The roles, responsibilities, and skills of engineers in the era of microservices-based architectures

Citation for the original published paper (version of record):
Michael Ayas, H., Hebig, R., Leitner, P. (2024). The roles, responsibilities, and skills of engineers in the era of microservices-based
architectures. Proceedings - 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering, CHASE 2024: 13-23.
http://dx.doi.org/10.1145/3641822.3641871

2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)

The Roles, Responsibilities, and Skills of Engineers in the Era of Microservices-Based Architectures

Hamdy Michael Ayas
ayas@chalmers.se
Chalmers | University of Gothenburg
Gothenburg, Sweden

Regina Hebig
University of Rostock
Rostock, Germany
regina.hebig@uni-rostock.de

Philipp Leitner
philipp.leitner@chalmers.se
Chalmers | University of Gothenburg
Gothenburg, Sweden

ABSTRACT

Enterprises often try to tame the complexity of their software using microservices and practitioners generally perceive the impact of microservices as positive. However, different responsibilities fall in the hands of practitioners and new skill-sets are required to address the challenges and reap the benefits of microservices. The objective of this study is to gather and organize what industry requires from microservices practitioners. To achieve this, we conduct a qualitative analysis of 125 job-ads related to microservices that are gathered from 7 different countries, across 5 continents, posted during 14 consecutive days, sampled from a total of 1351 job-ads. We contribute with detailed taxonomies on roles, responsibilities, soft- and hard-skills that are necessary for microservices practitioners. Specifically, we detail 5 families of responsibilities, 3 of which are human-centered, describe 8 themes of popular soft-skills and describe 11 themes of popular hard-skills, along with how they relate to soft-skills. Our results indicate the importance of human-centered responsibilities and skills in microservices practitioners, and point to the existence of a job market for microservices software architects with a high demand on human aspects. Hence, our findings can help unravel organizational structures in microservices, improve training programmes, and understand the manifestation of human aspects in microservices.

CCS CONCEPTS

• **Software and its engineering → Programming teams**; **Software creation and management**; *Software design engineering*; *Software architectures*; *Software development process management*.

KEYWORDS

Microservices, Job market, Responsibilities, Skills, Human-aspects

ACM Reference Format:
Hamdy Michael Ayas, Regina Hebig, and Philipp Leitner. 2024. The Roles, Responsibilities, and Skills of Engineers in the Era of Microservices-Based Architectures. In *2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE '24), April 14–15, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641822.3641871

CHASE '24, April 14–15, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0533-5/24/04.
https://doi.org/10.1145/3641822.3641871

1 INTRODUCTION

Enterprises try to tame the complexity of their software using microservices [21]. The intention is to, among others, scale up the development, operations and maintainability of software [10]. As different responsibilities fall into the hands of practitioners, new skill-sets are required to address the challenges and reap the benefits from using microservices [22]. To this end, practitioners generally perceive the impact of microservices as positive for their systems [5]. Hence, microservices are increasingly adopted in different domains (e.g., software companies, cyber-physical systems and automotive software) [3, 9, 21]. The journey of micoservices adoption can often contain different activities and solutions [18] on a variety of topics [26].

However, organizations face challenges in adopting microservices due to the lack of relevant skills and solid understanding of several non-functional aspects needed to effectively implementing and operating microservices-based applications [2]. It is evident that microservices require from practitioners new technical development capabilities [22, 26] as well as informed judgement on the new architecture's fit with the system's context [14, 17]. Unfortunately, little empirical evidence exists on what is required from practitioners to thrive in modern software engineering (SE) environments that develop decoupled, distributed and individually contained pieces of software based on microservices.

Existing research describes the skills that practitioners need in other domains of SE, such as requirements engineering [8], testing [7] and AI engineering [13]. However, research lacks the detailed specification of the main responsibilities that engineers have in microservices-based systems. In addition, even though recent work indicates the roles and competences of practitioners in microservices [15], there is room to further specify and connect identified roles such as web-based software engineers and DevOps engineers with specific hard- and soft-skills that organizations require practitioners to have.

Consequently, the objective of this study is to gather and organize the requirements that industry has on practitioners, when asked to work on microservices-based architectures. Specifically, this study extends the taxonomy of microservices practitioners roles identified in existing literature [15] and further details the roles with specific responsibilities, hard skills and soft skills that are required from industry. Therefore, in this paper we address the following research questions:

***RQ1**: What are the roles and responsibilities of practitioners working with microservices?*

***RQ2**: What are the soft skills that microservices practitioners with different roles are required to have?*

***RQ3**: What are the hard skills that microservices practitioners with different roles are required to have, and how do those hard-skills relate to soft-skills?*

To achieve this study's objective, we conduct a large scale qualitative analysis of job-ads posted on Glassdoor. Specifically, we analyze 125 job-ads, sampled from a total of 1351 job-ads related to microservices that are gathered from 7 different countries, across 14 days. Our analysis uses techniques from Grounded Theory and the full replication package including data gathering script, raw data and analyzed data can be found at [16].

This study contributes with:

- A taxonomy of 21 responsibilities, organized in 5 families of responsibilities that microservices practitioners have.
- Organizing skills with themes of 8 soft-skills, 11 hard-skills, their relationships and their appearance in different roles.
- Definitions, descriptions and importance of the 3 human-centered families of responsibilities, focusing on the team (e.g., culture cultivation), governance (e.g. assure quality and compliance), or customer (e.g. maintaining partnerships).
- Definitions and evidence on the importance of the top 5 demanded soft-skills themes for microservices practitioners: (1) Articulation and transferability of knowledge, (2) Stakeholders management, (3) Problem solving, (4) Communication, presentation and negotiation, and (5) Leadership.
- Evidence of a job market for software architects specializing in microservices, that is distinct from other known roles of engineers associated with microservices, as they tend to relate with more human aspects compared to other roles.

2 RELATED WORK

The theoretical foundations of this study are around Microservices, their human-centric challenges, and the field of investigating job-ads to understand the skills expected from software engineers.

2.1 Microservices-based Architectures

Many enterprises increasingly use microservices to modernize their systems [12] and attempt to tame the complexity of their software using such software architectures [21]. Microservices is a type of service-oriented software architecture structuring software based on independent features and their domain boundaries [19]. The intention with microservices is to improve maintainability, decrease time to market of features, improve requirements coverage, and scale both operations and development [10]. To this end, practitioners generally perceive the impact of microservices as positive for their systems [5]. Consequently, microservices are increasingly adopted in different domains. Early adopters were software companies such as Netflix and Spotify [21], but there is evidence that organizations developing cyber-physical systems also adopted microservices [9] as well as automotive software [3]. However, different adopters use microservices differently, depending on scale and familiarity with different technologies and thus, their use cases cannot necessarily be repeated by others.

Naturally, as the technology of implementing microservices becomes clearer, text books indicate tasks to develop software using microservices [6, 19, 25]. In addition, empirical research investigating the journey towards microservices indicates the systemic and technical change that take place when adopting microservices [18]. Moreover, there is evidence of the plethora of new technologies and tools that practitioners need to master for microservices [26]. However, both text books and existing research do not help practitioners to prioritize the plethora of new technologies and plan their personal development based on how industry requirements evolve.

2.2 Human-centric challenges in adopting microservices

Practitioners are increasingly being challenged by microservices-related issues [27], face difficulties with a variety of technologies that come with developing software in microservices [22], and adopt new approaches in designing and developing software systems [26]. Besides the technical challenges that practitioners come across, there are important organizational challenges that need to be tackled for microservices to operate successfully, such as the coordination of autonomous teams that make local decisions [11]. This can lead to structural silos within an organization, which is considered the biggest obstacle of adopting microservices [2]. In fact, the challenge of coordination between teams can manifest itself in problematic technical decisions e.g., by adopting solutions that are difficult to communicate and reuse across services [11]. Consequently, there is a need for research that indicates what practitioners need to address such organizational challenges.

Furthermore, practitioners do not receive professional trainings and it is still not clearly defined what trainings practitioners should have to be prepared for microservices-related work [26]. Moreover, there is a lack of expertise among practitioners, due to difficulties in recruiting skilled personel and due to the fact that technical aspects are not the main challenges of microservices [10]. Hence, there is a need to further investigate and understand the skills of practitioners in microservices specifically, not less than other sub-fields of SE.

2.3 Analyzing skills in Software Engineering sub-domains

Existing research describes the skills that practitioners have or need in different SE sub-domains [7, 8, 13, 20], but not in microservices. For example, automated approaches exist that identify automatically hard- and soft-skills of software engineers in general [20, 24]. However, automated analysis might miss contextual information of job-ads. Hence, research exists that analyzes 101 Dutch Requirements Engineering (RE) job openings, aggregating the skill-set needed from Requirements engineers and organizing them into RE-related tasks, non-RE related tasks, competencies and soft skills that are needed from engineers [8]. In addition, a large-scale analysis of job-ads indicated which testing skills are more important for coders and which for testers, as well as the importance of automated testing in industry along with the most valuable testing tools and frameworks [7]. Finally, existing literature presents the profiles of AI engineers that companies ask for, along with the technical and soft skills that engineers are required to have [13]. Few works cover Software Architecture aspects and thus, there is a gap in understanding the skills of microservices practitioners.

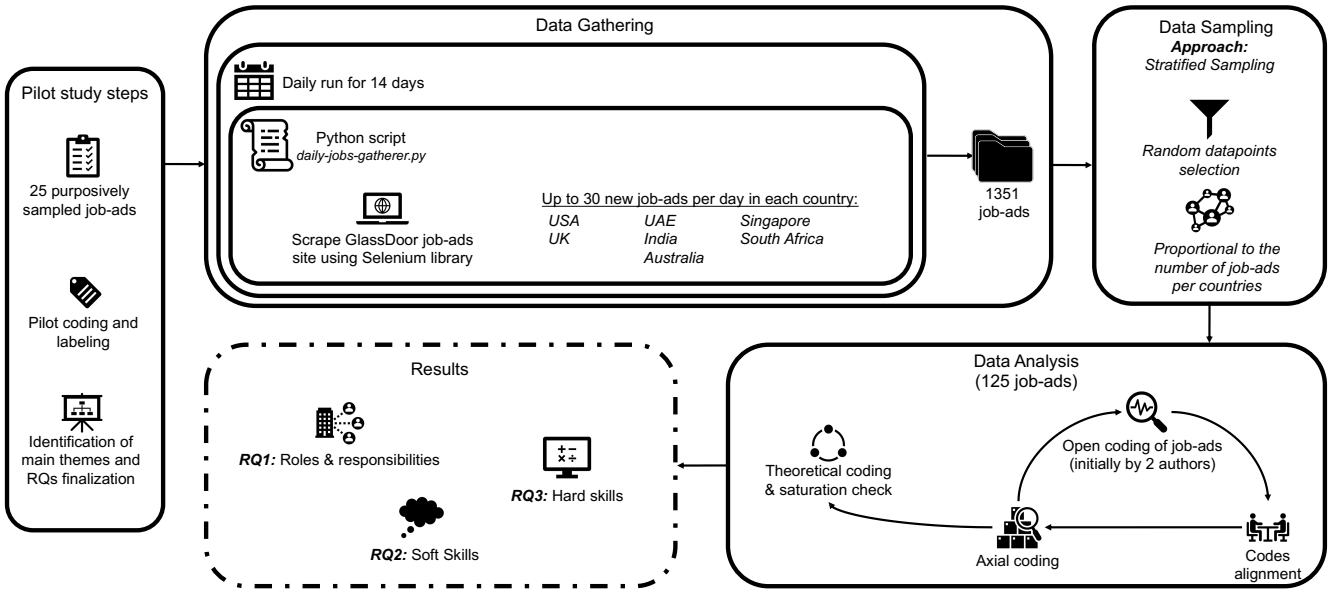

Figure 1: Methodology including all the study steps taken

3 METHODOLOGY

This study is using qualitative analysis methods, based on techniques from classic Grounded Theory as summarized by Stol et al. [23]. The GT techniques are applied on a data-set of job-ads that were posted on Glassdoor[1]. Glassdoor is selected as it is one of the largest sites worldwide specializing on recruiting, with millions of job openings[2]. As shown in Figure 1, the study has four phases (the raw data, analysis, and intermediary results can be seen in the replication package [16]). At first, a pilot study takes place where the authors gather 25 job-ads and conduct a preliminary analysis of the job-ads. The objective of this step is to identify if there are valuable insights from the job-ads on the required skills in microservices and define the research questions [23]. Then, data gathering takes place, resulting in 1351 job-ads systematically gathered across 14 working days. Next, we sample a sub-set of the gathered data, using stratified sampling and 125 job-ads are qualitatively analyzed. In this section we provide details on the data gathering, sampling, and analysis phase of the study.

3.1 Data Gathering

The objective of the data gathering phase was to systematically retrieve job-ads that have a high representation of microservices-related skills across the world. Therefore, we created a script that automatically gathers job-ads posted in Glassdoor, by automatically opening the live page of Glassdoor using Selenium WebDriver[3]. The script filters the job-ads to only gather up to 30 job-ads posted in a given day (less than 1 day old), sorts them chronologically, and is run for a duration of 14 working days (25/05/2023 - 14/06/2023) by the first author. In this way, we ensured the systematic gathering of job-ads, following the natural order they were posted, without any potential selection bias of Glassdoor's presentation of job-ads. For example, we prevent the gathering of paid ads or ads that are presented based on glassdoor's mechanisms that consider search history or usage cookies related to the user-profile. In order to ensure representativeness, we gather data from 7 predominantly English speaking countries (at least in their professional contexts), from 5 continents. Therefore, as shown in Figure 1, we create a data-set with 1351 job-ads, gathered across 14 days from different regions and specifically: (1) the Americas (USA), (2) Europe (UK), (3) Oceania (Australia), (4) Middle East (UAE), (5) South Asia (India), (6) South-East Asia (Singapore), and (7) Africa (South Africa). The authors recognize that the selected countries do not represent all regions of the world. A selection was made though, considering the language requirement of having English speaking job-ads. Metadata of the job-ads are gathered automatically in the same script, to enrich our dataset with contextual information on the recruiting organizations.

Country	Job-ads gathered	Job-ads sampled
Australia	102	8
India	415	31
Singapore	96	7
South Africa	27	2
UAE	22	2
UK	288	22
USA	377	28
Pilot sample	25	25
Total		*125*

Table 1: Job-ads data gathered and sampled per country

[1]https://www.glassdoor.com/Job/index.htm
[2]https://help.glassdoor.com/s/article/What-is-Glassdoor
[3]https://www.selenium.dev/documentation/webdriver/

Size (FTEs)	Job-ads	Year founded	Job-ads	Industry	Job-ads
1 - 50	11	2020-present	2	Information Technology	32
51 - 200	15	2010-2019	15	Financial Services	23
201 - 500	7	2000-2009	14	HR & Staffing	7
501 - 1000	8	1990-1999	14	Management & Consulting	4
1001 - 5000	11	1980-1989	1	Energy, Mining & Utilities	3
5001 - 10000	4	1970-1979	3	Media & Communication	3
10000+	44	1960-1969	10	Manufacturing	3
N/A	25	1959-older	30	Retail & Wholesale	3
		N/A	36	Telecommunications	3
				Insurance	2
				Aerospace & Defense	2
				Pharma & Biotech	2
				Real Estate	2
				Restaurants & Food Service	2
				Legal	1
				Hotels & Accommodation	1

Table 2: Meta-data overview of gathered job-ads, summarizing size, age and industries of the sampled job-ads.

3.2 Data Sampling

After gathering the data, a sample had to be selected to conduct the manual, qualitative analysis of the job-ads. Due to the diverse origin of the gathered data, the selected sampling technique on the 1351 gathered job-ads is Stratified Sampling, as per the guidelines of sampling in SE [1]. Specifically, we randomly select job-ads from the different countries. The proportion of the random selection is in relation to the amount of jobs that were gathered per country. Hence, we sample the job-ads so that our data-set ends up with the same percentage of job-ads from each country as is the case for the entire population of gathered data. The sampled data per country can be seen in Table 1. The sampled data include job-ads from several organizations of different characteristics. As shown in Table 2, job-ads from organizations of all sizes are included, as well as organizations of different age, meaning that we cover organizations with different legacy in their systems. Finally, Table 2 showcases that we gather job-ads from 16 different industry domains, covering a wide range of organizations using microservices.

3.3 Data Analysis

The data analysis phase of the study is based on the techniques of classical Grounded Theory [23]. Specifically, the analysis took place in iterations, with the authors: (1) coding 30-35 job-ads in each iteration, (2) deliberating and agreeing on the codes, (3) selecting and analyzing the codes into themes, and (4) organizing the codes into themes and a theory that explains the responsibilities and skills that are required from microservices practitioners. In the first iteration, the first two authors analyzed the same sub-set of 30 job-ads, re-using and evolving the codes identified in the pilot study. The objective of multiple authors analyzing the same sub-set of job-ads is two-fold. On the one hand, the aim is to broaden the interpretation scope of the data. On the other hand, the discussion sessions followed on aligning the codes helped to develop and maintain the same standards on the classification of content.

In the following iterations, the first author carried out the coding of the remaining data-points, with alignment sessions after every iteration. The coding included both open, axial and theoretical coding. This resulted to an incremental development of the theory, as per the guidelines of GT. After every iteration, an alignment session took place among all authors for deliberating on the axial coding results and further developing the theory. During open coding, the first author also characterized what role every job-ad was targeting, from a predetermined list of roles (presented in Section 4.1), based on the job title and job description.

Finally, we conduct a calculation to identify the relationship between identified hard- and soft-skills. Specifically, the calculation substracts the average amount of soft-skills that occur (average occurrence of soft-skills) when a hard-skill is missing from job-ads out of the average amount of soft-skills that occur when a hard-skill is included in job-ads. Thus, if the presence of a hard-skill in job-ads typically occurs with in average more soft-skills, this number will be positive. We highlight the hard-skills that are accompanied with 0.5 or more soft-skills on average, to exclude random variations.

3.4 Threats to validity

The study decisions made in the methodology entail threats to validity that the readers should be aware of.

3.4.1 External validity. First, we cannot claim complete representativeness of microservices practitioners in the entire software industry, since we only gather data of the job openings that are published in Glassdoor. We recognize that the specific platform might not be the job-posting choice of the entire industry and therefore, job-ads that are not posted in Glassdoor are not included in our study. To mitigate this challenge, we broadened our search to multiple geographic locations, including countries that have a strong presence in the specific platform. Moreover, we recognize that a big part of the industry require practitioners to work with microservices, but do not necessarily reveal this information in their job-ads. This threat is mitigated by having as broad search as possible, collecting all job-ads with "microservices" in their description.

3.4.2 Internal validity. A threat to internal validity is that authors were pre-exposed to literature in microservices, as well as to the hard skills of practitioners working with microservices. A mitigation to this threat is the focus on human-aspects in both the identified responsibilities as well as the technical (hard-skills). In addition, a limitation of our results is that we cannot claim that the list of identified responsibilities and skills is complete. We recognize that practitioners have responsibilities and skills that cannot be captured in the descriptions of job-ads. Consequently, the results presented can be seen as complementary to existing works detailing the work of microservices practitioners and can act as a ground to further investigate microservices practitioners' work and identify more responsibilities and skills. Moreover, a threat is that job-ads include the work that needs to take place in the circumstance that exist in the timing of publishing the job-ad, as it is perceived from the hiring manager or the recruiter who composed the job-ad. However, in reality there is the possibility that the circumstance change when practitioners are hired or that the perception of the responsibilities and skills needed differs from reality. To mitigate this threat, we aggregate the identified responsibilities and skills and focus our analysis to those that are most re-occurring across a large proportion of the data.

4 RESULTS

The analysis reveals details on the roles of practitioners working with microservices, in respect to their responsibilities, as well as

the skills they require to have. We stratify all our results with descriptions on the roles of *(1) Web-based software engineers, (2) DevOps engineers, (3) Data engineers, and (4) Software Architects*, as defined by literature [15] and extended in this study. First, we derive the *Responsibilities matrix* of microservices practitioners, organizing the 21 identified responsibilities in 5 families of responsibilities. Then, we provide detailed views of the skills that practitioners working with microservices are expected to have. We distinguish between hard- and soft-skills, since the data gathered reveal a great importance on the soft-skills of potential hires.

4.1 The roles and responsibilities of microservices practitioners (RQ1)

Four roles in microservices development are identified, each of them having different potential specialties. The taxonomy of roles includes *(1) Web-based software engineers, (2) DevOps engineers, and (3) Data engineers* as determined in literature [15]. Additional to these existing roles, our analysis reveals the role of the *Microservices-focused Software Architect*. The Microservices-focused Software Architect has specialties on *(1) Software engineering strategy, (2) Software product delivery, or (3) Cloud engineering*. For simplicity, we refer to this role in the rest of the paper as Software Architect. Furthermore, our analysis reveals an extension on the initial taxonomy with the specialty of *DevSecOps* that DevOps engineers can have (in addition to the existing Software Security specialty that Web-based software engineers have).

The 21 ***different responsibilities of microservices practitioners*** we identify are organized in 5 families of responsibilities, as shown in Figure 2. The families of responsibilities are derived based on the scope of their artifacts at hand, or based on the scope of the main stakeholder using the artifact. On the one hand, the scope refers to whether the responsibilities are about the *software product* or about the *software development organization*. On the other hand, the scope refers to whether the responsibilities are about the *software engineering team* or about the *Customers (internal or external)*. As shown in Figure 2, the 5 families of responsibilities are *software development support and infrastructure*, *software development process and team development*, *software product delivery*, *professional services delivery*, and *software engineering governance*. Moreover, 14 responsibilities are marked as human-centric, as a human (e.g., engineer or customer) is their main protagonist and they have to do with human aspects of software engineering (SE).

Software development support and infrastructure entails responsibilities on delivering software products to SE teams. The objective of such responsibilities is to support software development and infrastructure configuration. For example, a key responsibility is to *Implement automation tools*, that the SE team can use to unburden their work from repetitive or low value adding tasks.

> "*You'll support the identification of opportunities to automate and simplify so we can deliver better quality products for our customers.*" - Job-ad #49

Another key responsibility is to *Configure and maintain infrastructure* on which the team deploys, executes and tests their software. In addition, engineers with such responsibilities can also *Test, monitor, resolve and optimize performance aspects* of software, *Make technical decisions*, *Analyze and manage legacy systems* as well as *ensure interoperability* of developed software with existing systems. As shown in Figure 2, such responsibilities appear in all roles, with 82% of DevOps and Web-based software engineers, 74% of Data engineers and 62% of Software Architects having at least one software development support and infrastructure responsibility.

Software development process and team development entails the responsibilities that are about facilitating the software development organization for SE teams. The objective of responsibilities in this family of responsibilities is to facilitate stability, structure and predictability in both the software development process, and the team development process. Therefore, this family includes human-centric responsibilities such as *Adopt, adhere and improve development process*, but also such as *Cultivate culture, collaboration and coordination*.

> "*Adds to team culture of diversity, equity, inclusion, and respect.*" - Job-ad #71

Other key responsibilities are to *Navigate organizational structures* and *Lead and coach different teams*. The responsibilities on leading and coaching different teams are highly human-centric, since they are mainly about enabling engineers to personally develop.

> "*This is a leadership level role and will blend both deep domain and technical expertise and great passion for coaching and developing people in a "player-coach" model.*" - Job-ad #92

As shown in Figure 2, such responsibilities are more apparent in Software Architects (95%) and Data engineers (100%). Many DevOps engineers also have such responsibilities (82%) but relatively fewer Web-based software engineers (69%) are responsible for the development process and the development of teams.

Software product delivery entails responsibilities that are about the delivery of software products and software features to customers. The objective of responsibilities about software product delivery is to make sure that the software product is delivered to customers, at the right time and in the right form. In Figure 2 we can see that mainly Software Architects (57%) are asked for such responsibilities, which is reasonable as many architects are specialized in software product delivery. In addition, 37% of job-ads about Web-based software engineers and 33% of DevOps engineers have to deliver software to customers.

Professional services delivery has responsibilities about the presence of the software development organization to customers. The objective of responsibilities on professional services delivery is to establish trusted relationships with customers, as well as deliver a complete customer experience, that is not only about the product delivered. Such responsibilities include active interactions with customers, like *Consulting* or *Maintaining customer and user partnerships*. These responsibilities are human-centric as the customer is the central protagonist.

> "*Leads the effort to provide technical support to customers during software warranty period.*" - Job-ad #110

In addition, professional services delivery include to *Promote technical advancements* with active presence in open source software as well as technological community-wide discussions and contributions. Similarly to product delivery responsibilities, responsibilities on professional services delivery are observed more on

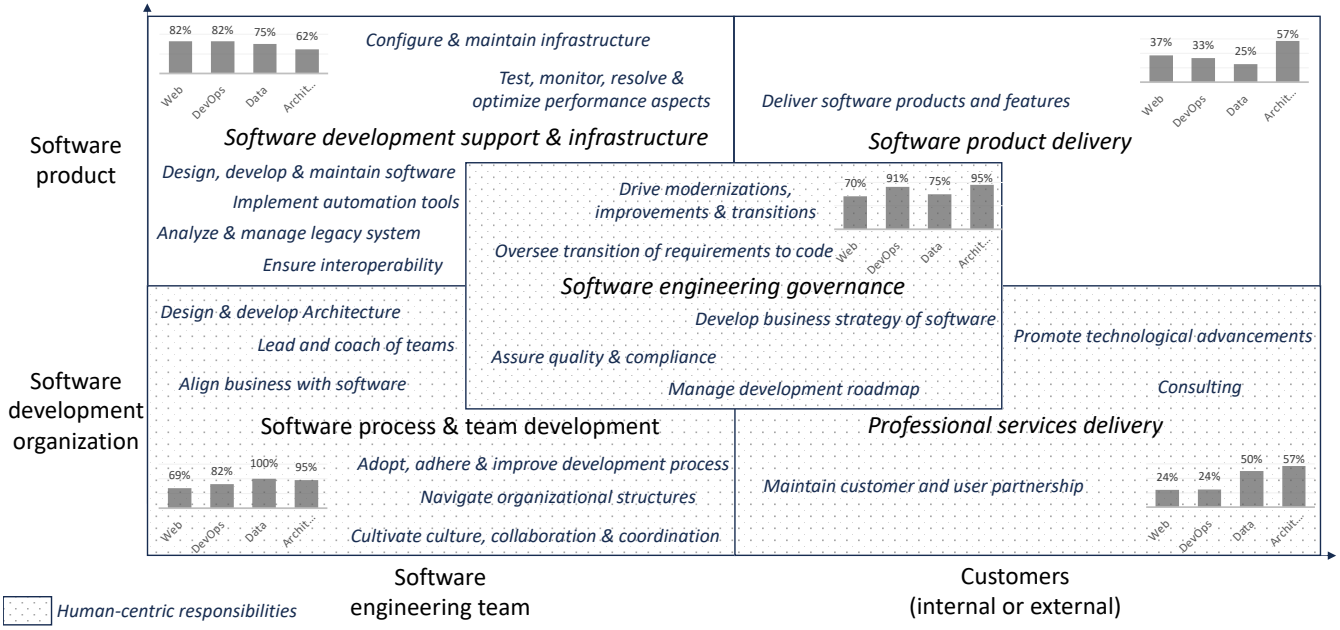

Figure 2: Responsibilities matrix, organizing the 21 identified responsibilities of microservices practitioners in 5 families of responsibilities and marking those that are human centric.

Software Architects, with 57% of them requiring such responsibilities. Relatively few DevOps and Web-based software engineers have such responsibilities (both 24%).

Software engineering governance entails responsibilities that implicitly or explicitly touch upon all dimensions. Hence, these responsibilities on the one hand influence both software products and the software development organization, and on the other hand influence both SE teams and customers. The goal of this family of responsibilities is to establish the framework on which the entire organization governs the SE activities that take place. Consequently, this family entails responsibilities focusing on high level aspects of development, for example to *Develop business strategy of software* and *Drive modernization, improvements and transitions*. Drive modernization, improvements and transitions is human-centric as a human is either a change agent in this responsibility, or the predisposed change influences human stakeholders directly.

> "*Ensures business and technology leaders have adopted an architectural approach for digital and technology change.*" - Job-ad #13

Furthermore, other key and human-centric responsibilities steering the development and delivery of software, are to *Manage development roadmap*, *Oversee the transition of requirements to code* and *Assure quality and compliance*. The responsibility of managing development roadmap is human-centric because it is centered around the engineers that develop software and overseeing the transition of requirements to code are human-centric since the focus is on manifesting the requirements of customers in the delivered software. Assuring quality and compliance is about instilling to engineers a mindset in which they follow or comply to quality and industry standards.

> "*Provides subject matter expertise and mentorship in developing system software code using commonly known best practices. Willing to adhere to [...] standards and policies regarding software development.*" - Job-ad #110

As shown in Figure 2, all roles are involved with the software engineering governance, but Software Architects and DevOps engineers have particularly high involvement, with 95% and 91% of their population having responsibilities about this family of responsibilities. 75% and 70% of data and web-based software engineers have such responsibilities.

4.2 The soft skills of microservices practitioners (RQ2)

Based on our data analysis, we identify a list of 14 soft skills that seem to be needed from microservices practitioners. Across all job ads, the 5 most demanded soft skills (as presented in Figure 3) are *(1) Articulation and transferability of knowledge, (2) Stakeholders management, (3) Problem solving, (4) Communication, presentation and negotiation, and (5) Leadership.* We refer to those skills as *the main soft skills of microservices practitioners* in the rest of the report.

The* 5 *main soft skills of microservices practitioners appear in at least 30 job-ads from those analyzed, as shown in Figure 3. Articulation and transferability of knowledge is the skill with which practitioners articulate verbally or in written form knowledge of the system and the technologies involved. The objective is to transfer the knowledge to other people (with either the same or different background). An example of a required skill on articulation and transferability of knowledge is:

> "*Strong ability to collaborate effectively within diverse teams, both technical and non-technical.*" - Job-ad #30

Soft skills	Details	All [125]	Web engineers [67]	DevOps engineers [33]	Software Architects [20]
Articulation and transferability of knowledge	Verbal or written articulation of technical knowledge for technical or non-technical stakeholders.	[49]39%	[16]24%	[17]52%	[14]70%
Stakeholders management	Handle needs and communication of stakeholders, to accommodate trade-offs and take decisions.	[35]35%	[18]26%	[12]36%	[13]65%
Problem solving	Understand and resolve a problem that is either defined or undefined.	[33]34%	[20]29%	[12]36%	[9] 45%
Communication, presentation & negotiation	Communicate with peers, make presentations and negotiate with stakeholders to realize trade-offs.	[34]31%	[17]25%	[10]30%	[10]50%
Leadership	Empowering others, support the team or being at the forefront of the team.	[32]30%	[15]22%	[13]39%	[8] 40%
Coaching / mentoring	Mentor colleagues with less experience for their technical as well as their personal development.	[27]24%	[8] 12%	[9] 27%	[12]60%
Personal development	Learning continuously or fast, as well as willingness to self-develop as a professional.	[20]23%	[10]15%	[11]33%	[8] 40%
Adaptability / growth mindset	Adapting ways of working based on environmental needs and personal growth and development.	[21]22%	[15]22%	[7] 21%	[6] 30%

Figure 3: The 8 most popular soft-skills themes extracted from job-ads, as they appear in the different roles.

Stakeholders management is the skill with which microservices practitioners handle the different requirements and needs of different stakeholders. The objective is to accommodate trade-offs and ensure that decisions are taken.

> "*Work with Leads, Engineers, Architects, Product Managers, and Business stakeholders to identify technical and functional needs of systems based on priority.*" - Job-ad #41

Furthermore, Problem solving is the skill that practitioners have to understand a problem that is either defined or undefined and find a (technical) solution to resolve it. Communication, presentation and negotiation is the important skill of practitioners to communicate with peers, present technical contents to the organization and negotiate with stakeholders to realize the right trade-offs on technical decisions.

> "*Write progress reports, conference papers, and deliver presentations to stakeholders.*" - Job-ad #21

Finally, leadership is the skill of practitioners that fuels the implementation with empowering others or being at the forefront of the team.

> "*Strong leadership, interpersonal skills and ability to work effectively with a wide range of audiences, technical and non-technical.*" - Job-ad #21

The soft skills of Web-based software engineers are distributed across all the main soft skills, without any of the soft skills being particularly apparent in most web-based software engineers. Specifically, as shown in Figure 3, *Problem solving* is the most popular soft skill with 29% of web-based software engineer job-ads requiring it. Next, *Stakeholders management, Communication, presentation and negotiation,* as well as *Articulation and transferability of knowledge* are popular with 26%, 25%, and 25% of job-ads requiring those skills. Furthermore, skills that are required for the specific role are *Leadership* and *Adaptability and growth mindset* with 22% of job-ads requiring these skills. *Adaptability and growth mindset* is the professional skill that engineers are required to have in adapting their ways of working based on the environments needs as well as growing as professionals and becoming more versatile in terms of conducting technical tasks.

> "*Flexible thinking, including the ability to pivot and try new approaches when faced with challenges. Ability to function effectively in a fast-paced environment and manage continuously changing business needs.*" - Job-ad #38

The soft skills of DevOps engineers include the 5 main soft skills of microservices practitioners as well as the soft skills of *Personal Development*. As presented in Figure 3, DevOps engineers tend to need the skill of Articulation and transferability of knowledge, with 52% of DevOps job-ads describing the specific skill. In addition, DevOps engineers need to have Leadership, Problem solving and Stakeholders management skills according to 39%, 36%, and 36% job-ads respectively. Finally, a highly demanded skill for DevOps engineers is *Personal development*, which is the soft skill containing a continuous and fast learning attitude as well as willingness to learn and self-develop as a professional.

> "*A self-driven approach to continuous learning, with the ability to work independently and make well-informed decisions even with limited supervision.*" - Job-ad #30

The soft skills of Software Architects appear at a higher percentage of job-ads than the rest of the identified roles. Specifically, from the 5 main soft skills of microservices practitioners, articulation and transferability of knowledge appears in 70% and stakeholders management appears in 65% of Software Architect job-ads. In addition, communication, presentation and negotiation appears at 50% of Software Architect job-ads, indicating the importance of this soft skill for the specific role.

> "*Negotiation, influencing, decision making and organisational skills for deployment in all areas of supplier and internal stakeholder management.*" - Job-ad #81

Problem solving and leadership appear in 45% and 40% of Software Architect job-ads respectively. Furthermore, it seems that *Coaching and mentoring* is a very important skill of Software Architects with 60% of the analyzed job-ads including this specific skill. This means that Software Architects often need to have skills in working with colleagues that they have to coach and mentor for the technologies they use as well as their development.

> "*Solid coaching expertise, working alongside our feature teams to assist them in understanding a DevOps approach to enable them to contribute themselves.*" - Job-ad #92

Finally, Software Architects are quite often asked to have a high capacity of personal development, with 40% of the analyzed job-ads requesting the specific skill.

Hard skills	Details	Soft-skills relationship	All [125]	Web engineers [67]	DevOps engineers [33]	Software Architects [20]
Quality Assurance	Code reviews, testing, setup contents of quality checks in CI pipelines	*0,08*	[88]70%	[44]66%	[31]94%	[10]50%
Cloud infrastructure	Configure infrastructure, usage of cloud services and networking	*0,02*	[85]67%	[41]61%	[26]79%	[15]75%
Data management	Setup databases, data warehouses and big data management solutions	*-0,10*	[72]58%	[41]61%	[16]48%	[11]55%
APIs and event-driven arch.	Design, implement and manage distributed or central APIs	*0,32*	[70]56%	[44]66%	[15]45%	[9] 45%
CI/CD engineering	Configure delivery and integration pipelines for continuous releases	*0,43*	[69]55%	[29]43%	[30]91%	[7] 35%
Agile	Comprehend and maintain agile process, adhere to agile principles and practices	*1,80*	[60]48%	[33]49%	[17]52%	[10]50%
Software Deployment	Configure ways of deploying software such as serverless and containerization	*0,04*	[54]43%	[26]39%	[19]58%	[7] 35%
Artifacts configuration	Manage configurations or configure the automation of repeated tasks	*0,04*	[54]43%	[18]27%	[25]76%	[7] 35%
Microservices design	Design, decompose services and define how services will operate	*0,76*	[40]32%	[13]19%	[15]45%	[11]55%
Development practices	Oversee and make decision about the software development life-cycle	*1,53*	[39]31%	[16]24%	[13]39%	[10]50%
Domain, standards & regulations	Understand domain and industry standards and comply with regulations	*1,82*	[30]24%	[15]22%	[4] 12%	[10]50%

Figure 4: The 11 most popular hard-skills themes extracted from job-ads, as they appear in the different roles. The column of soft-skills relationship indicates how many more soft-skills appear on average when each hard-skill is included in job-ads.

4.3 The hard skills of microservices practitioners (RQ3)

Besides the soft-skills identified, our analysis reveals 27 hard-skills that microservices practitioners are required to have based on what companies ask in their recruitment. Hard-skills range from technical skills that are related to developing software (e.g., *Front-end and visualization, Cloud infrastructure* etc.), to process skills that are related to software development in general (e.g., *Agile, Domain standards and regulations* etc.). We first detail the appearance of hard-skills in the roles of practitioners and then we give an overview of the relationships of hard-skills to soft-skills.

The 5 main hard skills of all microservices practitioners recruitments are *(1) Quality assurance, (2) Cloud infrastructure, (3) Data Management, (4) APIs and event-driven architectures, (5) CI/CD engineering.* As shown in Figure 4, each of the main hard skills appear in more than half (> 50%) of the job-ads. Quality assurance is the skill with which practitioners are asked to either assure software quality through code reviews and testing, or setup the contents of software quality checks in Continuous Integration pipelines.

> "*Extensive working knowledge of automation for deployment/configuration of different Application servers.*" - Job-ad #86

Cloud infrastructure is referring to the skillset that are required to configure infrastructure on the cloud, ranging from the use of cloud services to configuring networking in private clouds. Data management is the skill to setup databases, data warehouses and big data management solutions.

> "*Using Terraform to maintain our video delivery platform, which uses the full suite of AWS video products, as well as S3 and Cloudfront, to process and deliver video to our customers.*" - Job-ad #83

APIs and event-driven architectures is the skillset for designing, implementing and managing distributed end-points in services to integrate with each other either via central APIs or event-driven architectural designs. CI/CD engineering is the skillset that companies ask for configuring software delivery pipelines as well as integration pipelines of continuously releasing software features seamlessly, without compromising quality.

> "*Three-plus (3+) years of experience building and maintaining continuous delivery pipelines.*" - Job-ad #102

The most prominent hard skills required from Web-based software engineers are the core hard skills of microservices practitioners, except CI/CD engineering, which is only mentioned in 43% of the analyzed job-ads. Instead, skills on *Agile* are highly popular with 49% of job-ads containing them. *Agile* skills is the skillset required from engineers to maintain an agile development process, adhering to the agile principles and following agile practices and ceremonies.

> "*Strong understanding of Scrum, Lean, XP, Kanban and other agile development techniques.*" - Job-ad #107

The most popular skills required for this role are APIs and event-driven architectures as well as Quality Assurance, with 66% of the job-ads requesting them. In addition, 61% of job-ads include Cloud infrastructure and Data management skills.

The most prominent hard skills required from DevOps engineers are partially from the main hard skills of microservices practitioners, along with 3 other skills. Quality Assurance, CI/CD engineering, Cloud infrastructure are very prominent in DevOps engineers job-ads with 94%, 91% and 79% including them, as shown in Figure 4. Additionally, *Artifacts configuration* is also a popular skill with 76% of job-ads describing this particular skill. *Artifacts configuration* is the skill needed for engineers to systematically manage configurations of technical artifacts (e.g., infrastructure) or configure the automation of repeated (configuration) tasks.

> "*In-depth knowledge of cloud automation and configuration management tools, such as Ansible, Terraform, or Azure Automation.*" - Job-ad #69

Furthermore, DevOps engineers are often required to have Agile skills, as mentioned by 52% of the analyzed job-ads. Finally, *Software deployment* is a popular (56%) skill in DevOps job-ads, which contain the skill-set to configuring different ways of deploying software such as serverless and containerization of software packages.

> "*Experience working with containerized workloads such as Docker and Kubernetes.*" - Job-ad #58

The hard skills required from Software Architects do not come only from the main hard-skills of microservices practitioners. Figure 4 showcases that from the 5 main hard-skills Cloud infrastructure, Data management, and Quality assurance are among the

regularly required skills with 75%, 55% and 50% respectively. Additionally, 50% of Software Architect job-ads require skills on Agile, *Software development practices*, and *Domain, standards and regulations*. Software development practices are skills that help to oversee software development and make general decisions about managing the software development life-cycle. Such decisions include the choice and adherence to specific best practices on the development process, optimizing the development process, or selecting tools and IDEs. Domain, standards and regulations are skills that predispose the experience and familiarity with the business domain at hand, industry standards that might influence the software and regulations that the software needs to comply with.

> "*Good understanding of common information security management standards, frameworks, and laws / regulations: E.g. BSIMM, ISO 27001, GDPR etc.*" - Job-ad #93

Finally, 50% of Software Architect job-ads require *Microservices design* skills, to decompose services and define how the services will operate and be orchestrated in containerized environments.

> "*Construct a solution architecture around Microservices and APIs, and be able to decompose monolith applications into meaningful implementable Microservices and APIs in a Cloud/DevOps environment.*" - Job-ad #67

In total, 4 ***of the identified hard-skills appear to be related to soft-skills***.

This indicates that human aspects play a role in utilizing these hard-skills in practice. As shown in Figure 4, skills about *Domain, standards and regulations knowledge* as well as *Agile* are hard-skills that appear in job-ads, with on average 1.82 and 1.8 more soft-skills that job-adds without these hard-skills. In addition, *Software development practices* is a hard skill that is accompanied by 1.53 more soft-skills on average. Therefore, these specific hard-skills seem to be more closely related to a need for soft-skills than the rest of the hard-skills. Finally, hard-skills on *Microservices design* are somewhat related to soft-skills. That is because when they are included in job-ads 0.76 more soft-skills are also included on average. For the rest of the hard-skills we do not observe any particular differences from whether they are included in job-ads or not. Specifically, the difference of the average amount of soft-skills occurring when they are included in job-ads is close to 0.

5 DISCUSSION

Our findings bring new details regarding the roles, responsibilities and skills of practitioners in the context of microservices.

Detailing the organizational structures of companies working with microservices. The derived taxonomies of responsibilities and skills, in relation to the roles of practitioners help to define the organizational structures of companies that use microservices. Firstly, our findings of the study confirm the existence of DevOps, Data and Web-based software engineer roles, as suggested in previous research [15, 26]. In addition, the role of Microservices-focused Software Architect is introduced, with specializations on forming the organization's software engineering strategy, product delivery and cloud engineering. Another specialization introduced in DevOps engineers is that of DevSecOps, focusing on the security infrastructure of software.

Disentangling skills and responsibilities from roles. The presented details of the responsibilities taxonomy indicate the responsibilities that different microservices practitioners can have. Specifically, our findings provide detailed descriptions on how all roles are responsible for support and infrastructure, product delivery, governance, process and team development, as well as professional services delivery. This is in-line with the premise that even though organizations have experts (e.g., DevOps engineers specialized on monitoring), there is collective ownership of responsibilities on engineering tasks (e.g., Web-based software engineers responsible for infrastructure configuration). We also complement existing research on tasks and activities related to microservices [18, 26], with the skills that are required to conduct those tasks.

The importance of soft-skills in training and personal development. Training and development programmes can take our results as input to determine their contents. The role of soft-skills in the required qualifications for the job-ads is prominent and thus, training programmes for microservices practitioners should embed soft-skills development. The findings of relating soft-skills to hard-skills, indicate which hard-skills should be prioritized to be accompanied with soft-skills training in programmes. For example, courses in "Agile" should focus on training soft-skills, which is not unexpected. However, "Domain, standards and regulations" are surprisingly highly related with soft-skills. Therefore, courses in such topics should also provide soft-skills training, rather than only their contents. Finally, our findings can help structuring and scoping training programmes, bringing the choice of specializing, for instance to a specific role, or a specific family of responsibilities, or a specific sub-set of skills, to help the learning curve of practitioners [26].

The responsibilities taxonomy supporting teamwork. The responsibilities taxonomy can help SE stakeholders to frame their expectations of who is responsible for different aspects of development. The taxonomy provides a topological organization of what microservices practitioners are responsible for. This can help set expectations among practitioners working together. In addition, the responsibilities can help in planning the work for setting up microservices architectures. Finally, having a clear view of the responsibilities that microservices practitioners have can also facilitate processes such as on-boarding of new recruits in their role.

Responsibilities and skills in microservices versus other SE subdomains. Existing research investigates the skills of SE professionals, and our findings present similarities and differences with such works. Many responsibilities we identified in our findings are aligned with the identified Requirements Engineering (RE) and non-RE tasks/sub-tasks [8], such as overseeing transition of requirements to code, coordination or project management. In addition, the ways that we decompose the work of microservices practitioners complements the ways that existing research decomposed the work of testing practitioners [7], with the organization of different activities of engineers. Our findings naturally extend the state-of the art, by revealing responsibilities that are not only about RE or Testing and our analysis organizes responsibilities into families that have a specific scope. Therefore, we contribute with a structure for understanding practitioners' responsibilities and providing a

framework regarding their scope (e.g., support and infrastructure, governance, professional services etc.).

The identification of soft- and hard-skills as introduced in existing research [20] appears also in RE [8] and profiling AI engineers [13]. Our collection of soft-skills largely aligns with existing collections, even though there are additions. Most importantly, the derived soft-skills taxonomy complements existing research with the relation to specific roles and hard-skills, as well as the creation of themes (e.g., Articulation and transferability of knowledge), that in the future can be further investigated in isolation.

Human aspects on a spectrum. The human aspects angle in responsibilities and skills is often taking place as a binary classification, both in existing literature [8, 13, 28] and the starting point of our analysis. For example, often a separation of skills into hard-skills and soft-skills takes place. However, our results showcase that the appearance of human aspects is not always binary, but rather on a spectrum across different skills and responsibilities. On the one hand, we have human-centric skills and responsibilities (about human relationships, communication etc.). On the other hand we have (technical) artifact-centric skills and responsibilities (about technologies, processes, tools etc.). Between the two extremes, we have skills and responsibilities sitting anywhere across the spectrum.

For example, human aspects are manifested differently in "Cultivate culture, collaboration and coordination" than in "Navigate organizational structures", even though they are both human-centric responsibilities. In the former, human aspects are consciously and explicitly defined and formally embedded in practice by design. Whereas in the latter, human aspects are implicitly added since they need to be considered or used to achieve a direct objective, (e.g., escalating the communication of a non-functional requirement across the organization). A similar realization exists in existing research for hard-skills [20] and we showcase how it applies also for soft-skills. Therefore our findings indicate the appearance of human-centric responsibilities with different focus.

Furthermore, calculating the relationship of soft-skills to hard-skills, revealed that different hard-skills require to be accompanied by soft-skills at a different degree. Specifically, hard-skills such as domain, standards and regulations, agile, development practices, and microservices design seem to be related to the existence of soft-skills (hard-skills that typically occur alongside more soft-skills than others). In addition, those hard-skills are related to soft-skills through a different amount of average soft-skills appearing alongside them. Therefore, human aspects manifest themselves at a different degree across responsibilities and hard-skills. Consequently, we argue that there is a spectrum of responsibilities and skills, ranging from human-centric to (technical) artifact-centric.

The contributions of the data gathering and analysis approach. We introduce a systematic approach of gathering online job-ads daily. Our approach can be used in other research directions, that can be addressed with data existing in SE job-ads. Furthermore, it can be combined with existing approaches for consistent and systematic investigation of job-ads in SE [7, 20]. In addition, the global scale that our approach covers, evolves existing works in SE focusing on specific regions [8, 13], with ways to collect worldwide data.

5.1 Future work

The presented study lays a foundation for future research to further investigate posts of job-ads, but also to follow-up with studies on the basis of the findings presented. For example, future works can use the data gathering script to collect job-ads automatically. The data gathering script can run for longer periods of time and record the evolution of skills required in microservices, or other sub-areas of SE. Moreover, the existing data-set can be used to gather a larger sampling and extend the analysis. For extending the quantity of the data analyzed, researchers can try to automate the analysis of the contents using text-analysis methods (e.g., LDA or other methods [4]). In addition, job-ads from different recruitment platforms can be gathered and analyzed to enrich the analysis.

Furthermore, we call for researchers to complement the analysis of job-ads with investigating the perspectives of recruiters or hiring managers regarding responsibilities and skills of microservices practitioners. Another important direction, is to compare the responsibilities and skills identified with professional training and education programmes. This can help to understand how graduates of training and education programmes actually prepare for their future responsibilities and required skills. Finally, since there are studies about skills and responsibilities in different areas of SE, systematic mapping studies or a systematic literature review can be beneficial to aggregating the state-of-the art in SE skills.

6 CONCLUSION

Microservices-based architectures are increasingly gaining popularity across different organizations that develop software [3, 9, 21]. The underlying technologies of microservices are becoming diverse [26] and further specification of roles and skills is needed [15]. In this study, we analyze 125 job-ads from Glassdoor and chart the different responsibilities and skills that are necessary for microservices practitioners. We present a taxonomy of 21 responsibilities in 5 families of responsibilities, including *Software development support and infrastructure, software process and team development, software product delivery, professional services delivery as well as software engineering governance.* Moreover, a taxonomy is provided with the main (8) soft- and (11) hard-skills required from practitioners. Finally, we indicate how hard-skills relate to soft-skills.

Responsibilities and skills are stratified in the context of Web-based software engineers, DevOps engineers, and Software Architects roles. We find that there is a job market for microservices software architects, distinct from other known roles. In addition, we indicate the importance of human-centered responsibilities and skills, since almost all job-adds include human-aspects. We present the human-centric families of responsibilities, focusing on the team, governance, or customer. Moreover, we describe the 5 most demanded soft skills: (1) Articulation and transferability of knowledge, (2) Stakeholders management, (3) Problem solving, (4) Communication, presentation and negotiation, and (5) Leadership. Furthermore, we indicate how human aspects play a role in utilizing hard-skills in practice, since some hard-skills tend to occur alongside more soft-skills than others. Finally, we discuss and give directions for future research on unraveling organizational structure, personal development of practitioners and understanding the manifestation of human aspects in microservices.

REFERENCES

[1] Sebastian Baltes and Paul Ralph. 2022. Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering* 27, 4 (2022), 94.

[2] Saša Baškarada, Vivian Nguyen, and Andy Koronios. 2018. Architecting microservices: Practical opportunities and challenges. *Journal of Computer Information Systems* (2018).

[3] Christian Berger, Björnborg Nguyen, and Ola Benderius. 2017. Containerized Development and Microservices for Self-Driving Vehicles: Experiences & Best Practices. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. 7–12. https://doi.org/10.1109/ICSAW.2017.56

[4] Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. 2020. Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5832–5842. https://doi.org/10.18653/v1/2020.coling-main.513

[5] Justus Bogner, Jonas Fritzsch, Stefan Wagner, and Alfred Zimmermann. 2019. Microservices in Industry: Insights into Technologies, Characteristics, and Software Quality. In *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. 187–195. https://doi.org/10.1109/ICSA-C.2019.00041

[6] Antonio Bucchiarone, Nicola Dragoni, Schahram Dustdar, Patricia Lago, Manuel Mazzara, Victor Rivera, and Andrey Sadovykh. 2020. Microservices. *Science and Engineering. Springer* (2020).

[7] Maura Cerioli, Maurizio Leotta, and Filippo Ricca. 2020. What 5 Million Job Advertisements Tell Us about Testing: A Preliminary Empirical Investigation. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (Brno, Czech Republic) *(SAC '20)*. Association for Computing Machinery, New York, NY, USA, 1586–1594. https://doi.org/10.1145/3341105.3373961

[8] Maya Daneva, Chong Wang, and Patrick Hoener. 2017. What the Job Market Wants from Requirements Engineers? An Empirical Analysis of Online Job Ads from the Netherlands. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 448–453. https://doi.org/10.1109/ESEM.2017.60

[9] Jonas Fritzsch, Justus Bogner, Markus Haug, Ana Cristina Franco da Silva, Carolin Rubner, Matthias Saft, Horst Sauer, and Stefan Wagner. 2023. Adopting microservices and DevOps in the cyber-physical systems domain: A rapid review and case study. *Software: Practice and Experience* 53, 3 (2023), 790–810. https://doi.org/10.1002/spe.3169 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.3169

[10] Jonas Fritzsch, Justus Bogner, Stefan Wagner, and Alfred Zimmermann. 2019. Microservices Migration in Industry: Intentions, Strategies, and Challenges. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 481–490. https://doi.org/10.1109/ICSME.2019.00081

[11] Pooyan Jamshidi, Claus Pahl, Nabor C. Mendonça, James Lewis, and Stefan Tilkov. 2018. Microservices: The Journey So Far and Challenges Ahead. *IEEE Software* 35, 3 (2018), 24–35. https://doi.org/10.1109/MS.2018.2141039

[12] Holger Knoche and Wilhelm Hasselbring. 2018. Using Microservices for Legacy Software Modernization. *IEEE Software* 35, 3 (2018), 44–49. https://doi.org/10.1109/MS.2018.2141035

[13] Marcel Meesters, Petra Heck, and Alexander Serebrenik. 2022. What is an AI Engineer? An Empirical Analysis of Job Ads in The Netherlands. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI* (Pittsburgh, Pennsylvania) *(CAIN '22)*. Association for Computing Machinery, New York, NY, USA, 136–144. https://doi.org/10.1145/3522664.3528594

[14] Nabor C Mendonça, Craig Box, Costin Manolache, and Louis Ryan. 2021. The monolith strikes back: Why istio migrated from microservices to a monolithic architecture. *IEEE software* 38, 5 (2021), 17–22.

[15] Hamdy Michael Ayas, Regina Hebig, and Philipp Leitner. [n. d.]. An Empirical Investigation on the Competences and Roles of Practitioners in Microservices-Based Architectures. *Available at SSRN: https://ssrn.com/abstract=4450113 or http://dx.doi.org/10.2139/ssrn.4450113* ([n. d.]).

[16] Hamdy Michael Ayas, Regina Hebig, and Philipp Leitner. 2023. The Roles, Responsibilities, and Skills of Engineers in the Era of Microservices-Based Architectures. https://doi.org/10.5281/zenodo.10100987

[17] Hamdy Michael Ayas, Philipp Leitner, and Regina Hebig. 2021. Facing the giant: A grounded theory study of decision-making in microservices migrations. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–11.

[18] Hamdy Michael Ayas, Philipp Leitner, and Regina Hebig. 2023. An empirical study of the systemic and technical migration towards microservices. *Empirical Software Engineering* 28, 4 (2023), 85.

[19] Sam Newman. 2021. *Building microservices.* " O'Reilly Media, Inc.".

[20] Maria Papoutsoglou, Nikolaos Mittas, and Lefteris Angelis. 2017. Mining People Analytics from StackOverflow Job Advertisements. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 108–115. https://doi.org/10.1109/SEAA.2017.50

[21] Cesare Pautasso, Olaf Zimmermann, Mike Amundsen, James Lewis, and Nicolai Josuttis. 2017. Microservices in practice, part 1: Reality check and service design. *IEEE software* 34, 01 (2017), 91–98.

[22] Jacopo Soldani, Damian Andrew Tamburri, and Willem-Jan Van Den Heuvel. 2018. The pains and gains of microservices: A Systematic grey literature review. *Journal of Systems and Software* 146 (2018), 215–232. https://doi.org/10.1016/j.jss.2018.09.082

[23] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. 2016. Grounded theory in software engineering research: a critical review and guidelines. In *Proceedings of the 38th International conference on software engineering*. 120–131.

[24] Sami SURAKKA. 2005. Analysis of Technical Skills in Job Advertisements Targeted at Software Developers. *Informatics in Education* 4, 1 (2005), 102–122. https://doi.org/10.15388/infedu.2005.07

[25] Chellammal Surianarayanan, Gopinath Ganapathy, and Raj Pethuru. 2019. Essentials of microservices architecture: Paradigms, applications, and techniques. (2019).

[26] Muhammad Waseem, Peng Liang, Mojtaba Shahin, Amleto Di Salle, and Gastón Márquez. 2021. Design, monitoring, and testing of microservices systems: The practitioners' perspective. *Journal of Systems and Software* 182 (2021), 111061.

[27] Menghan Wu, Yang Zhang, Jiakun Liu, Shangwen Wang, Zhang Zhang, Xin Xia, and Xinjun Mao. 2022. On the way to microservices: Exploring problems and solutions from online q&a community. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 432–443.

[28] Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank. 2022. SkillSpan: Hard and Soft Skill Extraction from English Job Postings. arXiv:2204.12811 [cs.CL]