Taking the Matter in their own Hands – Can Business Unit Developers Fullfill their Digital Demands with Low-Code Development Platforms?

Katharina Frosch
Department of Economics
Brandenburg University of Applied Sciences
Brandenburg an der Havel, Germany
katharina.frosch@th-brandenburg.de

Olga Levina
Department of Economics
Brandenburg University of Applied Sciences
Brandenburg an der Havel, Germany
olga.levina@th-brandenburg.de

***Abstract*— Low Code Development Platforms (LCDP) often promise an easy and fast way to include data processing and support into the otherwise non-digital process. Nevertheless, it often remains unclear, besides anecdotal evidence, how business users are getting on with transformation of business requirements into the software. This research explores the potential low code development has for business users to address their needs for process support via software tools. The experiment was chosen as the research method to assess the feasibility of software development with LCDP by novices. The results point towards a dedicated LCDP implementation approach if the technology is to be implemented in the business context. Hence, the research provides suggestions on how Business Unit Developers (BUD) can be supported to efficiently deliver productive results and how to assess LCDP-based development process and points towards potential challenges of LCDP implementation.**

Keywords- low code development platforms; software development proces; digital novices; socially-aware software; performance indicators

I. INTRODUCTION

Routine data processing within a process can take up time, which might be required in a more expert context. Nevertheless, integration of a specific software or data layer might not be enough to start a development process due, among others, to the lack of resources in the developer teams.

Low Code Development Platforms (LCDP) promise an easy and fast possibility to include data processing and data exchange support in the otherwise non-digital process [1]. The terms “citizen developer” [2] or “business unit developer” [3] are often used in the LCDP context to underline the potential of the software tools to involve programming novices in the development of the solution for their needs [4].

LCDP allow platform users to develop applications based on a Graphical User Interface (GUI) without creating code and thus to develop programming skills [4]. Each GUI object is programmed in a hard code that can be adapted to some degree of personalization.

This research follows up on this promise with the goal of exploring the potential low code development has for business users to answer their need for support via software tools. The research questions were as follows: Are LCDP feasible for programming novices? And: What aspects need to be considered if an LCDP is provided for the user? Hence, this research provides suggestions on how Business Unit Developers (BUD) can be supported to efficiently provide productive results. This research was not previously published and is the first to be presented here.

The paper is structured as follows: First, the current literature on LCDP and its use in a business context is reviewed, and the research questions are derived in Section 2. Research methods description in Section 3 and results descriptions in Section 4 lead to recommendations on how to implement LCDP in productive environment. Section 5 provides summary and outlook on future research as a conclusion.

II. RELATED WORK

The use of the LCDP in different business domains has been increasingly the focus of research in the last years. Sanchis et al. [5] showed that rapidity and the cost reduction through intuitive development and management can be attributed to the LCDP in manufacturing context. Nowak et al. [6] show case the usage of LCDP in the context of the internal logistics processes in a company from the E-Commerce industry. This case study is meant to display the use of LCDP in the context of process improvement as is allows for direct eliminations of found limitations in processes. The authors argue that the implementation of the IT support using LCDP was effective, #, an enhancement in terms of time and costs needed for its realization.

Bies et al. [7] conduct a mixed-method study to identify challenges and promising perspectives for digital innovations in Small and Middle Enterprises (SME). The authors found the LCDP application areas are mostly of the supportive nature such as creation of application for resource management or creation of customized digital forms. Nevertheless, the majority of the surveyed SMEs stated LCDP to be of high to very high relevance. Factors that diminish the relevance for low-code in SME are according to the authors: limited human resources, as personnel is still necessary to develop and maintain the application, knowledge transfer between the platforms as well as training in dealing with IT structures and detailed knowledge of the platforms.

Lethbridge [8] also explores the development process of the software product as well as the aspects of implementation and maintenance of the LCDP software within the existing enterprise architecture. His findings suggest that LCDPs create “technical debts” that can be overcome by the

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-077-3

development of the LCDP towards "scaling, understandability, documentarily, usability, vendor-independence and user experience for the developers". Hintsch et al. 2021 [9] also identify threats and opportunities in the LCDP development concerning the security and availability of the created applications. Nevertheless, the authors also uncover success factors for LCDP use in a business context by novices.

Kermanchi et al. [10] focus in their research on software development methods and the use of LCDP. In their experiment, they explored the episodic experience with different LCDP among software developers with varying levels of programming experience but no experience in the specific LCDP. The findings show that previous programming experience seems to have a significant impact on developers' performance, experience, and tool preferences, yet most developers continue to have doubts about the scalability and maintainability of applications created with LCDPs. Opinions on the effectiveness of the instruments vary among the participants.

Bernsteiner et al. [11], conduct expert interviews in their research to investigate what skills developers with little or no software development experience, i.e., novices, need to successfully develop software on LCDP platforms. Several of the interviewed experts mention that successfully developing an LCDP solution requires at least basic programming skills. This is in line with research findings stating that LCDPs still require some prerequisites in software development [12] or in database structures [13], which hampers the adoption of LCDP by non-programmers without any further training.

Krejci et al. [13] report in a case study how non-IT employees were involved in the process of digital innovation while making efficient use of their IT resources. These citizen developers, i.e., employees who are working outside of the Information Technology (IT) department and are not professional programmers, as users of LCDP are in the focus of the analysis by Lebens et al. [14]. The authors conducted a survey about the use of LCDP in organizations. The results show that companies both large and small are making use of low- and no-code platforms. Additionally, the majority of the surveyed organizations have employees outside of the IT department who are creating IT solutions.

Bock and Frank [15] provide a critical overview of the LCPD features, architecture, and opportunities, while pointing out research directions for information systems research in this domain. They state that although both professional developers and citizen developers use LCDP, there is a lack of research on how to make LCDPs fit cognitive capabilities and personal working styles of these two groups [p. 739]. This is in line with other studies pointing out that successfully developing software on LCDP requires at least basic programming skills.

The use of development templates in the context of software creation is analyzed by Boot et al. [16]. The authors compare instructional software products made by developers with low production experience and high production experience, working with a template-based authoring tool. The analysis showed that the technical and authoring quality was equal for both groups, indicating that templates enable domain specialists to participate successfully in the production process. Research in agile software development shows that SCRUM projects profit from having a coach on the team [17]. The same is visible in software engineering education [18].

BUD and job crafting, i.e., proactive strategies to improve work processes according to one's own needs and goals, are subjects of the analysis by Li et al. [3]. The authors found that using LCDP provides positive jobs crafting consequences such as meaningfulness, for the employees using these tools [3][19]. In what follows, we prefer to use the term BUD instead of citizen developers, stressing that they might make up for the lack of programming skills by their large expertise in the respective business domain. Nerveless, the research does not focus on the description of how much support was needed for BUD to finish their application.

Despite these first attempts to understand the "human side" of LCDPs, research is still scarce with respect to acceptance and successful adoption by domain experts outside corporate IT departments. To our knowledge, there are no empirical studies yet to gain a deeper understanding of how BUD fare when using LCDPs. With our study, we contribute to closing this gap and explore:

- Whether BUD can develop functional applications based on LCDPs to improve their business operations
- Whether the amount of time invested and developing behavior differ between BUD and IT experts when using an LCDP to develop applications

III. RESEARCH METHOD

To gain evidence for answering our research questions, we draw upon a field experiment where BUD and IT experts build apps in the business domain of Human Resource Management (HRM) based on an LCDP given a finite time frame of few weeks. The experiment was divided in three challenges with modified compositions of participants. The challenges are described below. For the experiment, BUD are Master students of business management with the specialization in HR (20 students). In the third challenge, BUD were included in teams with experts. The experts were Master students of Information Systems Research (ISR) (18 students). All of the ISR students had already taken at least one course in advanced software engineering within their master program at the time of the experiment, thus gaining the definition as "IT experts". None of the participants was familiar with or has heart of the LCDP selected for the experiment.

The LCDP used for the experiment was Joget [20], an open-source LCDP with the promise to easily build, run and maintain apps. A visual builder allows drag-and-drop for pages, forms, views, data lists, menus, and a process builder to automate workflows. It also offers user management and role-based authentication. We used the community edition that can be self-hosted at no license cost.

The experiment is divided into three self-contained challenges. Challenge #1 was run with a few BUD only, in order to have a pretest and check whether business students are, at all, able to use the LCDP to develop simple apps. The pretest was run between April 21 and June 6, 2021. To kick start app development, BUD were provided with links to tutorials as well as with a basic app template and a 30-min-video showing exemplarily how an app can be built starting from this template. In this context, they were also explicitly pointed to the open-source character of app development in this setting, and about the possibility to share and reuse app elements from other groups. In the pretest, BUD managed to develop apps, but pointed out that they would have enjoyed working in teams in order to solve problems collaboratively. Furthermore, support by one student who previously had graduated from a bachelor program in software engineering and acted as informal coach for his fellow students has been acknowledged as extremely helpful.

Based on the insights gained in the pretest, we recruited the informal coach from challenge #1 to act as a formally appointed coach in challenge #2 and decided to run development in teams. For Challenge #2: BUD teams (with three to four students) developed their apps within six weeks (April 21 – June 6, 2021, 42 days). The team members cooperated online, due to the restrictions because of the COVID-19 pandemic. Developers got the same kick start as in the pretest and were also pointed towards the template and the possibility to share and reuse apps. Furthermore, a coach with experience in software development was available to get help with questions on tool usage and minor development questions. In Challenge #3, expert teams (including four to seven students) developed their apps between May 20 and June 7, 2022 (19 days). During the development challenge, two teams joined forces within the development process, resulting in a seven members team working on the challenge. The first day of the development phase (May 20, 2022) was organized as a face-to-face daylong hackathon. The introductory video and tutorials were made available beforehand, but no template or coach were provided for the teams.

IV. Results

The experiment has shown that in all three challenges, BUD were able to create a software application using LCDP in a given amount of time without any additional training in software development. All 15 apps created during the challenges have been successfully developed and implemented. Successful means that they met the requirements depicted in the conceptual papers, and that apps worked when tested. The technology readiness of the resulting apps corresponds to level 3 (experimental proof of concept) according to the European Union Technology Readiness Levels [21].

To address the second research question, the logs were archived and anonymized to calculate time spent at the platform and number of actions taken to create an app. The development activities of challenge #2 and #3 were then compared using indicators for time spent on platform as well as number of actions taken (per developer and per app, respectively). Overall, our data comprises 320 logins by BUD and 206 logins by experts, resulting in 4895 and 4094 actions taken, respectively. Figure 1 shows that the distribution of time spent on the platform, logins and actions taken is right-skewed, with most developers investing not more than 10 hours in development. Moreover, in both the BUD and the expert group, we observe one outlier with more than 60 (BUD) and more than 30 (expert) hours, respectively. As comparing means given such a data structure may lead to misleading results, we use modal values to compare development activities between BUD and experts.

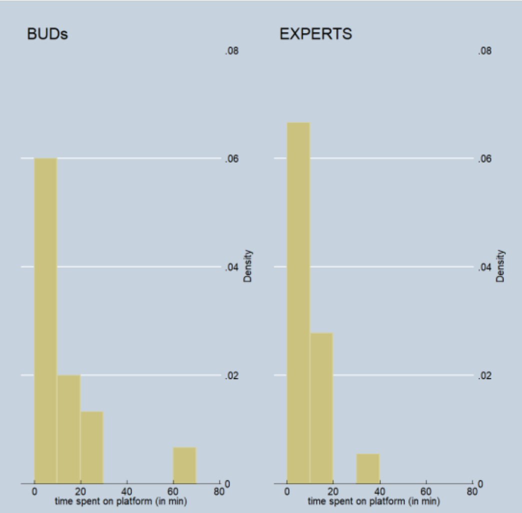

Figure 1. Time spent on platform

Table 1 shows that BUD tend to spend more time on the LCDP platform, but the total time investment per app is somewhat higher in the expert group. This result may be partly explained by the fact that on average, BUD teams were smaller than expert teams. BUD take fewer actions per app and per person as compared to experts. None of the indicators, however, shows statistically significant differences between the expert and the BUD groups when conducting a t-test on mean differences, also not when leaving out the extreme outlier in the BUD group (not displayed in Table 1).

TABLE I. Modal Values for Development Activity Indicators

Indicator	BUD	Experts
Time spent on platform (h), per developer	6.5	4.3
Time spent on platform (h), per app	23.7	30
Number of actions taken, per developer	152	165
Number of actions taken, per app	807	946

To conclude, we can state that BUD can create their own software applications in their business domain using an LCDP, and that time and effort invested in development are not significantly different from that of developers with programming knowledge.

V. Conclusion

The research question of this paper was whether LCDPs are a suitable tool for BUD to develop a digital solution that meets their requirements for information and data management within the business process without the involvement of the IT department. To answer this question, an experiment with three different challenges was conducted. All the solutions for the challenges led to an app that was ready to be implemented in the business context. Although, the quality of the created artifacts was not measured, and the size of the developer groups varied, the research offers valuable insights on the development process using LCDP by both non-IT and IT-trained users. Furthermore, some approaches were identified that might support BUD in the first steps during their engagement with the tool. Here, the use of templates and the availability of a coaching person is suggested.

In addition, this paper presented some indicators to measure LCDP performance within the software development process. The results can be used by managers and practitioners to support an effective and successful LCDP implementation. The applied research method can be expanded by HR and ISR researchers to support their conceptual artifacts in a low-code development context with data. Also, the suggested indicators can be used to assess the process performance of the software development with LCDP.

The experimental setting provided a near real life situation that allowed assessing the interaction with LCDP as well as resulted in interactional data that will be used to derive further insights on the LCDP-based business software development. Nevertheless, the group work made it more difficult to derive explicit indicators, so that future research will be based on individual software creation in collaborative stings. We will focus on the development of further interaction metrics for the assessment of the impact of LCDPs on the working styles of BUD and experts. Here, we will look at the engagement and interaction efficiency with the LCDP across the groups of experts and BUD. Another future research direction will focus on the job crafting effects of LCDP-based development for BUD and experts. Here, motivational and engagement aspects will be guiding the development of metrics to allow further comparative analysis.

REFERENCES

[1] S. Rafi, M. A. Akbar, M. Sánchez-Gordón, and R. Colomo-Palacios, “DevOps Practitioners’ Perceptions of the Low-code Trend,” *Int. Symp. Empir. Softw. Eng. Meas.*, pp. 301–306, Sep. 2022, doi: 10.1145/3544902.3546635.

[2] K. Rokis and M. Kirikova, “Challenges of Low-Code/No-Code Software Development: A Literature Review,” in *Lecture Notes in Business Information Processing*, 2022, vol. 462 LNBIP, pp. 3–17, doi: 10.1007/978-3-031-16947-2_1.

[3] M. M. Li, C. Peters, M. Poser, K. Eilers, and E. Elshan, “ICT-enabled job crafting: How Business Unit Developers use Low-code Development Platforms to craft jobs,” *ICIS 2022 Proc.*, Dec. 2022, Accessed: Feb. 03, 2023. [Online]. Available: https://aisel.aisnet.org/icis2022/is_futureofwork/is_futureofwork/16

[4] K. Talesra and N. G. S., “Low-Code Platform for Application Development,” *Int. J. Appl. Eng. Res.*, vol. 16, no. 5, p. 346, May 2021, doi: 10.37622/IJAER/16.5.2021.346-351.

[5] R. Sanchis, Ó. García-Perales, F. Fraile, and R. Poler, “Low-code as enabler of digital transformation in manufacturing industry,” *Appl. Sci.*, vol. 10, no. 1, p. 12, Dec. 2020, doi: 10.3390/app10010012.

[6] F. Nowak, J. Krzywy, and W. Statkiewicz, “Study on the Impact of the Use of No-code Application on Internal Logistics Processes in a Company from the E-Commerce Industry - Process Analysis,” *Eur. Res. Stud. J.*, vol. XXV, no. Issue 2B, pp. 59–71, Aug. 2022, doi: 10.35808/ERSJ/2936.

[7] L. Bies, M. Weber, T. Greff, and D. Werth, “A Mixed-Methods Study of Low-Code Development Platforms: Drivers of Digital Innovation in SMEs,” *Int. Conf. Electr. Comput. Commun. Mechatronics Eng. ICECCME 2022*, 2022, doi: 10.1109/ICECCME55909.2022.9987920.

[8] T. C. Lethbridge, “Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13036 LNCS, pp. 202–212, 2021, doi: 10.1007/978-3-030-89159-6_14/COVER.

[9] J. Hintsch, D. Staegemann, M. Volk, and K. Turowski, “Low-code Development Platform Usage: Towards Bringing Citizen Development and Enterprise IT into Harmony,” *ACIS 2021 Proc.*, Jan. 2021, Accessed: Feb. 06, 2023. [Online]. Available: https://aisel.aisnet.org/acis2021/11.

[10] A. Kermanchi, S. P. Fabian, F. Advisor, and A. Teronen, “Developer Experience in Low-Code Versus Traditional Development Platforms - A Comparative Experiment,” Dec. 2022, Accessed: Feb. 03, 2023. [Online]. Available: https://aaltodoc.aalto.fi:443/handle/123456789/118413.

[11] F. Bernsteiner, R., Schlögl, S., Ploder, C., Dilger, T., and Florian Brecher, “Citizen vs. Professional developers: Differences and Similarities of Skills and Training Requirements for Low Code Development Platform,” in *ICERI2022 Proceedings*, 2022, pp. 4257–4264.

[12] A. Sahay, A., Indamutsa, A., Di Ruscio, D., and Alfonso Pierantonio, “Supporting the understanding and comparison of low-code development platforms.,” in *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2020, pp. 171–178.

[13] D. Krejci, L. Küng, and S. Missonier, “A Case Study of Enterprise-wide Digital Innovation: Involving Non-IT Employees,” *ECIS 2022 Res. Pap.*, Jun. 2022, Accessed: Jan. 23, 2023. [Online]. Available: https://aisel.aisnet.org/ecis2022_rp/55.

[14] M. Lebens, R. J. Finnegan, S. C. Sorsen, and J. Shah, “Rise of the Citizen Developer,” *Muma Bus. Rev.*, vol. 5, pp. 101–111, 2021, doi: 10.28945/4885.

[15] A. C. Bock and U. Frank, “Low-Code Platform,” *Bus. Inf. Syst. Eng.*, vol. 63, no. 6, pp. 733–740, Dec. 2021, doi: 10.1007/S12599-021-00726-8/FIGURES/1.

[16] E. W. Boot, J. J. G. Van Merriënboer, and A. L. Veerman, “Novice and experienced instructional software developers: Effects on materials created with instructional software templates,” *Educ. Technol. Res. Dev.*, vol. 55, no. 6, pp. 647–666, 2007, doi: 10.1007/s11423-006-9002-9.

[17] C. Bunse, I. Grützner, C. Peper, S. Steinbach-Nordmann, and G. Vollmers, “Coaching professional software developers an experience report,” *Softw. Eng. Educ. Conf. Proc.*, vol. 2006, pp. 123–130, 2006, doi: 10.1109/CSEET.2006.11.

[18] H. I. Akyüz and M. Kurt, “Effect of teacher’s coaching in online discussion forums on students’ perceived self-efficacy for the educational software development,” *Procedia - Soc. Behav. Sci.*, vol. 9, pp. 633–637, Jan. 2010, doi: 10.1016/J.SBSPRO.2010.12.209.

[19] E. Elshan, E. Dickhaut, and P. Ebel, “An Investigation of Why Low Code Platforms Provide Answers and New Challenges,” Accessed: Mar. 01, 2023. [Online]. Available: https://hdl.handle.net/10125/103380.

[20] Joget, “Open Source Platform to Easily Build, Run and Maintain Apps”, Accessed: Mar 7, 2023. [Online]. Available: https://www.joget.org.

[21] European Commission, *Technology readiness levels (TRL)*, 2014, Accessed: Mar 7, 2023 [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf