist that employers can use when specifying their job ads for software testers.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 discusses the methodology for data selection and analysis. Section 4 reports and discusses our findings, and Section 5 concludes.

2. Related work

A study by Saldaña-Ramos et al. (2012) revealed that testing personnel often lacks the competencies needed to carry out their tasks efficiently. Even though software testers' main activity is often related to bug-finding, they frequently engage in activities such as demonstrating divergence from requirements, checking compliance with standards, evaluating the ability to withstand stressful load conditions, detecting attempts to gain unauthorized access, and assessing attributes such as usability (Bertolino, 2007).

Itkonen et al. (2012) proposed a classification of knowledge required in testing, which includes domain knowledge, system knowledge, and general software engineering knowledge. Good domain knowledge was also emphasized by Merkel and Kanij (2010). Further, Assyne et al. (2021) conducted a literature review on essential software professional competencies and found that the most important hard skill for a software test engineer was to write and automate tests with code.

Recently, Garousi et al. (2019) conducted a survey among software practitioners and found software testing to be one of the main knowledge gaps in software engineering education. The majority of software engineering graduates enter the workforce with good development skills but lack the necessary testing skills, partly because academic curricula place less emphasis on teaching software testing as a formal software engineering subject (Astigarraga et al., 2010). In an investigation of skill gaps between graduating IT students and employers' expectations, Radermacher et al. (2014) found that the greatest challenges lay in using configuration management systems, writing unit tests, and communicating with colleagues and customers.

Testing processes and practices may benefit from standards and models. Wong et al. (2017) present an overview of 59 catastrophic events caused by undetected software vulnerabilities due to a lack of adequate testing. They suggest that testing should be conducted systematically, for example, by applying various testing standards. Garcia et al. (2014) identified 23 test process models in a systematic literature review, out of which most were adapted or extended versions of the two models Test Maturity Model integrated (TMMi) and Test Process Improvement (TPI). Garousi et al. (2017) identified 58 maturity models and found that such models help advise companies on improving their testing process, where TMMi and TPI are the most widely used. TMMi (van Veenendaal, 2022) is a framework that describes best practices for software testing to help software engineering companies improve their test activities and processes. The model includes five levels of test process maturity and describes how to implement test practices in each of the levels systematically. The TPI model (Sogeti, 2022), with TPI Next as the newest version, specifies how to measure effectiveness and support test process improvement in areas such as test strategy, planning, coordination, and evaluation.

The variety of test processes and activities and the associated needed knowledge motivate defining specialized roles in software testing. Kassab et al. (2021) analyzed 1000 job ads for software testers, and found that the most common roles asked for were testing/QA engineer, tester and analyst. The most requested skill was in the area of test automation.

Saldaña-Ramos et al. (2012) conducted a review of test competence models, a job analysis in a focus group, and a survey among practitioners. They identified four roles of the software tester: test contract manager, test manager, test engineer, and tester. The authors emphasized that a challenge is that many models, including TMMi, do not include the roles involved in the testing processes or the competencies required.

3. Research method

To explore the role of the software tester, we collected 400 job ads, interviewed 19 software testers, identified software tester roles through thematic analysis and investigated how they were reflected in the job ads; see Fig. 1.

3.1. Interviews

The interviews were semi-structured and conducted in October–November 2019. We used purposeful sampling of the interviewees (Patton, 2014) to obtain rich data on the practices and tasks of experienced software testers. For the topics on which we received brief answers, we followed up with additional questions and encouraged the interviewees to reflect on their remarks. The interview guide is shown in Appendix A.

Additionally, we collected the CVs of the interviewees and public online information from their professional profiles. We analyzed the interviewees' academic backgrounds and professional paths in different workplaces, industries, roles, and job responsibilities.

The interviewees worked within software development departments in various industries—accounting (3), banking (3), logistics (3), telecommunications (2), automotive (2), IT services (2), audit (2), tourism (1), and healthcare (1)—and performed their roles in Norway, Romania, Sweden, Denmark, France, the United Kingdom, Ireland, Pakistan, Poland, Croatia, and Ukraine. Sixteen interviewees worked in internationally distributed large companies (more than 250 employees), while three worked in small to medium-sized companies. Table 1 gives an overview of the interviewees.

When possible, we conducted the interviews face-to-face (7) and otherwise via Skype (12). The conversations lasted from 40 min to 1 h and 30 min (55 min on average). All the interviewees consented to an audio recording and the subsequent publication of the results. The interviews were conducted in English, Romanian, or Norwegian. The first author transcribed the interviews and translated the Romanian transcriptions into English. A preliminary analysis of the educational background, skill acquisition, and learning preferences of the 19 software testers was reported by Florea and Stray (2020). Seven of the

Table 1
The interviewees.

Id	Age	Educational attainment*	Year work exp.	Year work exp. in IT (T, D, ITM)***	No. jobs in IT	Current company size**	Current role in testing***	Prior roles in testing***	Identified role ****
Id1	46	BA, Economics	21	**16** (16,0,0)	3	L	TM	T, TM	1
Id2	44	MSc, Logistics	16	**13** (13,0,0)	3	L	TM	T, QA, BA	5
Id3	29	MSc, Computer Science	8	**8** (7,1,0)	2	S/M	T	QA	2
Id4	44	MA, Economics	19	**19** (19,0,0)	3	L	TM	T, QA, BA	3
Id5	35	BSc, Mathematics	12	**12** (12,0,0)	3	L	QA	T	3
Id6	39	BSc, Accounting & Finance	11	**9** (9,0,0)	4	L	Sr. TA	T, QA, BA, TAu	2
Id7	51	BSc, Biochemistry	29	**19** (19,0,0)	2	L	TM	T	5
Id8	55	Undergrad.	32	**19** (19,0,0)	2	S/M	Sr. T	T, QA	1
Id9	40	MA, Business & Economics	18	**8** (8,0,0)	3	S/M	BA	QA	1
Id10	36	MA, Economics	14	**9** (8,1,0)	2	L	QA	T	1
Id11	40	MA, Linguistics	15	**9** (8,0,0)	2	L	QA, BA	T	2
Id12	41	BSc, Computer Science	21	**18** (13,5,0)	3	L	T	T	2
Id13	33	MA, Economics	11	**7** (7,0,0)	2	L	QA, BA	T	4
Id14	59	BSc, Computer Science	35	**35** (27,8,0)	5	L	Sr. TA	T, TM, OT, UAT, QA, BA	4
Id15	49	Undergrad.	24	**14** (14,0,0)	3	L	QA, T	T, QA, DT	5
Id16	36	MSc, Computer Science	11	**11** (11,0,0)	2	L	Sr. T	T, QA	5
Id17	50	MSc, Informatics	26	**26** (10,5,11)	2	L	VE	T	3
Id18	41	MSc, Physics	17	**15** (11,0,4)	4	L	DTM	T, TM, C, TAn, QAM	5
Id19	43	MA, Banking	15	**9** (9,0,0)	4	L	AuA	T, TM, TAu, ST	3

* BA = Bachelor of Arts; BSc = Bachelor of Science; MA = Master of Arts; MSc = Master of Science
** S/M = Small / Medium; L = Large
*** AuA = Automation Architect; BA = Business Analyst; C = Consultant; D = developer; DT = data tester; DTM = Domain Test Manager; ITM = Manager in IT; 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
**** The role the interviewee preferred. See Section 4 for an explanation of the roles.

transcripts were previously used in a separate study on human factors in agile software development (Stray et al., 2021).

In this study, thematic analysis was used to extract qualitative information through explicit codes, defined as patterns of meaning (themes) across the qualitative data sets (Braun and Clarke, 2006; Patton, 2014). We analyzed the transcripts using the NVivo12 software package. The analysis was performed inductively using open and axial coding (Stol et al., 2016). We followed the steps for thematic analysis as recommended by Braun and Clarke (2006). In *phase 1*, we first transcribed the interviews and read and re-read the transcripts. We also gathered and read the CVs of the interviewees. We regularly recorded our thoughts and ideas. In *phase 2*, we applied open coding on all the transcripts. Examples of our initial codes were "test-scripts issues", "prioritizing work", "lack of knowledge", and "changing jobs". This analysis resulted in 388 codes. In *phase 3*, we used axial coding to collate the codes into themes related to tasks, collaborators, issues, and preferences. In *phase 4*, we refined the codes and themes in three iterations. For example, we removed candidate themes of little relevance for this study, such as "perceived job safety". In *phase 5*, we identified and named 20 themes related to four aspects ("tasks preferred", "tasks avoided", "main collaborators", and "perceived issues in testing") of five areas of software testing activities, which combined gave rise to five named roles of the software tester.

3.2. Ads

To study employers' requirements regarding testing jobs, we used a curated set of 400 job ads collected online from four global employment websites (Monster.com, Glassdoor.com, Simplyhired.com, Indeed.com) as a data source. The advertisers were small and large organizations operating in both the public and private sectors, including Amazon, Expedia, Nasdaq, Texas Instruments, Verizon, Motorola Solutions, Fujitsu, VISA, IBM, Nokia, New South Wales Government, National Bank of Canada, Accenture, and Sogeti. From the job ads, we extracted, among other things, job titles, required or preferred background (education, certifications, job experience, skills), and job responsibilities and tasks. Results of this analysis were reported in Florea and Stray (2019a,b, 2018).

For the purpose of this study, we analyzed the ads regarding the roles that emerged from the interviews with the testers. For example, a job ad asking the candidate to *"collaborate with the product owner to understand the project requirements"* was coded as a request for a domain-specific tester. *"Implement and configure test environment"* was coded as a request for a test infrastructure specialist. See more examples in Table 2. To access the curated job ads and data analysis material, see Appendix B.

We also investigated the job ads to identify additional roles beyond what was found in the interviews. We found some additional requirements in a few ads, for example, pen testing, but no new clear roles occurred in the ads.

4. Results

The two research questions are answered through the results of the analyzed interviews and job ads, respectively, RQ1 in Section 4.1 and RQ2 in Section 4.2.

Table 2
Excerpt of coding of one job ad.

Role description	Role description	Requirements	Roles identified: 4
[...] is now looking for a highly motivated QA Test Senior Specialist specialized in Non-Functional Testing to join the [...]QA team in Stockholm. As a QA Test Engineer Senior Specialist you will be integrally involved in meeting our challenge to develop robust high quality software, with ever increasing performance requirements and simultaneously contribute to improve agility and time to market in a changing business and regulatory environment.	• Review business requirements and hardware configuration specifications (*Role 1*) • Create test effort estimates, test plans, test specifications and test reports (*Role 5*) • Design, plan, coordinate, lead and test system conformance to Service Level Agreements (*Role 5*) • Execute all types of tests in the non-functional area such as transaction latency, throughput, robustness, failover, operational tests, exploratory, etc. (*Role 3*) • Automate non-Functional tests (using internal test frameworks in java or perl or other scripting languages such as batch, python, etc.) *(Role 2)* • Track and maintain defect lists and aid development teams to ensure resolution of reported defects • Provide customer support and lead investigations of non-functional issues • Maintain and improve working processes to gain efficiency and effectiveness *(Role 5)*	• Experienced in x86 server hardware, or similar • Experienced in Linux administration including server installation • Proven experience as HW/SW performance engineer including familiarity with proofing, debugging and networking tools such as: SAR analysis, use of performance hardware counters, GRPROF, STRACE, ITRACE, Traceroute, etc. *(Role 3)* • Solid knowledge of TCP/IP Networking with proven experience • Big Data knowledge with experience in statistics and using tools to analyze large data sets • Knowledge of scripting in bash, perl, python or similar script languages *(Role 2)* • Knowledge in Java programming is a merit • Ability to build good relationships with peers, business partners, and clients • Excellent written and verbal communication skills in English • Experience from working with financial systems especially trading/clearing/settlement systems would be a merit *(Role 1)*	*Role 1* *Role 2* *Role 3* *Role 5*

Table 3
Description of the roles of the software tester.

Role	Description
Role 1: Domain-specific tester	
Tasks preferred	Transforming the business requirements into system specifications; validating that the implementation follows the specification.
Tasks avoided	Writing automated test scripts; tasks related to the test infrastructure.
Main collaborators	Project stakeholders, customers and product owners (to elicit business requirements); developers to implement and verify the system requirements in short iterations.
Perceived issues in testing	An insufficient number of product owners allocated to the teams leads to poor-quality implementation of the business requirements.
Role 2: Test automation specialist	
Tasks preferred	Designing and implementing automated test scripts and test suites; executing the test suites on targeted test configurations; evaluating the execution results and 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); Role 3 testers (practicalities regarding the infrastructure for testing).
Perceived issues in testing	A lack of resources allocated to automated test scripts, resulting in test suites not being groomed and execution failures not being 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); deployment engineers (setting up test infrastructure solutions); Role 2 testers (deployment of test scripts, availability and specifics of the test environments).
Perceived issues in testing	A lack of dedicated resources to create and maintain up-to-date test infrastructure; outdated test environments, resulting in a 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 the use of the software).
Perceived issues in testing	A lack of resources for testing 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 for testing activities; monitoring to ensure 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-upon testing prioritizations (leading to insignificant testing on par with business-critical testing); lack of clear responsibility for prioritizing test activities and resource allocations.

4.1. Interviews

Our first research question was: *"What are the different software tester roles?"* Through the data analysis, we identified five roles in software testing: domain-specific tester (Role 1), test automation specialist (Role 2), test infrastructure specialist (Role 3), user experience tester (Role 4), and test manager (Role 5). Table 3 describes the roles regarding tasks preferred and avoided, the roles' main collaborators, and the main issues perceived by the testers in the respective roles.

We categorized each interviewee in only one role, as they preferred to work in only one of the roles. Four of the interviewees acted in Role 1 (Interviewee Id1, Id8, Id9, Id10), four in Role 2 (Id3, Id6, Id11, Id12), four in Role 3 (Id4, Id5, Id17, Id19), two in Role 4 (Id13, Id14), and five in Role 5 (Id2, Id7, Id15, Id16, Id18); see also the last column in Table 1.

4.1.1. Role 1: Domain-specific tester

Most of the domain-specific testers had a non-IT educational background and were initially hired to perform testing with a domain-specific focus. Additionally, they played a part in connecting the business and development sides of the project.

One Role 1 tester stated, *"I found that my accountant background is really useful because I knew the accountant language, and the developer didn't know it. I was in the middle and translated between the customer and the developer—and that was helpful"*. Another recalled, *"At the first job, I was truly useful. Their implementation was technically correct but did not produce as intended. So they needed someone to know how it should work, and I had the right domain knowledge"*.

Depending on how the project was organized, the Role 1 testers collaborated with the project stakeholders, the product owners, or directly with customers to transform business requirements into testable system specifications. An interviewee explained the need for these testing activities with the high frequency of software releases: *"The communication and the contact with the customers is now more important than before. If you release twice a year, you contact people twice a year. But here, you can be in contact with the customers on a daily basis"*. Further, the Role 1 tester spent extensive time with software developers on implementing and checking system specifications.

Since the Role 1 testers were often the only testers on their teams, they were all required, at some point, to perform test automation or perform test infrastructure-related tasks. While most of our respondents in Role 1 did not decline to manually write step-by-step scenarios for test automation specialists, they felt they performed automation-related tasks inefficiently because they did not have the required skills.

Some Role 1 testers were able to automate basic test scenarios with help from other colleagues, but they were overwhelmed when asked to perform advanced test scripting and test infrastructure services independently, with possibly technically demanding test environments. As a tactic to avoid test automation responsibilities, those in Role 1 prioritized domain-specific work tasks and postponed the script-related tasks, which were eventually picked up by other colleagues with more technical skills. One interviewee explained, *"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, and I need to talk to these servers. All these things are just to get started. […] And since I don't have the skill set, it takes a really long time. The problem is that the company doesn't even think to check that all the testers are able to contribute to test automation"*.

Most of the testers in Role 1 aimed to complete software-related certification programs, such as those provided by the International Software Testing Qualifications Board (ISTQB) or the Project Management Institute (PMI). They were eager to become certified to enhance and document their software-related skills, which might be useful for further careers in software development since they did not have formal education in testing. Additionally, they attended software conferences to learn about current technological trends and grow their network of software professionals.

4.1.2. Role 2: Test automation specialist

The Role 2 testers preferred technology-related tasks to domain-related tasks. They enjoyed transforming manual test scenarios into automated test scripts, checking test execution logs, and identifying unexpected results. An interviewee stated, *"The industry type should not be that important; it's absolutely not that relevant. As an automation engineer, it's the technology that is different"*.

Their closest collaborators were developers, with whom they analyzed the execution logs and troubleshot the software under test. Further, they collaborated with the Role 1 testers to transform the software specifications into step-by-step scenarios ready to be automated. The Role 2 testers were not comfortable acting in Role 1 because the role required both domain-specific and interaction soft skills, which they did not excel at: *"I find it hard with interaction. I take a lot of time until I decide to talk to people, even if people don't see me like this."*

Several Role 2 interviewees mentioned that performing manual testing and pointing out errors in the software written by the developers sometimes led to tensions or conflicts. However, when they could refer to errors as part of the test script results, communicating the mistakes became less personal and thus less stressful. Therefore, they appreciated developing skills in script automation. One interviewee explained, *"It's totally different to do automatic than manual testing. In automated testing, you are building something, not destroying it. And, you know, it feels good."*

The Role 2 interviewees said that they felt pressed to take over the management of test environments (Role 3), as the project management regarded it as a normal duty of the technical testers. However, the interviewees considered the skills needed to administer testing environments too complex for their skill range, as described by a Role 2 tester: *"When they ask us to take on operations tasks, I get scared because you have to handle infrastructure and test the infrastructure, too"*. Nonetheless, Role 2 testers exchanged practical information regarding the testing infrastructure with the Role 3 testers.

4.1.3. Role 3: Test infrastructure specialist

Role 3 tasks involved setting up and managing test environments, which included mastering fast-changing testing tools and implementing patches, updates, and integrations. One tester stated, *"I think that if test environments are in decent control with good quality, the testing would be much easier and smoother"*.

The challenges described in the literature above were also identified in our study. One interviewee explained the need for up-to-date testing tools as follows: *"The tools evolve very rapidly nowadays; we have many problems with maintenance"*. Another tester commented: *"The biggest source of problems, misunderstandings, and quarrels concerning testing lies actually, as far as my experience tells, in bad test environments. If they are not properly maintained, topped with bad test data management, many defects arise"*.

The testers in Role 3 collaborated most with the software architects and other software infrastructure engineers to discuss and agree on the test environments and tools. However, they were reluctant to engage in domain-specific product testing, as expressed by one interviewee: *"I had over a year when they insisted that I should strengthen my accounting part. And I said 'Yes,*

yes,' but always, when there was a task to be done that required more technical skills, I would choose that one first". Furthermore, two Role 3 interviewees mentioned that they avoided performing test scripting because they considered such tasks doable by testers with fewer technical skills.

4.1.4. Role 4: User experience tester

The Role 4 testers were those that focused on the user experience. They acknowledged the need for the well-established roles of the domain-specific tester (Role 1) and the technical roles (Roles 2 and 3). However, they saw a further need for a role of testers who would be the user's advocates and focus on specifications, guidelines, and assessments of usability aspects. An interviewee stated: *"Technical aspects like automation are important in testing, but there is much more than that: It's about having good error messages, creating interfaces, and so on. For a tester, it is important to give feedback on these usability aspects. I think there is a lot of focus on the technical part, but the emotional part is also very important".* The Role 4 testers described their mission as bringing up end-user issues, such as the frequent use of a sub-optimal feature, unclear messages, and annoying interruptions.

One Role 4 tester stated that testers often compensated for the lack of available UX developers, for example, by checking the implementation of the UX guidelines.

The Role 4 testers were skeptical of using automated tests as the only or dominant indicator of software quality. They were concerned that the UX guidelines would not be the focus of the developers. One interviewee stated: *"I have been very happy and lucky that we have reorganized how we work. I am part of the UX group. So we do bug-fixing from a UX perspective and test new features. Trying to be a bit more customer-centric".*

The Role 4 interviewees avoided domain-specific testing because they did not have sufficient domain-specific expertise to dive into the details of the requirements. They mentioned major stress caused by both domain-specific and technically demanding testing activities assigned to them, particularly if the documentation was incomplete or outdated. One interviewee stated, *"I told my manager, 'You're putting me on this all by myself without documentation or specs.' And I told him that I found it really stressful. It was a hard time for me".*

4.1.5. Role 5: Test manager

Those who worked in Role 5 focused on test coordination and prioritization. The testers preferring this role attended to the issues arising from the lack of coordination and prioritization in testing, such as duplicated tests, gaps, and misguided priorities.

We found that several testers were concerned about companies' lack of attention to test management. In a few of the companies, there was not even a formal test manager. In these companies, the interviewees took on the role of a test manager because they considered that project management alone could not have a good enough grip on testing. They attempted to balance test priorities and customer needs and ensure that the various team members did their part of the testing. However, being in the management role often became frustrating because they had no formal mandate to manage the team members and the testing activities.

One interviewee, who was formally a test manager, felt that the test managers were unjustly disregarded: *"I see that some people think that the need for test managers is no longer there. They believe we don't need test managers because the technical testers can do the job".*

The need for dedicated practitioners to coordinate and prioritize the test activities was brought up by the interviewees acting as test managers (Role 5). For example, deciding which tests to automate.

The practitioners in Role 5 worked together with many other roles, including project managers, product owners, customer support, other testers, and developers. One interviewee stated, *"Regardless of whether you are a developer or tester, you need to talk to each other to ensure a good product for the customers".*

4.2. Job ads

Our second research question was: *"How are the software tester roles reflected in job advertisements?"* Table 4 shows the number of ads where each of the identified roles was requested in the job ads (the percentages exceed 100 because one ad may include several roles). The majority of the ads seek test automation specialists (72%), domain-specific testers (63%), and test infrastructure specialists (45%). For each role, Table 4 shows one example of a description from an ad. Most ads ask for several roles in the same ad, as shown in Table 5.

Table 5 shows the combinations of roles asked for in the same advertised job. Among the ads, 6% did not ask for any of the identified roles, 23% asked for one role. Approximately one-quarter of the ads describe responsibilities of two roles, with the combination of test automation specialist and test infrastructure specialist being the most popular, followed by the combination of domain-specific tester and test automation specialist. Almost half of the employers asked for three or more roles in the same job ad.

The request for several roles in the same job ads is in contrast to the preference of our interviewees to specialize in one role only. Employers may not know who will apply, and in a job market where companies compete for qualified testers, employers may want to attract testers with different qualifications. In addition, it may be too costly for companies to have dedicated testers in each role.

If employers need testers to fill the technical Roles 2 and 3 only, it is a given that a technical background is required. However, when requesting a combination of these roles with non-technical roles, the ads must clearly state that technical competencies are needed. As shown in Table 5, the most common combination was that of Roles 1, 2, and 3; that is, a candidate is supposed to handle domain testing, test automation and test infrastructure. Further, combining technical and non-technical skills may cause people without a technical background to refrain from applying.

5. Discussion

Many studies support the need for the different testing roles that we identified. Beer and Ramler (2008) emphasized that substantial domain knowledge was needed for particular aspects of software testing, which would require the tester to have had work experience from the particular domain. In agile teams, product owners typically possess domain knowledge, and practice has shown they are involved in testing tasks (Bass et al., 2016). A domain-specific tester would be crucial in teams without a product owner.

One explanation why most employers request test automation specialists (Role 2) might be that automated testing may greatly reduce the testing costs (Khari, 2020). A recent survey of practices regarding test automation (Wang et al., 2020) identified a lack of guidelines for designing and executing automated tests, which was perceived as a great source of dissatisfaction for software practitioners. Such guidelines must be developed by specialists on test automation. However, finding people with sufficient skills has been reported as a challenge in test automation (Rafi et al., 2012). Garousi et al. (2020) found in a survey among practitioners that test automation was considered the most challenging

Table 4
Roles requested in job ads.

#	Role	Number	Curated excerpt from the ad
1	Domain-specific tester	254 (63%)	You are the key figure between business and IT, and you work in an intense dialogue with internal business customers on viable IT solutions. Your main responsibilities are: business solutions development together with the process owners, user stories design, functional solutions design. You translate, where necessary, business objectives to a clear requirements specification for application engineers. You define a test strategy and develop test scenarios based on user stories.
2	Test automation specialist	286 (72%)	You need to create test scenarios and test cases from business and functional requirements in TFS/Microsoft Test Manager or HP Quality Center. You need to execute test cases, record and track all defects in Microsoft Test Manager, participate in planning sessions, and provide daily status updates with the QA team at 8am. You are able to interpret detailed technical and functional specifications and use this information to plan and develop test scenarios and cases. You have experience in developing and executing SQL scripts for validation purposes.
3	Test infrastructure specialist	181 (45%)	You will be developing automation frameworks and infrastructure in our quest for continuous delivery. We aim to build quality at the same time as increasing the speed of our release cycle. You possess excellent analytical and diagnostic skills and are both logical and creative in your testing approach. You have an agile mind-set and thrive in a fast-paced environment. Most important is hands-on experience in the following areas: automation frameworks such as Selenium, NightWatch, Appium, Node JS, build and deployment automation systems (e.g., Jenkins, Maven, Puppet, Foreman), knowledge and understanding of unit test frameworks such as Mocha, Chai, Sinon.
4	User-experience tester	84 (21%)	You will analyze data from multiple sources, translate research findings into specific, prioritized recommendations to quickly improve digital experiences, collect information to analyze and evaluate existing or proposed functionality across a variety of platforms. (e.g., desktop, smartphone, tablet), conduct user research activities (including review of analytics, user interviews, ethnographic observation, workflow analysis and contextual inquiry), perform usability testing, run a User Testing Lab, and create journey maps, personas and user scenarios based on individual initiatives and for different clients.
5	Test manager	121 (30%)	The role is newly established. The main responsibilities are developing, implementing and maintaining overall test frameworks, processes, and guidelines and continuously improving test activities. The candidate will act as a test manager within software development, implementation, and upgrade projects, manage virtual teams of testers, implement methods for automated testing, and ensure the quality of test cases.

testing activity. To be proficient in developing automated tests, testers need to master automation frameworks and tools (Beer and Ramler, 2008) and the programming languages of the system under test (Imtiaz et al., 2019). Furthermore, deciding which test to automate may require a cost–benefit analysis (Ramler and Wolfmaier, 2006).

Karhu et al. (2009) found that while running automated testing scripts reduced execution costs and improved software quality, new costs were introduced when implementing and maintaining automation infrastructure. Wiklund et al. (2014) reported studies that identified poor configuration and management of test environments. Wang et al. (2020) also identified poor management of test environments in a survey among practitioners. Such costs and management challenges require a particular role responsible for the testing infrastructure.

Garousi (Garousi et al., 2018) conducted a systematic literature review of testing embedded software and categorized typical test activities. Two of these were for example, test automation and test management

Traditional development methodologies such as Waterfall, including the variant V-model, specify separate test teams (Najihi et al., 2022). In agile and DevOps development, the testers are integrated into the development team and should work continuously on testing increments. The knowledge transfer between developers and testers (Li et al., 2010) increases the mutual understanding of the activities of the other team members. One principle of Scrum is that the teams should be cross-functional. Gregory and Crispin (2014) state that the competencies needed should be available within a team and that specific roles or titles are unnecessary. Their view is consistent with Scrum (Schwaber and Sutherland, 2020), which does not recognize a *tester* as a title in the development team.

In contrast, our view is that having specific named roles with given areas of responsibility and corresponding qualifications will help ensure that teams have the right competencies. In practice, a company may not afford to have all the five tester roles within one team if a person fills only one or two roles and at the same time work in only one team. However, in large-scale agile, separate test teams may work with several development teams (Dikert et al., 2016) to justify the cost of having specialized testers. Specialized testing teams may also help overcome the problem that testers within a Scrum team may have little to do at the start of a sprint and too much work to do at the end of a sprint, with the consequence that work items may not be sufficiently tested before the deadline. This negative consequence of the time-boxing of Scrum has been an argument for switching to Kanban (Sjøberg et al., 2012).

5.1. Limitations

One limitation is that the interviewees represented a much smaller scope of companies than the job ads. For example, most interviewees worked in Norwegian organizations, where low-hierarchy organizational structures might affect how the tester role is defined and practiced. Further, the interviewees of our study worked with information systems development. However, other subfields may need additional specific tester roles. For example, in embedded systems development, there may be a need for a hardware–software integration tester who ensures that the hardware and software components of the embedded system work together seamlessly.

Another limitation is that the analysis of the interviewees and job ads was conducted primarily by the first author and may therefore be particularly subjective, even though the codes and themes were discussed with the two co-authors. To further mitigate the threat of subjectivity, we presented preliminary results to all our participants. We asked them to provide feedback on erroneous or possibly missing information and to reflect on

Table 5
Software tester roles demanded in the same role.

Number of roles found in ads	Total number of ads	Percentage of ads	Number of ads by role	Role 1	Role 2	Role 3	Role 4	Role 5
0 role	**23**	6%	23	-	-	-	-	-
1 role	**93**	23%	45		✓			
			33	✓				
			8			✓		
			5					✓
			2				✓	
2 roles	**111**	28%	37		✓	✓		
			34	✓	✓			
			15	✓				✓
			9	✓			✓	
			7	✓		✓		
			6		✓			✓
			2			✓		✓
			1		✓		✓	
3 roles	**96**	24%	48	✓	✓	✓		
			13	✓	✓		✓	
			11	✓	✓			✓
			9		✓	✓		✓
			5	✓			✓	✓
			3	✓		✓		✓
			3		✓	✓	✓	
			3		✓		✓	✓
			1	✓		✓	✓	
4 roles	**62**	15%	30	✓	✓	✓		✓
			15	✓	✓	✓	✓	
			14	✓	✓		✓	✓
			2		✓	✓	✓	✓
			1	✓		✓	✓	✓
5 roles	**15**	4%	15	✓	✓	✓	✓	✓
Total	400	100%	400					

Table 6
Checklist.

Check:	Recommended action:
…that the primary tasks and responsibilities of the job are clearly stated and made explicit. Possibly focus on one of the Roles 1–5.	The applicants should be pointed to the essential parts of the job since the tasks listed in the ad may not be equally important. Be explicit. Generic tasks such as "automate tests" may conceal myriad tasks beyond scripting, such as creating automation specifications, implementing a testing tool, and maintaining the test environment.
…that the expectations for the applicant reflect a realistic amount of work.	Ensure that the tasks for the advertised position do not represent more than what one person can handle in a full-time job.
…that the coherence between the task descriptions and the list of skills required. In particular, if the job involves technical tasks, check that IT education is requested.	Ensure that the primary skills required of the applicant match the primary tasks stated in the ad. Requirements should be marked as mandatory or "nice-to-have." Employers may expect testers to perform tasks they do not have the education or experience to handle. For example, candidates with only domain-specific knowledge may be unable to perform technically-demanding tasks.

additional input. Ten of the 19 interviewees responded to our request and reviewed the preliminary results. We received no disagreements with the results of the analysis.

Based on the identified roles in the interviews, we investigated how and to what extent these roles were reflected in the job ads. If we instead had first explored roles in the job ads based on the listed test activities, other roles might have emerged.

5.2. Implications for practice

Table 3 describes the typical activities of the different roles. Implementing these roles will help ensure that a company perform the associated activities.

Based on the interviews and ads analysis, we propose the checklist shown in Table 6 to be used by companies when specifying requirements for open testing jobs. For example, one job ad stated the following:

> *You will be working closely with the different development teams to define and map out the Test strategies for your product area, and create Test plans to execute those strategies. Automate everything!! We want you to be creative in pioneering new tools to build high quality systems. The ideal candidate will essentially be a Full stack QA testing both the front and back-end. (...) You have a good eye for UI/UX design, understand methodology behind creating amazing user workflows.*

Some tasks are clearly specified and made explicit, while, for example, "automate everything" is imprecise. Furthermore,

the ad lacks focus because it tries to encompass three roles (domain-specific tester, test automation specialist, UX tester), thus violating checklist point one.

Even though "automate everything" should not be interpreted literally, the broad area of tasks indicates an unrealistic amount of work, thus violating checklist point two. Deak et al. (2016) found that many testers struggle because of their overloaded schedules.

It is unlikely that the company will find someone with all the required skills. The extensive list of responsibilities and requirements may even discourage potential candidates from applying. We advise companies to prioritize responsibilities and tasks and mark competence as mandatory or "nice-to-have", given the priority.

Ideally, different persons should fill different testing roles. Therefore, a job ad should clearly specify different testing roles so the job seeker can choose which role(s) to apply for. Consequently, such a distinction between roles may simplify and make the recruitment of testers more systematic and attract more suitable candidates for each role. We acknowledge that smaller companies may not afford to have several software testers, so the candidate would have to fill many roles. When formulating job ads, one should also be aware of potential gender bias in the text (Kanij et al., 2022).

6. Conclusion and future work

Through a qualitative analysis of 19 interviews, we identified five roles in software testing: domain-specific tester, test automation specialist, test infrastructure specialist, user experience tester, and test manager. We then analyzed how these roles were represented in 400 job ads. The most in-demand roles were the test automation specialist and domain-specific tester.

Most of the ads (71%) requested that testers perform in multiple roles. In contrast, most of our interviewees wished to work in only one role. They were inclined to develop skills and become proficient in a specific area to work efficiently and produce results of high quality.

Finally, the ads frequently combined requirements for highly technical skills (test automation and infrastructure) with less technical skills (domain, UX, and management). Our interviewees found this combination challenging. The testers with limited technical skills avoided assuming responsibilities relying on technical competencies, such as test scripting and test infrastructure implementation. On the other hand, the technically skilled testers felt that the non-technical roles in testing would be a suboptimal use of their capabilities.

Future work will explore among employers how the identified roles are present in both small and large companies, and how well our descriptions of the activities fit the roles. The proposed checklist will also be introduced to obtain feedback on its applicability and possible improvements.

CRediT authorship contribution statement

Raluca Florea: Conceptualization, Methodology, Validation, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Viktoria Stray:** Conceptualization, Methodology, Validation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Dag I.K. Sjøberg:** Conceptualization, Methodology, Validation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

We would like to thank the interviewees who participated in our study, and the editor and the three anonymous reviewers for their valuable feedback.

Appendix A

General information.

- What is your current job title?
- Give me some examples of tasks and responsibilities that you have.

Experience.

- How long have you worked as a software tester?
- How many companies have you worked in?
- What were those software testing jobs about?
- Have you switched jobs? If yes, why?

Education.

- Tell me about your educational background. Do you hold an academic degree? From which institution?
- Describe courses that were useful to you in your current job.
- Do you feel you should have studied software testing more? How would that have helped?

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?

Skills.

- What kind of testing do you like best?
- Which of your skills are most important in your job?
- How do you discover a significant bug?
- 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?
- Can you name some easy testing skills to acquire and practice?
- Are some testing skills more complicated to acquire? Which ones?
- What are the biggest challenges to your learning?
- Are your skills measured and followed up at work? How?
- Do you wish for more support from the company in the development of your skills? What kind of support?

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?

Learning.

- What sources of learning do you use?
- Which learning sources do you like best? Why?
- What determines your choice of learning source?
- Did you have to take any extra certifications? On what topic? On whose initiative? Who paid for it?
- 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 have to acquire skills?
- How do you prepare yourself for the future, in terms of skills?

Closing.

- Would you like to continue working in your role as a software tester, or are you considering switching roles in the future?
- If you had a magic wand to fix one thing in testing, what would that be?
- Is software testing a respectable job?
- Do you have any questions for me?
- Is there anything else you would like to discuss that was not covered by the questions asked?

Appendix B

The 400 processed job ads and data analysis material are available online, https://doi.org/10.5281/zenodo.7845798.

References

Assyne, N., Ghanbari, H., Pulkkinen, M., 2021. The state of research on software engineering competencies: A systematic mapping study. J. Syst. Softw. 111183.

Astigarraga, T., Dow, E.M., Lara, C., Prewitt, R., Ward, M.R., 2010. The emerging role of software testing in curricula. In: 2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments. IEEE, pp. 1–26.

Bass, J.M., Beecham, S., Noll, J., Razzak, M.A., 2016. All hands to the pumps: The product owner role in small companies, Lero technical report: 2017_1. Lero Technical Report.

Bath, G., 2020. The next generation tester: Meeting the challenges of a changing IT world. In: The Future of Software Quality Assurance. Springer, Cham, pp. 15–26.

Beer, A., Ramler, R., 2008. The role of experience in software testing practice. In: 2008 34th Euromicro Conference Software Engineering and Advanced Applications. IEEE, pp. 258–265.

Bertolino, A., 2007. Software testing research: Achievements, challenges, dreams. In: Future of Software Engineering (FOSE'07). IEEE, pp. 85–103.

Braun, V., Clarke, V., 2006. Using thematic analysis in psychology. Qual. Res. Psychol. 3 (2), 77–101. http://dx.doi.org/10.1191/1478088706qp063oa.

Capretz, L.F., Varona, D., Raza, A., 2015. Influence of personality types in software tasks choices. Comput. Hum. Behav. 52, 373–378.

Cunningham, S., Gambo, J., Lawless, A., Moore, D., Yilmaz, M., Clarke, P.M., O'Connor, R.V., 2019. Software testing: a changing career. In: European Conference on Software Process Improvement. Springer, pp. 731–742.

Deak, A., Stålhane, T., 2013. Organization of testing activities in Norwegian software companies. In: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops. IEEE, pp. 102–107.

Deak, A., Stålhane, T., Sindre, G., 2016. Challenges and strategies for motivating software testing personnel. Inf. Softw. Technol. 73, 1–15.

Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. J. Syst. Softw. 119, 87–108.

Fernández-Sanz, L., Villalba, M.T., Hilera, J.R., Lacuesta, R., 2009. Factors with negative influence on software testing practice in Spain: A survey. In: European Conference on Software Process Improvement. Springer, pp. 1–12.

Florea, R., Stray, V., 2018. Software tester, we want to hire you! An analysis of the demand for soft skills. In: International Conference on Agile Software Development. Springer, pp. 54–67.

Florea, R., Stray, V., 2019a. A global view on the hard skills and testing tools in software testing. In: 2019 ACM/IEEE 14th International Conference on Global Software Engineering. ICGSE, IEEE, pp. 143–151.

Florea, R., Stray, V., 2019b. The skills that employers look for in software testers. Softw. Qual. J. 27 (4), 1449–1479. http://dx.doi.org/10.1007/s11219-019-09462-5.

Florea, R., Stray, V., 2020. A qualitative study of the background, skill acquisition, and learning preferences of software testers. In: Proceedings of the Evaluation and Assessment in Software Engineering. pp. 299–305.

Garcia, C., Dávila, A., Pessoa, M., 2014. Test process models: Systematic literature review. In: International Conference on Software Process Improvement and Capability Determination. Springer, pp. 84–93.

Garousi, V., Felderer, M., Hacaloğlu, T., 2017. Software test maturity assessment and test process improvement: A multivocal literature review. Inf. Softw. Technol. 85, 16–42.

Garousi, V., Felderer, M., Karapıçak, Ç.M., Yılmaz, U., 2018. Testing embedded software: A survey of the literature. Inf. Softw. Technol. 104, 14–45.

Garousi, V., Felderer, M., Kuhrmann, M., Herkiloğlu, K., Eldh, S., 2020. Exploring the industry's challenges in software testing: An empirical study. J. Softw.: Evol. Process 32 (8), e2251.

Garousi, V., Giray, G., Tuzun, E., 2019. Understanding the knowledge gaps of software engineers: An empirical analysis based on SWEBOK. ACM Trans. Comput. Educ. (TOCE) 20 (1), 1–33.

Gregory, J., Crispin, L., 2014. More Agile Testing: Learning Journeys for the Whole Team. Addison-Wesley Professional.

IEEE Standards Association, 1990. IEEE 610-1990 - IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries. URL https://standards.ieee.org/standard/610-1990.html.

Imtiaz, J., Sherin, S., Khan, M.U., Iqbal, M.Z., 2019. A systematic literature review of test breakage prevention and repair techniques. Inf. Softw. Technol. 113, 1–19, Publisher: Elsevier.

International Electrotechnical Commision, 2010. ISO/IEC/IEEE 24765:2010 systems and software engineering - vocabulary. URL https://www.iso.org/standard/50518.html.

Itkonen, J., Mäntylä, M.V., Lassenius, C., 2012. The role of the tester's knowledge in exploratory software testing. IEEE Trans. Softw. Eng. 39 (5), 707–724.

Kanij, T., Grundy, J., McIntosh, J., Sarma, A., Aniruddha, G., 2022. A new approach towards ensuring gender inclusive se job advertisements. In: 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). IEEE, pp. 1–11.

Karhu, K., Repo, T., Taipale, O., Smolander, K., 2009. Empirical observations on software testing automation. In: 2009 International Conference on Software Testing Verification and Validation. IEEE, pp. 201–209.

Kassab, M., Laplante, P., Defranco, J., Neto, V.V.G., Destefanis, G., 2021. Exploring the profiles of software testing jobs in the United States. IEEE Access 9, 68905–68916.

Khari, M., 2020. Empirical evaluation of automated test suite generation and optimization. Arab. J. Sci. Eng. 45 (4), 2407–2423.

Li, J., Moe, N.B., Dybå, T., 2010. Transition from a plan-driven process to scrum: a longitudinal case study on software quality. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–10.

Merkel, R., Kanij, T., 2010. Does the individual matter in software testing? Swinburne University of Technology, Centre for Software Analysis and Testing, Technical Report, 1.

Montandon, J.E., Politowski, C., Silva, L.L., Valente, M.T., Petrillo, F., Guéhéneuc, Y.-G., 2021. What skills do IT companies look for in new developers? A study with Stack Overflow jobs. Inf. Softw. Technol. 129, 106429. http://dx.doi.org/10.1016/j.infsof.2020.106429.

Najihi, S., Elhadi, S., Abdelouahid, R.A., Marzak, A., 2022. Software testing from an agile and traditional view. Procedia Comput. Sci. 203, 775–782.

Patton, M.Q., 2014. Qualitative Research & Evaluation Methods: Integrating Theory and Practice. Sage publications.

Radermacher, A., Walia, G., Knudson, D., 2014. Investigating the skill gap between graduating students and industry expectations. In: Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014. ACM Press, pp. 291–300. http://dx.doi.org/10.1145/2591062.2591159, event-place: New York, New York, USA.

Rafi, D.M., Moses, K.R.K., Petersen, K., Mäntylä, M.V., 2012. Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In: 2012 7th International Workshop on Automation of Software Test. AST, IEEE Press, pp. 36–42.

Ramler, R., Wolfmaier, K., 2006. Economic perspectives in test automation: balancing automated and manual testing with opportunity cost. In: Proceedings of the 2006 International Workshop on Automation of Software Test, pp. 85–91.

Saldaña-Ramos, J., Sanz-Esteban, A., García-Guzmán, J., Amescua, A., 2012. Design of a competence model for testing teams. IET Softw. 6 (5), 405–415.

Schwaber, K., Sutherland, J., 2020. The 2020 scrum guide. p. 13, URL https://scrumguides.org/scrum-guide.html.

Sjøberg, D.I., Johnsen, A., Solberg, J., 2012. Quantifying the effect of using kanban versus scrum: A case study. IEEE Softw. 29 (5), 47–53.

Sogeti, 2022. Test process improvement (TPI). URL https://www.tmap.net/building-blocks/test-process-improvement-tpi.

Stol, K.-J., Ralph, P., Fitzgerald, B., 2016. Grounded theory in software engineering research: a critical review and guidelines. In: Proceedings of the 38th International Conference on Software Engineering. pp. 120–131.

Stray, V., Florea, R., Paruch, L., 2021. Exploring human factors of the agile software tester. Softw. Qual. J. http://dx.doi.org/10.1007/s11219-021-09561-2.

van Veenendaal, E., 2022. TMMi specification (reference model), release 1.3. p. 224, URL https://www.tmmi.org/download/tmmi-framework-r1-3-pdf/.

Wang, Y., Mäntylä, M., Demeyer, S., Wiklund, K., Eldh, S., Kairi, T., 2020. Software test automation maturity: A survey of the state of the practice. In: Proceedings of the 15th International Conference on Software Technologies - ICSOFT. SciTePress, pp. 27–38. http://dx.doi.org/10.5220/0009766800270038.

Waychal, P., Capretz, L.F., Jia, J., Varona, D., Lizama, Y., 2021. Practitioners' testimonials about software testing. In: 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 582–589. http://dx.doi.org/10.1109/saner50967.2021.00070.

Wiklund, K., Sundmark, D., Eldh, S., Lundvist, K., 2014. Impediments for Automated Testing–An Empirical Analysis of a User Support Discussion Board. IEEE, pp. 113–122.

Wong, W.E., Li, X., Laplante, P.A., 2017. Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures. J. Syst. Softw. 133, 68–94.