Required Knowledge, Skills and Transversal Competences for a Career in Software Engineering

Nikolaos Mittas[1], Dimitrios Trygoniaris[1], Apostolos Ampatzoglou[*2], Elvira-Maria Arvanitou[3], Christina Volioti[4], Alexander Chatzigeorgiou[2], and Lefteris Angelis[5]

[1] Hephaestus Laboratory, School of Chemistry, Faculty of Sciences, Democritus University of Thrace, Kavala, Greece

[2] Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece

[3] Department of Information and Electronic Engineering, International Hellenic University, Thessaloniki, Greece

[4] School of Philosophy and Education, Aristotle University of Thessaloniki, Thessaloniki, Greece

[5] School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

[*] Corresponding Author

Contact: apostolos.ampatzoglou@gmail.com

Context: Possessing up-to-date knowledge, skills and transversal competencies (KSTs) is essential for both the successful delivery of software projects and a career in software engineering (SE). However, the technological landscape is changing rapidly, posing continuous challenges: for professionals entering the market or pivoting careers, for organizations hiring and monitoring workforce expertise and for educational institutes designing or updating their curricula. **Objectives:** We study job requirements within and across SE occupations (Applications Programmers, Software Developers, Systems Analysts, Web and Multimedia Developers) to assist software organizations to better face skill mismatch and skills' gap problems, software engineers in upskilling and reskilling endeavors and software education institutes in providing more industrially relevant curricula. **Method:** In this study, we leverage a large corpus of online job advertisements, which are jointly collected by CEDEFOP and Eurostat. The dataset is analyzed through the lens of concepts and techniques from the study of biodiversity of species to assess the variation of expertise and identify skills that are transferable or unique in these occupations. Specifically, we adopt established diversity indices, such as alpha diversity, beta diversity, ordination methods, and indicator species analysis, aiming to quantify both the variety of skills within occupations and the differences across them. This approach highlights both the breadth and distinctiveness of expertise across occupations, rendering the biodiversity perspective a central and practical part of our methodology. **Results:** The results reveal that the complete list of KSTs that is used to characterize the profiles of OJAs for SE-related occupations is very broad and that skillset required for each occupation is quite distinct, since there are statistically significant differences in the composition of the skillsets. Transversal Skills and Competences (T) appear to be the most transferable qualification; or "adapt to change" and "work in teams" are the KSTs that appears more uniformly to all studied software occupations, and "computer programming" is the top hard-skill that appears more uniformly to all occupations. However, each occupation shows some specific qualifications. **Conclusion:** The results are contrasted against the literature, are interpreted, various implications to researchers and practitioners are provided, and a retrospective analysis of the tailoring of the biodiversity approach to SE labor landscape is provided. Overall, the proposed biodiversity analysis adds value by providing a novel, theory-driven methodology to assess skill variation, identifying both common and occupation-specific KSTs, and supporting evidence-based workforce and curriculum design.

Keywords: job requirements profiling, transferable skills/competences, diversity indices, ordination methods, indicator species analysis, CEDEFOP, Eurostat

1. Introduction

Seeking to work as a *software engineer* (SE) implies that you are selecting a career route that resembles trying to hit a moving target by continuously changing your arsenal. Technological shifts in software development are so frequent, and in some cases so radical, that regular upskilling and reskilling of software professionals is required, to survive in a constantly changing environment [65, 69]. These changes can stem from various sources, such as new hardware that hosts the software (e.g., embedded, IoT, etc.), new platforms for deploying the software (e.g., web, mobile, cloud, etc.), new languages and frameworks (e.g., Python, Rust, ReactJS, Angular, etc.), new businesses for the software (e.g., healthcare, data analytics, games / multimedia, social, etc.). However, older technologies are very "*hard to die*", since legacy applications are in continuous maintenance and in use for various decades (e.g., various banking systems are still maintained in COBOL, and successful systems are maintained in system-specific languages, like SAP ABAP). Consequently, software development organizations face more abruptly the "*skills' gap*" and "*skills' mismatch*" phenomena that are rapidly growing in any domain in EU [72, 85]. HR departments of software development organizations need to be sure that they are specifying the correct job requirements for a given occupation (e.g., web developer) and assess if their workforce is "*covering*" all the job requirements for an up-to-date software house [75]. The (newcomer) software engineers face symptoms of frustration when inspecting diverse job advertisements, whose qualifications cannot be met and are confused when attempting reskilling or upskilling [60] in the context of a career pivot. Finally, software education providers are seeking to develop industrially relevant curricula that provide all the required knowledge, skills and competences to "*shape*" the future software professionals [31].

Based on the above considerations (Problem Statement Conceptualization), in this study, we analyze the diversity of *Knowledge, Skills and Transversal Competences* as structured in the *European Skills / Competences, Qualifications and Occupations* (ESCO) taxonomy for the purpose of comparing and profiling job requirements within and across four *Software and Applications Developers and Analysts occupations* as classified by the *International Standard Classification of Occupations* (ISCO-08) taxonomy in the context of the software engineering workforce, from three different points of view: (a) software development organizations, (b) newcomer software engineers, and (c) software education providers. To address the dual objective, we formulated the following *Research Questions* (RQs):

[**RQ$_1$**] *How does the variety and distribution of Knowledge, Skills and Transversal Competences qualifications differ within ($RQ_{1.1}$) and among ($RQ_{1.2}$) Software and Applications Developers and Analysts occupations*?

[**RQ$_2$**] *Which Knowledge, Skills and Transversal Competences signify ($RQ_{2.1}$) are transferable across multiple Software and Applications Developers and Analysts occupations, and which Knowledge, Skills and Transversal Competences imply ($RQ_{2.2}$) prototypical specialization within specific occupations*?

As the means to respond to this need (Solution Conceptualization, see Figure 1), we borrowed concepts from ecology on the biodiversity of species to match the context of software engineering workforce. *Biodiversity* is fundamental to ecology, as it represents the variety and heterogeneity of life forms within an ecosystem encompassing a plethora of aspects including the number of unique species, their relative abundance, and the interactions and ecological processes that sustain them over different spatial and temporal scales [66]. The key concepts in biodiversity research are: (a) the ecosystem; (b) the habitats; (c) the community; (d) the species; and (e) the environment [21]. In ecological studies, the term ecosystem refers to the interaction between a *community* of living organisms and their environment. The ecosystem is of primary focus regarding biodiversity issues, and it is analogous in our study to the *software industry* (see Figure 1). Within an ecosystem, specific places where interconnected species (community) live are called habitats. In our context, habitats represent the software professional occupations that are in demand within the software industry. These occupations (habitats) host skillsets (communities) of interconnected job requirements described as skills and competencies (species) in online job advertisements. Although the investigation of the term environment is beyond the research scopes of this study, factors such as temporal, geographical, employer, salary, contract type and other related information can be considered as external conditions that shape the career landscape of SE professionals.

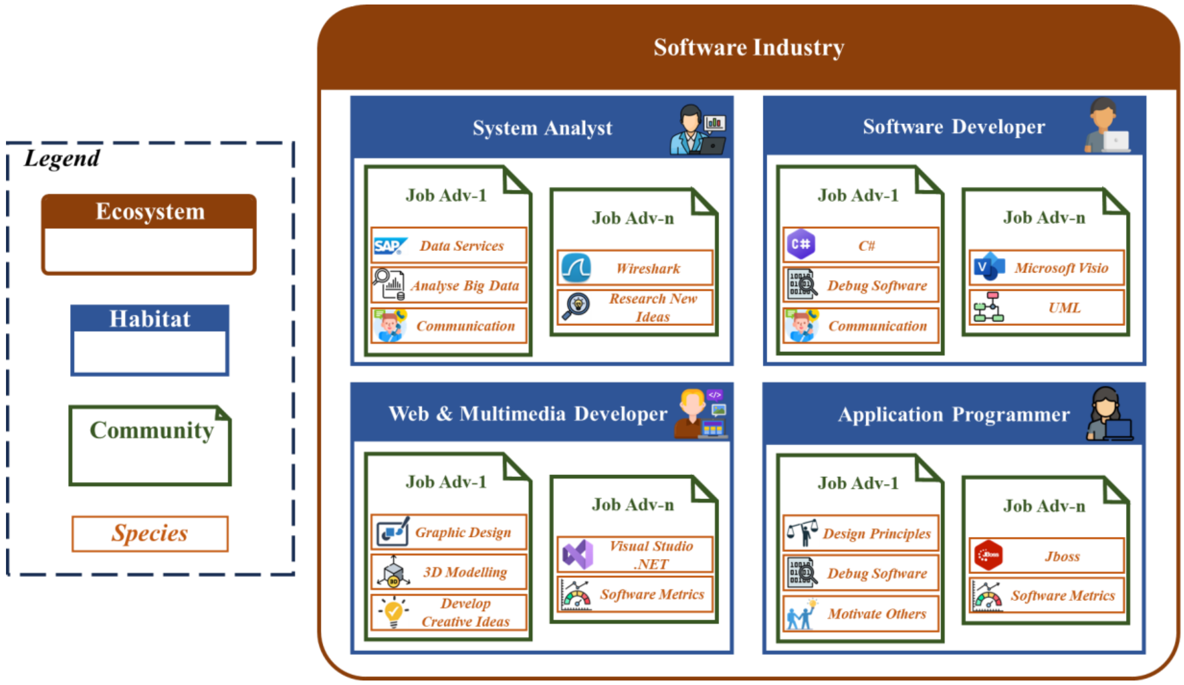

Figure 1. Solution Conceptualization

To answer RQs (Methodological Conceptualization), we conducted an empirical study using a large corpus of *Online Job Advertisements* (OJAs) from the first three available quarters (or the first nine months) of 2024 provided by CEDEFOP[1] and Eurostat[2] that maps job requirements to the four specific software-related occupations. Regarding the analysis, we synthesized a "*suite*" of well-defined data-driven approaches that have been introduced and widely used in ecology studies. Specifically, to quantify the diversity of job requirements within software-related occupations ($RQ_{1.1}$), we made use of appropriate *alpha* diversity indices [58, 66, 69], while the examination of diversity among software-related occupations ($RQ_{1.2}$) were performed via the *Bray-Curtis* (BC) dissimilarity index [17] complemented by ordination methods [39] such as (a) *principal coordinates analysis* (PCoA) [41] and (b) *permutational multivariate analysis of variance* (PERMANOVA) [4] on the composition data computed on OJA collections. To identify transferable ($RQ_{2.1}$) and prototypical job requirements ($RQ_{2.2}$), we employed *Indicator Species Analysis* (ISA) [34] that offers a robust statistical approach to reveal differences and commonalities in the job requirements among the four software-related occupations. The commonalities capture the fundamental job requirements that are needed from software professionals to be hired, regardless of the intended occupations. On the other hand, differences spotlight unique requirements demanded by specific occupations.

Our study according to Gregor [89] can be classified as a mix of theory for analyzing and explaining phenomena in software engineering, using as primary constructs biodiversity indices from ecology. The rest of the paper is organized as follows: in Section 2, we present the necessary background information from ecology that enables a deeper understanding of the concepts of the study, as well as related work. In Section 3, we present the experimental setup of our data-driven approach and the posed research questions, and in Section 4, the results of our empirical study. In Section 5, we discuss the results by providing interpretations and comparisons to the litera-

[1] https://www.cedefop.europa.eu/en
[2] https://ec.europa.eu/eurostat

ture, as well implications for researchers and practitioners and threats to validity. Finally, Section 6 concludes the paper by presenting a recap of the study.

2. Related Work and Background Information

2.1 Background: Key Taxonomies and Frameworks for Skills

In this section, we present the key taxonomies, frameworks and a glossary of terms that provide the necessary background for our study. In particular, it is essential to understand the established frameworks and taxonomies used to classify occupations and skills. In the contemporary European landscape, several key frameworks provide a standardized language for this purpose.

The *European Skills, Competences, Qualifications and Occupations* (ESCO)[3] framework provides a standardized taxonomy and individual pillars for describing occupations, skills, and competences across diverse sectors and industries. The Occupations Pillar is anchored upon the *International Standard Classification of Occupations* (ISCO-08)[4], serving as the hierarchical framework for organizing occupational data. Each occupation within ESCO is meticulously mapped to a specific ISCO-08 code, facilitating precise categorization and comparison of roles within the European labor market. Complementing the Occupations Pillar, the Skills Pillar of ESCO offers a comprehensive taxonomy of skills and competences, distinguishing between different skill types and encompassing a wide array of knowledge domains. This pillar not only identifies essential hard skills, such as language proficiency and technical expertise, but also delineates soft skills, including communication, teamwork, and self-management abilities. With over 13,000 concepts organized within a hierarchical structure, the Skills Pillar provides a detailed framework for understanding the competences demanded by various occupations and sectors. Crucially, ESCO adopts a position on knowledge and skills that recognizes knowledge as a primary and essential element, an accumulation of facts, principles, theories and practices, which are necessary in any branch of work or study.

Complementing ESCO, the *European e-Competence Framework* (e-CF)[5] is a standard specifically designed for the *Information and Communication Technology* (ICT) sector. It provides a shared reference of 41 competences as required and applied at the ICT workplace. The e-CF is structured across five competence areas and relates them to the eight proficiency levels of the *European Qualifications Framework* (EQF)[6]. While ESCO provides a broad vocabulary of all skills, the e-CF focuses specifically on the professional competences required in the ICT domain, making it a valuable tool or HR departments, education provides, and ICT professionals for defining job profiles and planning career paths.

Finally, to ensure a common understanding of the terminology used throughout this paper, we refer to the official glossary developed by the European Centre for the Development of Vocational Training (CEDEFOP)[7]. In Table 1, we present the terms that can be considered as an essential background. Each term is presented along with a short description as provided in the official glossary of CEDEFOP in alphabetical order.

Table 1. Terminology of Knowledge, Skills and Competences management[7]

Term	***Description***
Competence	Demonstrated ability to use knowledge, know-how, experience and – job-related, personal, social or methodological – skills, in work or learning situations and in professional and personal development. Competence can be

[3] https://esco.ec.europa.eu/en/about-esco/what-esco
[4] https://ilostat.ilo.org/methods/concepts-and-definitions/classification-occupation/
[5] https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/european-e-competence-framework-e-cf
[6] https://europass.europa.eu/en/europass-digital-tools/european-qualifications-framework
[7] https://www.cedefop.europa.eu/en/tools/vet-glossary/glossary

Term	*Description*
	further categorized as: cognitive competence, emotional competence, entrepreneurship competence, life competence, mathematical competence, multilingual competence, thinking competence, transferable competence, transversal competence
Education and Training Needs Analysis	Process of identifying skills' gap in the workforce and – current and future – skill needs of the economy, to implement an education and training strategy that meets the needs of society (competitiveness of businesses, personal and professional development of individuals)
European Skills / Competences, Qualifications and Occupations (ESCO)	EU multilingual taxonomy, identifying and categorizing skills / competences, qualifications and occupations useful to the EU labor market and education and training, and which provides occupational profiles showing the relationships between occupations, skills / competences and qualifications
International Standard Classification of Occupations (ISCO-08)	Tool for organizing occupations into a clearly defined set of groups, according to the tasks and duties undertaken in the job. The current version (ISCO-08) consists of ten major occupational groupings sub-divided into further occupational groups
Knowledge	Outcome of assimilation of information through learning. Knowledge is the body of facts, principles, theories and practices related to a field of study or work
Labor Market / Job Market	Real or virtual meeting point, within an economy or area, where people selling their labor (workers) negotiate and may reach an agreement with those who buy it (employers)
Occupation	Set of jobs whose main tasks and duties are characterized by a high degree of similarity
Online Job Advertisement	Publication on the Web of a vacant position to inform and to attract potential candidates
Reskilling	Training enabling individuals to acquire new skills, giving access either to a new occupation or to new professional activities
Skill	Ability to apply knowledge and use know-how to complete tasks and solve problems. Skills can be categorized as: (a) cognitive, (b) emotional, (c) life, (d) mathematical, (e) multilingual, (f) thinking, (g) transferable, and (h) transversal skills
Skills' Gap	Situation where an individual does not have the type or level of skills required to perform adequately the tasks associated with a job
Skill Needs	Demand for types of knowledge and aptitudes on the labor market (total demand within a country or region, economic sector, etc.)
Skill Supply	Volume and type of skills or qualifications available on the labor market, and number of people who have these skills and qualifications
Transversal Skills / Competencies	Proven ability to use knowledge, skills and personal, social or methodological abilities, in work / study and in professional and personal development.
Upskilling	Short-term targeted training typically provided following initial education or training, and aimed at supplementing, improving or updating knowledge, skills and competences

2.2 Related Work in Software Engineering Management of Skills

To structure our analysis of the existing literature, we adopt the concept-centric approach advocated by Webster and Watson [83]. The labor market terminology used in this section has been specified in Table 1, and the interested reader can correspond there for clarifications. This methodology emphasizes prior work around key concepts rather than presenting a chronological list of authors. To identify the studies included in the Related Work section, we performed a structured search in Scopus, focusing on seven well-established SE venues[8]. We applied our search string (*skill OR labour OR labor OR competence OR competencies*) on the title of candidate studies, published in the aforementioned venues. This process initially yielded 50 candidate papers, including paper published up to 2025 to include the latest research on the topic, and thus the newest articles. All identified articles were independently handled by the third and fourth authors. In cases of disagreement, the fifth author was consulted until consensus was reached. In total, only 2 articles have been discussed to reach consensus. After applying this selection process, we retained 34 papers as presented in Table 2. To organize the selected studies into conceptual groups, we followed the approach suggested by Webster and Watson [83], complemented with an open card sorting method [77]. Through an extensive review of 35 papers, we identified three themes (T) that structure the research field on SE skills management: (T1) *Landscape of Software Engineering Skills and Competencies*—which covers any research reporting on a high-level analyses of the skills market in the domain of SE, explaining which KST are needed; (T2) *Profiling and Differentiating Specific SE Roles and Practices*—which focuses on defining and comparing the skillsets that specialized for specific roles and practices in the SE sector; and (T3) *Methods for Eliciting, Measuring and Modeling SE Skills*—which covers the methodological contributions in this domain. We note that in our study we found that the literature clusters neatly into these distinct concepts, with each reviewed paper making its primary contribution to one of these core concepts. Table 2 presents the concept matrix, mapping each of the reviewed papers to these central concepts in an alphabetical order. The subsequent sections present the literature synthesized within each concept. In principle, studies that are related to T1 are conceptually closer to RQ_1, whereas studies that are related to T2 are closer to RQ_2.

Table 2. Concept Matrix of Related Work

Reference	Published Year	T1: Landscape of SE Skills and Competencies	T2: Profiling and Differentiating Specific SE Roles and Practices	T3: Methods for Eliciting, Measuring and Modeling SE Skills
Ajimati et al. [2]	2022			√
Assyne et al. [7]	2022	√		
Assyne et al. [8]	2022	√		
Ayas et al. [9]	2024		√	
Berenbach [13]	2008		√	
Bergersen et al. [14]	2014			√
Borges and Gratão de Souza [16]	2024	√		
Carrington et al. [20]	2005	√		
Creighton and Singer [22]	2008	√		
De Morais Leca and De Souza Santos [29]	2025		√	
Dorofeev [32]	2020			√
Downey and Ali Babar [33]	2008		√	
Duarte [34]	2017			√
Fucci et al. [36]	2015		√	
Gafni et al. [37]	2024			√

[8] Information and Software Technology (IST), Journal of Systems and Software (JSS), Transactions on Software Engineering (TSE), Transactions on Software Engineering and Methodology (TOSEM), Empirical Software Engineering (EMSE), International Conference on Software Engineering (ICSE), and International Symposium on Empirical Software Engineering and Measurement (ESEM).

Reference	Published Year	T1: Landscape of SE Skills and Competencies	T2: Profiling and Differentiating Specific SE Roles and Practices	T3: Methods for Eliciting, Measuring and Modeling SE Skills
Galster et al. [38]	2023	√		
Gren et al. [42]	2018	√		
Heldal et al. [44]	2024	√		
Heggen and Cody [45]	2018	√		
Holtkamp et al. [46]	2015		√	
Jørgensen et al. [47]	2021			√
Kapitsaki et al. [48]	2024	√		
Liang et al. [53]	2022			√
Loufek et al. [54]	2025	√		
Misic and Graf [63]	2004		√	
Montandon et al. [64]	2021	√		
Orsted [68]	2000		√	
Rose et al. [73]	2007		√	
Santos [74]	2023			√
Turley and Bieman [79]	1996		√	
Wang et al. [82]	2018	√		
Zanatta et al. [86]	2018		√	
Zieris and Prechelt [87]	2014		√	
Zieris and Prechelt [88]	2021		√	

2.2.1 *Theme 1: The Landscape of Software Engineering Skills and Competencies*

A substantial body of research has been dedicated to investigating the landscape of skills and competencies required for modern software engineers, motivated by the persistent gap between industrial demands and workforce capabilities. Borges and Gratão de Souza [16] conducted a systematic literature review to explore the soft skills needed by software engineers, as well as the teaching methodologies that can contribute to developing the identified soft skills. The results of their study revealed 23 soft skills. Additionally, the authors provided a definition of the soft skills considered most relevant for software engineers, including ten soft skills desired for software professionals, which were not yet clearly defined in the literature. Borges and Gratão de Souza [16] confirmed that skills development is a central concern for both industry and academia. In addition to this, Assyne et al. [8] performed a systematic mapping study on software engineering competencies (SEC). In particular, the goal of this study focused on identifying: (a) the research areas; (b) the SEC frameworks; (c) the essential competencies of software professionals; and (d) the changes in SEC research over the last three decades. The authors identified two main reach areas, namely personnel competence (i.e., focuses on software professional competencies) and organizational competence (i.e., focuses on tools or instruments). Regarding SEC frameworks, the authors identified 14 different SEC models and explored 49 essential competencies of software professionals.

This research is often driven by the "skills’ gap", which manifests in two key areas: upskilling the existing workforce and preparing newcomers. To address the former, studies have explored industry-university collaborations aimed directly at upgrading the knowledge and skills of software professionals already in the industry [20]. For the latter, the challenge is particularly important for students transitioning into professional roles [45]. Heggen and Cody [45] described the Student Software Developers Program that offers a year-long, immersive experience where students develop internal software solutions for real stakeholders. The program not only strengthens technical proficiency but also enhances critical soft skills—such as communication, teamwork, and client interaction—through authentic, team-based work environments.

To understand these industrial demands, researchers frequently turn to large-scale data sources like Online Job Advertisements (OJAs). A notable example is the work of Montandon et al. [64], who performed an empirical study to identify the hard and the soft skills that are required in IT companies. In particular, the authors analyzed more than 20,000 job posts from the Stack Overflow Jobs portal. Montandon et al. [64] categorized posts into 14 IT roles and identifying 1,916 distinct hard skills, their work provided a detailed snapshot of the most in-demand roles (e.g., full-stack developer) and technical skills (e.g., programming languages), while also confirming that communication and collaboration are top-demanded soft skills.

Moreover, numerous studies have highlighted the rising relevance of social competence, especially for leadership roles in development teams [22]. Galster et al. [38] identified specific soft skills required by software professionals in specific markets like New Zealand. In particular, Galster et al. [38] analyzed 530 New Zealand job advertisements, identifying that 82% of postings specified at least one soft skill, with communication, teamwork, and problem-solving being the most frequently required. Building on this line of research, Kapitsaki et al. [48] conducted a quasi-replication of Galster et al. work [38] in the labor market of Cyprus. Kapitsaki et al. [48] analyzed 689 software job advertisements published in 2023 and 2024 and identified 36 distinct soft skills. While communication and teamwork remained the most frequently required, the study also highlighted regional differences in how employees frame collaboration and adaptability. Adding a managerial perspective, Loufek et al. [54] conducted an exploratory study at Hewlett Packard Enterprise, interviewing 12 hiring managers and analyzing 7 entry-level SE job postings. Their findings revealed that employers prioritize non-technical skills such as adaptability, communication, and problem-solving, over technical competencies which are often considered teachable on the job.

Gren et al. [42] performed a survey with 113 participants, in order to investigate the relationship between individual-level non-technical skills and the maturity of agile practices in software development teams, finding that the relationship is weaker than often assumed. Beyond identifying lists of skills, some studies aim to create structure or explore contextual factors. Assyne et al. [7] propose a unified competence framework (UComGSP) in order to define and categorize the essential hard and soft competencies required of software professionals. Through a rigorous mixed-method, Assyne et al. [7] identified 125 competencies—62 hard and 63 soft—of which 25 were deemed essential. The skills landscape is also shown to be continuously expanding, with studies identifying emerging competency requirements in new domains such as software sustainability [44]. Finally, Wang et al. [82] performed an empirical study in order to explore socio-technical factors, such as the competence-confidence gap, that can act as significant barriers affecting the contributions and career progression of software professionals.

2.2.2 *Theme 2: Profiling and Differentiating Specific SE Roles and Practices*

In this section, we present studies that explore the distinct skill sets and competencies associated with specific software engineering roles and development practices. A common approach is to create detailed profiles for established or emerging professional roles. For example, multiple studies have focused on identifying the necessary technical and non-technical skills for software architects [33], emphasizing not only design and systems thinking, but also cross-functional collaboration, communication, and leadership in large-scale and distributed development environments. In addition, attention has been drawn to the “other skills”, often less visible, skills that contribute to their effectiveness—such as political awareness, negotiation, and the ability to mediate between conflicting stakeholder interests—which are frequently overlooked in traditional training but critical to navigating the organizational and social dimensions of architectural work [13]. In the same way, Misic and Graf [63] investigate how the systems analyst role has evolved in modern IT programs, emphasizing that success now hinges as much on behavioral competencies—such as interpersonal communication, collaboration, and stakeholder engagements on technical knowledge. This type of profiling extends to modern, specialized roles such as data scientists, with studies identifying their essential soft skills (e.g., critical thinking [29]).

Beyond official job roles, researchers differentiate software engineers based on other important dimensions. Turley and Bieman [79] conducted a two-phase study to compare the competencies of exceptional versus non-exceptional software engineers. Turley and Bieman [79] identified 38 core competencies across task accomplishment, personal attributes, situational skills, and interpersonal skills. Holtkamp et al. [46] performed a survey to investigate how skill requirements change across different phases of the development lifecycle, with studies identifying specific soft competency requirements for requirements engineering, software design, implementation, and testing. Additionally, Fucci et al. [36] conducted a quasi-experimental study with 30 industry practitioners to evaluate the impact of developers' test-driven development-related skill sets on code quality and productivity, whereas Zieris and Prechelt [88] conducted qualitative analyses of industrial pair programming sessions to identify the core elements that distinguish effective pairs, with earlier work [87] highlighting the role of structured knowledge transfer and incremental explanations in facilitating successful collaboration.

Orsted [68] provided subjective views on the soft skills required for a Software Development Engineer within a major company like Microsoft. After the interviews, Orsted [68] claimed that technical skills alone are insufficient and that roles like Software Development Engineer require strong soft competencies. Rose et al. [73] conducted a grounded theory study with project managers at WM-data, to challenge the dominant tool-centered perspective in software project management. Through interviews, focus groups, and causal mapping, they demonstrated that project success depends far more on managerial competences, such as leadership, communication, stakeholder negotiation, and adaptability, than on the use of specific tools or methodologies. Ayas et al. [9] analyzed 13,517 software developer profiles from Stack Overflow and other public sources to empirically determine key technical competencies and role specializations in Microservices-based architectures. Ayas et al. [9] identified 3 collections (i.e., Web Technologies, DevOps, and Data Technologies) and 11 clusters of competences of microservice practitioners. Their study also identified the predominant practitioner roles within MSA teams, such as API, service integration, full-stack, monitoring, and CI/CD engineers. Finally, Zanatta et al. [86] conducted a grounded-theory study to investigate why crowd workers on software development platforms struggle to contribute consistently. They found that competence gaps, poor collaboration, and time-management challenges were major barriers leading to high dropout rates.

2.2.3 *Theme 3: Methods for Eliciting, Measuring and Modeling SE Skills*

In addition to profiling specific roles, another important line of research focuses on the development and application of innovative methods for identifying, assessing, and modeling software engineering skills. A key challenge in this area is the initial elicitation of skill data from complex software artifacts and software developer activities. To address this, Liang et al. [53] developed a tool namely Disko for mining developer repositories like GitHub to automatically extract evidence of skills from contribution histories. Bergersen et al. [14] developed a psychometric instrument to assess programming skills reliably. These instruments are then used in further studies to investigate the relationships between measured skill, software developer effort estimates, and other performance indicators [47]. Santos [74] proposed a skill recommendation system to support newcomers in identifying relevant skills needed to contribute effectively to open-source projects. Some studies have also explored the use of artificial intelligence, proposing AI-driven approaches for the objective evaluation of employees' soft skills, aiming to reduce bias in assessment [37]. Moreover, Dorofeev [32] proposed a model-driven approach to skill representation in the industrial automation domain. The approach defines how skills can be formalized, reused, and orchestrated within engineering workflows, bridging gaps between human and automated competencies. Finally, software engineering researchers tend to adapt powerful analytical methods from other scientific disciplines. Ajimati et al. [2] investigated how developers' positions within advice networks relate to their problem-solving competence, finding that greater connectedness is associated with higher competence. Their work highlights social network analysis as a valuable method for assessing and understanding software engineering skills. Duarte [34] examined the relationship between software quality maturity levels and labor productivity in Brazilian software companies. The study serves as an example of using rigorous quantitative analysis to test complex, long-held assumptions within the software industry.

2.3 Biodiversity Indices, Ordination Methods and Indicator Species Analysis

This section is dedicated to the presentation of background information on concepts and terminology used in ecology and biodiversity. From a historical perspective, the term "*ecology*" was, first, coined by Ernst Haeckel in 1870 with its roots in the Greek words "*oikos*" and "*logos*" meaning "*house*" and "*knowledge*", respectively, and it was used for "*characterizing those sciences concerned with the relations of animals to the outside world*" [19]. Since then, over the past centuries and decades, the term has been adapted and applied to investigate a wide range of research problems from diverse perspectives and goals. This study aligns more closely with the definition given by Andrewartha and Birch back to the 60s that prioritizes the focus on the explanation of "*the distribution and abundances of species by studying the environments of individuals in natural populations*" [6].

In this context, a *biodiversity index* can be, simply, defined as a quantitative measure used to evaluate the general properties and characteristics of communities [66]. Despite this straightforward definition, biodiversity is a complex multifaceted concept, and thus, there has been an extensive debate and, more importantly, no consensus on the most appropriate index for capturing all aspects of diversity [66]. Ideally, a biodiversity index should account for two key factors: (a) the number of unique species present in the ecosystem, known as species *richness*, and (b) the relative abundance of these detected species, referred to as *evenness* [49]. Moreover, Whittaker [84] emphasized the imperative need to take into consideration that species diversity can be decomposed and assessed at various scales guided by the specific requirements of the research objectives. In this regard, he pointed out that *alpha diversity* should be considered, when the objective is the evaluation of the local diversity within a particular habitat, community or ecosystem, whereas *beta diversity* focuses on "*the degree of community differentiation*" between habitats, communities or ecosystems.

Concerning the alpha diversity, it is usually expressed by species *richness* (S) (Table 3) representing the total number of unique species within all levels of the hierarchy of life (habitat, community, ecosystem) [58, 66]. However, an ecosystem with high species richness alone cannot be considered as highly diverse. For example, if two ecosystems are similar in terms of their richness but one is dominated by a single species, the other ecosystem would be considered more diverse, since it has a more even distribution of species populations. Thus, species *evenness* is another important aspect that should be incorporated into the evaluation of alpha diversity resulting to the development of a wide range of compound indices. Among them, *Shannon's diversity* (H') and *Simpson's evenness* (E) indices are two of the most widely applicable in ecology studies capturing both richness and abundance in a unified measure (Table 3). In practice, these compound indices have been designed to serve the same objective, but they are different in terms of their theoretical foundation and interpretation [58].

On one hand, the Shannon's diversity index has its origins in information theory expressing the uncertainty in the prediction of species [58]. Inspecting the rationale and definition of the Shannon's index, it should be noted that it prioritizes richness as a more important component in the assessment of alpha diversity with higher values representing greater evenness in distribution of individuals among species and thus, higher amount of alpha diversity. Although the Shannon's index ranges from a theoretical minimum value of zero (no diversity, i.e. only one species), its upper limit depends on the total number of unique species (S). To offer a more straightforward interpretation, Shannon's diversity indices can be standardized to $[0,1]$ range. To achieve this standardization, one must divide the computed Shannon's diversity index by the natural logarithm of richness ($H'_{Std} = H'/\ln(S)$) that represents the theoretical maximum diversity, resulting, in fact, into the evaluation of *Pielou's Evenness* index [76]. On the other hand, the Simpson's evenness index (Table 3) is a probabilistic indicator based on the Simpson's diversity index that expresses the probability that two randomly selected individuals from a given level of the hierarchy of life will belong to the same species putting emphasis on evenness rather than richness. The Simpson's evenness index yields standardized values within [0,1], with higher values indicating a more evenly distribution among species, whereas lower values represent communities dominated by a few species.

Shifting the interest to the level of differentiation between a predefined level of the hierarchy of life in terms of species composition, a plethora of beta diversity indices have also been proposed in the ecology literature, whereas the selection of the most appropriate one is not a trivial task. In this study, we opted to make use of a well-known beta diversity index introduced by Bray and Curtis [17] that evaluates the dissimilarity in species relative abundance data between two examined habitats, communities or ecosystems exploiting the information of both the presence or absence of species and their abundances. Higher values of Bray-Curtis (BC) index represent greater species turnover (beta diversity) and less overlap in species composition [50].

Table 3. Alpha- and beta-diversity indices

Biodiversity Index	Formula	Description
Alpha diversity		
Richness	$S = \lvert\Omega\rvert$	Total number of distinct species within an ecosystem [a]
Standardized Shannon's diversity	$H'_{Std} = -\sum_{i}^{S} p_i \ln(p_i) \Big/ \ln S$	Quantifies both richness and evenness. More emphasis on richness, less sensitive to evenness [b]
Simpson's evenness	$E = \left(1 - \sum_{i}^{S} p_i^2\right) \Big/ S$	Quantifies both richness and evenness. More emphasis on evenness (abundance), less sensitive to richness [b]
Beta diversity		
Bray–Curtis dissimilarity	$d_{ab} = \frac{\sum_i^S \lvert p_{ai} - p_{bi}\rvert}{\sum_i^S (p_{ai} + p_{bi})}$	Quantifies the diversity of species relative abundances between habitats[c]
Notes:		
[a] $\lvert\Omega\rvert$ is the cardinality of the set $\Omega = \{s_1, \dots, s_n\}$, where s_i represents a distinct species in the habitat		
[b] p_i is the relative abundance of species i and S is the total number of species present in the habitat		
[c] p_{ai} and p_{bi} are the relative abundances of species i in habitats a and b, respectively, and S is the total number of species present in habitats a and b		

Even though all these indices provide valuable information regarding the biodiversity across habitats, communities or ecosystems, they face significant challenges, as they oversimplify the complexity of examined phenomenon to a single statistic derived from samples. To better understand complicated ecological relationships and gain deeper insights into the patterns and composition of species in each level of the hierarchy of life, there is a need to use appropriate multivariate statistical approaches that fall under the general umbrella of *ordination analysis* [39]. Ordination analysis was, firstly, introduced in ecological literature by Curtis and McIntosh [23] and it can be considered as a "*suite*" of statistical tools addressing various scopes and objectives. In this study, we make use of (a) *principal coordinates analysis* (PCoA) [41] and (b) *permutational multivariate analysis of variance* (PERMANOVA) [4] that serve quite different but complementary objectives.

PCoA belongs to the general branch of dimensionality reduction techniques aiming to project the ordination (scaling) of the objects from a full-dimensional space into a low-dimensional space, usually encompassing two (2D) or three (3D) dimensions, facilitating, in turn, the visual exploration of patterns and relationships among them. PCoA is considered an unconstrained ordination method in the field of ecology [62], where the term "un-constrainted" refers to the fact that the ordination is constructed by exploiting information reflecting the internal relationships within objects without taking into consideration the effect of external explanatory variables (i.e., constraints). Regarding the algorithmic details of the approach [52], PCoA initiates by computing the distance matrix of objects in the full-dimensional space via an appropriate dissimilarity measure (Bray-Curtis in our case). Next, the distance matrix is transformed into a centered matrix of scalar products. Finally, through an eigenvalue decomposition process, the algorithm resulted in a set of orthogonal axes (or dimensions), whose contribution is computed by their corresponding eigenvalues [18].

The exploratory identification of hidden patterns in data can be further augmented with statistical hypothesis testing procedures to enhance the inferential process when investigating factors influencing the composition and relative abundances of species in ecosystems [4]. Traditional univariate and multifactor statistical approaches, such as *Analysis of Variance* (ANOVA), are well-established methods grounded in strong theoretical foundations from the applied scientific domain of design of experiments. However, these parametric approaches rely on strict assumptions (e.g., normally distributed data, homogeneity of variance etc.) that are rarely met in most ecology experimental studies [4]. Due to this limitation, Anderson [5] proposed PERMANOVA, as a non-parametric analogue to the traditional ANOVA, that partitions variation within a dissimilarity matrix (Bray-Curtis in our case). This statistical method provides a robust approach for the examination of various types of effects (e.g., main / interaction effects, random effects in mixed effects and hierarchical designs etc.) and the estimation of statistical significance exploiting distribution-free permutation approaches [5].

Based on the previous considerations, PERMANOVA can be considered a valuable tool for inferential purposes, when the objective is to examine differences in the composition of habitats, communities or ecosystems. Beyond this, it is of great practical importance to identify species that are strongly associated with specific habitats, communities or ecosystems, as these species often serve as indicators of ecological health or change [70]. *Indicator Species Analysis* (ISA) can, further, extend the body of knowledge regarding the dynamics within complex ecological systems, as it enables the identification of species whose presence or abundance is closely related to environmental conditions, communities or habitat types [35]. The approach is based on the evaluation of a compound index [35], called the *Indicator Value* (IV), which quantifies the strength of the association between species and habitats under examination. Specifically, IV is calculated as the product of two components, *specificity* and *sensitivity* (or *fidelity*), that combines both the occurrence (or abundance) of a species and its frequency of occurrence in the set of the examined habitat types. Each species is, then, assigned to the habitat with the highest IV ($IV_{\max}$), whereas a randomization permutation approach is applied to test the statistical significance of this association ($IV_{\max}$).

In brief, the first component, specificity (A_{ij}), is defined as the ratio of the number of occurrences of the species i within sites (land units in ecology) belonging to the target group j divided by the number of occurrences of the species across all sites. The definition of specificity was, further, modified by De Caceres et al. [26] to give equal weight to all sites within a group ignoring the total number of sites each group contains. Thus, they proposed the following formula (Eq. 1) for the computation of specificity (A_{ij}^{g}):

$$A_{ij}^{g} = \frac{n_i/N_j}{\sum_{j=1}^{J} n_i/N_j} \tag{1}$$

where n_i/N_j is the relative frequency of the species i in the target group j, and $\sum_{j=1}^{J} n_i/N_j$ is the sum of relative frequencies of the species i over all J groups. The second component, sensitivity B_{ij}, is equal to the nominator of A_{ij}^{g} and is defined as:

$$B_{ij} = \frac{n_i}{N_j} \tag{2}$$

Summarizing, the IV_{ij} for species i in group j is given by the product of the specificity A_{ij}^{g} and sensitivity B_{ij} multiplied by 100.

$$IV_{ij} = A_{ij}^{g} \times B_{ij} \times 100 \tag{3}$$

De Cáceres et al. [25] have been further extended the original approach proposed by Dufrêne and Legendre [35] to account for the fact that a subset of species may be associated to more than a single group. In ecology, this is often a realistic scenario, as species may exhibit different niche breadths, thus, there is an imperative need to apply ISA mechanisms that are able to identify species that are strongly related, simultaneously, with multiple groups. This is a critical task to avoid overlooking potentially meaningful insights that are exhibited between species and combinations of groups that may not be reflected by single groups analysis. Finally, we note that instead of reporting the raw IVs (Eq. 3), the square root transformation is applied to each IV [24] to mitigate threats related to giving more importance to relative abundance over relative occurrence, as well as limitations, such as the presence of heavily skewed IV distributions [25].

2.4 Biodiversity in Software Engineering Research

Biodiversity has been considered of critical importance in ecology as an indicator of sustainability and resilience. In recent years, SE research has employed various ecological methodologies based on diversity measures to investigate the dynamics of software systems and communities.

Previous works compared the natural ecosystems with specific software ecosystems. Dhungana et al. [27] pointed out that, just as biodiversity enables ecosystems to maintain their vital characteristics after disturbing events, diversity in SE and user groups strengthens the sustainability of software platforms such as Eclipse. Similarly, Baudry and Monperrus [11] introduced the concept of ecology-inspired software engineering, aiming to map ecological diversity types (e.g., genetic, functional, spatio-temporal) onto key aspects of software development such as robustness, productivity, and stability. The DIVERSIFY project [12] further advanced this perspective by proposing automated mechanisms to sustain software diversity, relying on a collection of software variants that act as a reservoir of adaptation solutions, thereby strengthening resilience against unforeseen failures. In parallel, the ECOS project [61] studied open-source ecosystems, where contributors (including developers and end-users) were considered as species while projects were viewed as resources that are produced and consumed. Based on this perspective, resilience increases with contributor increasing diversity (e.g. developers proficient in different programming languages and/or contributors specialized in different roles such as testing and debugging), enabling the ecosystem to better withstand "environmental" changes such as technology shifts or the obsolescence of legacy projects.

Biodiversity indices have been previously employed in SE studies for the investigation of gender/national diversity impact on project outcome. For example, Torchiano et al. [78] applied the Shannon and Blau indices to study gender and nationality diversity in student Agile teams, finding gender diversity positively correlated with outcomes. Azman and AlDhaheri [10] analyzed large GitHub projects, applying the Simpson index (and Shannon for gender diversity) to assess demographic, geographical, and technical diversity of contributors, reporting positive correlations between geographical/commit diversity and software quality, though mixed results for gender and employment diversity. These studies demonstrate the versatility of biodiversity indices in capturing different forms of heterogeneity in software projects and their impact on performance and quality.

Beyond diversity measures and indicators, plenty statistical methodologies originally developed for ecological purposes have also been transferred into SE. One example is the use of the Mantel test [58], a multivariate statistical method originally developed in ecology, biology, and population genetics to evaluate correlations between dissimilarity matrices. Mantel test has been applied for the monitoring of technical debt [3] and the investigation of the relation between principal and interest by examining whether artifacts with similar levels of TD principal also exhibited similar amounts of TD interest.

These previous works illustrate how biodiversity has been used in SE research both as an analogy for resilience in ecosystems of software and contributors, and as a quantitative framework for assessing diversity in teams and projects. While these studies focus on the dynamics of software systems, contributors, or teams, our work examines a completely different aspect: the labor market for software engineers. Specifically, we employ biodiversity indices to quantify the diversity of job requirements across software engineering occupations. This perspective allows us to extend the biodiversity methods into the domain of Labor Market Analysis (LMA), providing novel insights into how KSTs vary within and across the different SE occupational categories.

3. Experimental Setup

3.1 Context, Data Collection, and Units of Analysis

In this study, we utilized data that comprises OJAs for job openings located in the EU27 Member States, EFTA[9] countries and the United Kingdom. The data are organized and provided by the *Web Intelligence Hub* (WIH), because of collaborative effort between CEDEFOP and Eurostat. The WIH-OJA database covers a period from 2019 onwards, offering a wealthy source of information encompassing various characteristics such as, the geographical location of the offered job (city, region, country), the first and last active date of the job posting (specified in days, months and years), the required experience, and more. More importantly, the EU organizations have developed effective mechanisms for cleaning and pre-processing unstructured textual content of OJAs collected from the web. A three-phase approach that makes use of metadata and fuzzing-matching techniques was employed to remove duplicate OJAs before the application of data-driven methodologies. The data are then passed through an ontology and an ML-based pipeline for the extraction of the set of job requirements and corresponding occupation category for each OJA. As a result, the final dataset, along with its metadata, contains a detailed list of identified job requirements from Level 4 of the multilingual of ESCO skills pillar[10], as well as the corresponding occupation category from Level 4 of the ISCO-08 pillar[11]. We note that the ESCO skills pillar is organized into a hierarchical structure with four sub-classifications that are: (*i*) *Knowledge*, (*ii*) *Skills*, (*iii*) *Transversal Skills and Competences* and (*iv*) *Language Skills and Knowledge*. In this study, we focused on requirements that belong to one of the three first sub-categories (Knowledge (K), Skills (S) and Transversal skills and competences (T)), which are referred to as KST throughout the rest of the study.

Given that the main objective of this study is the investigation of the state-of-the-art in-demand job requirements in the software engineering workforce, we decided to focus, solely, on OJAs from the first three available quarters of 2024, whereas the subset of OJAs was, further, filtered to include only those relevant to SE professionals. Therefore, the inclusion criterion for an OJA to be included in the final dataset was based on the ISCO-08 occupations pillar, which classified each OJA into the "`251-Software and Applications Developers and Analysts`" occupation concept. Thus, the experimental setup for providing answers to the posed RQs was based on four distinct job occupations (Level 4 of ISCO-08): (a) "`2511-Systems Analysts`"; (b) "`2512-Software Developers`"; (c) "`2513-Web and Multimedia Developers`"; and (d) "`2514-Applications Programmers`".

The final dataset contains 1,198,791 OJAs, whereas a total set of 550 unique KSTs from Level 4 of ESCO skills pillar were identified. Figure 2 presents the frequency distribution of job openings across the 27 EU Member States and the United Kingdom, showing that France (30.3%), Germany (26.28%) and United Kingdom (13.62%) hold the lion's share, cumulatively accounting for over 70% of job openings for SE professionals. In

[9] European Free Trade Association
[10] https://esco.ec.europa.eu/en/classification/skill_main
[11] https://esco.ec.europa.eu/en/classification/occupation_main

terms of the four occupational categories, `Systems Analysts` (46.30%) and `Software Developers` (43.34%) are the most in-demand occupations (Table 4).

Table 4. Distribution of OJAs across the SE-related occupation categories (Level 4) from the first three quarters of 2024

Job Occupation	***N* (%)**
Systems Analysts	562,562 (46.30%)
Software Developers	519,608 (43.34%)
Web and Multimedia Developers	84,786 (7.10%)
Applications Programmers	31,835 (2.66%)

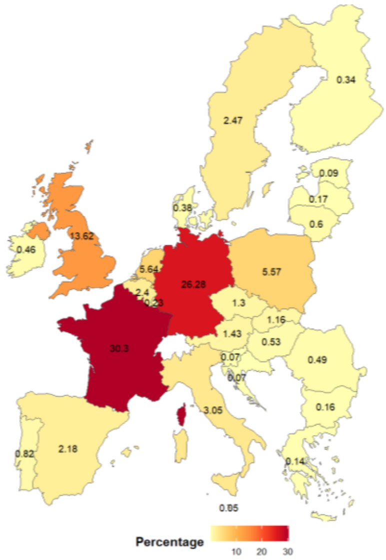

Figure 2. Frequency distribution of OJAs across EU countries from the first three quarters of 2024

3.2 Research Questions and Data Analysis

To address the aim of this study, which is to gain insights into the diversity of demanded KSTs (goal-1) and to identify the most-in-demand KSTs and profile them for SE professionals (goal-2), we followed a top-down approach. At the macro-level, we focused on the quantification and examination of the KST diversity in terms of richness and evenness: (a) within (goal-1a), and (b) among SE occupations (goal-1b). At the micro-level, we performed a fine-grained analysis to identify: (a) transferable KSTs that are needed across SE occupations (goal-2a), and (b) specialized KSTs differentiating occupations, being prototypical for specific occupations (goal-2b). In the next paragraphs, we present the motivation for the posed RQs of the study, while Table 5 provides a mapping of the goals, the associated RQs and guidelines on the biodiversity approaches, indices and algorithms (Methodological Conceptualization) applied to macro- and micro-level KST data for inferential purposes.

Table 5. Methodology Conceptualization

Goal	**RQ**	**Approach**	**Index / Algorithm**	**Used Data**
[RQ$_1$] *How does the variety and distribution of Knowledge, Skills and Transversal Competences qualifications differ within ($RQ_{1.1}$) and among ($RQ_{1.2}$) Software and Applications Developers and Analysts occupations*?				
Quantification of KST diversity Macro-level Goal-1	Within Occupations ($RQ_{1.1}$)	Alpha Diversity	Richness (S), Standardized Shannon's diversity (H'), and Simpson's evenness (E)	Composition (relative abundance) KST data computed on collections of OJAs
	Among Occupations ($RQ_{1.2}$)	Beta Diversity	Bray–Curtis dissimilarity index (BC) [17], Principal Coordinates Analysis (PCoA), and Permutational Analysis of Variance (PERMANOVA)	

Goal	RQ	Approach	Index / Algorithm	Used Data
[**RQ$_2$**] *Which Knowledge, Skills and Transversal Competences signify (RQ$_{2.1}$) are transferable across multiple Software and Applications Developers and Analysts occupations, and which Knowledge, Skills and Transversal Competences imply (RQ$_{2.2}$) prototypical specialization within specific occupations*?				
Identification and Profiling of KSTs Micro-level Goal-2	Transferable across Occupations (RQ$_{2.1}$)	Indicator Species Analysis (ISA)	Indicator Value (IV)	Individual OJAs
	Prototypical within Occupations (RQ$_{2.2}$)			

[**RQ$_1$**] *How does the variety and distribution of Knowledge, Skills and Transversal Competences qualifications differ within (RQ$_{1.1}$) and among (RQ$_{1.2}$) Software and Applications Developers and Analysts occupations*?

Motivation [RQ$_1$]: Given that the SE industry faces continuous and radical changes driven by the complex nature of software [81], the "bloom" of emerging technologies[12] and the lack of consensus on which KSTs shape the roadmap for the next generation of professionals [8], there is a persistent need for empirical evidence from data-driven approaches that keep a finger on the pulse of the workforce landscape. Understanding how the distribution of KST qualifications varies within and among SE professionals is critical for unveiling trends in future workforce demand, aligning training and educational programs and safeguarding the establishment of effective talent management strategies.

To answer RQ$_1$, which focuses on a macro-level analysis of the KST demand, we made use of the proposed biodiversity inspired approach applied to the composition KST data that were expressed as relative abundance by occupation category from OJA. This decision was driven by the need to investigate the variety and distribution of KSTs across the EU landscape with the aim of deriving conclusions regarding potential differences both within and among SE occupations. RQ$_{1.1}$ is addressed by using the alpha diversity indices (1st row in Table 5), along with the computation of appropriate measures of central tendency and variation, which can also serve for inferential purposes and decision-making. By considering the four SE occupations as different habitats and the KST qualifications as the species existing in these habitats, the alpha diversity analysis (Richness (S), standardized Shannon's diversity (H'_{Std}) and Simpson's evenness (E) are valuable indices for quantifying how rich, diverse and evenly populated these habitats are. Based on this metaphor, potential variation in richness, computed by the total number of distinct KSTs associated with each SE occupation, reflects broader or narrower must-have qualifications for positioning a specific occupation. The diversity and evenness of the detected KSTs within the four occupations are evaluated through the computation of the standardized Shannon's diversity and the Simpson's evenness indices on the relative abundance KST data.

The results that will be derived from the alpha diversity analysis can serve for exploratory purposes within a specific occupation but should not be used for comparisons between (among) occupations, as they are not, directly, comparable due to the significant differences in the sample sizes (number of OJAs) collected for each

[12]https://metapress.com/exploring-the-role-of-software-engineering-in-emerging-technologies/

occupation. Additionally, the unequal number of OJAs within each occupation at the predefined time intervals may also be an extra source of variability that needs to be controlled. To account for this variance due to "*sampling effort*", a well-known limitation in ecological studies, we made use of a *rarefaction* approach [30] that mitigates, efficiently, this threat. The rationale behind this standardization mechanism is to randomly sub-sample without replacement the sets of OJAs collected at the pre-defined time intervals so that the size of each set equals the smallest observed sample size within an examined occupation. To examine, whether there are significant differences among SE occupations ($RQ_{1.2}$), the Bray-Curtis dissimilarity matrix calculated on composition KST data was the main input for assessing the beta diversity index (2nd row in Table 5). This dissimilarity matrix was also used for exploratory purposes by the projection of potential differences into a 2D space via PCoA (2nd row in Table 5). Finally, a PERMANOVA model was fitted to statistically test; whether the observed differences can be attributed to the effect of the Occupation factor on job requirements composition (2nd row of Table 5).

[**RQ_2**] *Which Knowledge, Skills and Transversal Competences signify ($RQ_{2.1}$) are transferable across multiple Software and Applications Developers and Analysts occupations, and which Knowledge, Skills and Transversal Competences imply ($RQ_{2.2}$) prototypical specialization within specific occupations*?

Motivation [RQ_2]: Shifting to a micro-level analysis, we aim to identify the required skillsets targeting the development of candidate profiles within SE professionals. This is, certainly, a non-trivial task, as SE professionals share common qualifications that are pivotal for entering the SE industry[13]. Apart from these transferable KSTs, the SE industry seeks employees who are equipped with qualifications that are aligned to the specialized needs of specific occupations[14].

To gain insights into the mapping of KSTs to job occupations, ISA was employed on the initial set of OJAs by examining the presence of KSTs within each job opening. Specifically, the extension of the original ISA approach, proposed by De Cáceres et al. [25], was used to identify KSTs that are transferable to combinations of SE occupations ($RQ_{2.1}$) (3rd row in Table 5). Through this one-step-analysis, ISA also provides straightforward takeaways regarding KSTs that are specific to individual SE occupations ($RQ_{2.2}$) (4th row in Table 5).

To facilitate ease adoption and independent replication of the proposed methodology, we have developed a replication package that includes all core analysis scripts (i.e., data pre-processing, computation of biodiversity indices, execution of ordination methods and indicator species analysis) along with synthetic data that mirrors the structure of the data used in this study. The replication package is hosted on GitHub[15], providing detailed instructions for installing the required R packages and for executing the analysis.

4. Results

In the following sections, we present the results of this study organized by research question. We answer sub-questions together so that we can synthesize the main findings. For simplicity, the results are presented in raw form in this section, while interpretations and implications are discussed in Section 5 (Discussion).

4.1 Variety and Distribution of KSTs Within and Among the Software-Related Occupations

Table 6 summarizes the measures of central tendency and dispersion for the three alpha diversity indices, which are presented with their corresponding 95% confidence intervals. The inspection of richness provides insights into the number of unique job qualifications found in job openings for each occupation. For `Applications Programmers`, the mean richness value is approximately 187 unique KSTs. In contrast, the corresponding statistical measures are significantly higher for `Systems Analysts` ($M = 366.472(\pm 15.301)$) and

[13] https://www.wgu.edu/blog/6-qualifications-needed-become-software-engineer2302.html
[14] https://www.equalture.com/blog/skill-based-hiring-transferable-skills-vs-job-specific-skills/
[15] https://github.com/dtrygoni/Biodiversity_JobProfiling

`Software Developers` ($M = 318.056\ (\pm 13.943)$). To assist interpretation, Table 6 also includes the first and third quartiles for each diversity index, offering a practical reference point for identifying relatively low or high values within the empirical distribution.

One needs to be cautious when interpreting these findings, as, the divergences may be due to the actual need for broader skillsets or variant types of job roles within these occupations (`Systems Analysts` and `Software Developers`) but, in parallel, we have also to keep in mind that the high differences in sample sizes of OJAs may be an additional factor that may influence the estimated alpha diversity indices. Moreover, as we have already mentioned in Section 2.3, that richness alone does not provide meaningful insights into the diversity of qualifications within SE occupations and for this reason, we turn our interest into the examination of standardized Shannon's diversity and Simpson's evenness indices that consider both richness and evenness, but under different weighting mechanisms.

Focusing on the diversity and evenness of KSTs, a closer examination of the computed indices reveals that the ranges of the standardized Shannon's diversity indices suggest relatively even distributions of KSTs within all occupations, as the mean values are higher than 0.75 approaching unity, which, theoretically, represents perfect evenness. On the other hand, this conclusion is not supported by the evaluation of the Simpson's evenness indices, as the generally observed low values demonstrate that all SE occupations are dominated by a few KSTs. This inconsistent finding stems from the fact that the two indices give emphasis on different aspects of alpha diversity (richness and evenness). Shannon's diversity index and its standardized version prioritize richness, and thus, it is sensitive to the presence of rare KSTs that are in-demand within SE occupations, while Simpson's evenness is highly affected by the presence of dominant KSTs [67]. The choice of the most appropriate alpha diversity index should be guided by the specific needs and scopes of stakeholders. The former is a rationale choice for the assessment of alpha diversity from a more general perspective that considers emerging, evolving or infrequently demanded KSTs within SE occupations. The latter provides meaningful insights into discovering whether certain skillset profiles with dominant KSTs shape the SE workforce landscape.

Table 6. Descriptive and Exploratory Analytics for Alpha Diversity Indices across SE Occupations

Index	Statistic	Applications Programmers	Software Developers	Systems Analysts	Web and Multimedia Developers
S (richness)	*M* (*SD*), 95% CI	186.661 (9.032), [182.169, 191.153]	318.056 (13.943), [311.122, 324.989]	366.472 (15.301) [358.863, 374.081]	243.289 (12.941), [236.853, 249.724]
	Mdn (Q_1, Q_3)	189.200 (186.725, 191.225)	319.500 (314.100, 326.225)	369.600 (363.050, 376.050)	247.800 (243.700, 250.075)
H'_{Std} Standardized Shannon's diversity	*M* (*SD*), 95% CI	0.828 (0.006), [0.826, 0.831]	0.762 (0.007), [0.759, 0.765]	0.751 (0.006), [0.748, 0.754]	0.809 (0.006), [0.806, 0.812]
	Mdn (Q_1, Q_3)	0.827 (0.824, 0.829)	0.761 (0.758, 0.765)	0.751 (0.748, 0.754)	0.808 (0.805, 0.810)
E Simpson's evenness	*M* (*SD*), 95% CI	0.279 (0.013), [0.272, 0.286]	0.174 (0.010), [0.170, 0.179]	0.147 (0.007), [0.144, 0.151]	0.242 (0.012), [0.236, 0.248]
	Mdn (Q_1, Q_3)	0.278 (0.271, 0.286)	0.173 (0.168, 0.179)	0.147 (0.143, 0.150)	0.239 (0.234, 0.244)

Notes: *M, SD, 95% CI, Mdn,* Q_1, Q_3 stand for mean, standard deviation, lower/upper limits of 95% Confidence Intervals for mean, median, first and third quartile, respectively

To provide a more detailed insight into the kind of unique KSTs that appear within the job openings of specific occupations, in the Figure 3, we present the proportional distribution of unique job requirements that fall in each KST category (as provided by ESCO) for all SE occupations; whereas in the Figure 4, we present the relative frequencies of the categories of the hard skills taxonomy as proposed by Montandon et al. [64] within the four SE occupations[16]. From the results, we can observe that for all SE-related occupations the types of KSTs follow

[16] The mapping of KST to the hard-skills' categorization has been performed by the 4th and 5th author. Minor conflicts have been resolved by the 3rd author.

a similar pattern concerning the required KSTs: unique *Skills* (S) cover 40%-50% of the unique qualifications that appear for these occupations; unique *Knowledge* (K) covers 42%-49% of the unique qualifications that appear for these occupations; and unique *Transversal Skills and Competences* (T) covers 8%-12% of the unique qualifications that appear for these occupations. In terms of ranking, for `Web and Multimedia Developers`, as well as `Application Programmers` *Knowledge* (K) is the most required kind of KSTs, whereas for `System Analysts` and `Software Developers`, *Skills* (S) are the most required kind of KSTs. In terms of hard-skills, `System Analysts` is the occupation for which Data Systems related KSTs are the most useful for. KSTs on Development tools appear to be an important part of the qualifications required for `Web and Multimedia Developers`, whereas KSTs on Processes & Methods, Frameworks and Libraries, and OS & Infrastructure appear as consistently important for all occupations. KSTs on Programming languages appear to be more relevant for `Web and Multimedia Developers` and `Application Programmers`.

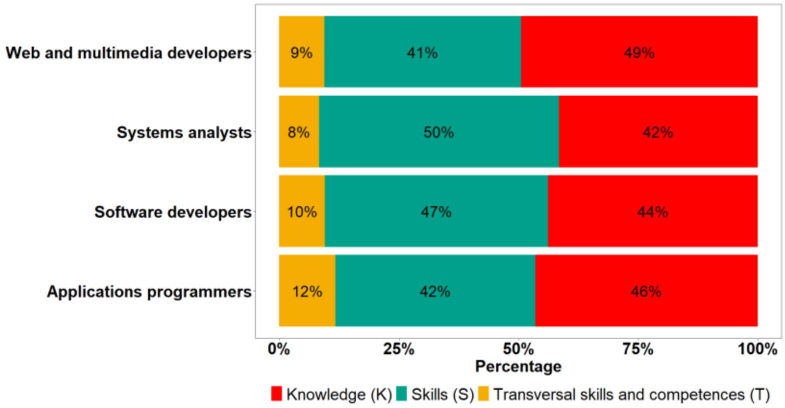

Figure 3. Classification of KST job requirements per SE-related occupation, grouped by ESCO categories.

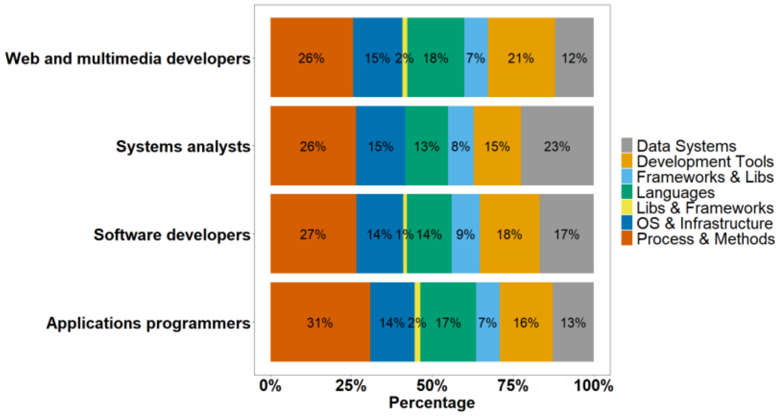

Figure 4. Classification of KST job requirements per SE-related occupation, grouped by hard skills taxonomy proposed by Montandon et al.[64]

Next, we proceed by focusing on the examination of the diversity in KST composition among different SE job occupations. To shed light on this second research objective, we first investigate potential fluctuations in composition of KSTs through exploratory visualization techniques with the aim of uncovering similarities and / or differences in KST composition across the examined occupations. Figure 5 depicts the biplot obtained from the deployment of PCoA after the computation of the Bray-Curtis dissimilarity index on the composition KST data. In this plot, each group of point represents a specific occupation category highlighted with different color, whereas the geometric position of the point "*quantifies*" the KST composition in terms of relative abundances at the predefined 15-day intervals of collection. The 2D projection explains about 80% of the total variation in the dissimilarity matrix exhibiting a satisfactory representation into the low-dimensional space. The inspection of the relative position of the points brings to the surface meaningful insights into the KST demand landscape. A remarkable finding concerns the existence of distinct clusters of points that belong to the same occupation category. This practically means that clusters of points that are far from each other can be considered as SE occupations, in which there are diverse qualifications in terms of KSTs. In this regard, the points representing `Systems analysts` are far from both the origin and the rest occupational categories, a fact that implies differences in qualifications of job occupations. In contrast, `Web and Multimedia Developers` and `Application Programmers` are related, indicating alternative KST profiles. Finally, the cluster of `Software Developers` is positioned near the center of the graph and very close to the origin point and due to this fact, one can infer that, while this specific occupation is distinct, it shares similar KSTs with all other occupational categories. Thus, the graphical inspection of the biplot signifies differences in KSTs demand among the SE-related occupations. However, while PCoA provides a "*big picture*" of the projected dissimilarities in a low-dimensional (2D) space, and it is considered an exploratory technique that may not fully uncover the observed patterns hidden within the raw dissimilarity matrix of the full multivariate space. Due to this fact, we conducted formal statistical hypothesis testing procedures to examine whether these observed differences are not due to chance.

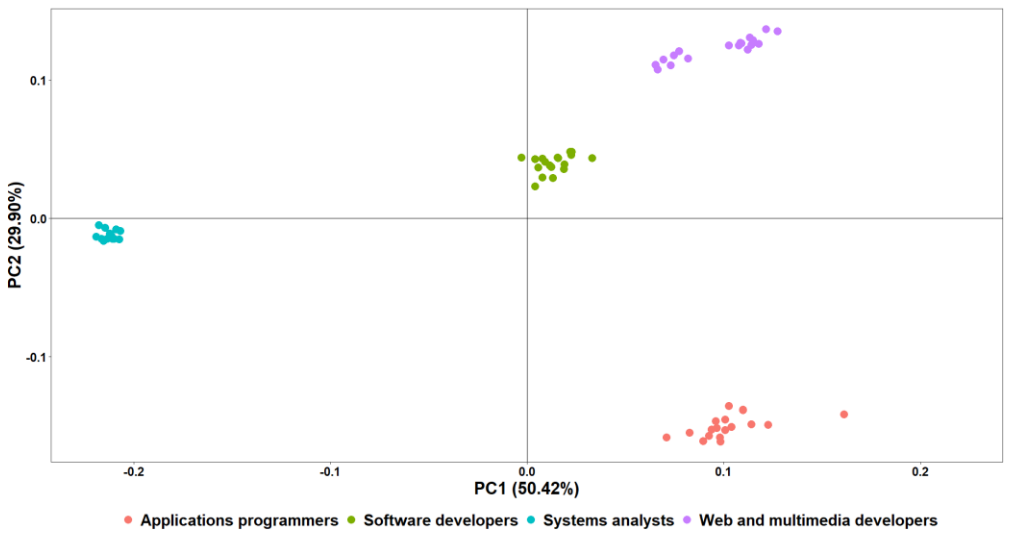

Figure 5. Biplot of PCoA based on Bray-Curtis beta diversity dissimilarity index

PERMANOVA constitutes a well-established inferential mechanism based on resampling techniques for examining the effect of multiple factors on relative abundance data computed by the Bray-Curtis dissimilarity index. Therefore, in our experimental setup, PERMANOVA was applied to estimate the variation between and within occupation categories through the computation of a test statistic, namely, pseudo-*F* ratio. Next, the distribution of the pseudo-*F* ratio is, repeatedly, evaluated on many randomly shuffled distance matrices by permuting the levels of the factors under investigation. This empirical distribution is used, in turn, to obtain the statistical significance of the hypothesis testing procedure.

To assess the potential temporal (*Time*) and occupational (*Occupation*) effects on the composition of relative abundances of KSTs, we fit separate one-way PERMANOVA models to measure the variance explained by each factor, independently. Regarding the examination of the temporal effect, the results indicated that *Time* did not present a statistically significant main effect on KST composition ($F(17,51) = 0.094, p = 1.000$) with the model explaining only 2.9% of the total variance ($R^2 = 0.029$). In simple terms, the *Time* variable did not introduce extra variability and / or spurious effects into the inferential process that should have been taken into consideration, when the interest shifted to the investigation of composition differences among occupations. Moreover, this finding indicates that the demanded qualifications for SE professionals remained intact during the examined period, a reasonable outcome given the short time-span of data collection. Certainly, any important technological shift in the demand for KSTs among SE professionals would, typically, require a longer time horizon to be meaningfully detected in the labor market. In contrast, the results for *Occupation* revealed a strong and statistically significant effect on KST composition ($F(3,68) = 406.967, p = 0.001$), as the model explained 94.7% ($R^2 = 0.947$) of the total variability highlighting the overwhelming importance of occupational differences in KST composition. Finally, post-hoc analysis showed that all occupation pairs differed significantly ($p = 0.001$).

- The complete list of KSTs that is used to characterize the profiles of OJAs for SE-related occupations is very broad, ranging from 180-360 KSTs.
- The skillset required for each occupation is quite distinct, since there are statistically significant differences in the composition of the skillsets.
- The skillset required for System Analysts is characterized by uniqueness among the SE-related occupations, whereas the skillset required for Software Developers seems to have the most central role.
- KSTs on Software Processes & Methods appear to be the kind of KST that is required primarily by all SE-related occupations; whereas KSTs on Programming languages appear to be more relevant for Web and multimedia developers and Application programmers and KSTs on Data Systems are the more useful for System Analysts.

4.2 Transferable and Specialized KSTs Across Multiple SE-Related Occupations

Next, we proceed with ISA aiming at the identification of KSTs that are either transferable or specialized across SE professionals. Figure 6 presents the joint distribution of KSTs for all combinations of SE occupations (rows) and the ESCO skills pillars (columns) under examination. A first remarkable finding concerns the examination of the identified KSTs within occupations that fall under the general ESCO pillar of *Transversal Skills and Competences* (T). In this regard, the relative frequency of transversal skills that are required in all occupations (22.1%) is the highest compared to the corresponding percentages observed in all triplets, pairs and individual categories. In contrast, the skillsets tailored to individual occupations are mostly (above 94%) KSTs that belong to the *Knowledge* (K) and *Skills* (S) pillars. Additionally, `Applications programmers` and `Web and multimedia developers` professionals need to be more familiar with "*the body of facts, principles, theories and practices that is related to a field of work or study*[17]" (K) than "*abilities to apply knowledge and use know-how to complete tasks and solve problems*" (S).

[17] https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/knowledge

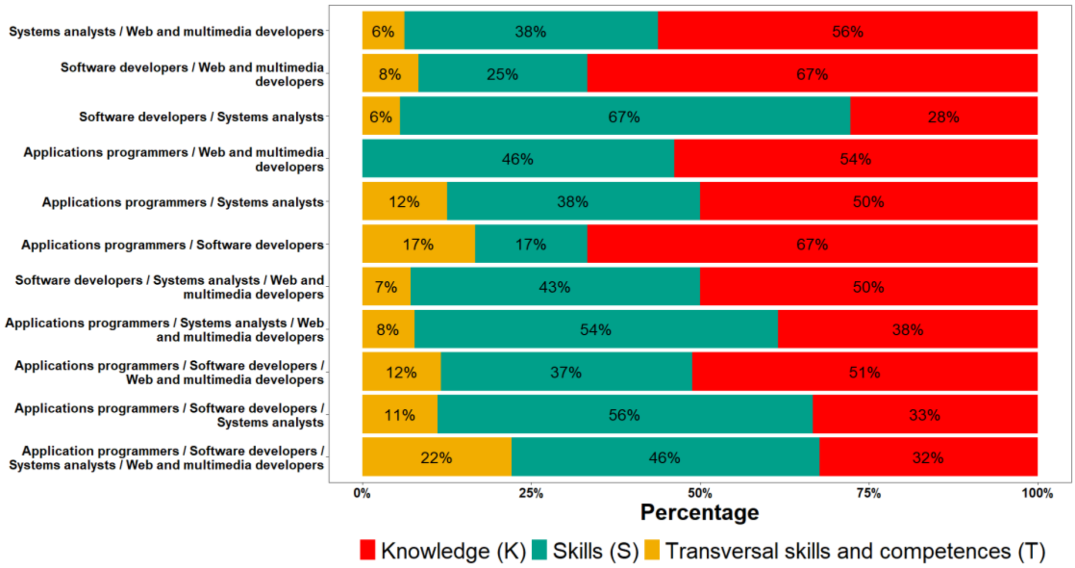

Figure 6. Distribution for combinations of job occupations and KSTs

Regarding the identification of transferable qualifications that are in-demand for various SE professionals, the tables in Appendix A summarize the findings extracted from the application of ISA on the identified KSTs within the total set of collected OJAs. The tables present *IV*s for each KST, along with the specificity / sensitivity metrics and the associated *p*-values for all possible combinations of the four SE occupations. For example, "*adapt to change*" is the top-most requested KST exhibiting an *IV* of 86.17, computed by taking the square root of product of specificity (1.00) and sensitivity (0.7425). This practically, means that this specific transversal competence can be considered as "*indicator*" or representative KST for all SE occupations. Moreover, in nearly 75% (sensitivity) of the total set of OJAs related to SE occupations, there is an indispensable demand for job candidates that demonstrate flexibility, willingness and openness to adapt, effectively, to new conditions, situations and challenges arising in workplace environment.

To visualize the information of the most-in-demand KSTs along with their importance in the EU landscape, we exploited Network Analysis visualization techniques based on the results that were derived from the application of ISA on OJAs. Figure 7 visualizes the resulting network, where the four SE occupations are represented by black-colored text (nodes), and KSTs are depicted with different font colors according to their ESCO pillar classification. Moreover, the font size of each KST is proportional to its *IV* to facilitate the identification of highly important qualifications. We note that KSTs with an *IV* lower than 20 were filtered out from the analysis to enhance the clarity and readability of the produced outcome. Interpreting the graph, nodes (KSTs) positioned near to each other and close to the four black occupation hubs demonstrate commonalities in the qualifications among the four SE occupations. The same information is visualized, split by ESCO pillar, in Appendix B.

Indicatively, the top-5 KSTs that are "*adapt to change*", "*work in teams*", "*have computer literacy*", "*computer programming*" and "*teamwork principles*" are connected by edges to all four occupations. Oppositely, nodes that are in isolation at the outer boundaries of the network represent prototypical KSTs to individual occupations, as they exhibit single connections (edges) to specific occupations. In this regard, "*quality standards*", "*sell services*", "*sell products*", "*mobile operating systems*", "*mobile device software frameworks*" and "*Android (mobile operating systems)*" form a specialized in-demand skillset for `Applications programmers`. Finally, "*develop creative ideas*", "*SQL*", "*Java (computer programming)*", "*implement front-end website design*", "*unified modelling language*", "*use query languages*" and other KSTs provide must-have qualifications for candidates seeking for a job opportunity related to the development of (web) applications and software (`Applications programmers`, `Web and multimedia developers`, `Software` `developers`). Visualizations by ESCO pillars allow us to perceive connections of hard-skills to occupations, since in Figure 7 they are subsumed by Transversal skills and competencies. More specifically, we can observe that the technologies with the most central role in the network are "*PHP*" and "*SQL*", followed (with quite some distance) by "*C#*", "*web programming*", and "*databases*". In terms of specific occupations, we can observe that `Web and multimedia developers` must be aware with technologies such as "*graphics editors design*", "*CSS*" and "*TypeScript*", "*design UIs*", and "*use creative suite software*". Furthermore, for `System Analysts,` we can observe a tendency to require KSTs related to the "*analysis of business requirements*" and "*analysis of business processes*". Finally, from the network analysis it becomes evident that "*scripting programming*" and "*query languages*" are outperforming more traditional programming paradigms and languages, such the "*object-oriented programming*".

- *Transversal Skills and Competences* (T) appear to be the ESCO pillar that contributes the most transferable qualifications among all the software occupations studied.
- "adapt to change" and "work in teams" are the KSTs that appears more uniformly to all studied software occupations
- "computer programming" is the top hard-skill that appears more uniformly to all studied software occupations. However, each occupation shows some specific qualifications. For instance, mobile or Android-related KSTs are more fitting to Application Programmers, whereas UI design, CSS, TypeScript and Graphics Design are more needed for Web and multimedia developers.
- System Analysts seem to have a dual focus, either on data-related KSTs or software analysis-related KSTs.
- Specific languages (such as PHP, SQL), programming paradigms (such as scripting and query programming), or domains (such as web programming) appear to be more central in the network of KSTs, compared to more traditional programming paradigms and languages.

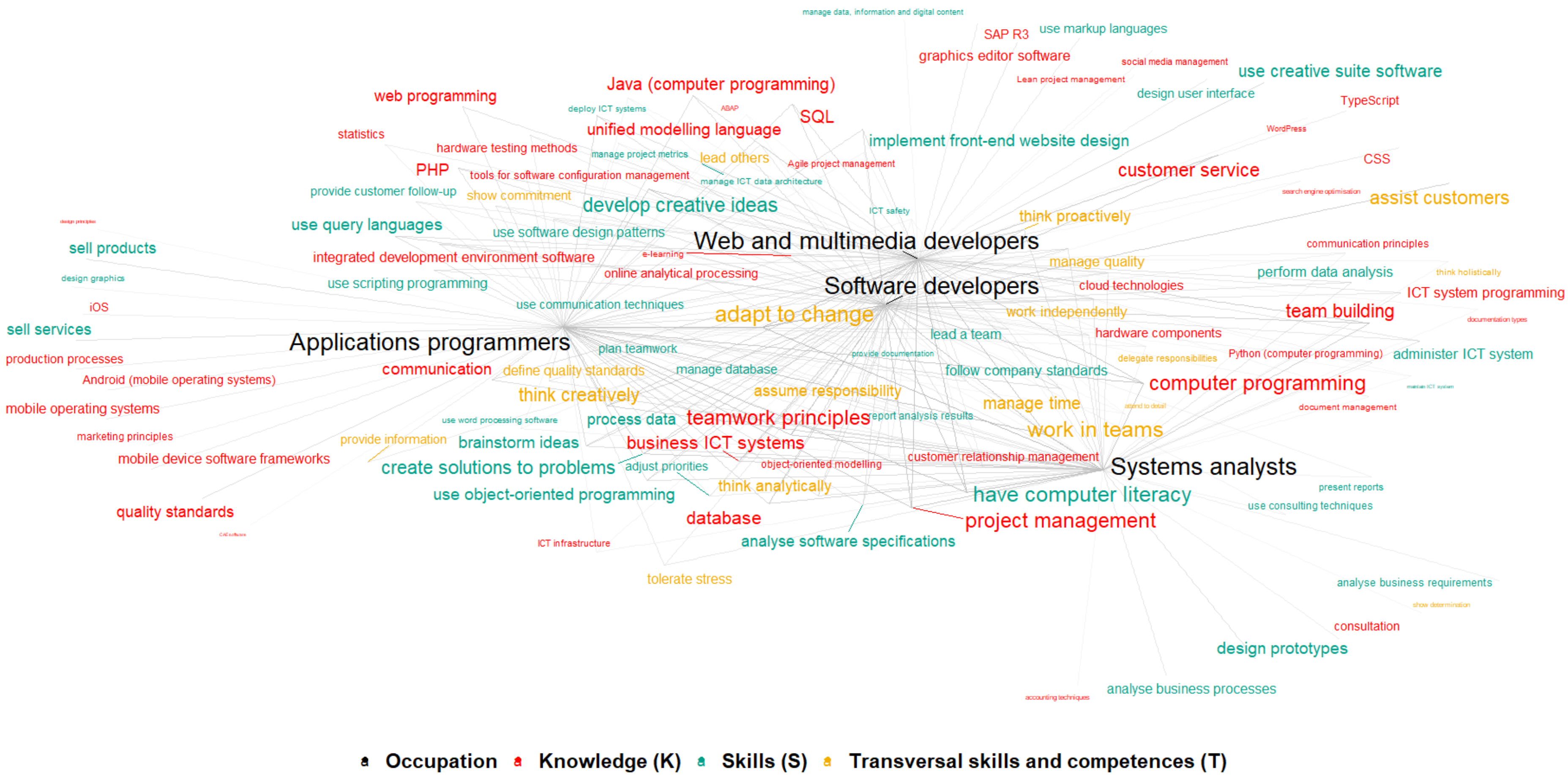

Figure 7. Distribution for combinations of job occupations and KSTs

5. Discussion

5.1 Interpretation of Results

In this section we discuss the main findings of this work, and interpret them either against previously reported results in the literature, or on our intuition. The main results from both research questions are synthesized and their presentation is organized by discussion topic.

Problem Scope and Relevance: The complete list of KST that is used to characterize the profiles of OJAs for SE-related occupations is very long, ranging from 180-360 KSTs. This result supports previous evidence, despite the fact that different research methods have been used [64]. This work extends previous knowledge on skills that IT companies look for, by showing that the skillset required for each occupation is quite distinct, since there are statistically significant differences in the composition of the skillsets. These findings suggest that software engineering is a domain for which skills management can be very challenging. Consequently, *skills management for SE (Skills4SE) is a research area that is industrially relevant, which can provide stimulating research problems, urging for novel and specialized solutions*. Furthermore, the demanding and evolving landscape of skills for software engineering professions calls for close industry-academia collaboration to regularly monitor and update university curricula.

Hard-Skills4SE: KSTs on Software Processes & Methods appear to be the kind of KST that is required primarily by all SE-related occupations. This finding is intuitive in the sense that regardless of the application domain (general, applications, web, multimedia, etc.) the baselines of SE (e.g., development methods, patterns, practices, etc.) are the tools to make software development more efficient. Also, in terms of the original taxonomy of hard skills [64] the class of Software Processes and Methods is the most inclusive in terms of ESCO KSTs. Additionally, our findings have suggested that KSTs on programming languages appear to be more relevant for `Web and Multimedia Developers` and `Applications Programmers`, underlying the dependence of applications (especially web and mobile apps) on the choice of the right language. Naturally, "*computer programming*" is the top hard skill that appears more uniformly to all studied software occupations. Nevertheless, each occupation shows some specific qualifications. For instance, mobile or Android-related KSTs are more fitting to `Applications Programmers`, whereas UI design, CSS, TypeScript and Graphics Design are more needed for `Web and Multimedia Developers`. Finally, KSTs on Data Systems are the most useful for `Systems Analysts`. Even more interestingly, the skillset required for `Systems Analysts` is characterized by uniqueness among the SE-related occupations, whereas the skillset required for `Software Developers` seems to have the most central role. These observations are also explained by the categorization of ISCO on SE-related occupations: (a) `Web and multimedia Developers` and `Applications Programmers` are the two SE-related occupations that focus solely on development, whereas the others are more inclusive; and (b) the occupation of `Systems Analysts` includes roles that related to Data Analysts, which are not found in the other categories. Given the rise in the demand of Data Analysts in the recent years [43, 80], we expect that this role tends to dominate the class of `Systems Analysts`, compared to its other roles—*temporal dimension*. The temporal dimension of the skills shift is also underlined by focusing on programming languages. Specific web-based languages (such as PHP, SQL), programming paradigms (such as scripting and query programming), or domains (such as web programming) appear to be more central in the network of KSTs, compared to more traditional programming paradigms and languages; to our understanding this can also be explained by the focus of software industry on data-driven solutions that are deployed on the cloud using web technologies, empowered by modern AI technologies [51]—underlying a shift of technologies in the recent OJAs that we have studied. Consequently, adhering to the 4Ps rule [15]: "*assigning the right (P)erson, with the right expertise / (P)erformance to the right (P)osition at the right time (P)eriod",* necessitates a formal and evidence-based identification and profiling of diverse skillsets that are specialized for each occupation.

Soft-Skills4SE: Additionally, our work suggested that Transversal skills and competences appear to be the ESCO pillar that contributes the most transferable qualifications among all the software occupations studied. Our findings extend previous results underscoring the importance of soft skills, both in the software industry [1, 57], but also in any collaborative environment [16, 59]. Among the soft skills the ones that are more transferable have proven to be “Adapt to change” and “Work in Teams”, appearing more uniformly to all studied software occupations, in agreement with previous findings about soft skill requirements for various software development positions [1]. It goes without saying that adapting to changes is a top skill for an employee in an environment that changes constantly, as is happening in software engineering [30, 64], especially if we consider the agile processes adopted by many development teams. In the software industry it is reported that people are changing job very frequently (every 3 to 5 years on average[18]); therefore, the quick adaptation in the new environment is an important skill, which becomes even more important that even within a company the software stack is changing [1] and it is needless to say that customer requests and requirements are very fluid and continuous changes are required [56]. Thus, software engineers with an adaptive mentality are highly admired. In terms of teamwork, we also believe that this is an intuitive finding, in the sense that collaborative software development is the prevalent (if not the only applicable) way of developing software [59]. In that sense, and by considering the importance of team spirit and team morale in productivity and work satisfaction, hiring employees that are skilled to work in teams is highly desirable. Concluding, *educating early SE in soft-skills is equally important as educating them in hard-skills*; also, from a research perspective *understanding and the effect of teams’ soft-skills in software engineering is an interesting research direction.*

5.2 Retrospective Assessment of Bio-Diversity Analysis and Lessons Learnt

The use of ecology and biodiversity-inspired concepts in modeling the software engineering labor market presents both promising opportunities and important challenges. One major consideration is that the dynamic and rapidly evolving nature of the software industry, with its constant introduction of new technologies, frameworks, and methodologies, can be difficult to fully capture using a model inspired by relatively stable natural ecosystems. Additionally, this approach may oversimplify the complexity of human factors, such as individual motivations, career aspirations, or decisions driven by personal or organizational contexts. These aspects are often not directly observable in job advertisements or codified in structured taxonomies. As a result, the approach may overlook important aspects such as personal career motivations, informal learning pathways, or emerging trends. Finally, questions about scalability remain, particularly as the size and complexity of the labor market continue to grow. Despite these limitations, ecological methodologies offer a valuable high-level perspective for comparative analysis, especially when combined with complementary approaches that capture detailed individual behaviors, learning trajectories, and context-specific dynamics. While a wide range of techniques in LMA rely on statistical modeling, machine learning, or artificial intelligence to extract skills and classify occupations from online job advertisements [71], these approaches primarily emphasize detection, classification, or prediction. In contrast, our biodiversity-inspired approach focuses on the composition, distribution, and evolution of skill sets within occupational categories, thereby complementing, rather than substituting, existing methodologies. Graph-based models [40], taxonomy-driven classifiers [28], and predictive techniques [55] offer valuable insights into structural mapping and forecasting. However, our approach provides a theory-driven aspect in ecological diversity, capturing specialization and role differentiation through measures like ISA. While we acknowledge that biodiversity theory does not address all dimensions of labor market dynamics, such as forecasting or semantic similarity, it adds explanatory depth to how skills are organized and transformed.

[18] https://www.linkedin.com/pulse/frequent-job-changes-its-benefits-rajesh-choubey/

5.3 Implications for Practitioners, Researchers, and Future Work Opportunities

Implications for Practitioners: The findings have multiple implications for various practitioners around the software engineering community. First, the results of the study can be used by *early-stage software engineers* or *software engineers that want to switch occupation*, to understand the skills that are required for their target occupation. This understanding can be used for guiding possible upskilling or reskilling endeavors, but also to realize if they are ready to perform a career change, in terms of possessed and desired skills. Additionally, *HR departments of SE-related organizations* can improve their hiring processes, and particularly the process of building job advertisements, by inspecting the skills that the competition is targeting at for each occupation. Furthermore, *educational stakeholders in the domain of SE* can inspect the findings of this study to evaluate the industrial relevance of their departments, placing special emphasis on KSTs that are transferrable among SE-related occupations (e.g., on team building, adaptation to change, etc.).

Implications for Researchers: The findings of this study underlined that the specific research field, i.e., skills management, can benefit from the presented biodiversity-inspired indices and approaches, regardless of the application domain. Researchers, especially in social sciences, can leverage the alpha diversity and beta diversity indices to identify and reason about skills that are frequent or unique in a set of professional areas. Indicator Species Analysis, as presented in the context of Software Engineering occupations, can be leveraged to identify crosscutting / transferrable skills (across occupations) or prototypical skills within certain professions. The data-agnostic nature of the proposed approach provides a straightforward mechanism for large-scale experimentation that is robust to the choice of: (a) data sources, (b) skills extraction techniques, (c) machine learning models for skills and occupations categorization, (d) skills/occupations taxonomies and (e) temporal and / or regional dimensions. The rationale of this paper complies with the "light" theory concept proposed by Avison and Malaurent [90], which champions replacing excessive focus on theory development, with the provision of empirical evidence. Relying on such evidence, we have provided empirical evidence suggesting policy makers and skill experts provide a taxonomy of occupations that are organized first per application domain (e.g., web, cloud, mobile, applications, etc.) and then a second level taxonomy based on the role (e.g., developer, analyst, data analyst, designer, tester, etc.). As the current taxonomy of ISCO stands the 4 SE-related occupations are mixing roles and domains at the same level, leading to an important overlap of required KSTs, which is confusing for both practitioners and researchers.

Future Work Opportunities: We encourage researchers to further work with the developed dataset to prioritize the links between KSTs and occupations using explainable AI techniques. Another interesting extension would be to develop competence and skills matrices for each occupation, along various KST levels. The findings from the application of biodiversity inspired approach open future research directions that could prove beneficial for monitoring the workforce of future professionals and occupations. One such direction involves replicating the study using alternative digital sources for collecting OJAs from industry. This would enable researchers and practitioners to empirically uncover insights at a more fine-grained level of analysis focusing on specialized job roles within occupations of interest. In addition, several critical research questions emerge within the proposed approach. For example, it would be of great importance to understand whether a high diversity of skills in the software ecosystem or within specific occupations has a positive effect or poses challenges for industry. This question has direct implications for policymaking, as it remains unclear whether stakeholders should promote skill variation or instead prioritize a narrower set of better-controlled competences. Furthermore, it is worth exploring whether the digital transformation of industry contributes to the extinction of traditional skills and competences, while simultaneously encouraging the transition into newly digital and green-related job requirements. Finally, another research direction would be to investigate whether it is even feasible to restrict the diversity of skills, given the extremely complexity of human nature and the continuously growing needs of society. These considerations lead to broader questions regarding the role of skills diversity in promoting individuality, inclusiveness, equality, innovation, freedom of choice, personal evolution or, conversely, in enabling biased evolution toward controlled stereotypes.

5.4 Threats to Validity and Limitations

Considering that the goal of this study was to explore the landscape of knowledge, skills and competences that characterize the profiles of OJAs for SE-related occupations, a valid threat to the generalizability (or external validity) of the findings can be identified. The retrieval of approximately 1.2 million OJAs, obtained from the collaborative data collection effort between CEDEFOP and Eurostat and compiled in the WIH-OJA database partially mitigates this threat; nevertheless, the results should be interpreted with caution, as they reflect a particular timeframe (i.e., the first nine months of 2024), specific SE-related professional occupations (Applications Programmers, Software Developers, Systems Analysts, Web and Multimedia Developers) and a particular geographical region (i.e., EU27 Member States, EFTA countries and the United Kingdom).

Resorting to the established ESCO skills framework for the classification of skills and the ISCO-08 occupations pillar mitigate construct validity threats, as these taxonomies provide structured definitions for 13,939 skills and 3,039 occupations. However, we acknowledge that certain newly emerging skills (e.g., PyTorch, Jupiter Notebooks, Flask etc.) and occupations (e.g., Prompt Engineer, Digital Twin Engineer etc.), that may characterize SE-related professionals may not yet be integrated into the ESCO taxonomy or may be classified under broader existing categories. We believe that this construct validity threat is mitigated by relying on the most recent version of the ESCO skills pillar available at the time of analysis. In any case, it should be underlined, that the data-agnostic nature of the proposed approach enables future replication using updated ESCO databases, or any real-world datasets containing presence / absence features for skill-related information, whether from OJAs, resumes, interview assessments, internal HR systems or other frameworks such as O*NET. Another construct validity threat arises from the fact that the study is entirely based on a pre-processed tabular format originally developed by CEDEFOP and integrated into Eurostat's WIH, where the mapping of raw skills demand expressed in textual form to ESCO skills and ISCO occupations was performed using its standardized data pipeline comprising several steps and processes[19]. As such, we did not perform any manual or algorithmic pre-processing on the raw text of the OJAs corpus, nor we developed any machine learning approach for skills extraction or mapping purposes. On one hand, this pre-processed tabular database may lead to certain limitations, since the original title and description of each OJA are not available, preventing more fine-grained analysis of industry-specific roles and job descriptions (e.g. IoT Developer, Data Analyst, etc.) that may appear in original OJA titles. It also limits the ability to investigate and uncover potential discrepancies or misuses within or across industry sectors. Moreover, the closed-source nature of various ontology-based and machine learning models used in the WIH-OJA database to classify each OJA to a particular occupation or to extract and map text to a specific skill, constitutes a distinct construct validity threat. On the other hand, these limitations and potential threats to construct validity are substantially mitigated by the rigorous quality control processes implemented by WIH members, since quality assurance constitutes a primary concern for the key stakeholders (CEDEFOP, Eurostat, European Commission) involved. The employed quality control processes, which include data validation and quality monitoring by WIH members, help ensure the reliability and validity of the dataset. In this regard, utilizing a custom dataset of OJAs would introduce even more threats to validity and bias, stemming from the subjectivity related to data source selection, as well as the implementation of data pre-processing, skills extraction techniques and mapping approaches to the ESCO skills and occupations frameworks.

Finally, the statistical validity of the findings has, to the best of our knowledge, been secured using multiple, complementary statistical analyses before drawing any conclusions and by the provision of a documented and

[19] https://cros.ec.europa.eu/wih/oja

navigable replication package.

6. Conclusions

This study aims to profile knowledge, skills and competencies required for a career in software engineering. To achieve this goal, we have analyzed more than 1 million job advertisements as collected by Eurostat and CEDEFOP. The means of analysis was an adapted biodiversity approach that calculated various indices, by mapping ecology terms to the SE workforce ecosystem. The analysis has unveiled that the complete list of knowledge, skills and competencies that is used to characterize the profiles of job advertisements for SE-related occupations is very broad, and that the skillset required for each occupation is quite distinct, since there are statistically significant differences in the composition of the skillsets. For instance: (a) the skillset required for System Analysts is characterized by uniqueness among the SE-related occupations, whereas the skillset required for Software Developers seems to have the most central role; and (b) KSTs on Software Processes & Methods appear to be the kind of KST that is required primarily by all SE-related occupations; whereas KSTs on Programming languages appear to be more relevant for Web and multimedia developers and Application programmers and KSTs on Data Systems are the more useful for System Analysts. Additionally, Transversal Skills and Competences (T) appear to be the most transferable qualifications among all the software occupations studied. More specifically, “adapt to change” and “work in teams” are the KSTs that appears more uniformly to all studied software occupations; whereas “computer programming” is the top hard-skill that appears more uniformly to all studied software occupations. However, each occupation shows some specific qualifications. For instance, mobile or Android-related KSTs are more fitting to Application Programmers, whereas UI design, CSS, TypeScript and Graphics Design are more needed for Web and multimedia developers. In terms of specific occupations, System Analysts seem to have a dual focus, either on data-related KSTs or software analysis-related KSTs; whereas in terms of skills we note that specific languages (such as PHP, SQL), programming paradigms (such as scripting and query programming), or domains (such as web programming) appear to be more central in the network of KSTs, compared to more traditional programming paradigms and languages.

Acknowledgements

This work has been funded by the European Union’s Horizon Europe Framework Programme under grant agreement No 101132663 (project SKILLAB).

References

[1] Ahmed F., Capretz L. F., Bouktif S., Campbell, P. (2015). Soft Skills and Software Development: A Reflection from the Software Industry. International Journal of Information Processing and Management. 4 (3). https://doi.org/10.1016/j.jss.2021.111183

[2] Ajimati, M. O., Whelan, E., Aalbers, R., Morgan, L., & van Kranenburg, H. (2022). The effect of advice network connectedness on problem-solving competence among software developers. Journal of Systems and Software, 194. https://doi.org/10.1016/j.jss.2022.111489

[3] Ampatzoglou A., Mittas N., Tsintzira A. A,, Ampatzoglou A., Arvanitou E. M., Chatzigeorgiou A., Avgeriou P., Angelis L. (2020). Exploring the Relation between Technical Debt Principal and Interest: An Empirical Approach. Information and Software Technology. 128. https://doi.org/10.1016/j.infsof.2020.106391.

[4] Anderson M. J. (2001). A new method for non-parametric multivariate analysis of variance. Austral ecology, 26(1): 32-46. https://doi.org/10.1111/j.1442-9993.2001.01070.pp.x

[5] Anderson M. J. (2014). Permutational multivariate analysis of variance (PERMANOVA). Wiley statsref: statistics reference online. https://doi.org/1-15. 10.1002/9781118445112.stat07841

[6] Andrewartha H. G., Birch L. C. (1986). The ecological web: more on the distribution and abundance of animals. University of Chicago Press.

[7] Assyne, N., Ghanbari, H., & Pulkkinen, M. (2022). The essential competencies of software professionals: A unified competence framework. Information and Software Technology. 151. https://doi.org/10.1016/j.infsof.2022.107020

[8] Assyne N., Ghanbari H., Pulkkinen M. (2022). The state of research on software engineering competencies: A systematic mapping study. Journal of Systems and Software. 185. https://doi.org/10.1016/j.jss.2021.111183

[9] Ayas, H. M., Hebig, R., & Leitner, P. (2024). An empirical investigation on the competences and roles of practitioners in microservices-based architectures. Journal of Systems and Software, 213. https://doi.org/10.1016/j.jss.2024.112055

[10] Azman S. K., Aldhaheri K. (2016). An Empirical Study of Contributors Diversity and Software Quality in GitHub Projects. UAE GSRC 2016. United Arab Emirates University, Al-Ain, United Arab Emirates. April 2016.

[11] Baudry B., Monperrus M. (2012). Towards ecology-inspired software engineering. CoRR.

[12] Baudry ., Monperrus M., Mony C.,Chauvel F., Fleurey F., et al.. (2014). DIVERSIFY - Ecology-inspired software evolution for diversity emergence. CSMR, Feb 2014, Antwerp, Belgium. IEEE, pp.395-398.

[13] Berenbach B. (2008). The other skills of the software architect. 1st International Workshop on Leadership and Management in Software Architecture (LMSA '08), International Conference on Software Engineering (ICSE ‘08). Leipzig, Germany, May 11, 2008. https://doi.org/10.1145/1373307.1373310

[14] Bergersen, G. R., Sjøberg, D. I. K., & Dybå, T. (2014). Construction and validation of an instrument for measuring programming skill. IEEE Transactions on Software Engineering, 40(12), 1163–1184. 10.1109/TSE.2014.2348997

[15] Bohlouli M., Mittas N., Kakarontzas G., Theodosiou T., Angelis L., Fathi M. (2017) Competence assessment as an expert system for human resource management: A mathematical approach. Expert Systems with Applications. 70: 83-102. https://doi.org/10.1016/j.eswa.2016.10.046

[16] Borges G. G., Gratão de Souza, R. C. (2024). Skills development for software engineers: Systematic literature review. Information and Software Technology. 168. https://doi.org/10.1016/j.infsof.2023.107395

[17] Bray, J. R., Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin. Ecological monographs. 27(4): 326-349. https://doi.org/10.2307/1942268

[18] Borcard D., Gillet F., Legendre P. (2011). Numerical ecology with R. New York: Springer. 2. https://doi.org/10.1007/978-1-4419-7976-6

[19] Bruhn J. G. (1974). Human ecology: A unifying science?. Human Ecology. 2: 105-125. https://doi.org/10.1007/BF01558116

[20] Carrington, D., Strooper, P., Newby, S., & Stevenson, T. (2005). An industry/university collaboration to upgrade software engineering knowledge and skills in industry. Journal of Systems and Software, 75(1–2), 29–39.

[21] Colwell, R. K. (2009). Biodiversity: concepts, patterns, and measurement. The Princeton Guide to Ecology. Princeton Univ. Press.

[22] Creighton O., Singer M. (2008). Who leads our future leaders? On the rising relevance of social competence in software development. 1st International Workshop on Leadership and Management in Software Architecture (LMSA '08), International Conference on Software Engineering (ICSE ’08). Leipzig, Germany,

May 11, 2008.

[23] Curtis, J. T., McIntosh, R. P. (1951) An upland forest continuum in the prairie-forest border region of Wisconsin. Ecology. 32(3): 476-496.

[24] De Cáceres M., Legendre P. (2009) Associations between species and groups of sites: indices and statistical inference. Ecology. 90(12): 3566-3574.

[25] De Cáceres M., Legendre P., Moretti M. (2010) Improving indicator species analysis by combining groups of sites. Oikos. 119(10): 1674-1684. https://doi.org/10.1111/j.1600-0706.2010.18334.x

[26] De Cáceres M., Legendre P., Wiser S. K., Brotons L. (2012) Using species combinations in indicator value analyses. Methods in Ecology and Evolution. 3(6): 973-982. https://doi.org/10.1111/j.2041-210X.2012.00246.x

[27] Dhungana D., Groher I., Schludermann E., Biffl S. (2010). Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. 4th European Conference on Software Architecture: Companion Volume (ECSA '10). Copenhagen, Denmark, August 23 - 26, 2010. https://doi.org/10.1145/1842752.1842777

[28] De Mauro, A., Greco, M., Grimaldi, M., & Ritala, P. (2018). Human resources for Big Data professions: A systematic classification of job roles and required skill sets. Information Processing & Management, 54(5), 807–817. https://doi.org/10.1016/j.ipm.2017.05.004

[29] De Morais Leca M., De Souza Santos R. (2025). Curious, critical thinker, empathetic, and ethically responsible: Essential soft skills for data scientists in software engineering. 47th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS 2025). Ottawa, ON, Canada, April 27 – May 3, 2025. 10.1109/ICSE-SEIS66351.2025.00021

[30] Dikert K., Paasivaara M., Lassenius C. (2016) Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software. 119: 87-108. https://doi.org/10.1016/j.jss.2016.06.013.

[31] Dobslaw F., Angelin K., Öberg L.-M., Ahmad A. (2023) The Gap between Higher Education and the Software Industry — A Case Study on Technology Differences. 5th European Conference on Software Engineering Education (ECSEE '23). Seeon/Bavaria Germany June 19 - 21 2023. https://doi.org/10.1145/3593663.3593690

[32] Dorofeev K. (2020). Skill-based engineering in industrial automation domain: Skills modeling and orchestration. 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). Seoul, South Korea, October 5–11, 2020.

[33] Downey J., Ali Babar M. (2008). On identifying the skills needed for software architects. 1st International Workshop on Leadership and Management in Software Architecture (LMSA '08), International Conference on Software Engineering (ICSE '08). Leipzig, Germany, May 11, 2008.

[34] Duarte, C. H. C. (2017). Productivity paradoxes revisited: Assessing the relationship between quality maturity levels and labor productivity in Brazilian software companies. Empirical Software Engineering, 22(2), 699–741. https://doi.org/10.1007/s10664-016-9453-5

[35] Dufrêne M.,Legendre, P. (1997), Species Assemblages and Indicator Species: The Need for a Flexible Asymmetrical Approach. Ecological Monographs, 67: 345-366. https://doi.org/10.1890/0012-9615(1997)067[0345:SAAIST]2.0.CO;2

[36] Fucci, D., Turhan, B., Juristo, N., Dieste, O., Tosun-Misirli, A., & Oivo, M. (2015). Towards an operationalization of test-driven development skills: An industrial empirical study. Information and Software Technology, 68: 82–97. Towards an operationalization of test-driven development skills: An industrial empirical

study

[37] Gafni, R., Aviv, I., Kantsepolsky, B., Sherman, S., Rika, H., Itzkovich, Y., & Barger, A. (2024). Objectivity by design: The impact of AI-driven approach on employees' soft skills evaluation. Information and Software Technology, 170. https://doi.org/10.1016/j.infsof.2024.107430

[38] Galster, M., Mitrovic, A., Malinen, S., Holland, J., & Peiris, P. (2023). Soft skills required from software professionals in New Zealand. Information and Software Technology. 160. https://doi.org/10.1016/j.infsof.2023.107232

[39] Gauch H. G. (1982). Multivariate analysis in community ecology (No. 1). Cambridge University Press. https://doi.org/10.1017/CBO9780511623332

[40] Giabelli, A., Malandri, L., Mercorio, F., & Mezzanzanica, M. (2022). GraphLMI: A data-driven system for exploring labor market information through graph databases. Multimedia Tools and Applications, 81, 3061–3090. https://doi.org/10.1007/s11042-020-09115-x

[41] Gower J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika, 53(3-4): 325-338.

[42] Gren, L., Knauss, A., & Stettina, C. J. (2018). Non-technical individual skills are weakly connected to the maturity of agile practices. Information and Software Technology, 99: 11–20. https://doi.org/10.1016/j.infsof.2018.02.006

[43] Handali J. P., Schneider J., Dennehy D., Hoffmeister B., Conboy K., Becker J. (2020). Industry Demand for Analytics: A Longitudinal Study. 28th European Conference on Information Systems (ECIS2020), A Virtual AIS Conference. Marrakech. Morocco. June 15-17, 2020.

[44] Heldal, R., Nguyen, N.-T., Moreira, A., Lago, P., Duboc, L., Betz, S., Coroamă, V. C., Penzenstadler, B., Porras, J., Capilla, R., Brooks, I., Oyedeji, S., & Venters, C. C. (2024). Sustainability competencies and skills in software engineering: An industry perspective. Journal of Systems and Software. 211. https://doi.org/10.1016/j.jss.2024.111978

[45] Heggen S., Cody M. (2018). Hiring Millennial Students as Software Engineers: A Study in Developing Self-Confidence and Marketable Skills. International Workshop on Software Engineering Education for Millennials (SEEM 2018). Gothenburg, Sweden, June 2, 2018.

[46] Holtkamp, P., Jokinen, J. P. P., & Pawlowski, J. M. (2015). Soft competency requirements in requirements engineering, software design, implementation, and testing. Journal of Systems and Software, 101: 136–146. https://doi.org/10.1016/j.jss.2014.12.010

[47] Jørgensen, M., Bergersen, G. R., & Liestøl, K. (2021). Relations between effort estimates, skill indicators, and measured programming skill. IEEE Transactions on Software Engineering, 47(12): 2892–2906. 10.1109/TSE.2020.2973638

[48] Kapitsaki G., Chatzivasili L., Papoutsoglou M. Galster M. (2024). An Exploratory Study on Soft Skills present in Software Positions in Cyprus: a quasi-Replication Study. 18th International Symposium on Empirical Software Engineering and Measurement (ESEM '24). Barcelona, Spain, October 24 - 25, 2024. https://doi.org/10.1145/3674805.3686681

[49] Kim B. R., Shin J., Guevarra R., Lee J. H., Kim D. W., Seol K. H., Lee J. H., Kim H. B., Isaacson R. (2017). Deciphering Diversity Indices for a Better Understanding of Microbial Communities. Journal of microbiology and biotechnology. 27(12):2089–2093. https://doi.org/10.4014/jmb.1709.09027

[50] Koleff P., Gaston K.J., Lennon J.J. (2003). Measuring beta diversity for presence–absence data. Journal of Animal Ecology. 72: 367-382. https://doi.org/10.1046/j.1365-2656.2003.00710.x

[51] Kommisetty P., Dileep V. (2022). Leading the Future: Big Data Solutions, Cloud Migration, and AI-Driven Decision-Making in Modern Enterprises. Educational Administration: Theory and Practice. 29(3): 352 – 364 Doi: 10.53555/kuey.v28i03.7290

[52] Legendre P. (2012) Numerical ecology. Elsevier. 3rd Edition, 24.

[53] Liang J. T., Zimmermann T., Ford D. (2022). Towards mining OSS skills from GitHub activity. 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '22). Pittsburgh, Pennsylvania, May 21–29, 2022. https://doi.org/10.1145/3510455.3512772 .

[54] Loufek S. B., Santos F., Trinkenreich B. (2025). Beyond the Job Posting: What Hiring Managers Seek in Entry-Level Software Engineering Candidates. 19th International Symposium on Empirical Software Engineering and Meas-urement (ESEM '25). Honolulu, Haweii, September, 28 - October 3, 2025

[55] Lovaglio, P. G., Cesarini, M., Mercorio, F., & Mezzanzanica, M. (2018). Skills in demand for ICT and statistical occupations: Evidence from web-based job vacancies. Statistical Analysis and Data Mining: The ASA Data Science Journal, 11(2), 57–68. https://doi.org/10.1002/sam.11372

[56] Madampe K., Hoda R., Grundy J. (2020) A Multi-dimensional Study of Requirements Changes in Agile Software Development Projects. arXiv:2012.03423.

[57] Magurran A. E. (2003). Measuring biological diversity. John Wiley & Sons.

[58] Mantel N. (1697). The detection of disease clustering and a generalized regression approach Cancer research. 27 (2). pp. 209-220.

[59] Matturro G., Raschetti F., Fontán C. (2019) A Systematic Mapping Study on Soft Skills in Software Engineering. Journal of Universal Computer Science. 25(1): 16-41. https://doi.org/10.3217/jucs-025-01-0016

[60] McGuinness S., Pouliakas K., Redmond P. (2018) Skills Mismatch: Concepts, Measurement and Policy Approaches. Journal of Economic Surveys. 32(4): 985-1015. https://doi.org/10.1111/joes.12254

[61] Mens T., Claes M., Grosjean P. (2014). ECOS: Ecological studies of open source software ecosystems. Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE'14). Antwerp, Belgium, 2014. doi: 10.1109/CSMR-WCRE.2014.6747205.

[62] Minchin P.R. (1987) An evaluation of the relative robustness of techniques for ecological ordination. Vegetatio 69: 89–107. https://doi.org/10.1007/BF00038690

[63] Misic, M. M., & Graf, D. K. (2004). Systems analyst activities and skills in the new millennium. Journal of Systems and Software, 71(1–2): 31–36. https://doi.org/10.1016/S0164-1212(02)00124-3

[64] Montandon J. E., Politowski C., Silva L. L., Valente M. T., Petrillo F., Guéhéneuc, Y.-G. (2021) What skills do IT companies look for in new developers? A study with Stack Overflow jobs. Information and Software Technology. 129. https://doi.org/10.1016/j.infsof.2020.106429

[65] Montandon J. E., Silva L.L., Valente, M.T. (2019) Identifying experts in software libraries and frameworks among GitHub users. 16th International Conference on Mining Software Repositories (MSR'19). Montreal. Canada. May 26 - 27 2019. https://doi.org/10.1109/MSR.2019.00054

[66] Morris E. K., Caruso T., Buscot F., Fischer M., Hancock C., Maier T. S., Meiners T., Müller C., Obermaier E., Prati D., Socher S. A., Sonnemann I,, Wäschke N., Wubet T., Wurst S., Rillig, M. C. (2014) Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories. Ecology and evolution, 4(18), 3514-3524. https://doi.org/10.1002/ece3.1155

[67] Nagendra H. (2002) Opposite trends in response for the Shannon and Simpson indices of landscape diversity. Applied geography. 22(2): 175-186. https://doi.org/10.1016/S0143-6228(02)00002-4

[68] Orsted M. (2000). Software development engineer in Microsoft: A subjective view of soft skills required.

International Conference on Software Engineering (ICSE 2000 the New Millennium). Limerick, Ireland, June 9, 2000. https://doi.org/10.1145/337180.337445

[69] Pedota M., Grilli L., Piscitello L. (2023) Technology adoption and upskilling in the wake of Industry 4.0. Technological Forecasting and Social Change. 187: 122085. https://doi.org/10.1016/j.techfore.2022.122085

[70] Power M. E., Tilman D., Estes J. A., Menge B. A., Bond W. J., Mills L. S., Daily G., Castilla J. C., Lubchenco J., Paine, R. T. (1996) Challenges in the Quest for Keystones. BioScience 46(8): 609-62. https://doi.org/10.2307/1312990

[71] Rahhal, I., Kassou, I., & Ghogho, M. (2024). Data science for job market analysis: A survey on applications and techniques. Expert Systems with Applications, 251, 124101. https://doi.org/10.1016/j.eswa.2024.124101

[72] Rikala P., Braun G., Järvinen M., Stahre J, Hämäläinen R. (2024) Under-standing and measuring skill gaps in Industry 4.0 — A review. Technological Forecasting and Social Change. 201: 123206. https://doi.org/10.1016/j.techfore.2024.123206

[73] Rose, J., Pedersen, K., Hosbond, J. H., & Kræmmergaard, P. (2007). Management competences, not tools and techniques: A grounded examination of software project management at WM-data. Information and Software Technology, 49(6): 605–624. https://doi.org/10.1016/j.infsof.2007.02.005

[74] Santos F. (2023). Skill recommendation for new contributors in open-source software. 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). Melbourne, Australia, May 14–20, 2023. 10.1109/ICSE-Companion58688.2023.00084

[75] Setúbal M., Conte T., Kalinowski M., Araújo A. A. (2024) Investigating the online recruitment and selection journey of novice software engineers: Anti-patterns and recommendations. Empirical Software Engineering. 29(5). https://doi.org/10.1007/s10664-024-10498-w

[76] Smith B., Wilson J. B. (1996). A consumer's guide to evenness indices. Oikos. 76(1): 70-82. https://doi.org/10.2307/3545749

[77] Spencer, D. (2009). Card Sorting: Designing Usable Categories, Rosenfeld Media.

[78] Torchiano M., Coppola R., Vetrò A., Musaj X. (2025). The Impact of Team Diversity in Agile Development Education. 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25). Trondheim, Norway, June 23 - 28, 2025. https://doi.org/10.1145/3696630.3728696

[79] Turley, R. T., & Bieman, J. M. (1995). Competencies of exceptional and nonexceptional software engineers. Journal of Systems and Software, 28(1): 19–38. https://doi.org/10.1016/0164-1212(94)00078-2

[80] Vassakis K., Petrakis E., Kopanakis I. (2018). Big Data Analytics: Applications, Prospects and Challenges. Lecture Notes on Data Engineering and Communications Technologies. 10. Springer, Cham. https://doi.org/10.1007/978-3-319-67925-9_1

[81] Venters C. C., Capilla R., Nakagawa E. Y., Betz S., Penzenstadler B., Crick T., Brooks, I. (2023). Sustainable software engineering: Reflections on advances in research and practice. Information and Software Technology. 164. https://doi.org/10.1016/j.infsof.2023.107316

[82] Wang Z., Wang Y., Redmiles D. (2018). Competence-Confidence Gap: A Threat to Female Developers' Contribution on GitHub. 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS 2018). Gothenburg, Sweden, May 27 – June 3, 2018.

[83] Webster, J., and Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly, 26(2), xiii–xxiii

[84] Whittaker R. H. (1960). Vegetation of the Siskiyou Mountains, Oregon and California. Ecological mono-

graphs, 30(3): 279-338. https://doi.org/10.2307/1943563

[85] Wodnicka M. (2024) Skills Gap and New Technologies: Bibliometric Analysis. Procedia Computer Science. 246: 3430-3436. https://doi.org/10.1016/j.procs.2024.09.214

[86] Zanatta A. L., Machado L., Steinmacher I. (2018). Competence, collaboration, and time management: Barriers and recommendations for crowdworkers. 5th International Workshop on Crowd Sourcing in Software Engineering (CSI-SE), 40th International Conference on Software Engineering (ICSE ‘18). Gothenburg, Sweden, May 27 – June 3, 2018.

[87] Zieris F., Prechelt L. (2014). On knowledge transfer skill in pair programming. 8th International Symposium on Empirical Software Engineering and Measurement (ESEM '14). Torino, Italy, September 18 - 19, 2014. https://doi.org/10.1145/2652524.2652529.

[88] Zieris F., Prechelt L. (2021). Two elements of pair programming skill. 43rd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '21). Virtual Event, Spain, May 25–28, 2021. https://doi.org/10.1109/ICSE-NIER52604.2021.00019

[89] Gregor, S. (2006). "The Nature of Theory in Information Systems." MIS Quarterly, 30(3), pp. 611-642.

[90] Avison, D., and Malaurent, J. (2014). "Is theory king?: questioning the theory fetish in information systems." Journal of Information Technology, 29(4), pp. 327-336.