5-2-2023

FASTER, CHEAPER, BETTER? ANALYZING HOW LOW CODE DEVELOPMENT PLATFORMS DRIVE BOTTOM-UP INNOVATION

Edona Elshan
University of St.Gallen, edona.elshan@unisg.ch

Dominic Germann
University of St.Gallen, dominicgermann@bluewin.ch

Ernestine Dickhaut
University of Kassel, ernestine.dickhaut@uni-kassel.de

Mahei Li
University of Kassel, mahei.li@uni-kassel.de

FASTER, CHEAPER, BETTER? ANALYZING HOW LOW CODE DEVELOPMENT PLATFORMS DRIVE BOTTOM-UP INNOVATION

Research Paper

Edona Elshan , University of St.Gallen, Switzerland, edona.elshan@unisg.ch

Dominic Germann, University of St.Gallen, Switzerland, dominicgermann@bluewin.ch

Ernestine Dickhaut, University of Kassel, Germany, ernestine.dickhaut@uni-kassel.de

Mahei Li, University of Kassel, Germany, mahei.li@uni-kassel.de

Abstract

Recent years have seen a growing adoption of Low Code Development Platforms (LCDPs) in organizations. The increasing affinity for technology development across all user groups, consumerization of development, and advancing digitalization are opening up a new target group for the low code movement. This change in software development allows bottom-up user innovators within a company to leverage their domain knowledge and quickly deploy much-needed digital services. However, a clear understanding of this paradigm of software development in organizations and the influence on end-user acceptance is still missing. In this paper, we present the results of an interview study conducted with 18 LCDP experts and discuss the implications of our findings, highlighting the role of LCDPs and context in bottom-up innovation as well as user-centricity. Our research contributes to the literature on LCDPs and offers valuable insights for organizations looking to leverage their workforce's innovative potential.

Keywords: Citizen Developer, Bottom-Up Innovation, Digital Innovation, Low Code Development Platforms.

1 Introduction

Keeping up with the ever-increasing pace of innovation in these digital times challenges many organizations to gain or maintain a competitive advantage in order to be a shaping factor of digitalization rather than being shaped and disrupted by it. Organizations, in particular, are faced with the need to drastically reduce time-to-market for digital products while their industries are disrupted by new market entrants and radical interindustry innovations, necessitating the development of organizational resilience through improved innovation processes (Cohendet et al., 2017; Wehking et al., 2021). Furthermore, the COVID-19 pandemic has accelerated the pace of digitalization and continues to present a number of social and economic challenges to society and business, requiring profound change and transformation (Gallagher, 2020; Waizenegger et al., 2020). This is reflected in a recent survey of business executives (Roth et al., 2020), which shows that the majority of executives believe the consequences of the pandemic will fundamentally transform the way they do business and will have a long-term impact on consumer behaviour in the coming years. In today's digital world, an organization's ability to innovate at the same rate as environmental, intra-, and inter-industry conditions change is more important than ever (Roth et al., 2020). Up to now, IT departments have been generally held responsible for developing digital innovation applications (Krejci et al., 2021), but they frequently lack the necessary skills to ensure both the exploitation of current systems and the exploration of new novel technologies and use cases

(Lee et al., 2015). 57% of manufacturing executives, for example, believe their company lacks the skilled workforce to support the smart manufacturing digitization plan (Gartner, 2020).

In this realm, practitioners and scholars increasingly are viewing business unit employees as powerful drivers of rapidly developing digital innovation capabilities (Opland et al., 2020). A novel type of technology has emerged in recent years that promises to encapsulate much of the technological complexities into easily utilizable modules in order to create enterprise-ready services and democratize IT for business units — so-called low code development platforms (LCDPs) could be the solution to this problem (Brinker, 2018; Brown, 2021). Citizen developers (i.e., employees developing applications with LCDPs) seem to be frequently better suited to come up with application ideas that improve business processes or constitute new user offerings because they have in-depth business knowledge. Unlike traditional development approaches, citizen developers can create and test a prototype of an innovative application on the LCDPs entirely on their own. Using the LCDPs properties, citizen developers can quickly add and remove features and functionalities, ultimately selecting only those that add the most value (Richardson and Rymer, 2014; Sanchis et al., 2019). Employees in business units can thus begin developing and evaluating their application ideas without the assistance of an IT developer, significantly reducing upfront investment (Gaggioli, 2017). Thus, LCDPs are a technology platform with the potential to revolutionize the early stages of digital innovation processes and thus empower employees in their creation. This is also in line with previous calls to better understand the role of digital technologies in the innovation process and how digital tools facilitate and inhibit innovation outcomes (Nambisan et al., 2017; Yoo et al., 2012). Given the high practical relevance of LCDPs, and estimates that by 2024, 65% of all application development within an organization will be executed with LCDPs (Vincent et al., 2019), we advocate that it is essential to revisit the intertwining between LCDPs and digital innovation. However, research on this topic is still in its early stages, thus, there is currently little scientific literature on it (Al Alamin et al., 2021; Maruping and Matook, 2020). When it comes to investigating LCPDs and organization, there are first studies that deal with LCDPs and their implementation and use in organizations. For example, Prinz et al. (2022) investigate two perspectives on LCDPs challenges, while Elshan et al. (2023) recently conducted an interview study on the opportunities and risks of LCDPs for organizations. Other studies, for example, Li et al. (2022) have looked at LCDP users, or employees who work with LCDPs and the need to re-/ or upskill them. However, none of the previous studies focused on the main goal that should be achieved by implementing LCDPs in organizations: faster development and meeting the needs of end users.

Bottom-up innovations (BUI) are often characterized by severe resource constraints and extreme experimentation and feedback cycles to solve functional requirements (Hughes and Lonie, 2007). By focusing on the advantages of LCDP, such as short development cycles, BUI is a useful theoretical framework for analyzing LCDP and the impact of user-centeredness. Understanding the relationship between LCD-infused digital innovation and acceptance is critical, because user-centric innovations and artifacts are more successful on the market than those that do not meet users' needs (Brenner et al., 2014). As a result, these digital innovations are generally well received by users.

Thus, LCDPs have enormous potential in this domain for integrating end users into the development of artifacts (Maruping and Matook, 2020), precisely because enduser needs and feedback can be addressed in this manner. At the same time, the use of LCDP strongly promotes user-centricity of artifacts (Clark, 2019). Kaur et al. (2020) investigate the possibility of combining low code development platforms with design thinking methods in this regard. According to Alamin et al. (2021), the advantage of platforms is that artifacts can be developed quickly and easily and then tested immediately with end users. Further, the use of LCDP can improve the quality of developed artifacts (Elshan et al., 2023). This is accomplished through the collaboration of various project stakeholders (such as UX people, business analyst, and frontend developers). However, there may be negative consequences for the end users of the developed artifacts. Even though the relationship between LCDPs and digital innovation acceptance has been studied in some depth, we still lack a comprehensive and clear understanding of the various topics that LCDPs are associated with, such as usability (see Bexiga et al., 2020; Pacheco et al., 2021; Silva et al., 2021). To the best of our knowledge, such an integrated conceptualization does not exist, so we attempt to condense the body of knowledge on LCDPs into a concise and meaningful manner that is useful for practitioners and researchers.

When developing LC-infused digital innovations, LCDPs providers provide some guidelines on how usability can be promoted, user needs can be met, and thus acceptance of digital innovations can be achieved. In order to do so, companies such as Appian, Mendix, and Outsystems have named various approaches. In their book with best practices for low code development, Mendix, for example, lists concepts related to good user interfaces (UI). Appian's website also includes a UX design guide. This, however, focuses more on the UI or usability aspects of components, such as buttons, or the overall layout, such as web page width. Outsystems also provides video tutorials for application development with an emphasis on the user interface. Nonetheless, none of the platform providers provide answers on what impact these measures will have on user-centricity, which is critical if businesses are to succeed in the digital age.

Thus, the following research questions need to be answered: *"How do low code development platforms drive or hinder bottom-up innovation?"*

In order to answer our research question, we draw on a qualitative study in which we interviewed 18 experts that have implemented LCDPs within their organizations and built digital products with them. Within the interviews, we aim to explore what organizations are trying to achieve with LCDPs and how their implementation has influenced the acceptance of digital innovations. To analyze our data, we draw on bottom-up innovation. Our results suggest that there are positive and negative aspects of LCDPs-based bottom-up innovation that drive the acceptance of with LCDPs developed digital innovations. We continue as follows. First, we present the theoretical background of our work. We then shed light on the research design, in particular, on how we conducted our interview study, followed by the presentation of our findings. Lastly, we discuss the implications and limitations of our work and provide a brief conclusion.

2 Theoretical Background

2.1 Low Code Development Platforms

Low code development platforms are a set of tools that are intended for both experienced programmers and users who have no prior programming knowledge or experience (Adrian et al., 2020; Bock and Frank, 2021; Kletti, 2021). Furthermore, they allow non-coders to create high-quality software in a short period of time (Sanchis et al., 2019). LCDPs are typically cloud-based services (also known as platform-as-a-service) that enable the development of applications using specially designed tools. LCDPs use prefabricated components and configurations to reduce the need for manual programming (Khorram et al., 2020). The application logic, user interface, or links to various data services are built using user-friendly visual tools and can be supplemented with manual code as needed (Di Sipio et al., 2020). Users with domain expertise but no specific IT knowledge, or so-called "citizen developers," are thus one of the primary target groups for such novel platforms (Tisi et al., 2019). A citizen developer, according to Gartner (2022), is "an employee who creates application capabilities for consumption by themselves or others." In general, the concept of low code is not new, as approaches that allow people without computer science knowledge to build systems independently have been around for a long time. However, external factors such as the rise of digital platforms have altered the landscape of IS development. As a result, platform-centricity is critical to the spread of low code in work environments. LCDP vendors (such as OutSys or Mendix) typically provide a variety of tools to aid in the application development process, from initial ideation and modeling to implementation and maintenance (Almonte et al., 2020).

Furthermore, a new generation of tech-savvy, digitally native workers is emerging with some of the necessary qualifications. Factors such as increased affinity for technology, consumerization, and advancing digitalization are allowing the low code movement to reach a whole new audience. LCDPs' broad application possibilities, as well as the benefits described in connection with them, are gaining traction.

2.2 Use Centricity and Bottom-Up Innovation

As previously stated, the use of LCDP can be justified by the fact that various benefits can arise. Among other things, the quality of developed artifacts may improve as a result of increased collaboration among various stakeholders. The integration of these stakeholders allows the end users' needs to be better met and their software acceptance to increase.

The intensive inclusion of users and their needs in development is referred to as a "user-centered approach" – Brenner et al. (2014) also refer to this as customer centricity and acknowledge that this has already been extensively studied in science under the terms "human-centered design" or "customer orientation." According to Loshin and Reifer (2013), when developing products or services, end users and their needs are always prioritized. This assists in determining who the customers are, what they truly desire, and how they can benefit from a company's offering. This, in turn, is intended to increase customer loyalty, decrease churn, and, ultimately, increase sales by providing superior products and services to customers. These user-centric approaches are incorporated in user-, use-, and utility-centricity, where the focus is on "digital users" and their data or user behavior (Brenner et al., 2014).

LCDP development processes aim to emphasize user-centricity by engaging users throughout the development lifecycle. This approach aligns with established methodologies such as agile, lean startup, and design thinking, which prioritize user needs and feedback (e.g., Parizi et al. 2022). In contrast to more traditional methods like waterfall, the LCDP process is characterized by iterative and incremental development, allowing for rapid adjustments based on user feedback and evolving requirements (Bock & Frank, 2021).

The increasing personalization of products, services, and processes (Gummesson, 2008) dramatically alters user-company interactions (Sheth et al., 2000). In addition to user centricity, use centricity - which includes user experience and usability - has a significant impact on an artifact's success. Use-centricity is an approach that focuses on meeting the needs of artifact users as best as possible (Leimeister, 2014). Thus, users expect satisfaction at all times and in all situations, and it is shaped by usability and user experience (UX) (Varco and Lush, 2016). For example, Rosenberg (2018) and Ruth (2021) acknowledge that a good user experience is an important driver of an artifact's market success. This is also helped by good usability. The two terms are defined in greater detail and placed in context below. According to Winter and Stevens (2020), the terms usability and user experience are clearly separated in science, but only partially in practice. Usability is defined as how extensively a product or service can be used in a specific context to effectively achieve desired goals (Leimeister, 2014). At the same time, Rajanen et al. (2017) state that there is still no agreement in science on a definition of the term, but that it should be oriented to the already known ISO standard: "Usability describes how a product can support its users to be effective, efficient, and satisfied in its use" (Bevan et al., 2015, p. 11). Coursaris and Kim (2006), Zhang and Adipat (2005), and Harrison et al. (2013) all include learnability and memorability in addition to the three dimensions of effectiveness, efficiency, and satisfaction (Tan et al., 2020). Simultaneously, they list additional dimensions based on the use case or via smartphone or computer. Fearns et al. (2015) considers usability to be a component of the user experience and categorize it into five additional factors: credibility, usefulness/value, desirability.

Simultaneously, Winter et al. (2015) recognize other factors that contribute to the user experience, such as timeliness, adaptability, transparency, or novelty. Based on the various descriptions, it is clear that there is no single definition in science; this is also true in practice (Hassenzahl and Tractinsky, 2006).

Innovation generation can be divided into top-down innovation (TDI), and bottom-up innovation (BUI) emerged. BUI focuses on the involvement of employees in the project, which in turn goes hand in hand with use centricity. Thus, BUI has become increasingly important since it is both promising and promotes employee integration and motivation. By involving employees, the entire project benefits from their know-how and, in turn, gives them back the feeling of being part of the big picture.

Information systems scholars have studied the role of BUI in the development of a plethora of information technologies (Bagayogo et al., 2014; Boudreau and Robey, 2005; Ciborra et al., 2000; Mazmanian et al., 2013; Oborn et al., 2011). Zorina and Karanasios (2021) have shown in their study how BUI motivate employees and developments take place that focuses on the needs of the users. While recent research stresses the importance of communities as useful and emergent organizational structure for BUI (e.g., Zorina and Karanasios, 2021), user-driven innovations address how lead users within a formal organization create new technological services and features (e.g., Morrison et al., 2000; Von Hippel, 2005). User-driven innovation is often characterized by a) strong resource limitations and b) extreme experimentation and feedback cycles to solve the functionality needs (e.g., Hughes and Lonie, 2007). Thus, focusing on the advantages of LCDPs, such as short development cycles BUI, is a useful theoretical framework for analyzing LCDPs and the impact of use-centricity.

3 Research Design

To address the research question, we applied a qualitative research design. In particular, we conducted our interview study following the approach and principles of Grounded Theory (Strauss and Corbin, 1998). Our research process is described in the following. First, we searched for and selected suitable interview partners. For a purposeful selection, we followed the recommendations of Gioia (2013) and interviewed "knowledgeable agents". After conducting and transcribing the interviews, we analyzed our interviews iteratively and derived drivers of use-centricity and withdraws of use-centricity that yield user-centricity of digital innovations.

3.1 Data Collection

To ensure that the research findings are as projectable as possible, care was taken in the selection of interviewees and experts to ensure that different industries and companies of all sizes (small, medium-sized, and large companies) were covered - but that they all had their headquarters or a registered office in the DACH (Germany, Austria and Switzerland) region (Paré, 2004). It was also considered that people from the three groups listed below were interviewed: users, consultants, and vendors. The first category includes businesses that use LCDP, while the second category includes businesses that provide LCDPs (e.g., Mendix, OutSystems). The term "consultants" refers to businesses that provide consulting services in conjunction with LCDP. All the selected experts have been working with low code development platforms for more than two years and consider it essential for the creation of digital innovations. Overall, data triangulation occurred in the expert interviews due to the diversity of people from different industries, company sizes, and application, consulting, or offering groups. The 18 expert interviews were conducted with 18 different companies, using semi-structured interviews, which followed predefined guidelines but allowed for naturally evolving conversations by incorporating variations in topics and adaptability to emerging themes (Longhurst, 2003). The interviews lasted, on average 51 minutes ranging from 34 to 70 minutes. Most of the interviews were conducted in German and some in English.

Interviewee	Group	Job Role	Domain	Duration
IP1	Consultant	Senior Manager Digital Transformation	Consulting	56 min
IP2	Consultant	Co-Founder Low Code Lab		61 min
IP3	Consultant	Senior Consultant		40 min
IP4	Consultant	Head of Consulting		43 min
IP5	Consultant	Senior Consultant		68 min
IP6	Vendor	Head of Product Management	Software Development	56 min
IP7	Vendor	Solution Sales Manager		50 min
IP8	Vendor	Head of Solution Advisor		41 min
IP9	Vendor	Executive Technology Strategist		70 min

IP10	Vendor	Account Executive		59 min
IP11	User	Head of Innovation	Manufacturing	56 min
IP12	User	Lead Web Development	Banking	36 min
IP13	User	IT Business Partner	Retail	60 min
IP14	User	Product Owner	Biotech	48 min
IP15	User	Project Lead	Banking	34 min
IP16	User	Chief Digital Officer	Utility Supplier	48 min
IP17	User	Head of IT	Retail	47 min
IP18	User	Chief Technology Officer	IT	54 min

Table 1. Interview Participants.

To elicit a rich amount of qualitative data, we utilized the established method of semi-structured interviews (Myers and Newman, 2007). For the interviews, an interview guide was created with the following topics to explore the experiences of the experts in depth: (1) use case of low code application, (2) understanding of user-centricity, (3) drivers of bottom-up innovation, (4) impact on use-centricity.

3.2 Data Analysis

Because the interviewing and analyzing phases are inextricably linked (Langley, 1999; Locke and Golden-Biddle, 1997), we conducted a qualitative analysis of transcripts and addressed the dimensions of credibility, dependability, reliability, and transferability to ensure rigor (Merriam and Grenier 2019). The careful selection of interviewed experts who are knowledgeable agents in their work environment ensured the credibility of our results in terms of agreement with reality. Two researchers extracted data from the material using the interview transcripts and engaged in open, axial, and selective coding (Saldaña, 2021). We followed the general principles of Grounded Theory in doing so (Charmaz and Belgrave, 2012; Strauss and Corbin, 1998). Going on, the coding iterations and deriving of the first-order constructs were iteratively validated by two researchers in research discussions (Forman and Damschroder, 2007). This shall account for stable, valid, and reproducible research results. However, we note that when deriving codings from interviews, there is always a certain residuum of subjective bias as individual human judgment is involved. In total, we coded 24 main categories (information about the experts, platforms, user groups, use case, etc.) and assigned 2306 code segments within the 18 interview transcripts Differences in coding were discussed and resolved using a consensus approach until a consensus view was reached. We used the program "MAXQDA" as a coding aid. Table 2 shows an example of an open code as well as illustrative interview data.

Open Code	Illustrative Data
Rapid Development	“And these kinds of tools should just be able to kind of quickly generate output for you. And it actually comes kind even more of a kind of dementia; we are developing something almost with “No code” but are getting the whole functionality. “
Software Quality	“And above all, if not only different people carry out the process by hand, but it is carried out by software, then I also have a certain quality assurance. Then I can be sure that if I carry out the process six times, it will be carried out six times in exactly the same way, and the same applies if I do it 200 times, if I somehow have a team of 15 people carry out 200 processes, then the result will always be a little different.”

Table 2. Expository Open Coding.

After openly coding the documents and assigning relationships among the open codes (axial coding), we set the core variable and iteratively evaluated the coding in discussions among two researchers to reach validity and reproducibility (Saldaña, 2021). Lastly, we had a look at all of the drivers that were identified in our findings and classified them either as positive or negative when it comes to BUI.

Axial Code	Sub-Category	Illustrative Data
Prototyping	• Agile development approaches collect requirements and pain points • Iterative cycles improve time and costs for MVPs • Non-familiarity with UX concepts	"And these kinds of tools should just be able to kind of quickly generate output for you. And it actually comes kind even more of a kind of dementia; we are developing something almost with "No code" but are getting the whole functionality. "
User Testing	• Users are involved during the development • Faster/ and more feedback • Compromises due to limited applications • Unnecessary applications due to enablement of citizen developer	"And above all, if not only different people carry out the process by hand, but it is carried out by software, then I also have a certain quality assurance. Then I can be sure that if I carry out the process six times, it will be carried out six times in exactly the same way, and the same applies if I do it 200 times, if I somehow have a team of 15 people carry out 200 processes, then the result will always be a little different."
Tool Development	• Citizen developers are enabled to develop their own solutions • Predefined/ pre-tested building blocks • Error prevention • Value of familiarity • Identification of improvement potential • Improved collaboration • Shorter development cycles	"One use case we have is for the transfer of goods by rail to the ports, and we have started to redevelop the entire business application with all components, from order to cash, so to speak. We have been developing this for two years now. This is a huge business application."
Patching	• Error prevention • Poor performance • Limited computer capacity or processing speed • Less maintenance	[...] If the data stays in the current database, how can we connect that? What technologies are available there? All of these questions are much easier to answer through LCDPs because we are all building on the same platform, and not just everyone is doing what they feel like doing

Table 3. Selective Coding.

Since the interviews and the analysis were done mostly in German, we translated the final coding into English while preserving the meaning. The findings are outlined in the following.

4 Findings

The majority of experts see various drivers of bottom-up innovations that can result from using LCDP for the development of applications. Furthermore, we have identified several tension areas within LCDP-based bottom-up innovations. Tension areas are focused on either the LCDP development process or the properties of the bottom-up innovation result. Having the tension areas in mind, the drivers (positive or negative) that lead to these tension areas are listed and discussed separately in the following sections.

4.1 Tension Areas for LCDP-enabled BUIs

Tension Area 1: User-driven innovations focus on developing digital functionalities (Hubbard and Ottoson, 1997). LCDPs provide the tools to quickly develop and deploy digital functional applications, which can lead to much shorter development cycles than traditional software development (Daniel et al., 2020) projects. LCDP developers are also often more in touch with user business needs, which allows them to quickly develop a multitude of applications (Bexiga et al., 2020). While users might profit from multiple useful applications, organizations are also at risk of losing a business-wide overview of their bottom-up innovations. Thus, the quick development cycles for developing individual functionalities are in tension with the risks associated with shadow IT (Haag and Eckhardt, 2017).

Tension Area 2: Low code development platforms provide citizen developers with pre-developed modules, oftentimes shown as graphical plug-and-play representations (Bock and Frank, 2021). This encapsulates software code that has been tried and tested and allows the “developer” to focus on the desired application output rather than code quality, trusting in the LCDP-provider’s responsibility to provide only high-quality software modules (Prinz et al., 2021). This diminishes the importance of commenting code or code refactoring while reducing time needed for debugging. While this form of extreme LCDP-enabled code encapsulation allows developers to focus on user-centricity, performance issues are relegated to the responsibilities LCDP providers. Thus, there is a possible tension between user-centricity and software performance.

Tension Area 3: Low code development platforms provide pre-designed UX design elements, which ensure a level of usability while addressing the limited resources of BUIs. BUI users have become accustomed to its look and feel. While advantageous, such LCDPs can also limit the innovation potential of their developers (Di Sipio et al., 2020). With pre-conceived modules, iterative innovation based on reconfigurations is strongly encouraged (Beverungen et al., 2018). Individual customizations and other more radical innovations require even more resources to adhere to the existing LCDP structure. Thus, the freedom reusable LCDP resources provide can also lead to restrictions on innovation potential.

4.2 Positive Aspects of LCDPs-based Bottom-Up Innovation

According to experts, the use of LCDP leads to greater satisfaction of the users' needs and habits. There are several reasons for this. For starters, people - particularly those in the business department - may develop their own solutions. Because they know exactly what they require, they are more likely to be satisfied. Second, when the LCDP is used by the business or IT department, collaboration with users improves. This means they are involved in the implementation process from the start and throughout further development, resulting in fewer "translation errors" between what is required and what is ultimately built. Working with end users directly on prototypes leads to sharpening the picture of exactly what they want and can then be implemented directly: *"[...] because [...] what the department and the customer want just becomes sharper and more right on target" (IP14).*

Third, LCDP development typically occurs in an agile environment and through requirements engineering; in workshops, surveys, or observations, the needs or pain points of the users are identified, and solutions are proposed step by step. Fourth, iterative feedback cycles can be used to create artifacts that are limited to the absolute benefit and eliminate unnecessary functions: *"And gets exactly what he wants" (IP22).*

Overall, this enables the development of solutions that are precisely tailored to the needs of the users - for example, through the hyper-personalization of user interfaces. The latter is made possible, above all, by the ability to specifically address the individual wishes of the user groups. The following is the context: Artifact development with LCDP takes less time, requires fewer resources, and can be done by users themselves. This frees up resources such as time and money, which can then be used to fulfill specific desires. As a result, (hypo)personalized or tailor-made solutions can be provided.

The fact that users' needs are better met is due in part to the fact that LCDP allows for faster, more iterative feedback. This means that the solutions can be tried and tested continuously. The experts see this as a benefit for end users: with increased collaboration and, thus, user involvement in development, they can provide feedback directly and quickly. This also shortens feedback cycles and allows for faster verification of assumptions made when creating an artifact. This ensures that the solutions are not "developed after" the users have been identified: *"Most of the time, you don't need more than two or three hours. Then you can already make something available to the customer or the user, who can then try it out and give initial feedback" (IP15)."The more people give feedback on an application, the better the user experience will be in the end" (IP12).*

During development, (rapid) prototyping is used in companies to validate the correctness of the assumptions. This means, for example, that a (clickable) mockup is created, displayed to users as a "showcase," and then discussed. This allows them to get a "feel" for the finished solution and provide direct feedback. Close collaboration and co-creation with customers from the start allows the artifact to be adapted and improved. Because of the simplicity of LCDP and the possibility of "drag and drop development," the entire development process takes less time than traditional programming. This is one of the reasons why experts see faster development / shorter development cycles as advantageous in LCDP. As a result, users receive what they require more quickly. Shorter development cycles are also valuable because user needs can change quickly - for example, due to a competitive environment - and adaptation must therefore occur quickly. The fact that development is accelerated frees up time for backlog processing. This, too, eventually results in a benefit for the end user: *"[...] because you simply work in the sprints in such a way that you can quickly, agilely provide results to the customers, to the users" (IP4).*

Aside from the platform's simplicity and the building blocks provided for development, working in an agile context and with short, iterative feedback cycles allows for faster implementation. Because of the close collaboration between individual stakeholders as well as users, they get what they really need more quickly. This saves time on future improvements after the solution has been published. Prototyping, in particular, provides advantages in this regard: These are also appropriate for comparisons of different versions, for example. As a result, developers can create a variety of solutions and then select the best variant based on user feedback.

According to the experts, using LCDP improves usability, UI, UX, user guidance, and artifact quality. As a result, a simple, understandable, and intuitive user interface can be created. This makes them more visually appealing, which improves the "look and feel" of the solutions in terms of usability and user experience. There are several reasons for this, including: First, the LCDP-specified/tested building blocks are used. Platform providers such as OutSystems, Mendix, and Appian have already tested and validated these with users. Simultaneously, OutSystems, for example, is focused on well-known solutions with good UX and usability and can replicate them: *"So OutSystems is like, they go into the market; they study all the web applications [...], they go into the Apple Store and Android Store, and they study all the top mobile apps; and then from there, they conclude that this is all the out-of-the-box user control that you need" (IP15).*

Second, using the blocks reduces errors during development because they have already been tested for functionality, and the scope for implementation is limited due to the limitations of the LCDP. As a result, the solution created with it is less prone to errors.

Third, the improved usability, UX, UI, user guidance, and quality can be attributed to the fact that the specified building blocks produce consistent user interfaces, usability, and user experience, resulting in a recognition value: *"You can basically design it so that it seems more like it's all of a piece" (IP11).*

As a result, the design or frontends appear similar, if not uniform. User guidance and underlying processes can be designed more consistently as well.

Fourth, tips provided by platforms during the development of artifacts improve usability and UX: *"So it's just somehow not like it used to be, [...] but really fancy apps [...]" (IP3).*

The previously mentioned increased collaboration or co-creation with users via iterative feedback cycles, as well as improved need satisfaction, result in additional benefits for end users: User acceptance and satisfaction are both increasing. This is aided by the users' understanding of the solution. Because they are involved in the development process from the start, they are aware of potential limitations and can understand compromises in the final solution: *"That allows for a collaborative discussion right at the open heart and right down at the base of everything, and that's where the big difference is" (IP6).*

Experts see the following benefits for end users: For starters, complexity is reduced. This specifically refers to two aspects: On the one hand, any use case can be handled by a (newly) developed IT application, and various standard applications are no longer required. Because several applications for user work are replaced by a single solution, the complexity of keeping track of the various applications is reduced. As a result, the work can become more efficient. Artifacts created with LCDP, on the other hand, cannot currently represent complex situations and processes due to their limited applicability. Therefore, they must be simplified. As a result, in the course of development, complexity is reduced. In terms of complexity, experts agree that certain standard applications are overly complex. These can be simplified in the course of a revision if they are now (re)built with LCDP.

Second, LCDP-created artifacts enable device independence. This means they can only be developed once and then run on multiple operating systems with the click of a button. This means that they can be used on a wide range of end devices. Third, closer collaboration between business, IT, and users increases mutual understanding and, for the latter group, affinity with IT or digitization. This enables them to identify more potential for improvement themselves. This, in turn, has various advantages, such as optimization of solutions, increased efficiency at work, and, if applicable, increased user satisfaction and acceptance.

Table 4 below lists the positive aspects of LCDPs in bottom-up innovation that lead to use-centricity and acceptance by end users, most frequently mentioned in the expert interviews.

Positive Aspects	Examples
Better usability/ software quality	*"[...] can be more responsive to customer needs [...]" and "[...]can very individually accommodate the needs of all external factors [...]."*
Faster development/ shorter development cycle	*"[...] even faster, even closer to me to get the feedback [...]”* *"Because I just get more feedback."*
Prototyping	*"Low-code/no-code gives me the ability to prototype super-fast."*
Customized solutions	*"They get what they really need at the end of the day [...]."*
Collaboration	*"[...] can slide into the role of Citizen Developer and participate in the development."*
Value of familiarity/ Consistent UX	*"That everything comes from a single mold", "[...], the recognition value is there [...]."*
Error prevention	*"[...]that automation is very good at error prevention, [...]."*
Reduction of complexity	*"[...]which of course also perhaps takes away a little bit of the complexity perhaps for the end customer."*
Review of assumptions	*"[...] they can quickly clarify what the thing that the business wants are."*
Predefined/ pre-tested building blocks	*"After all, you work with a modular system [...] and with ready-made system parts that are all fully tested. "*
Device independence	*"[...] because he can use that on any device."*
Hyper personalization	*"[...] create hyper-personalized apps, that you have a separate app for each skill level."*
Identification of improvement potential	*"[...] people are thinking more about it [...] and more actively asking for change."*
Comparison of different versions	*"[...] make different, actual versions or something of the same thing, and give it to customers to compare [...]."*

Table 4. Positive Aspects and Expository Quotes.

4.3 Negative Aspects of LCDPs-based Bottom-Up Innovation

According to the experts, however, LCDPs does not only offer advantages for the users of the artifacts developed with them. Various negative aspects (see Table 5) are identified: Among other things, the experts state that the usability / UX / UI / user guidance/quality can be poorer in the application.

There are several reasons for this: Firstly, larger compromises sometimes have to be made during development due to the limited applications. This is especially the case with more complex use cases. This can lead to a failure to meet the needs of the users. This, in turn, can result in frustration and lower user acceptance by the end users. At the same time, the discomfort regarding the unsatisfied needs can also be due to the fact that the users are bombarded with unnecessary applications - due to the enablement of the employees for programming. This can lead to an overstimulation of the end users: *"[...] if the question of meaning does not arise, they are filled up with any apps where someone has thought, I know what the users want, [and just] clicks something together, deploys [...]" (IP2).*

Secondly, poorer usability and UX are also a consequence of the fact that, for example, the business and IT departments, as users of the platform, are usually not familiar with concepts relating to usability / UX / UI and therefore neglect them during development. The lack of experience and neglect of (design) principles is thus also recognized as a disadvantage for the end users: *"With low code, you slapped a UI on very quickly. But many customers didn't focus on the UI issue at all" (IP6).*

Best practices in software development are neglected, or important aspects of concepts such as corporate identity or corporate design are simply forgotten. This is also due to the fact that in some cases, no UX / UI specialists are integrated into (smaller) development projects - among other things, because there are none at all due to the size of the company or the management does not recognize the importance of the role.

Third, poor usability and user experience results from the fact that the templates in terms of building blocks of LCDP do not look "very nice" and can only be customized by means of high effort. The building blocks also result in the artifacts developed with them looking very generic and providing a similar user experience. The sum as well as the individual parts of these negative points then results in poorer use-centricity. As a result, companies can no longer distinguish themselves from the competition with their solutions.

Fourth, in the development of artifacts using LCDP, there is sometimes a neglect of testing. There are two main reasons for this: On the one hand, user acceptance testing is performed in a reduced form. This is due to the fact that Citizen Developers are not aware of the importance of testing due to their inexperience. At the same time, they think that it is not necessary because they have developed the solution for themselves or together with the users. On the other hand, neglect of testing is also due to the fact that it is difficult - especially for people without IT knowledge: For example, with some platforms, the generated code and thus the underlying architecture cannot be checked. This in turn leads to the fact that tests can be automated less and are therefore neglected more.

Fifth, poor performance and greater susceptibility to errors in the developed artifacts can be identified as disadvantages for users. The experts explain this by pointing out that the scaling of solutions - due to the then limited computer capacity or processing speed - represents a weakness of LCDP. On the other hand, errors can also arise due to insufficient or missing concepts. This results from artifacts being developed without much thought and without the support of the IT department with its expertise. This in turn can lead to the disadvantage that the user: needs more effort and support because the solution is poor.

Furthermore, the experts recognize poorer maintenance in the solutions developed by LCDP. Further development can be difficult because compromises were made in the original implementation and detours were, therefore "tinkered with". Continuous integration and continuous deployment also suffer as a result. If this is complicated or made impossible, solutions are less likely to be maintained or developed further.

Negative Aspects	Examples

Poorer usability/ UX/ User guidance	*"[...]also often the result of low code poor usability."* *"[...] functionality can feel a bit "rough" and not quite "polished"*
Non-satisfaction of needs	*"[...]that he then does not get his absolutely suitable solution, [...]"* *"[...] are then simply unusable and unmaintainable."*
Neglect of established design guidelines	*"[...] The absence of typical disciplines in low code projects, [...]"*
Compromising UX	*"[...] does not get its absolutely suitable solution but has to live with a certain compromise."*
Poor performance	*"[...] that simply the performance itself falters, [...]"*
Poor maintenance	*"[...] and maintainability just goes down the drain."*
Neglecting testing	*"Because with low code platform, you can't write unit tests. You can't test code. "*
Higher susceptibility to errors	*"[...] is perhaps not as well designed, somewhat more error-prone [...]"*
Development of unnecessary features and apps	*"[...], these are poured on with any apps, [...]"*

Table 5. Negative Aspects and Expository Quotes.

5 Discussion

The expert interviews' findings highlight the various advantages and disadvantages of using LCDPs (Low-Code Development Platforms) in bottom-up innovation. Better user-centricity, increased collaboration, shorter development cycles, and improved usability and software quality are among the benefits. LCDPs also enable device independence, rapid prototyping, and the creation of customized and hyper-personalized solutions. These advantages help to increase user satisfaction and acceptance.

However, there are a number of disadvantages associated with the use of LCDPs. Low usability, user experience, and software quality can be attributed to development compromises, a lack of experience and disregard for design principles, and limitations in the customization of building blocks. Furthermore, the experts identified potential issues with the testing, performance, and maintenance of LCDP-based solutions, which could result in greater error susceptibility and increased effort required from end users. The experts also note that the joint development of the artifacts leads to a better understanding of the solution, especially among the users. This, in turn, strengthens the advantage that they see potential for improvement in the applications themselves or in the corporate landscape in general: *"[...] it has this positive effect that you think about it more, also as an end user, also as a department and ask more actively for changes" (IP5).*

This can be understood as the enablement of employees and ultimately makes an active contribution to the aforementioned "democratization of digitization: Through their (new) knowledge, the various stakeholders can become part of the digital transformation of a company and actively promote it. At the same time, companies must be aware that the use or manual creation of applications for the automation of processes by employees can lead to additional work, especially because the applications have to be maintained and further developed. On the one hand, employees benefit from less repetitive tasks and can use their time elsewhere. On the other hand, they have to invest the freed-up time in support or in improving the artifact: *"And suddenly you're doing the old job that you were hired for; you're not doing it afterward" (IP14).*

6 Conclusion and Limitations

LCDP research is still in its early stages, with only a few studies delving deeply into this novel development method. To the best of our knowledge, our work provides the first comprehensive overview of LCDP-driven bottom-up innovation. By conducting exploratory qualitative research, we provide valuable insights into the impact of LCDP use on bottom-up innovation and user-centricity in this paper. Our findings highlight tensions that arise when employees use LCDPs for bottom-up innovation, and we contribute to both the LCDP and bottom-up innovation literatures by identifying specific drivers, obstacles, and measures for promoting user-centricity in the context of LCDPs.

We recognize the need for additional research and have planned a qualitative interview study with a larger sample of actual end-users, also known as "citizen developers," who have firsthand experience working with LCDPs. This method will allow us to gain a more comprehensive understanding of LCDP usage and its impact on innovation while minimizing potential consultant and vendor biases. In addition, we intend to conduct a series of case studies that will look at specific instances of LCDP usage, focusing on the successes and failures in driving innovation. We hope to gain valuable insights and identify patterns that can be generalized to a broader context through these case studies, addressing the reviewer's concerns and deepening our understanding of the complexities of LCDP-driven innovation.

Given the limitations of our study, such as the small sample size, focus on three perspectives (consultants, vendors, and users), and focus on companies that chose to work with LCDPs, we believe that these future research directions will significantly improve our understanding of the topic. We will be better positioned to explore the various use cases of LCDPs within and beyond the DACH region, as well as understand the challenges, opportunities, and implications of LCDP adoption in diverse contexts, by broadening the range of industries and incorporating case studies.

In terms of practical contribution, for realizing the benefits of LDCP, our results recommend that companies must be a) aware of the potential benefits and strengths of LCDPs, b) leverage them by proactively providing the environment to use them and c) provide and communicate proof-of-value. Our assessment of LCDP weaknesses, challenges, and risks should be constantly managed and taken in consideration by businesses. Furthermore, they should promote user-centricity among end users in order to leverage LCDP to gain a competitive advantage—whether by optimizing internal processes or developing customer solutions.

References

Adrian, B., Hinrichsen, S., Nikolenko, A. (2020). “App development via low-code programming as part of modern industrial engineering education,” in: Nunes, I.L. (ed.) *International Conference on Applied Human Factors and Ergonomics,* Virtual Event, USA.

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 Thongtanunam. P., & Rastogi, A., Demeyer, S., & Khomh, F.(eds.). *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, Madrid, Spain.

Almonte, L., Cantador, I., Guerra, E., de Lara, J. (2020). “Towards automating the construction of recommender systems for low-code development platforms,” in Association for Computing Machinery (eds.). *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, Virtual Event, Canada.

Bagayogo, F.F., Lapointe, L., Bassellier, G. (2014). “Enhanced use of IT: A new perspective on post-adoption, ” *Journal of the Association for Information Systems* 15 (3), 361-387.

Bevan, N., Carter, J., Harker, S. (2015). “ISO 9241-11 revised: What have we learnt about usability since 1998?,” in: Kurosu M., (ed.) I*nternational Conference on Human-Computer Interaction*, Los Angeles, USA.

Beverungen, D., Lüttenberg, H., Wolf, V., 2018. “Recombinant service systems engineering, ” Business & Information Systems Engineering 60, 377–391.

Bexiga, M., Garbatov, S., Seco, J.C. (2020). “Closing the gap between designers and developers in a low code ecosystem,” in Association for Computing Machinery (eds.). *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, Virtual Event, Canada.

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

Boudreau, M.-C., Robey, D. (2005). “Enacting integrated information technology: A human agency perspective, ” *Organization Science* 16(1), 3–18.

Brenner, W., Karagiannis, D., Kolbe, L., Krüger, J., Leifer, L., Lamberti, H.-J., Leimeister, J.M., Österle, H., Petrie, C., Plattner, H. (2014). “User, use & utility research, ”*Wirtschaftsinformatik* 56, 65–72.

Brinker, S. (2018). *Democratizing martech: distributing power from IT to marketing technologists to everyone*. URL: https://chiefmartec.com/2018/05/democratizing-martech-marketing-technologists/ (visited on June 16, 2021).

Brown, H. (2021). *Low-Code is revolutionising the software industry: what type is dominating?* URL: https://cyclr.com/blog/low-code-is-revolutionising-the-software-industry (visited on November 15, 2022).

Charmaz, K., Belgrave, L. (2012). “Qualitative interviewing and grounded theory analysis,” *The SAGE handbook of interview research: The complexity of the craft* 2, 347–365.

Ciborra, C., Braa, K., Cordella, A., Dahlbom, B., Hepsø, V., Failla, A., Hanseth, O., Ljungberg, J., Monteiro, E. (2000). F*rom control to drift: the dynamics of corporate information infrastructures*, Oxford: Oxford University Press.

Clark, L. (2019). *Low-code maturity boosts efficiency and helps user acceptance*. URL: https://www.computerweekly.com/feature/Low-code-maturity-boosts-efficiency-and-helps-user-acceptance (visited on November 17, 2022).

Cohendet, P., Parmentier, G., Simon, L. (2017). *Managing knowledge, creativity and innovation, in: The Elgar Companion to Innovation and Knowledge Creation*, Northhampton, MA: Edward Elgar Publishing.

Coursaris, C., Kim, D. (2006). “A qualitative review of empirical mobile usability studie*s*,” in: AIS (eds.) *American Conference on Information System*s, Acapulco, Mexico.

Daniel, G., Cabot, J., Deruelle, L., Derras, M. (2020). “Xatkit: a multimodal low-code chatbot development framework,” *IEEE Access* 8, 15332–15346.

Di Sipio, C., Di Ruscio, D., Nguyen, P.T. (2020). “Democratizing the development of recommender systems by means of low-code platforms,” in Association for Computing Machinery (eds.). *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, Virtual Event, Canada.

Elshan, E., Dickhaut, E., Ebel, P. (2023). “An Investigation of Why Low Code Platforms Provide Answers and New Challenges, ” in: Bui, T.(ed.) Hawaii International Conference on System Sciences (HICSS), Mauii, USA.

Fearns, N., Graham, K., Johnston, G. (2015). “Improving the user experience of patient versions of clinical guidelines: user testing of a Scottish Intercollegiate Guideline Network (SIGN) patient version,” *BMC health services research* 16(1), 1–10.

Forman, J., Damschroder, L. (2007). Qualitative Content Analysis, in: Jacoby, L., Siminoff, L.A. (eds.), *Empirical Methods for Bioethics: A Primer* (Vol.11). Bingley: Emerald, pp. 39–62.

Gaggioli, A. (2017). “The No-Code Revolution May Unlock Citizens’ Creative Potential. Cyberpsychology, ” *Behavior, and Social Networking* 20(8), 508–509.

Gallagher, J. (2020). Analysis: How do we get out of lockdown? URL: https://www.bbc. com/news/ health-52183295 (visited on November 15, 2022).

Gartner. (2022). *Definition of Citizen Developer*. URL: https://www.gartner.com/ en/information-technology/glossary/citizen-developer (visited on November 17, 2022).

Gartner. (2020). Smart Manufacturing Strategy and Implementation Trends Survey.Gartner.

Gioia, D.A., Corley, K.G., Hamilton, A.L. (2013). “Seeking qualitative rigor in inductive research: Notes on the Gioia methodology,” *Organizational research methods* 16(1), 15–31.

Gummesson, E. (2008). “Extending the service-dominant logic: from customer centricity to balanced centricity,” *Journal of the Academy of Marketing Science* 36, 15–17.

Haag, S., Eckhardt, A. (2017). “Shadow it,” *Business & Information Systems Engineering* 59, 469–473.

Harrison, R., Flood, D., Duce, D. (2013). “Usability of mobile applications: literature review and rationale for a new usability model,” *Journal of Interaction Science* 1, 1–16.

Hassenzahl, M., Tractinsky, N. (2006). “User experience-a research agenda,” *Behaviour & information technology* 25(2), 91–97.

Hubbard, L.A., Ottoson, J.M. (1997). “When a bottom-up innovation meets itself as a top-down policy: The AVID untracking program,” *Science Communication* 19(1), 41–55.

Hughes, N., Lonie, S. (2007). “M-PESA: mobile money for the “unbanked” turning cellphones into 24-hour tellers in Kenya,” *Innovations: technology, governance, globalization* 2(1-2), 63–81.

Kaur, V., Berardi, V., Thacker, D., Blundell, G. (2020). Towards a Citizen Development Andragogy: Low-Code Platforms, Design Thinking, and Knowledge-Based Dynamic Capabilities.

Khorram, F., Mottu, J.-M., Sunyé, G. (2020). “Challenges & opportunities in low-code testing, ” in: Association for Computing Machinery (eds.). *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, Virtual Event, Canada.

Kletti, N.-L. (2021). “Trends der Fertigungs-IT 2021,” *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 116(5), 276–278.

Krejci, D., Iho, S., Missonier, S. (2021). “Innovating with employees: an exploratory study of idea development on low-code development platforms,” in: Matook, S., Rosenkranz, C., & Whitley, E. (eds.) *European Conference on Information Systems*, Virtual Event, Morocco.

Langley, A. (1999). “Strategies for theorizing from process data,” *Academy of Management Review* 24(4), 691–710.

Lee, O.-K., Sambamurthy, V., Lim, K.H., Wei, K.K. (2015). “How does IT ambidexterity impact organizational agility?,” *Information Systems Research* 26(2), 398–417.

Leimeister, J.M. (2014). *Collaboration Engineering: IT-gestützte Zusammenarbeitsprozesse systematisch entwickeln und durchführen*, Berlin Heidelberg: Springer-Verlag.

Li, M.M., Peters, C., Poser, M., Eilers, K., Elshan, E. (2022). “ICT-enabled job crafting: How Business Unit Developers use Low-code Development Platforms to craft jobs,” in: Böhmann, T., Hui, K.-L., & Venkatesh, V. (eds.) *International Conference on Information Systems*, Copenhagen, Denmark.

Locke, K., Golden-Biddle, K. (1997). “Constructing opportunities for contribution: Structuring intertextual coherence and “problematizing” in organizational studies,” *Academy of Management Journal* 40(5), 1023–1062.

Longhurst, R. (2003). “Semi-structured interviews and focus groups. Key methods in geography 3, 143–156.

Loshin, D., Reifer, A. (2013). *Using information to develop a culture of customer centricity: customer centricity, analytics, and information utilization*, Waltham, MA: Elsevier.

Maruping, L.M., Matook, S. (2020). “The evolution of software development orchestration: current state and an agenda for future research,” *European Journal of Information Systems* 29(5), 443–457.

Mazmanian, M., Orlikowski, W.J., Yates, J. (2013). “The autonomy paradox: The implications of mobile email devices for knowledge professionals,” *Organization Science* 24(5), 1337–1357.

Morrison, P.D., Roberts, J.H., Von Hippel, E. (2000). “Determinants of user innovation and innovation sharing in a local market,” *Management Science* 46(12), 1513–1527.

Myers, M.D., Newman, M. (2007). “The qualitative interview in IS research: Examining the craft,” *Information and Organization* 17(1), 2–26.

Nambisan, S., Lyytinen, K., Majchrzak, A., Song, M. (2017). “Digital Innovation Management: Reinventing innovation management research in a digital world,” *MIS Quarterly* 41(1), 223–238.

Oborn, E., Barrett, M., Davidson, E. (2011). “Unity in diversity: Electronic patient record use in multidisciplinary practice,” *Information Systems Research* 22(3), 547–564.

Opland, L.E., Jaccheri, L., Pappas, I.O., Engesmo, J. (2020). “Employee-driven digital innovation: A systematic review and a research agenda,” in: Newell, S., Pouloudi, N., & van Heck, E. (eds.) *European Conference on Information Systems*, Virtual Event, Morocco.

Pacheco, J., Garbatov, S., Goulão, M. (2021). “Improving Collaboration Efficiency Between UX/UI Designers and Developers in a Low-Code Platform, ” in: IEEE (eds.) *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. virtual, Japan.

Parizi, R., Prestes, M., Marczak, S., & Conte, T. (2022). “How has design thinking being used and integrated into software development activities? A systematic mapping,” *Journal of Systems and Software*, 187, 111217.

Paré, G. (2004). “Investigating information systems with positivist case research,” *Communications of the Association for Information Systems* 13(1), 18.

Prinz, N., Huber, M., Riedinger, C., Rentrop, C. (2022). “Two Perspectives of Low-Code Development Platform Challenges – An Exploratory Study,” in: Huang, M.-H., Gable, G., Cheung, C. & Xu, D. (eds.) *Pacific Asia Conference on Information Systems* 2022, Virtual Event.

Prinz, N., Rentrop, C., Huber, M. (2021). “*Low-code development platforms–a literature review,* ” in: Aubert B., Paré, G., & Chin, W. (eds.) *American Conference on Information Systems*, Virtual Event.

Rajanen, D., Clemmensen, T., Iivari, N., Inal, Y., Rızvanoğlu, K., Sivaji, A., Roche, A. (2017). “*UX professionals' definitions of usability and UX–A comparison between Turkey, Finland, Denmark, France and Malaysia*,” in: Bernhaupt, R., Dalvi, G., Joshi,A., Balkrishan, D. K., O'Neill, J., & Winckler, M. (eds.) *Ifip Conference on Human-Computer Interaction*, Berlin Heidelberg, Germany.

Richardson, C., Rymer, J.R. (2014). *New Development Platforms Emerge For Customer-Facing Applications.* Forrester Report.

Rosenberg, D. (2018). “The business of UX strategy,” *Interactions* 25(2), 26–32.

Roth, E., Am, J.B., Furstenthal, L., Jorge, F. (2020). *Innovation in a crisis: Why it is more critical than ever*. URL: https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/innovation-in-a-crisis-why-it-is-more-critical-than-ever (visited on August 13, 2020).

Saldaña, J. (2021). *The coding manual for qualitative researchers*, Thousand Oaks, CA: Sage.

Sanchis, R., García-Perales, Ó., Fraile, F., Poler, R. (2019). “Low-code as enabler of digital transformation in manufacturing industry,”*Applied Sciences* 10(1), 12.

Sheth, J.N., Sisodia, R.S., Sharma, A. (2000). The antecedents and consequences of customer-centric marketing,” *Journal of the Academy of Marketing Science* 28(1), 55–66.

Silva, C., Vieira, J., Campos, J.C., Couto, R., Ribeiro, A.N. (2021). “Development and validation of a descriptive cognitive model for predicting usability issues in a low-code development platform,” *Human Factors* 63(6), 1012–1032.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*, Thousand Oaks, CA: Sage

Tan, M.L., Prasanna, R., Stock, K., Doyle, E.E., Leonard, G., Johnston, D. (2020). “Understanding end-users' perspectives: Towards developing usability guidelines for disaster apps,” *Progress in Disaster Science* 7, 100118.

Tisi, M., Mottu, J.-M., Kolovos, D.S., De Lara, J., Guerra, E.M., Di Ruscio, D., Pierantonio, A., Wimmer, M. (2019). “Lowcomote: Training the next generation of experts in scalable low-code engineering platforms,” In STAF 2019 Co-Located Events Joint Proceedings: *1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019) (CEUR Workshop Proceedings (CEUR-WS.org))*. Eindhoven, Netherlands.

Von Hippel, E. (2005). “Democratizing innovation: The evolving phenomenon of user innovation,” *Journal für Betriebswirtschaft* 55, 63–78.

Waizenegger, L., McKenna, B., Cai, W., Bendz, T. (2020). “An affordance perspective of team collaboration and enforced working from home during COVID-19,” *European Journal of Information Systems* 29(4), 429–442.

Wehking, C., Kaiser, L.F., Schenk, B., Vom Brocke, J. (2021). “Orchestration of employees' creativity: A phased approach,” in *Proceedings of the 54th Hawaii International Conference on System Sciences*.

Winter, D., Schrepp, M., Thomaschewski, J. (2015). “Faktoren der User Experience-Systematische Übersicht über produktrelevante UX-Qualitätsaspekte,” in Fischer, H., Endmann, A., & Krökel, A. (eds.) *Mensch und Computer 2015–Usability Professionals*, Berlin: De Gruyter, 31–41.

Yoo, Y., Boland Jr, R.J., Lyytinen, K., Majchrzak, A. (2012). “Organizing for innovation in the digitized world,” *Organization Science* 23(5), 1398–1408.

Zhang, D., Adipat, B. (2005). "Challenges, methodologies, and issues in the usability testing of mobile applications," *International Journal of Human-Computer Interaction* 18(3), 293–308.
Zorina, A., Karanasios, S. (2021). "When IT evolves beyond community needs: Coevolution of bottom-up IT innovation and communities," *Journal of the Association for Information Systems* 22(4), 1146–1175.