10-9-2023

Low-Coders, No-Coders, and Citizen Developers in Demand: Examining Knowledge, Skills, and Abilities Through a Job Market Analysis

Björn Binzer
University of Hagen, Chair of Information Management, Hagen, Germany, bjoern.binzer@fernuni-hagen.de

Till J. Winkler
University of Hagen, Chair of Information Management, Hagen, Germany; Copenhagen Business School, Department of Digitalization, Frederiksberg, Denmark, till.winkler@fernuni-hagen.de

Low-Coders, No-Coders, and Citizen Developers in Demand: Examining Knowledge, Skills, and Abilities Through a Job Market Analysis

Research Paper

Björn Binzer[1] and Till J. Winkler[1,2]

[1] University of Hagen, Chair of Information Management, Hagen, Germany
[2] Copenhagen Business School, Department of Digitalization, Frederiksberg, Denmark
{bjoern.binzer,till.winkler}@ fernuni-hagen.de

Abstract. The emergence of low-code/no-code (LCNC) platform technologies and the resulting increase in citizen development programs are facilitating the democratization of the design, development, and deployment of digital solutions. *Citizen developers*, non-technical employees who leverage LCNC platforms, are at the heart of this trend. While many firms perceive LCNC and citizen development as a crucial component of their digital transformation strategy, little is known about the evolving roles in this field or the necessary knowledge, skills, and abilities (KSA). To address this knowledge gap, we processed 113,106 job postings published on Indeed.com. Our topic modeling methodology identified 34 KSA topics and classified them into the three domains *platform*, *business*, and *technology*. We contribute to research by empirically demonstrating which competencies are required to successfully work in the LCNC field. Our findings can guide individual professionals and organizations alike.

Keywords: Low-Code/No-Code Skills, Citizen Developer, Topic Modeling, Job Postings Analysis

1 Introduction

~ "(...) the future of coding is no coding at all." ~ Chris Wanstrath, former CEO at GitHub (Swinhoe, 2017; Peterson, 2017)

While driving the digital transformation is a top priority for organizations of all types (Matt et al., 2016), only few initiatives achieve their intended goals (Forth et al., 2020; Wade & Shan, 2020). Firms struggle to meet the ever-increasing demands for digital solutions and automation whilst facing a shortage of digital talent (Breaux & Moritz, 2021; Carroll & Maher, 2023). To address this challenge, the concept of citizen development has gained momentum in practice and academia. Citizen development empowers non-IT professionals within organizations to design, develop, and deploy lightweight digital solutions to solve specific work-related problems based on IT tools that are provided, recommended, or at least tolerated by core IT units (Binzer & Winkler, 2022). These IT tools are popularly referred to as low-code or no-code (LCNC) platforms (e.g., Carroll et al., 2021). Employees who leverage these LCNC platforms but work outside of IT functions are often called *citizen developers*. Digital solutions built on LCNC platforms can range from small business transactional applications (e.g., booking management) to more sophisticated enterprise-wide solutions (Johannessen & Davenport, 2021). Notably, business process and workflow automation are also closely linked to LCNC and citizen development (Johannessen & Davenport, 2021; Lebens et al., 2021; Eggers et al., 2023).

According to recent market analyses, the global LCNC platform market size is expected to reach between US$ 94.8 and US$ 169.2 billion by 2030 (Fortune Business Insights, 2022; GreyViews, 2023). More than two-thirds of all new applications in organizations are likely to be based on low-code platforms by 2025 (Gartner, 2021). It is thus hardly surprising that the number of citizen developers is rising and expected to exceed the number of professional developers by a factor of four in 2024 (Drakos & Wong, 2021). As evidenced by industry reports, firms are actively training their workforce to become LCNC users (e.g., Kappeyne, 2021). Moreover, recent job postings indicate that firms are explicitly seeking candidates with LCNC qualifications in the job market. In light of these developments, the significance of LCNC and citizen development in the broader context of digital transformation becomes evident.

Despite the increasing relevance of the phenomenon, the literature is still limited in terms of the knowledge and skillset of citizen developers (Binzer & Winkler, 2022). While citizen developers are often referred to as non-technical business users that leverage LCNC platforms to address operational pain points and inefficiencies (Carroll et al., 2021), they have also been characterized as tech-savvy coders with specialized IT expertise (Steele, 2021). Contrasting with this, McKendrick (2017) noted that the majority of citizen developers are actually power users and professional developers embedded within business units. Given these conflicting portrayals and the limited exploration of the required knowledge, skills, and abilities (KSA) in prior studies, there exists a gap in understanding the profiles of the evolving roles in the context of LCNC. No research has yet placed emphasis on unraveling KSAs that are demanded from individual LCNC users. Building on this observation, we aim to contribute to research by addressing the following research question:

RQ: What knowledge, skills, and abilities do employers seek in prospective employees for effective utilization of low-code and no-code technologies?

We pursued this question by processing a large collection of crawled job postings published at Indeed.com. Job postings are useful sources of information because they represent the most important and valued characteristics about a candidate (i.e., KSAs) from the viewpoint of organizations (Walsh et al., 1975; Todd et al., 1995; Cegielski & Jones-Farmer, 2016). As a result, we identified 34 KSA topics with our topic modeling approach, classified into three domains: platform, business, and technology. Our findings suggest that organizations currently require a broad spectrum of business and technical expertise to successfully approach the LCNC and citizen development field.

The remainder of this study is structured as follows: The next section describes related work. Section 3 illustrates our research methodology. Next, in Section 4, we present our results. In Section 5, we discuss our work by addressing implications and limitations before highlighting our theoretical and practical contributions in Section 6.

2 Background

2.1 Low-Code/No-Code and Citizen Development

Low-code and *citizen development* are practice-driven phenomena that are closely intertwined. While the term *low-code development platform* was first introduced by Forrester Research in 2014 (Richardson & Rymer, 2014), the term *citizen development* has already been coined in 2009 by Gartner (Weisinger, 2011). Since then, the innovativeness and relevance of the concepts have been debated (Bock & Frank, 2021; Di Ruscio et al., 2022). However, researchers also acknowledge the new emerging potential (Tisi et al., 2019; Phalake & Joshi, 2021), which is occasionally justified by recent advances in technologies such as artificial intelligence or machine learning (Carroll & Maher, 2023). As commonly defined, low-code platforms represent cloud-based environments that provide features such as advanced graphical user interfaces, visual representations, drag-and-drop functionality, reusable components, and declarative languages (e.g., Tisi et al., 2019; Sahay et al., 2020).

The basic idea behind both concepts is to hide the complexity of coding through a high level of abstraction while ensuring full functionality of the code (Sahay et al., 2020; Carroll et al., 2021). This, in turn, leads to an enhanced ease-of-use allowing subject matter experts with minimal or no coding skills to design, develop, and deploy their own lightweight applications (Sahay et al., 2020; Carroll & Maher, 2023). In this context, and to reinforce the idea of empowering employees without any IT-knowledge, the term *no-code* has recently emerged. However, low- and no-code are conceptually closely related, which is why they are regularly used interchangeably or combined under LCNC (e.g., Carroll & Maher, 2023). Against this backdrop, citizen development can be understood as an organizational strategy, that strives to democratize digital solution delivery through the large-scale adoption of LCNC (Binzer & Winkler, 2022).

More recently, research on citizen development and LCNC has gained increasing momentum. Binzer & Winkler (2022) conducted a multivocal literature review to identify dominant themes and propose an agenda for future citizen development research. Other studies investigated organizational factors that influence citizen development adoption decisions (Hoogsteen & Borgman, 2022), explored potential challenges that may occur during the implementation and ongoing use of LCNC (Prinz et al., 2022), and identified the key building blocks for establishing a citizen development program as part of a digital transformation initiative (Carroll & Maher, 2023).

2.2 Previous Research Analyzing Job Postings

Management research has long recognized job postings as an important source of information for the analysis of occupational qualifications and skills (Walsh et al., 1975; Todd et al., 1995). Online job platforms such as Indeed and Monster, and the emergence of advanced text-mining algorithms, recently fueled the interest of researchers to study increasingly large collections of job postings. Such an approach bears the potential to identify relevant KSAs that employers value most in a particular domain. In this context, text-mining research can be classified into three categories (Pejic-Bach et al., 2020): 1) developing classification schemes, 2) improving matching quality with potential candidates, and 3) exploring skillsets and developing job profiles.

Since our study falls into the third category, we focus on previous studies within this research stream in the following. Debortoli et al. (2014), for example, developed a taxonomy of big data and business intelligence competencies by analyzing a corpus of 1,807 job postings. Studies from the recent past demonstrate that scholars similarly approached related fields such as Industry 4.0 (Pejic-Bach et al., 2020), analytics (Handali et al., 2020), and data science (Michalczyk et al., 2021). While their applied methodological approaches are partly different, topic modeling techniques such as Latent Dirichlet Allocation (LDA) are particularly popular (e.g., Michalczyk et al., 2021). The basic assumption behind topic modeling is that text documents can be expressed in terms of a specific number of underlying topics that the algorithm is striving to uncover (Řehůřek & Sojka, 2010). In the context of job postings, these identified topics are then believed to represent required job qualifications and knowledge domains.

3 Methodology

By drawing on prior text-mining research, in particular on the works of Debortoli et al. (2016) and Michalczyk et al. (2021), we analyzed current job market demands around LCNC and citizen development. We therefore have chosen a multi-stage methodological approach as illustrated in our topic modeling pipeline in Figure 1. The subsequent sections explain the steps of our topic modeling pipeline.

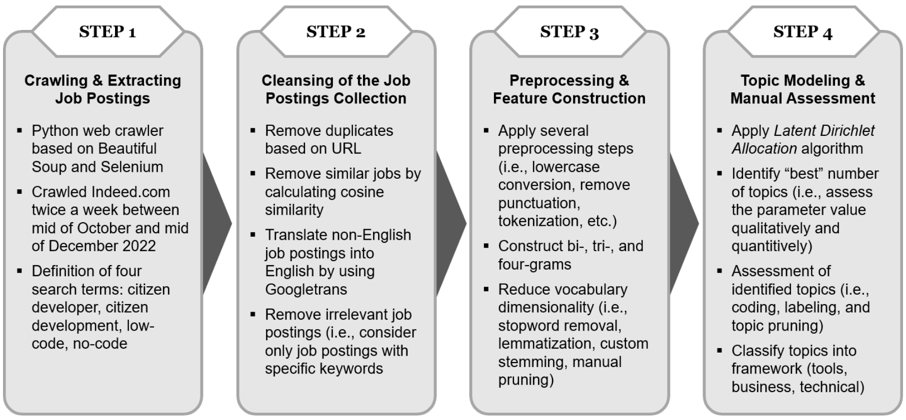

Figure 1. Topic Modeling Pipeline

3.1 Crawling, Collecting, and Extracting the Job Postings

We started our research by seeking relevant and appropriate job titles in the context of LCNC and citizen development. By screening common online job platforms and examining diverse combinations of keywords, we came up with a set of six search terms that delivered satisfying search results. Accordingly, our set of search terms consists of *citizen developer, citizen development*, *low-code*, *no-code, low code*, and *no code*.

To retrieve online job postings, we developed a web crawler in Python using the packages *Beautiful Soup* (Richardson, 2022) for the purpose of web scraping and *Selenium* (2022) as web driver and browser automation framework. As job platform, we selected *Indeed.com* which is one of the largest platforms for job seekers and recruiters from all over the world (Michaeli, 2023; Polner, 2023). Previous studies have demonstrated the suitability and usefulness of Indeed.com for research (e.g., Michalczyk et al., 2021; Sainju et al., 2021; Kortum et al., 2022). To gain a global perspective and to avoid any country-specific bias from crawling only one online job platform, we considered all 62 subdomains of Indeed.com. [1] By manually querying our six pre-defined search terms, we found that Indeed.com's search engine also considered the non-hyphenated counterparts ('low code' and 'no code') when searching for the hyphenated words ('low-code' and 'no-code'), and vice versa. Thus, we limited our set of search terms to four (*citizen developer*, *citizen development*, *low-code*, *no-code*). All four search terms have been queried individually and in quotation marks and have been applied to the full text of the job postings. As job postings are highly volatile and deleted once a job is filled, we followed the suggestion of Michalczyk et al. (2021) and crawled Indeed.com twice a week. As a result, we ran our crawler sixteen times between October and December 2022, which resulted in extracting a total of 113,106 job postings.

Our automated data extraction process included several meta data such as job title, location, and information about the recruiting organization. However, in line with previous research, our main focus was on extracting the full-text job description to capture the descriptive nature of the tasks, responsibilities, and requirements. These reflect the hiring organization's expectations of the job role and the desired qualifications of a potential candidate (Todd et al., 1995; Cegielski & Jones-Farmer, 2016).

3.2 Cleansing and Preparing of Extracted Job Postings

After completing our crawling and extraction activities, we merged our separate datasets into one large dataset. Overall, we relied on the Python package of *pandas* (pandas, 2022) for data preparation, data analysis, and data visualization. As we crawled Indeed.com twice a week, we started our cleansing process by removing redundant exact matches based on the extracted URL of the job postings. As a result, our dataset of job postings was reduced from 113,106 to 21,085. However, since some of the URLs were parameterized and therefore inherently unique, removing duplicates based on the URL was only of partial utility. Therefore, we additionally assessed the similarity of job postings by calculating the cosine similarity (Manning et al., 2009; Jurafsky & Martin, 2023). For calculation, we relied on the in-built *pairwise.cosine_similarity* function of *Scikit-learn* (Pedregosa et al., 2011). Whilst testing different thresholds, we identified a threshold of 0.9 as best-fitting for our dataset. This means that all pairs with a similarity of more than 90% were treated as duplicates and one of them was removed randomly, resulting in a set of 8,902 unique job posting documents.

While text mining studies often tend to filter out non-English documents to rely on a common language, we aimed to reflect a global perspective. Thus, we first identified

[1] https://www.indeed.com/worldwide

the language of each job posting and then translated all non-English job postings into English using the Python library *Googletrans* (2021). Our cross-checking of translations yielded sufficient results to accept this approach as reasonable.

Lastly, while we became more familiar with our dataset, we noticed job postings that were not relevant to our research endeavor. After manual investigation, we identified several false-positive phrases such as *'no Code of Conduct'*, '*jobtarget code: no code*', '*low code locations*', or '*dress no code*'. Moreover, similar to Debortoli et al. (2014), we observed that our search terms often only appeared in the company description (e.g., '*we are building a no-code solution*'), indicating that these firms were not looking for candidates that are intended to work with LCNC, but for a communications specialist or customer success manager, for example. We therefore decided to consider only job postings that included specific keywords in their job title. These keywords comprise *low-code*, *no-code*, *citizen dev**, and the names of the 25 most famous LCNC platforms [2] (Richardson & Rymer, 2014; Sahay et al., 2020; Bratincevic & Koplowitz, 2021). Following this procedure narrowed down our collection of job postings to a total of 1,632 unique documents that remained for further analysis.

3.3 Preprocessing and Feature Construction

Next, we performed the feature construction step of our topic modeling pipeline. Since traditional data mining techniques are not designed to deal directly with textual data, feature construction serves to convert raw textual data into mathematical representations (Brank et al., 2011; Verdonck et al., 2021). In this study, each job posting is represented as a vector of frequencies of each term, thus assuming the principles of the vector space model (Manning et al., 2009; Jurafsky & Martin, 2023). However, since this approach discards the words' grammatical ordering and focus solely on the frequencies of words within a document (i.e., bag-of-words assumption), we followed the procedure of previous text mining studies (e.g., Wallach, 2006; Debortoli et al., 2016; Michalczyk et al., 2021; Jurafsky & Martin, 2023) and constructed bi-, tri-, and four-grams in addition to unigrams (i.e., single words). Thus, we take contextual information into account, resulting in the inclusion of more meaningful features [3] such as *software_development* (bi-gram) or *analytical_problem_solving* (tri-gram).

Before constructing the bi-, tri-, and four-gram features, we applied several common data preprocessing steps (Silva et al., 2021). First, we converted all text to lowercase. We furthermore tokenized the text, removed punctuation, and replaced all non-alphanumeric characters. Second, we removed irrelevant standard stop words such as '*you*', '*this*', and '*any'* using the *NLTK* package (Bird et al., 2009). The resulting vocabulary was then systematically reviewed to correct unambiguous typos such as *developpment*, and to separate words written together, such as *softwaredevelopment*. Next, a lemmatization dictionary (Měchura, 2017) was applied to account for the dictionary form of the

[2] We encountered a lot of specialized job postings (e.g., *power platform* or *mendix developer*).

[3] Following common naming conventions of text-mining research (e.g., Manning et al. (2009); Jurafsky and Martin (2023)), we use the overarching term *feature* for referring to both, single words (i.e., unigrams) and word phrase patterns (i.e., bi-, tri-, and four-grams).

words (Manning et al., 2009). For instance, *analyze* is the lemma of *analyze, analyzes,* and *analyzed*. In addition, because our collection included both American English and British English words (e.g., *analyse* versus *analyze*), we harmonized our vocabulary by converting all spellings to American English. Finally, to reduce the feature dimension of the remaining vocabulary, we normalized similar features by grouping them together under a new custom lemma (Manning et al., 2009; Michalczyk et al., 2021; Silva et al., 2021). For example, we merged *entrepreneurial*, *entrepreneurship*, and *entrepreneur* into the lemma *entrepreneu*, which is still open to human interpretation. While our initial non-processed vocabulary would have contained more than 1.4 million unique features, our described preprocessing steps reduced the amount to a total of 1.1 million.

Next, to further trim the number of uninformative features, we followed the recommendations of previous research (e.g., Grimmer & Stewart, 2013; Debortoli et al., 2014; Carter et al., 2016; Székely & vom Brocke, 2017; Michalczyk et al., 2021). Consequently, we decided to drop all features that are present in less than 1% and in more than 50% of the analyzed job postings. While the lower limit of 1% ensures that a larger number of uncommon and sparse features (e.g., headquarter location) are not included in the subsequent analysis, the upper limit of 50% suppresses the inclusion of prevalent and non-differentiating job posting features (e.g., *experience*, *work*, or *business*). The remaining vocabulary contained 6,199 unique features, which we then manually reviewed to identify and exclude irrelevant features to the field of LCNC and citizen development. This final pruning process is a common and effective technique in topic modeling studies (e.g., Debortoli et al., 2014; Székely & vom Brocke, 2017). Overall, the manual review and cleansing resulted in 1,952 highly relevant features, which were then used as the final dictionary for applying the topic modeling algorithm.

3.4 Topic Modeling & Topic Coding

Following the data preprocessing, we chose to analyze our job postings dataset using *Latent Dirichlet Allocation* (LDA; Blei et al., 2003) as topic modeling approach. LDA is an probabilistic unsupervised machine learning algorithm that strives to discover hidden topics by running through a set of text documents (Blei, 2012). The algorithm generally assumes that 1) each document can be characterized by a mixture of different (latent) topics, and 2) each topic, in turn, can be represented by a certain distribution of features, while each individual feature may occur in more than one single topic (Blei et al., 2003; Airoldi et al., 2008; Debortoli et al., 2016). Prior research has used LDA in various contexts (e.g., Lukins et al., 2010; Chen et al., 2016; Yang et al., 2017), thereby demonstrating LDA's capability to extract semantically meaningful topics from large amounts of text (Debortoli et al., 2016). In this study, we applied the online variational Bayes algorithm (Hoffman et al., 2010) by using *gensim* (Řehůřek & Sojka, 2010).

The most crucial parameter when running the LDA topic modeling is the number of topics to obtain (*num_topic*). When setting *num_ topic* too high, the algorithm might disclose a plenty of minimally distinct and meaningless topics; on the other side, when setting *num_ topic* too low, there is a risk to severely constrain the exploratory potential of the LDA algorithm (Debortoli et al., 2016; Diegmann et al., 2018). Researchers have therefore proposed different approaches to identify the optimal number of topics

(Griffiths & Steyvers, 2004; Cao et al., 2009; Arun et al., 2010; Mimno et al., 2011), but there is no general rule to apply. We followed the recommendations as proposed by Debortoli et al. (2016). Hence, we qualitatively evaluated different numbers of topics (i.e., we trained various models and gradually increased *num_topic* from 10 to 80). While 30 topics describe our dataset imprecisely, 50 topics lead to numerous meaningless topics. Therefore, qualitatively evaluating the meaningfulness of the different models resulted in defining the ideal number of topics at *num_topic* = 38.

Next, we manually assessed and interpreted each topic by searching for patterns and assigning meaningful topic labels based on the associated features. In general, we used the visualization tool *pyLDAvis* for our explorative analysis (Sievert & Shirley, 2014). As suggested by Diegmann et al. (2018), we reviewed not only the most *frequent* features, but also examined the relatively most *salient* (relevant) features. Salient features are those that have the highest frequency within a particular topic compared to all other topics. Similar to previous studies, we discarded five topics that were irrelevant for our context, resulting in 33 relevant topics. Building upon the classification of Todd et al. (1995), we then grouped our labeled topics into the three domains: *platform*, *business*, and *technology*. Here, we applied *pattern-coding* to group our KSA topics into meaningful 'summative labels' (Saldaña, 2021). Apart from that, one topic contained both business-related and technology-related components. Therefore, we split this topic into two (see BR12 and TR7) and report a total of 34 relevant KSA topics in this study.

4 Results

4.1 *Platform-specific* Knowledge, Skills, and Abilities in Demand

As shown in Table 1, our analysis revealed nine platform-specific topics. Most of them point relatively clearly to a specific LCNC platform. For example, we derived topic P1 (Microsoft Power Platform) from its highest-loading features *power_platform*, *microsoft*, and *power_apps*. Features such as *dataverse* and *virtual_agent* supported our assumption. We similarly identified ServiceNow, OutSystems, Mendix, Oracle APEX, Salesforce, Pega, Appian, and SAP as predominant platform topics in our job postings collection. While multiple high-loading features within the topics P1 to P8 indicate their respective topic label, we derived P9 mainly from its highest-loading feature.

4.2 *Business-related* Knowledge, Skills, and Abilities in Demand

Table 2 presents the twelve business-related topics that emerged from our analysis. BR1 and BR2 highlight work and educational experience through high-loading features such as *+_year_work_experience*, *bachelor,* and *master_degree*. Topics BR3 and BR4 refer to interpersonal KSAs. For topic BR3, we interpret the high-loading features *cooperate*, *partner*, *exchange*, and *interdisciplinary* as essential work collaboration, in which professional language skills in English and German are required. BR4 complements this by highlighting appropriate communication skills and problem-solving abilities. Next, topic BR5 relates to the adoption and scaling of LCNC and citizen development, indicated by features such as *adoption*, *ensure*, and *center_excellence*. The following four

Table 1. Identified platform-specific KSA topics in LCNC and citizen development

Topic Label	**High-Loading Features** (excerpt)
P1: Microsoft Power Platform	power_platform, microsoft, power_apps, power_automate, dataverse, canvas, virtual_agent, power_bi, consultant, flow
P2: ServiceNow Platform	servicenow, servicenow_platform, service_management, integrate, workflow, lead, script, module, itil, servicenow_developer
P3: Oracle APEX	sql, database, oracle, programming, apex, web, pl_sql, oracle_apex, server, erp, java, software_development
P4: Salesforce Platform	salesforce, cloud, integrate, lightning, apex, salesforce_developer, practice, component, salesforce_platform, mulesoft, visualforce
P5: OutSystems Platform	outsystems, outsystems_developer, consultant, dutch, learn, budget, agile, challenge, scrum, outystems_platform, consult
P6: Pega Platform	pega, business_process_mangement, process_management, architect, system_architect, pega_system, sap, lead, pega_platform
P7: Appian Platform	appian, innovate, business_process, lead, empower, experience_appian, business_application, business_process_management
P8: Mendix Platform	mendix, scrum, mendix_developer, innovate, challenge, dutch, user, expert, enthusiasm, consultant, experience_mendix
P9: SAP	sap, innovate, management_system, operational, active, sustainable, specialist, challenge, process_technology, audit

topics, "Change Management" (BR6), "Business Analysis & Project Management (BR7), "Consulting" (BR8), and "Business Understanding" (BR9) were derived through their respective high-loading features. Further, due to the features *marketing*, *web_interaction*, and *customer_relationship_management*, we labeled BR10 as "Marketing & Customer Relationship Management". Finally, as topic BR11 contains the features *startup*, *entrepreneu*, and *business_development*, we labeled this topic "Entrepreneurial Mindset". Similarly, the high-loading features *enthusiasm*, *creative*, *creative_problem_solve*, and *creativity* lead us to the assumption to name topic BR12 "Enthusiasm & Creativity". In summary, while topics BR1, BR2, BR3, BR4, BR11 and BR12 highlight skills, abilities, and personal traits, topics BR5 to BR10 can be considered as business functions and knowledge areas in which LCNC and citizen development, as our results reveal, plays an important role.

4.3 *Technology-related* Knowledge, Skills, and Abilities in Demand

We identified 13 technology-related KSA topics as depicted in Table 3. The first topic (TR1) refers to automation. In particular, the highest-loading features *automation*, *process_automation*, and *intelligent_automation* justify our assumption. Next, topic TR2 represents the related field of robotic process automation. While *rpa* itself is a high-loading feature, this topic also includes platform-specific features around RPA (e.g., *uipath*, *blue_prism*, and *automation_anywhere*). The next topic, TR3, was labeled "Data Analytics & Business Intelligence" due to several high-loading features such as *dashboard*, *report*, *sql*, and *business_intelligence*, thus clearly referring to common keywords within these two analytical disciplines. We titled TR4 "Computer Science & Programming Skills" due to the features *computer_science*, *programming*, *software_developer*, *programming_language*, and *learn*. Similarly, the features *scrum*, *agile*, *product_owner*, *agile_method_scrum*, and *scrum_master* led us to the decision to

Table 2. Business-related KSA topics in LCNC and citizen development

Topic Label	**High-Loading Features** (excerpt)
BR1: Professional Work Experience	+_year, two_+_year, two_+, test, deliver, microsoft, user, +_year_experience, +_year_work_experience, power_apps
BR2: Academic Education	master, master_degree, bachelor_master, bachelor_master_degree, bachelor, user, digital_product, function, bank, degree
BR3: Collaboration & Professional Language Skills	consultant, microsoft, cooperate, german, specialist, innovate, english, partner, concept, advice, exchange, interdisciplinary
BR4: Communication & Problem-Solving Ability	lead, application_development, solve, excellent_write, outsystems, problem_solve, communication_skill, analytic_problem_solve
BR5: Adoption & Governance	power_platform, architect, adoption, microsoft, ensure, strategy, center_excellence, cloud, lead, community, stakeholder, governance
BR6: Change Management	change, agile, ensure, user, method, monitor, change_management, analyst, improvement, operation, risk, agile_method
BR7: Business Analysis & Project Management	analyst, manager, business analyst, project manager, user, internal, airtable, learn, project_management, autonomous, agile
BR8: Consulting	consultant, consult, multidisciplinary, analytic_skill, combine, german, architecture, agile, specialist, strategy, senior_consultant
BR9: Business Understanding	architecture, data, security, zerocode, code, understand_business, architecuture_design, meet_business, understand_business_need
BR10: Marketing & Customer Relationship Management	marketing, automation_specialist, google, customer_relationship_management, web_design, web, interaction, master_business
BR11: Entrepreneurial Mindset	english, startup, consultant, internal, programming, fluent, plan, problem_solve, communicate, entrepreneu, business_development
BR12: Enthusiasm & Creativity	web, enthusiasm, software_developer, ict, solve, mobile, creative, creative_problem_solve, creativity, mobile_application, web_app

Table 3. Technology-related KSA topics in LCNC and citizen development

Topic Label	**High-Loading Features** (excerpt)
TR1: Automation	automation, intelligent_automation, learn, process_automation, lead, business_process, artificial_intelligence, machine_learn, deliver
TR2: Robotic Process Automation	rpa, uipath, rpa_developer, power_automate, robot, blue_prism, analyst, automation_anywhere, business_process, artificial_intelligence
TR3: Data Analytics & Business Intelligence	data, report, power_bi, data_analytic, partner, manage, visualization, business_intelligence, data_model, dashboard, sql, aws, data_science
TR4: Computer Science & Programming Skills	appian, programming, challenge, software_developer, creativity, java, collaboration, learn, programming_language, computer_science
TR5: Agile Methodology	mendix_developer, scrum, innovation, dutch, expert, challenge, agile, agile_scrum, product_owner, agile_method_scrum, scrum_master
TR6: Web Development	web, javascript, css, html, frontend, html_css, user, code, framework, practice, api, react, fullstack, development_experience, backend
TR7: Mobile Application Development	web, enthusiasm, software_developer, ict, solve, mobile, creative, creative_problem_solve, creativity, mobile_application, web_app
TR8: User Experience & User Interface	user, interface, user_interface, user_experience, manager, product_manager, plan, bubble, user_experience_design, react_javascript
TR9: DevOps & Continuous Integration/Delivery	continuous, integration, delivery, devops, continuous_integration, cloud, continuous_delivery, agile, quality, azure, programming, git
TR10: Digital Solution Design & Architecture	architect, solution_architect, leadership, lead_team, manage, technical_architect, cloud_solution, architecture_solution, modernization
TR11: Cloud Architecture	microsoft, cloud, azure, integration, microsoft_azure, azure_cloud, azure_service, javascript_typescript, infrastructure, manage, deploy
TR12: Testing & Quality Management	test, test_automation, quality, automate, automation_engineer, framework, qa, automate_test, script, test_automation_engineer, selenium
TR13: Integration Knowledge	odata, outsystems_consultant, integration, degree, solve, complex, web, knowledge_integration, soap_rest, integration_system, soap

label TR5 "Agile Methodology". The subsequent three topics TR6, TR7, and TR8 relate to software development. As TR6 contains the features *html*, *web*, *css*, *code*, and *javascript*, we labeled this topic "Web Development". We identified TR7 as a related field and labeled it "Mobile Application Development". Moreover, we identified topic TR8 as "User Experience & User Interface" and TR9 as "DevOps & Continuous Integration/Delivery" because the highest-loading features represent the same. The architecture-related features in TR10 and TR11 encouraged us to label these topics with "Digital Solution Design & Architecture" and "Cloud Architecture". While TR10 refers to *architect*, *solution_architect*, and *technical_architect*, topic TR11 contains features such as *cloud*, *azure*, and *infrastructure*, thus supporting our assumption. Finally, the features *test*, *quality*, *framework*, *qa*, and *test_automation* underline the labeling of the topic TR12 as "Testing & Quality Management". Lastly, we labeled TR13 "Integration Knowledge" because of its high-loading features *odata*, *integration*, *soap*, *soap_rest*, and *knowledge_integration*. In summary, our 13 technology-related KSA topics highlight the need for automation knowledge (TR1, TR2), analytical skills (TR3), computer science and (agile) software development (TR4, TR5, TR6, TR7), customer-facing applications (TR8), and software practices (TR9, TR12). Moreover, architecture (TR10, TR11) and integration (TR13) seem to play an important role in LCNC.

5 Discussion

Given the lack of empirical research on the knowledge and skillset of individual citizen developers in the area of LCNC, we shed light on much demanded KSAs associated with the field. Our three-pronged framework covering platforms, business, and technology supports the understanding of these KSAs. We identified that LCNC users should have a substantial professional and academic background (BR1, BR2). Consistent with prior findings (e.g., Whitmore, 2021), LCNC users are valued for their creativity and problem-solving abilities. Additionally, as intermediaries between business and IT (e.g., Krejci et al., 2021), successful utilization of LCNC tools relies on collaboration (BR3), effective communication (BR4), and a profound understanding of business functions and activities (BR7, BR9). However, our findings also reveal that employers primarily demand general KSAs, rather than requiring KSAs that are specific to the domain of LCNC technologies. For instance, KSAs in change and project management, consulting, or familiarity with agile methodologies are prevalent and characteristic for various professional IT roles (Todd et al., 1995; McMurtrey et al., 2008; Gallagher et al., 2010). In turn, this raises the discussion on whether and how LCNC-related jobs are distinct from other IT-related professions.

Furthermore, our findings challenge the claims made by numerous LCNC vendors by revealing that LCNC is associated with a wide spectrum of technical expertise. This implies that utilizing LCNC platforms requires specialized technical knowledge or, in some cases, even programming experience (TR3-TR12). Consequently, the notion of non-IT professionals building digital solutions by solely leveraging LCNC platforms is being questioned. This contributes to the ongoing discourse surrounding the extent to which LCNC platforms truly fulfill the envisioned promise of "democratization of IT", highlighting the findings of prior research (e.g., Sahay et al., 2020; Luo et al., 2021;

Hoogsteen & Borgman, 2022). Moreover, our topic modeling approach revealed "Integration" (TR13) as a standalone topic, which is seen as difficult challenge in the LCNC area (Al Alamin et al., 2021). This finding further indicates the presence of diverse LCNC user types, including professionals, semi-professionals, and citizen developers.

Lastly, it is important to note that due to the absence of unified standards (e.g., language), each LCNC platform possesses its own distinctive features and peculiarities that users must acquaint themselves with (Sahay et al., 2020). Our identification of nine distinct platform-specific topics reinforces this observation, underlining the fact that the LCNC job market currently necessitates a profound understanding and proficiency in platform-specific knowledge and skills. This finding, again, contrasts the claims put forth by vendors that LCNC platforms are easy to use without requiring prior knowledge or experience.

We also recognize the limitations of this work. Limitations of our study arise from our applied crawling strategy and the topic modeling methodology itself. We may have missed jobs that are not in scope of our keyword search strategy. For instance, scholars began to use terms such as *zero-code*, *hyperautomation,* or *citizen automation* (Luo et al., 2021; Herm et al., 2022). Second, although we carefully cleaned our dataset by following best practices and previous research recommendations, there is always a risk of bias. Third, the LDA parameter settings we used may have biased the results of our topic modeling approach. Lastly, other scholars may have interpreted, labeled, and grouped the topics differently than we did.

6 Conclusion

In summary, the contribution of this work is two-fold. Our theoretical contribution lies in the empirical investigation of a large dataset of job postings in the area of LCNC and citizen development. Hence, we respond to calls for exploring the nature of LCNC users (Prinz et al., 2021; Binzer & Winkler, 2022). This study is thus the first empirical work that focuses solely on individual LCNC users, thereby extending the literature with a framework of 34 identified KSA topics.

However, since our findings revealed a broad spectrum of demanded KSAs, we encourage future research to examine (1) which types of LCNC users exist, and (2) how these types utilize and leverage the respective LCNC platforms. In addition, our findings emphasize the importance of automation in the context of LCNC and citizen development. Further research on bringing these literature streams into harmony may thus be an important and useful avenue.

From a practical viewpoint, we highlight much demanded KSAs in which individual professionals may upskill in order to successfully leverage LCNC platforms and prepare for new emerging job opportunities. Moreover, organizations can utilize our framework of identified KSAs to effectively identify and evaluate their workforce needs when defining citizen development strategies. On this basis, existing employees can be trained, or new ones be recruited. In addition, academic institutions can use our findings to design innovative course programs for teaching the field of LCNC and citizen development.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E. & Xing, E. P. (2008) Mixed Membership Stochastic Blockmodels. *Journal of Machine Learning Research* **9** (65), 1981–2014.

Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B. & Iqbal, A. (2021) An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. In: *18th International Conference on Mining Software Repositories (MSR)*. IEEE.

Arun, R., Suresh, V., Veni Madhavan, C. E. & Narasimha Murthy, M. N. (2010) On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. In: Hutchison, D., Kanade, T. & Kittler, J. et al. (eds.) *Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference*, Berlin, Heidelberg, pp. 391–402.

Binzer, B. & Winkler, T. J. (2022) Democratizing Software Development: A Systematic Multivocal Literature Review and Research Agenda on Citizen Development. In: Carroll, N., Nguyen-Duc, A., Wang, X. & Stray, V. (eds.) *Proceedings of the 13th International Conference on Software Business (ICSOB)*. Springer, Cham, pp. 244–259.

Bird, S., Loper, E. & Klein, E. (2009) Natural Language Processing with Python. https://www.nltk.org/book/. Accessed: 01.02.23.

Blei, D. M. (2012) Probabilistic topic models. *Communications of the ACM* **55** (4), 77–84.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003) Latent Dirichlet Allocation. *Journal of Machine Learning Research* **3** (Jan), 993–1022.

Bock, A. C. & Frank, U. (2021) Low-Code Platform. *Business & Information Systems Engineering* **63** (6), 733–740.

Brank, J., Mladenić, D. & Grobelnik, M. (2011) Feature Construction in Text Mining. In: *Encyclopedia of Machine Learning*. Springer, Boston, MA, pp. 397–401.

Bratincevic, J. & Koplowitz, R. (2021) The Forrester Wave™: Low-Code Development Platforms For Professional Developers, Q2 2021. https://www.forrester.com/report/The-Forrester-Wave-LowCode-Development-Platforms-For-Professional-Developers-Q2-2021/RES161668. Accessed: 10.02.23.

Breaux, T. & Moritz, J. (2021) The 2021 software developer shortage is coming. *Communications of the ACM* **64** (7), 39–41.

Cao, J., Xia, T., Li, J., Zhang, Y. & Tang, S. (2009) A density-based method for adaptive LDA model selection. *Neurocomputing* **72** (7-9), 1775–1781.

Carroll, N. & Maher, M. (2023) How Shell Fueled a Digital Transformation by Establishing DIY Software Development. *MIS Quarterly Executive* **22** (2), 99–127.

Carroll, N., Móráin, L. Ó., Garrett, D. & Jamnadass, A. (2021) The Importance of Citizen Development for Digital Transformation. *Cutter IT Journal* **34** (3), 5–9.

Carter, J. D., Brown, J. & Rahmani, A. (2016) READING THE HIGH COURT AT A DISTANCE: TOPIC MODELLING THE LEGAL SUBJECT MATTER AND JUDICIAL ACTIVITY OF THE HIGH COURT OF AUSTRALIA, 1903–2015. *UNSW Law Journal* **39** (4), 1300–1354.

Cegielski, C. G. & Jones-Farmer, L. A. (2016) Knowledge, Skills, and Abilities for Entry-Level Business Analytics Positions: A Multi-Method Study. *Decision Sciences Journal of Innovative Education* **14** (1), 91–118.

Chen, C., Huang, E. & Yan, H. (2016) Detecting the association of health problems in consumer-level medical text. *Journal of Information Science* **44** (1), 3–14.

Debortoli, S., Müller, O., Junglas, I. & vom Brocke, J. (2016) Text Mining for Information Systems Researchers: An Annotated Topic Modeling Tutorial. *Communications of the Association for Information Systems* **39**, 110–135.

Debortoli, S., Müller, O. & vom Brocke, J. (2014) Comparing Business Intelligence and Big Data Skills. *Business & Information Systems Engineering* **6** (5), 289–300.

Di Ruscio, D., Kolovos, D., Lara, J. de, Pierantonio, A., Tisi, M. & Wimmer, M. (2022) Low-code development and model-driven engineering: Two sides of the same coin? *Software and Systems Modeling* **21** (2), 437–446.

Diegmann, P., Dreesen, T., Binzer, B. & Rosenkranz, C. (2018) Journey Towards Agility: Three Decades of Research on Agile Information Systems Development. In: *Proceedings of the 39th International Conference on Information Systems (ICIS)*.

Drakos, N. & Wong, J. (2021) *Future of Work Trends: Tinkerers Become Mechanics*.

Eggers, J., Wewerka, J., Viljoen, A. & Krcmar, H. (2023) Supporting Subject Matter Experts as Developers: Towards a Framework for Decentralized Robotic Process Automation Development. In: *Proceedings of the 56th Annual Hawaii International Conference on System Sciences (HICSS)*.

Forth, P., Laubier, R. de, Reichert, T. & Chakaborty, S. (2020) Flipping the Odds of Digital Transformation Success. https://www.bcg.com/de-de/publications/2020/increasing-odds-of-success-in-digital-transformation. Accessed: 10.03.23.

Fortune Business Insights (2022) Low Code Development Platform Market. https://www.fortunebusinessinsights.com/low-code-development-platform-market-102972. Accessed: 17.02.23.

Gallagher, K. P., Kaiser, K. M., Simon, J. C., Beath, C. M. & Goles, T. (2010) The requisite variety of skills for IT professionals. *Communications of the ACM* **53** (6), 144–148.

Gartner (2021) Gartner Says Cloud Will Be the Centerpiece of New Digital Experiences: Global Cloud Revenue to Total $474 Billion in 2022, Up from $408 Billion in 2021. https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences. Accessed: 17.02.23.

Googletrans: Free and Unlimited Google translate API for Python — Googletrans 3.0.0 documentation (2021). https://py-googletrans.readthedocs.io/en/latest/. Accessed: 11.02.23.

GreyViews (2023) Low Code Development Platform Market– Growth | Global Analysis to 2030. https://greyviews.com/reports/low-code-development-platform-market/578. Accessed: 17.02.23.

Griffiths, T. L. & Steyvers, M. (2004) Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* **101** (Suppl 1), 5228–5235.

Grimmer, J. & Stewart, B. M. (2013) Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis* **21** (3), 267–297.

Handali, J., Schneider, J., Dennehy, D., Hoffmeister, B., Conboy, K. & Becker, J. (2020) INDUSTRY DEMAND FOR ANALYTICS: A LONGITUDINAL STUDY. In: *Proceedings of the 28th European Conference on Information Systems (ECIS)*.

Herm, L.-V., Janiesch, C., Helm, A., Imgrund, F., Hofmann, A. & Winkelmann, A. (2022) A framework for implementing robotic process automation projects. *Information Systems and e-Business Management*, 1–35.

Hoffman, M. D., Blei, D. M. & Bach, F. (2010) Online learning for Latent Dirichlet Allocation. In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pp. 856–864.

Hoogsteen, D. & Borgman, H. (2022) Empower the Workforce, Empower the Company? Citizen Development Adoption. In: *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)*.

Johannessen, C. & Davenport, T. H. (2021) When Low-Code/No-Code Development Works — and When It Doesn't. https://hbr.org/2021/06/when-low-code-no-code-development-works-and-when-it-doesnt. Accessed: 16.02.23.

Jurafsky, D. & Martin, J. H. (2023) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. https://web.stanford.edu/~jurafsky/slp3/. Accessed: 09.02.23.

Kappeyne, N. (2021) Citizen software development | Shell Global. https://www.shell.com/energy-and-innovation/digitalisation/news-room/do-it-yourself-software-development-the-power-is-in-your-hand.html. Accessed: 17.02.23.

Kortum, H., Rebstadt, J. & Thomas, O. (2022) Dissection of AI Job Advertisements: A Text Mining-based Analysis of Employee Skills in the Disciplines Computer Vision and Natural Language Processing. In: *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)*.

Krejci, D., Iho, S. & Missonier, S. (2021) Innovating with employees: an exploratory study of idea development on low-code development platforms. *ECIS 2021 Research Papers*.

Lebens, M., J Finnegan, R., C Sorsen, S. & Shah, J. (2021) Rise of the Citizen Developer. *Muma Business Review* **5**, 101–111.

Lukins, S. K., Kraft, N. A. & Etzkorn, L. H. (2010) Bug localization using latent Dirichlet allocation. *Information and Software Technology* **52** (9), 972–990.

Luo, Y., Liang, P., Wang, C., Shahin, M. & Zhan, J. (2021) Characteristics and Challenges of Low-Code Development. In: Lanubile, F. (ed.) *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM, New York, NY, USA, pp. 1–11.

Manning, C., Raghavan, P. & Schütze, H. (2009) Introduction to Information Retrieval. https://nlp.stanford.edu/IR-book/information-retrieval-book.html. Accessed: 08.02.23.

Matt, C., Hess, T., Benlian, A. & Wiesbock, F. (2016) Options for Formulating a Digital Transformation Strategy. *MIS Quarterly Executive* **15** (2).

McKendrick, J. (2017) The Rise of the Empowered Citizen Developer. https://dbta.com/DBTA-Downloads/ResearchReports/THE-RISE-OF-THE-EMPOWERED-CITIZEN-DEVELOPER-7575.pdf. Accessed: 05.08.22.

McMurtrey, M. E., Downey, J. P., Zeltmann, S. M. & Friedman, W. H. (2008) Critical Skill Sets of Entry-Level IT Professionals: An Empirical Examination of Perceptions from Field Personnel. *Journal of Information Technology Education: Research* **7**, 101–120.

Měchura, M. (2017) Machine-readable lists of lemma-token pairs in 23 languages. https://github.com/michmech/lemmatization-lists. Accessed: 08.02.23.

Michaeli, A. (2023) Indeed Cements its Lead as the Most Popular Job Search App. https://appfigures.com/resources/insights/20230203?f=2. Accessed: 13.02.23.

Michalczyk, S., Nadj, M., Maedche, A. & Gröger, C. (2021) Demystifying Job Roles in Data Science: A Text Mining Approach. In: *Proceedings of the 29th European Conference on Information Systems (ECIS)*.

Mimno, D., Wallach, H., Talley, E., Leenders, M. & McCallum, A. (2011) Optimizing Semantic Coherence in Topic Models. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 262–272.

pandas (2022) pandas documentation — pandas 1.5.1 documentation. https://pandas.pydata.org/docs/index.html. Accessed: 30.10.22.

Pedregosa, F., Varoquaux, G. & Gramfort, A. et al. (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* (12), 2825–2830.

Pejic-Bach, M., Bertoncel, T., Meško, M. & Krstić, Ž. (2020) Text mining of industry 4.0 job advertisements. *International Journal of Information Management* **50**, 416–431.

Peterson, B. (2017) The CEO of GitHub, which caters to coders, thinks automation will bring an end to traditional software programming. *Business Insider,* October 11. https://www.businessinsider.com/github-ceo-wanstrath-says-automation-will-replace-software-coding-2017-10. Accessed: 16.02.23.

Phalake, V. S. & Joshi, S. D. (2021) Low Code Development Platform for Digital Transformation. In: *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. Springer, Singapore, pp. 689–697.

Polner, E. (2023) Best Job Search Websites. https://www.thebalancemoney.com/top-best-job-websites-2064080. Accessed: 13.02.23.

Prinz, N., Huber, M., Riedinger, C. & Rentrop, C. (2022) Two Perspectives of Low-Code Development Platform Challenges – An Exploratory Study. In: *Proceedings of the 26th Pacific Asia Conference on Information Systems (PACIS)*.

Prinz, N., Rentrop, C. & Huber, M. (2021) Low-Code Development Platforms – A Literature Review. In: *Proceedings of the 27th Americas Conference on Information Systems (AMCIS)*.

Řehůřek, R. & Sojka, P. (2010) Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50.

Richardson, C. & Rymer, J. (2014) New Development Platforms Emerge For Customer-Facing Applications. https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411. Accessed: 15.02.23.

Richardson, L. (2022) Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Accessed: 23.10.22.

Sahay, A., Indamutsa, A., Di Ruscio, D. & Pierantonio, A. (2020) Supporting the understanding and comparison of low-code development platforms. In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, pp. 171–178.

Sainju, B., Hartwell, C. & Edwards, J. (2021) Job satisfaction and employee turnover determinants in Fortune 50 companies: Insights from employee reviews from Indeed.com. *Decision Support Systems* **148**, 113582.

Saldaña, J. (2021) *The coding manual for qualitative researchers,* 4e. SAGE, Los Angeles, London, New Delhi, Singapore, Washington DC, Melbourne.

Selenium (2022) The Selenium Browser Automation Project. https://www.selenium.dev/documentation/. Accessed: 11.02.23.

Sievert, C. & Shirley, K. (2014) LDAvis: A method for visualizing and interpreting topics. In: Chuang, J., Green, S., Hearst, M., Heer, J. & Koehn, P. (eds.) *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 63–70.

Silva, C. C., Galster, M. & Gilson, F. (2021) Topic modeling in software engineering research. *Empirical Software Engineering* **26** (6).

Steele, T. (2021) Democratizing Innovation: Low-Code / No-Code Platforms & The Citizen Developer. https://www.capco.com/intelligence/capco-intelligence/democratizing-innovation. Accessed: 17.02.23.

Swinhoe, D. (2017) GitHub CEO: "The future of coding is no coding at all". https://www.idgconnect.com/article/3578431/github-ceo-the-future-of-coding-is-no-coding-at-all.html. Accessed: 17.02.23.

Székely, N. & vom Brocke, J. (2017) What can we learn from corporate sustainability reporting? Deriving propositions for research and practice from over 9,500 corporate sustainability reports published between 1999 and 2015 using topic modelling technique. *PloS one* **12** (4), e0174807.

Tisi, M., Mottu, J. M. & Kolovos, D. S. et al. (2019) Lowcomote: Training the next generation of experts in scalable low-code engineering platforms. In: *Software Technologies, Applications and Foundations (STAF 2019)*.

Todd, P. A., McKeen, J. D. & Gallupe, R. B. (1995) The Evolution of IS Job Skills: A Content Analysis of IS Job Advertisements from 1970 to 1990. *MIS Quarterly* **19** (1), 1.

Verdonck, T., Baesens, B., Óskarsdóttir, M. & vanden Broucke, S. (2021) Special issue on feature engineering editorial. *Machine Learning*, 1–12.

Wade, M. & Shan, J. (2020) Covid-19 has accelerated digital transformation, but may have made it harder not easier. *MIS Quarterly Executive* **19** (3), 199–211.

Wallach, H. M. (2006) Topic modeling: Beyond Bag-of-Words. In: Cohen, W. W. & Moore, A. W. (eds.) *Proceedings of the 23rd International Conference on Machine Learning*. ACM, New York, pp. 977–984.

Walsh, J., Johnson, M. & Sugarman, M. (1975) *Help wanted: Case studies of classified ads*. Olympus Pub. Co, Salt Lake City.

Weisinger, D. (2011) Open Data: Rise of the Rogue Citizen Developer and End-User Application Development. https://formtek.com/blog/open-data-rise-of-the-rogue-citizen-developer/. Accessed: 15.02.23.

Whitmore, R. (2021) Citizen Development Insights: Citizen Development Part 2: Getting Started. https://www.projectmanagement.com/blog/blogPostingView.cfm?blogPostingID=69464&thisPageURL=/blog-post/69464/citizen-development-part-2--getting-started#_=_. Accessed: 05.03.23.

Yang, X., Lo, D., Li, L., Xia, X., Bissyandé, T. F. & Klein, J. (2017) Characterizing malicious Android apps by mining topic-specific data flow signatures. *Information and Software Technology* **90**, 27–39.