



A Qualitative Study of the Background, Skill Acquisition, and Learning Preferences of Software Testers

Raluca Florea
University of Oslo
Oslo, Norway
ralucamf@ifi.uio.no

Viktoria Stray
University of Oslo
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 skills, the impediments they face, and their perception of the software testing role. **Objective:** In the current paper, we report on the background, skills, learning preferences, and role profiles as described by 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 with software testing practitioners, across eight industries. We performed a content and thematic analysis of the collected data. **Results:** The practitioners in software testing had diverse educational backgrounds, 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, whom they perceived as a learning source and 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 their skill development but need to allocate sufficient time for the learning. We identified five specialties of the role: product owner in testing, UX tester, DevOps tester, test-script automator, 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

EASE 2020, April 15–17, 2020, Trondheim, Norway
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7731-7/20/04...\$15.00
<https://doi.org/10.1145/3383219.3383252>

KEYWORDS

Software Testing, Skill Acquisition, Software Tester, Hiring Software Testers, Software Testing Profiles

ACM Reference Format:

Raluca Florea, Viktoria Stray. 2020. A Qualitative Study of the Background, Skill Acquisition, and Learning Preferences of Software Testers. In *Evaluation and Assessment in Software Engineering (EASE 2020)*, April 15–17, 2020, Trondheim, Norway. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3383219.3383252>

1 Introduction

Everywhere in the world, across industries, highly innovative companies post hiring advertisements for software testers, asking for people with a wide range of skills and competencies [1]. A previous study showed 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, in our current study, we dive into the skills acquisition and learning preferences of the professionals in software testing, to provide recommendations to the industry and academia on how to better assist these professionals in acquiring 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 what are their learning preferences?

The remainder of this paper is structured as follows: In Section 2, we discuss the methodology for data selection and analysis undertaken in our study. In Section 3, we summarize and structure our findings. We discuss and interpret the results in Section 4. We present the limitations of the study in Section 5. In Section 6, we wrap up the research with the conclusions and discuss 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, 16 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 put in contact by the companies they currently work for; we directly contacted 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 to which the respondents provided brief answers, we followed up with additional questions 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.

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 software testing more? How would that have helped?

SKILLS 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?

Fig 1. Excerpt from the interview guide

We steered 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, and probing on their remarks. At the same time, the interviews were to a great extent protected from interruptions from the outside, from competing distractions, and by not asking inhibiting questions. The atmosphere of the interview was relaxed and invited communication.

We conducted the interviews in October-November 2019, scheduling one to two interviews in a day, allowing us time to

prepare, to take notes, to write the transcripts, and to identify new tracks to follow. We conducted the interviews in two ways: face-to-face when possible and through 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 1 hour and 30 minutes. All the respondents consented to the recording of the dialogue and the publication of the results. During the interviews, we used the language with which the respondents felt most comfortable: English, Romanian, and parts in Norwegian. The first author of the paper transcribed and translated the material into English; therefore, all the data was processed in English.

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

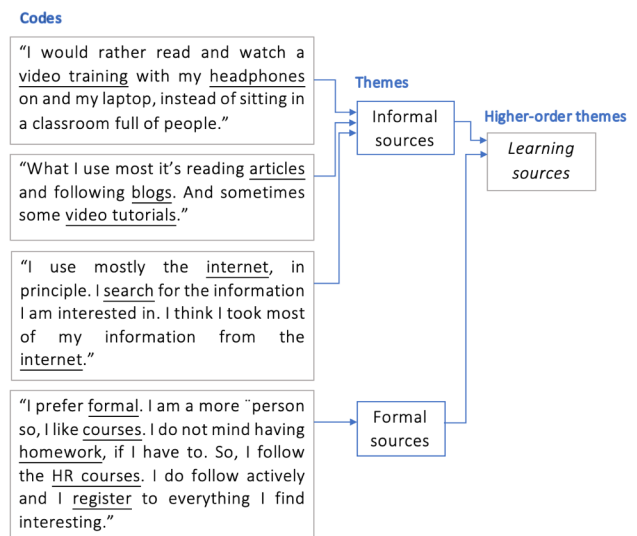


Fig. 2 Example of the thematic coding of the collected data

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

3 Results

Background of the interviewees

Because 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 had 20-30 years' testing experience, eleven participants had 15-20 years' experience, four participants had 10-15 years' experience, and two participants had 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 had worked for more than three companies in the software testing field. All of our participants had changed roles in software testing at least once.

The interviewees had a diverse tertiary educational background, ranging from economics (7) to informatics (4), military (2), mathematics (2), engineering (1), physics (1), biology (1), and 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), and health (1), and they 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.

Slightly over half of those interviewed did not have an 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 tertiary IT 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 to a software testing role for the first time seemed to read the hiring advertisements rather superficially, 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 through 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 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), and video tutorials on test techniques (8). Additionally, some of the respondents used online training providers (3) and books (2). In terms of 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 network of contacts: *"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. Additionally, the conferences proved to be useful to synthesize knowledge that the respondents were otherwise 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, some of our subjects held scrum master certifications (4), project management certifications (4), or advanced test manager certifications (2). For most of the certifications, the

employer offered the possibility to attend the courses and to take the examination, and also paid 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 spending 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 these stakeholders to have goals that otherwise might be overlooked.

Support from the employers

In general, the feedback was that their employers provided them with good support in 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 a great deal on 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. One of our respondents mentioned experiencing intense stress when they could not access the developers to obtain information needed for testing: *“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 zero bugs rejected. These metrics inhibited the exploration of the system under testing, which came inherently with the tester’s mistakes. On the other hand, imposing a minimum number of bugs put the software testers in 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 relieve some of the daily stress imposed 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 time, 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 acquiring broad software testing skills. One respondent stated, *“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 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-script automator (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, checking the viability, legality, opportunity, and costs of a requirement. However, they assessed that they were 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 toward the extensive use of the automated tests, as they perceived a product release based only on their execution results 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-script automators liked transforming manual test scenarios into automated test scripts, while they strongly felt they should not take on 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 toward 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, synthesized 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 educational backgrounds points to a diversity in the landscape of software testing, an essential factor in high-quality software development [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 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 the software testing role to the domain experts. Therefore, we report an ongoing need for universities to include information or courses on software testing.

The finding that hiring ads with too many requirements were intimidating, especially for the domain experts, indicates that 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, a fact also pointed out as beneficial in paper [8].

- RQ2: How do the software testers acquire their skills, and what learning predilections do they have?

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 therefore recommend the companies allow the time and facilitate the practice of it.

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

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 achieves 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 because there was a strong request for more concrete examples, we recommend the training providers include more samples in the teaching, and at the same time 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, came 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 successful software testing.

The determination manifested by a portion of our respondents to get 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 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 good enough support at the company level to acquire new skills indicates that employers generally focus on professionals in software testing and allocate budgets for it. However, at the team level, we recommend accounting in the task, planning the time needed for 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. Because this practice might have an additional impact on the product quality itself, we recommend employers 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 to not be imposed on to undertake multiple specialties, points to the need for employers to 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, in 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 for both establishments to consider shaping their tuition around 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 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 with the interview guide beforehand, to allow time for reflection on the questions. We were open to all input and insight, and we kept in 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 led to feedback, which we included in the study.

6 Conclusions and future work

Through 19 in-depth interviews, we conducted a qualitative analysis of the background, skills, learning preferences, and profiles of the practitioners in software testing. We determined that the software testers have a varied educational background, learned mostly from informal sources, and saw their own skills as a rather undefined heap of knowledge, increasing with each work task. They preferred hands-on and task-based approaches to acquire testing skills. Generally, employers provided good skill-development support; however, the practitioners needed sufficient time allocated 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-script automator, 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.

REFERENCES

- [1] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," *IT professional*, vol. 12, no. 1, 2010.
- [2] R. Florea and V. Stray, "The skills that employers look for in software testers," *Software Quality Journal*, pp. 1-31, 2019.
- [3] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, "Motivation in Software Engineering: A systematic literature review," *Information and software technology*, vol. 50, no. 9-10, pp. 860-878, 2008.
- [4] L. Guzmán, C. Lampasona, C. Seaman, and D. Rombach, "Survey on research synthesis in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, p. 2: ACM.
- [5] G. R. Gibbs, "Thematic coding and categorizing," *Analyzing qualitative data*, vol. 703, pp. 38-56, 2007.
- [6] T. Dybå, D. I. Sjøberg, and D. S. Cruzes, "What works for whom, where, when, and why?: on the role of context in empirical software engineering," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, 2012, pp. 19-28: ACM.
- [7] *ISTQB*. Available: <https://www.istqb.org/>
- [8] C. Kaner, B. Johnson, J. Falk, H. Q. Nguyen, and B. Lawrence, "Recruiting Software Testers," presented at the Software Development, San Francisco, 1999.
- [9] B. Zhu and S. Zhang, "Curriculum reform and practice of software testing," in *2013 the International Conference on Education Technology and Information System (ICETIS 2013)*, 2013: Atlantis Press.
- [10] L. Anderson and T. Anderson, "Online professional development conferences: An effective, economical and eco-friendly option," *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie*, vol. 35, no. 2, 2010.
- [11] E. Bjarnason, K. Wnuk, and B. Regnell, "Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development," in *2011 IEEE 19th international requirements engineering conference*, 2011, pp. 37-46: IEEE.
- [12] R. Black, "ISTQB Certification: Why You Need It and How to Get It," *Testing Experience The Magazine for Professional Testers*, vol. 1, pp. 26-30, 2008.
- [13] R. Florea and V. Stray, "Software Tester, We Want to Hire You! an Analysis of the Demand for Soft Skills," in *International Conference on Agile Software Development*, 2018, pp. 54-67: Springer.
- [14] D. Graham and M. Fewster, *Experiences of test automation: case studies of software test automation*. Addison-Wesley Professional, 2012.
- [15] P. Raulamo-Jurvanen, M. Mäntylä, and V. Garousi, "Choosing the right test automation tool: a grey literature review of practitioner

- sources,” in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2017, pp. 21-30: ACM.
- [16] C. Kaner, “Don’t use bug counts to measure testers,” *Software Testing & Quality Engineering*, vol. 80, 1999.
 - [17] A. Bai, H. C. Mork, and V. Stray, “A Cost-Benefit Analysis of Accessibility Testing in Agile Software Development Results from a Multiple Case Study,” *International Journal on Advances in Software Volume 10, Number 1 & 2*, 2017, 2017.
 - [18] L. S. Crespo, D. G. Rosado, C. B. Bueno, E. Fernández-Medina, and M. Patón, “Professional orientation in Software Engineering,” in *INTED2010 Proceedings*, 2010, pp. 3897-3908: IATED.