Developing a novel application to digitalize and optimize construction operations using low-code technology

Eder Martinez[1,2], Louis Pfister[2], and Felix Stauch[2]

[1]School of Architecture, Civil Engineering and Geomatics, University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Switzerland
[2]Division Civil Engineering, Implenia AG, Switzerland
eder.martinez@fhnw.ch, louis.pfister@implenia.com, felix.stauch@implenia.com

Abstract

Low-code is an emerging technology allowing subject experts with no information technology background to develop their own digital solutions to operational challenges. Low-code has shown potential to support digitalization in different business scenarios. Nevertheless, there is a lack of scientific literature exploring how this technology could be leveraged in the construction industry. This paper addresses this gap using case study research to analyze the development and implementation of a novel low-code application to monitor production in tunneling construction. The outcomes reveal that the involvement of people closer to operation in the low-code development process supports productive development practices enabling the delivery of an application meeting project requirements and supporting on-site adoption. The introduction of the application resulted in an increase in productivity during the reporting process. The innovative use of low-code technology in the construction industry also reveals opportunities for future research.

Keywords –

Low-code, Citizen developer, Low-code development platform, Lean construction, Building Information Modeling, Construction industry, Digitalization.

1 Introduction

Low-code development is an emerging technology aiming to enable people without coding or programing expertise to develop applications and digital solutions without the support of Information Technology (IT) professionals. Richardson et al. [1] coined the term low-code in 2014 arguing that it emerged as a response to rapid changing business environments and requirements for faster and cheaper technology development [2, 3].

In comparison to traditional programming methodologies, low-code seeks to reduce manual coding efforts and lower technical entry barriers in technology development, thereby fostering a more efficient and productive development. Low-code abstracts complex programming into an application modeler equipped with an intuitive graphical interface containing pre-defined components/templates to support the application development process. Users can drag and drop elements onto the graphical interface or connect directly to related databases without the need of writing code [4]. Embracing this programming approach empowers a distinct group of individuals, commonly identified as "citizen developers", to actively create their own digital solutions for addressing operational challenges [5, 6].

Low-code is usually seen an evolution of principles within Computer-Aided Software Engineering (CASE) [7]. For instance, Bucaioni et al. [8] argue that low-code should regarded as a set of tools and techniques within the domains of Model-Driven Engineering (MDE). Some authors even contend that the rise in popularity of the low-code trend may not in itself offer significant technical innovation, suggesting that its rise may be due to a rebranding of related CASE/MDE concepts [9].

Low-code offers an alternative to the ever-increasing demand of digitalization, particularly in light of the IT skills shortage [4, 10, 11]. This considering that in the next 5 years, the digitalization of different industries will demand the development of more than 500 million applications and digital solutions, exceeding the delivering capacity of IT organizations and developers [12, 13]. In this regards, market reports are very adventurous forecasting that by 2025 nearly 70% of business applications will be developed using some sort of low-code technology [14], with the number of “citizen developers” largely surpassing the number of professional developers [15]. This translates into an expected market growth from USD 13.89 billion in 2021 to USD 94.75 billion by 2028 [16].

If these projections materialize, low-code will influence not only digitalisation of various industries but also the way projects are managed. In fact, the Project Management Institute (PMI) has proactively initiated

educational programs aiming to train project managers to leverage the use of this technology [17].

Low-code is also considered in international research consortiums, such the Horizon 2020 Lowcomote project [18]. Lowcomote aims to equip the next generation of professionals, empowering them to emerge as leaders in the future engineering of low-code development platforms. There are also emerging competence centres such as LowCodeLab@OST [19] in Switzerland seeking to focus on building up strategic citizen development and low-code skills.

Since the introduction of the concept in 2014, low-code technology has been used to support digitalization in a variety of business scenarios. Nevertheless, there is a lack of literature delving into low-code use in the context of processes and operations in the construction industry. This paper contributes to filling this gap by describing and analyzing a case study where low-code was used to develop an application that supports specific construction tasks in the context of tunneling operations.

2 Related literature

Prinz et al. [20], Bucaionni et al. [8], and Pinho et al. [21] offer recent literature reviews about low-code. These studies highlight the limited literature on low-code as well as underline the growing interest of the academic community to fill the gap and assimilate the knowledge generated by the industry. Prinz et al. [20] point out that most of research focuses on the technical aspects of low-code platforms. However, social aspects related to roles and responsibilities, alignment of processes, and communication related to the deployment of low-code in business environment has received little attention. Bucaionni et al. [8] reveal that almost half of sources correspond to grey literature, with the first scientific publication appearing in 2018, four years after the introduction of the low-code term. The authors recommend further investigating the characteristics and nuances of applications created through low-code development. This aims to assess the current state of low-code development practices and analyze broader topics such as scalability and performance metrics. Pinho et al. [21] focus the analysis into advantages and disadvantages in low-code development with particular emphasis on platforms' usability. The authors make notes of caution related to the prominent presence of industry and low-code vendors in literature. This considering that low-code vendors have a strong incentive to enhance their products, making them potentially influential in driving research in this field. Although recognizing them as catalysts for advancing research in this domain can be advantageous, it is crucial to prioritize scientific rigor in peer-reviewed literature, and the primary motivation for their involvement should not solely revolve around business success.

Based on a systematic review, Martinez and Pfister [22] provide an overview of relevant low-code literature and classify articles using thematic analysis. The articles were clustered in 5 themes: technical, interfaces with contemporary topics, implications to software development, literature reviews, and business applications. This analysis a) confirms the growing scientific interest on low-code, b) confirms that most of the literature focuses on technical aspects of low-code development platforms, c) consolidates a variety of use cases describing low-code use in different business scenarios, and d) reveals the lack of scientific work exploring low-code in the context of the construction industry.

Only a handful of studies address the implications of low-code to business operations, and overall digital transformation. Waszkowski [3] describes the use of a low-code platform for automating business processes in a manufacturing environment. The author emphasizes that one of the main benefits of low-code is the reduction of time in the transfer of requirements from the end-user to the IT developers. This is because with low-code, the citizen developer can play the dual developer/end-user role. In the same direction, Wolff [23] states that manufacturing is very suited for low-code because most of the engineers running business processes are familiar with programming language, thus low-code environment is not fully foreign to them. The author also describes popular low-code manufacturing applications in supply chain, inventory tracking, logistics, and sales. From a broader perspective, Sanchis et al. [24] explore the suitability of low-code to enable digitalization in manufacturing. The authors analyze the status of research and benchmark different low-code platforms to identify challenges and opportunities.

The use of low-code in business scenarios has enabled the identification of different benefits. Practitioners identify faster development, ease of use, and low development cost as the main benefits of low-code [25–27]. On top of reducing hand-coding, the assisted low-code process facilitates data integration, provides space for testing and experimenting new ideas, as well as becomes a single control point for configuration and application maintenance [1]. Richardson and Rymer [28] state that low-code development process can be 5 to 10 times faster than traditional approaches. Industry surveys report that software developers are embracing low-code to accelerate the digital transformation, increase response to changing business requirements, and reduce reliance on hard-to-hire IT developers [29].

One of the main limitations of low-code platforms relates to restricted customization options [25, 30]. The lack of customization derives from the fact that low-code

development is naturally less powerful than traditional programming, and users are limited to the options offered in the low-code platform. Tisi et al. [31] also identify scalability and fragmentation as low-code shortcomings. Scalability refers to the ability to expand the use of a given application to larger systems, projects, or organizations. Fragmentation relates to limited interoperability among low-code platforms and supporting databases. Some organizations hesitate to fully embark into low-code because of the high cost per user [25], and concerns about potential vendor lock-in [29]. Although low-code platforms are intended to be intuitive, users and even IT professionals also require investing time in learning and getting familiarized with a platforms user interface [26]. In more complex scenarios, a citizen developer may be unable to meet demanding requirements, eventually requiring support from software developers [29].

Literature on low-code use in the construction industry context is particularly limited. Martinez et al. [32] proposed the idea of using low-code to support the delivery of construction project in a book chapter involving lean and construction industry 4.0. Expanding on this idea, Martinez and Pfister [22] introduced the first academic work exploring the use of low-code in the context of construction base analyzing the outcomes of multiple case studies. Martinez and Cisterna [33] provide another use case integrating low-code with artificial intelligence.

This study contributes to the emerging body of knowledge by extending the analysis and lessons learned from a prior case study, now considering information available after 18-months of application use.

3 Research methodology

This study uses Case Study research design and adheres to the methodologies suggested by Yin [34]. Case study research is an empirical inquiry that investigates a contemporary phenomenon within a real-life context leveraging the use of multiple sources of data and evidence [34]. In this context, the authors acted as citizen developers supporting the project team in the collection of requirements and developing the application using a low-code platform. The authors carried out a comprehensive study of the interactions that took place throughout the entire development process. This involved analysing various project documents, including meeting minutes and paper data collection forms. They also undertook a comparative assessment, contrasting the current manual process with the improved process incorporating the application developed by the project team.

4 Case study

4.1 Context

The case study involves a tunnelling project using different underground construction techniques. The initial construction phase of the project considers earthmoving operations to remove soft soil in front of the planned tunnel section opening. In a second phase, the project considers 500 m. of tunnelling excavation using the umbrella pipes technique. The final phase, considering more stable geology, involves 3.500 m. of tunnelling construction using drill and blast technique.

Traditionally, the reporting of activities and resources used in different operations is done using paper forms. The information included in these forms considers the different activities and corresponding execution times, as well as data about different resources used (e.g., machines and personnel hours, key materials, and subcontractors). This data is then manually transferred to a spreadsheet where different analyses can be performed.

This manual process to capture and process data has the typical shortcomings of a manual paper-based approach. Manual data processing is time-consuming and error-prone. In addition, it often results in delays in data processing, preventing managers and production teams from accessing real-time project performance information that is critical for timely and fact-based decision making. In this project, three daily shifts are tasked with managing the different paper forms.

4.2 Application development process

The project team aimed to tackle the shortcomings of paper-based project reporting by digitalizing and automating the process of data capturing and handling through a customized mobile application. For this purpose, the project team conducted a series of workshops to understand the data capturing process, identify relevant data inputs, and define the requirements of a digital solution. Along with reducing manual effort for data processing, the team aimed to have real time information about activities and resource utilization which facilitates the identification of improvement opportunities to increase productivity.

The authors acted as “citizen developers” supporting the team in the elicitation of requirements and the materialization of the application in the project digital collaboration environment. In this exercise participated the project manager, site manager, and several key project members. From the workshops, the team was able to consolidate requirements to replace the paper forms used in the project and derive all relevant relationships to build a data model that enables real time generation of project performance reports. The team also defined iterative loops to review prototypes to verify the data

model and improve the application's user interface.

4.3 System architecture

The architecture of the system is depicted in Figure 1. The digital solution is built on Microsoft Power Platform. Microsoft Power Platform is a set of cloud applications that allows automation of business processes and create application with low-code [35]. The data capturing and processing concept developed in this case study combines different components of this platform including PowerApps, Dataverse, and PowerBI. The application development process did not require manual code writing; rather, it involved integrating various functions to establish connections between data and facilitate interactions among pre-defined components.

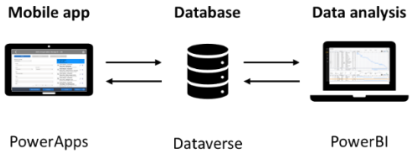

Figure 1: System architecture

In Figure 2, primary users report activities and resources through a Graphical User Interface (GUI) built in PowerApps. PowerApps is a suite of apps, services, connectors, and a data platform offering a rapid application development environment to realize custom applications tailored to specific business needs [36]. PowerApps allows connecting data stored Microsoft Dataverse (or in related online data sources such as SharePoint, Dynamics 365, and SQL Server).

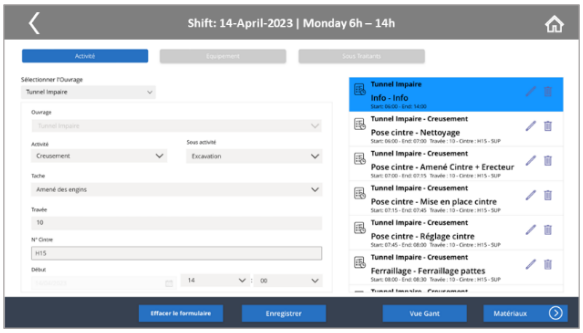

Figure 2: Graphical User Interface

The GUI is connected to the Microsoft Dataverse via integrated data connectors predefined in the Power Platform. From the mobile application data inputs are therefore sent and stored in a Dataverse. Dataverse allows to structure data in a set of cloud-based tables including relationships. In Figure 3, the data model includes two groups of data: master and transactional. Master tables contain predefined project data such as materials, type of machines, shifts identification number, personal, etc. This data can only be edited by certain key users at the project site and must be updated on a regular basis according to project needs. For example, a new machine arriving to the project must be added to the corresponding table to appear in the GUI.

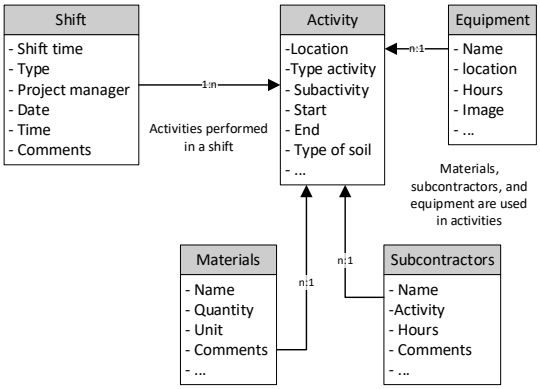

Figure 3: Data model

Transactional data complements master data and corresponds to all process inputs captured onsite through the mobile application. For instance, an application user selects a machine stored in the master data and complements this information with the number of hours it was used during the shift. Transactional data is captured though a series of views in the mobile application and sent to the database where relationships among different data entries are defined via unique identification numbers.

The different views included in PowerApps GUI emulate the paper forms intended to be replaced. The GUI is designed to walk the user through a logical workflow to input data in an organized way. Users begin by entering data about the shift (e.g., shift number, start and end time, location, project manager, etc.). In the background the creation of a shift generates a unique ID number where all the other inputs are attached. The user then can add further transactional data connected to the shift entry by entering the details in the corresponding GUI screen. This allows establishing several one-to-many relationships between the shift and activity/resources tables in the data model.

PowerBI is used for report generation, data analysis, and overall business intelligence. Through interconnected queries and established data relationships, PowerBI retrieves and combines data to analyze and generate the visual insights needed by the project team.

4.4 Outcomes and findings

The project team, working part-time over three months, successfully delivered a functional application through iterative development loops. The authors

highlight efficient requirement elicitation and meeting end-user needs as the primary benefits of low-code. Citizen developers with operational knowledge played a key role, articulating requirements, optimizing development, and bridging exchange between the project team and operations, resulting in faster validation loops.

In terms of process optimization, the materialization of the application removed manual steps from the process, enabling significant saving in personnel time. Table 1 presents estimated times to perform different tasks in the data capturing, controlling, and analysis process. These times were obtained through semi-structured interviews with users on-site. Site engineers, who are responsible for reporting data onsite, save approximately 35 min per shift. That is, 105 min of time savings per day (3 shifts / day). The project manager performs the task of controlling, analyzing data, and reporting daily. For these tasks, the project manager saves approximately 110 min per day. Approximately, the project saves a total of 215 min per day. That translated into an estimate of 235.425 min (3.924 hours) for the 3 years of project duration (215 min/day × 365 days × 3 years). Furthermore, there is also the implicit value of having project data processed in real time to make informed decisions about the course of the project. For instance, by allowing the team to promptly define corrective actions when detecting important deviation in the execution of activities and use of related key resources.

Table 1: Estimated time savings

Personnel	Savings (min/day)	Description
Foreman		
Report filling & data processing	105	No need to write on paper and manually digitalize data
Project Manager		
Controlling data	20	Control quality of data inputs
Extracting relevant data	60	Machine, materials, data is readily available
Report creation	30	Processing of data to generate report
Total	**215**	

In terms of challenges, a limitation associated with low-code development is the restricted ability for customization. The development process is restricted to the predefined options and structures available in a specific platform, potentially limiting the flexibility of development. In this case study, the project's requirements were consistently evaluated in relation to the capabilities of the low-code platform and the expertise of citizen developers. This approach was explicitly adopted to avoid the need for customizing applications through manual coding, which would require professional IT support. The requirements for digitizing the data capture process in this case did not necessitate customizations beyond what the platform could offer, which may not be the case in other projects.

In terms of scalability, the project team initially considered the development of the data model through basic online lists (SharePoint lists). However, as the data model's complexity grew, a shift to a relational database became essential to streamline the management of relationships among tables. This challenge led to subsequent development iterations and additional licensing costs. This reveals the necessity for careful consideration during the development process, addressing not only user interface requirements but also the comprehensive system architecture defined for the digitalization initiative.

5 Discussion

The authors explore broader industry implications of low-code, specifically its role in supporting the digitalization in the construction sector.

5.1 One-of-a-kind nature of construction

Low-code development offers opportunities to facilitate digitalization of the construction industry processes since it has the potential to cope with the so-called one-of-a-kind nature of construction business. Processes in the construction industry lack standardization because they are usually influenced by site conditions. As a result, information technology tools are unable to cope with a variety of requirements to match every business case, affecting their applicability on site and broader use in other projects. In this aspect, low-code has the potential to support the development of tailored applications with low effort and advanced engagement of people in the field. This could enable different degrees of digitalization of specific construction operations. Related literature highlights the potential of low-code to enable a bottom-up innovation process [37].

5.2 Digitalization and innovation

Several processes on construction sites are still planned and controlled using pen and paper, thus low-code could facilitate their digitalization. Traditionally, the data generated in a project is not properly used and learnings are hardly transferred from one project to the other. In this aspect, low-code offers an alternative to capture key process data and use it to analyze and improve operations. At a project level, teams can have real time access to project performance metrics, identify bottlenecks, and define actions to improve overall

productivity of key processes. At an enterprise level, construction firms could consolidate data coming from different projects and leverage the use of this information to improve performance in other phases of the project life cycle. For example, by consolidating and analyzing data from similar projects, an organization can generate a solid performance database and productivity indicators to improve accuracy when bidding on new projects. In this direction, the authors suggest exploring opportunities for further research on the impact of low-code in areas like Lean Construction, Integrated Project Delivery, Building Information Modeling (BIM), and Construction 4.0. For instance, it would be interesting to analyze how low-code platforms can interact with the Industry Foundation Classes in BIM.

5.3 Computing literacy

Although low-code platforms are intended to be intuitive, the citizen developer still requires some level of technology literacy and basic data modelling knowledge to deliver a working digital solution. This seems to be contradictory with low-code platform vendors and market research statements implying that basically anyone can build applications without software expertise. It is certainly true that any individual could use a low-code platform to build digital solutions. However, the realization of the value of a particular solution relies on how it fits the organizational context, processes, and people interacting with the technology [38]. The role of the citizen developer is therefore key for this purpose and discussions about training programs to equip them with the required skills already started in modern industrial engineering and project management education [17, 39]. In this aspect, exploring what are the necessary skills and competences of a citizen developer to leverage low-code in the context of the construction industry offers an interesting area for further research.

5.4 Traditional versus low-code programing

Low-code development is commonly positioned as a faster and more cost-effective alternative to conventional coding practices. While it is undeniable that low-code offers apparent advantages when compared to traditional approaches, such as the expedited elicitation of requirements leading to potential reductions in both development time and costs, it is crucial to acknowledge the limited body of literature establishing robust foundations for these statements. Estimating software development time and cost is challenging considering the many variables to consider [40], and traditional approaches may not suit the emerging low-code development paradigm [41]. Conducting a comparative analysis between traditional and low-code programming, with a focus on predefined parameters such as time and cost, presents a great opportunity for future research.

6 Limitations

The study's findings are derived from a single case using a specific low-code platform. To generalize these results, additional studies with similar contexts are needed. In the case study, the project team streamlined the digitalization process by already having a low-code platform integrated into their internal organization’s IT system, eliminating the need for evaluation and introduction.

7 Conclusions

The paper contributes to the emerging body knowledge related to low-code use in the construction context by documenting the development and implementation of a novel low-code based application to track project progress.

By lowering the technical barriers to programing, low-code enabled the involvement of people closer to operation in the development process. This enables productive development practices and agility to derive and application that meets the requirements of the project user and facilitates the adoption of the technology on-site. The implementation of the low-code based application allowed significant reduction of the manual work required to capture and process data in the reporting process. Promising results emerge in leveraging low-code for construction operations digitalization. However, it's crucial to address and explore limitations in customization, scalability, and organizational aspects associated with low-code implementation. The novelty of this topic also allows the identification of several areas for further research. For instance, overlaps with Lean Construction, Integrated Project delivery, Building Information Modeling, and Construction 4.0. are interesting areas for future research.

8 Acknowledgments

The authors would like to thank the project team members that participated in the development process of the application discussed in this study. Their contributions to the digitalization initiative, as well as their inputs to the whole process, significantly enriched the findings of this study.

9 References

[1] Richardson, C., Rymer, J. R., Mines, C., Cullen, A., & Whittaker, D. (2014). *New development platforms emerge for customer-facing applications*.

On-line: https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411

[2] Fryling, M. (2019). Low code app development. *Journal of Computing Sciences in Colleges*, *34*(6), 119. https://dl.acm.org/doi/abs/10.5555/3344051.3344061

[3] Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, *52*(10), 376–381. https://doi.org/10.1016/j.ifacol.2019.10.060

[4] Sahay, A., Indamutsa, A., Ruscio, D. D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. In *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 171–178). https://doi.org/10.1109/SEAA51224.2020.00036

[5] Oltrogge, M., Derr, E., Stranksy, C., Acar, Y., Fahl, S., Rossow, C., … Backes, M. (2018). The Rise of the Citizen Developer: Assessing the Security Impact of Online App Generators. In *IEEE Symposium on Security and Privacy (SP)* (pp. 634–647). https://doi.org/10.1109/SP.2018.00005

[6] Everhard, J. (2019). The Pros And Cons Of Citizen Development. Retrieved August 3, 2021, On-line: https://www.forbes.com/sites/johneverhard/2019/01/22/the-pros-and-cons-of-citizen-development/?sh=4d04dfae84fd

[7] Bock, A. C., & Frank, U. (2021). Low-Code Platform. *Business & information systems engineering*, *63*(6), 733–740. https://doi.org/10.1007/s12599-021-00726-8

[8] Bucaioni, A., Cicchetti, A., & Ciccozzi, F. (2022). Modelling in low-code development: a multi-vocal systematic review. *Software and systems modeling*, *21*, 1959–1981. https://doi.org/10.1007/s10270-021-00964-0

[9] Cabot, J. (2020). Positioning of the low-code movement within the field of model-driven engineering. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (pp. 1–3). https://doi.org/10.1145/3417990.3420210

[10] Breaux, T., & Moritz, J. (2021). The 2021 software developer shortage is coming. *Communications of the ACM*, *64*(7), 39–41. https://doi.org/10.1145/3440753

[11] Carroll, N., & Maher, M. (2023). How Shell Fueled Digital Transformation by Establishing DIY Software Development. *MIS Quarterly Executive*, *22*(2). On-line: https://aisel.aisnet.org/misqe/vol22/iss2/3

[12] Moore, S. (2015, December). How to Deliver Enterprise Mobile Apps Faster. On-line: https://www.gartner.com/smarterwithgartner/how-to-deliver-enterprise-mobile-apps-faster

[13] FutureScape, I. D. C. (2019, October). IDC FutureScape: Worldwide IT Industry 2020 Predictions. On-line: https://www.idc.com/getdoc.jsp?containerId=US45599219

[14] Wong, J., Iijima, K., Leow, A., Jain, A., & Vincent, P. (2021). Magic Quadrant for Enterprise Low-Code Application Platforms. *Gartner*. Retrieved August 3, 2023, On-line: https://www.gartner.com/en/documents/4005939

[15] Wong, J., Driver, M., & Ray, S. (2019). *The Future of Apps Must Include Citizen Development*. Gartner. On-line: https://www.gartner.com/en/documents/3970067

[16] Fortune Business Insights. (2022). Low Code Development Platform Market. Retrieved August 29, 2023, On-line: https://www.fortunebusinessinsights.com/low-code-development-platform-market-102972

[17] Project Management Institute. (2021). Introducing PMI Citizen Developer. Retrieved August 3, 2021, On-line: https://www.pmi.org/citizen-developer

[18] Lowcomote. (2023). Lowcomote. Retrieved August 21, 2023, On-line: https://www.lowcomote.eu/

[19] 1LowCodeLab. (2023). LowCodeLab@OST: Kompetenzzentrum für innovatives Low-Coding und strategisches Citizen Development. Retrieved August 21, 2023, On-line: https://www.ost.ch/de/forschung-und-dienstleistungen/interdisziplinaere-themen/themencluster/lowcodelab

[20] Prinz, N., Rentrop, C., & Huber, M. (2021). Low-Code Development Platforms - A Literature Review Completed Research. In *Proceedings of the 27th annual Americas Conference on Information Systems (AMCIS)*. On-line: https://aisel.aisnet.org/amcis2021/adv_info_systems_general_track/adv_info_systems_general_track/2

[21] Pinho, D., Aguiar, A., & Amaral, V. (2023). What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, *74*, 101185. https://doi.org/10.1016/j.cola.2022.101185

[22] Martinez, E., & Pfister, L. (2023). Benefits and limitations of using low-code development to support digitalization in the construction industry. *Automation in Construction*, *152*, 104909. https://doi.org/10.1016/j.autcon.2023.104909

[23] Wolff, I. (2019). Making In-House Apps with Low-Code, No-Code Platforms, *2021*(Nov.). On-line: https://www.sme.org/technologies/articles/2019/oc

tober/software-making-in-house-apps-with-low-code-no-code-platforms/

[24] Sanchis, R., García-Perales, Ó., Fraile, F., & Poler, R. (2020). Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences*, *10*(1), 12.

[25] Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021). Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1–11). https://doi.org/10.1145/3475716.3475782

[26] Alsaadi, H. A., Radain, D. T., Alzahrani, M. M., Alshammari, W. F., Alahmadi, D., & Fakieh, B. (2021). Factors that affect the utilization of low-code development platforms: survey study. *Romanian journal of information technology and automatic control-revista romana de informatica si automatica*, *31*(3), 123–140. https://doi.org/10.33436/v31i3y202110

[27] Sahinaslan, E., Sahinaslan, O., & Sabancioglu, M. (2021). Low-Code Application Platform in Meeting Increasing Software Demands Quickly: SetXRM. In *Fourth International Conference Of Mathematical Sciences (ICMS)* (Vol. 2334, p. (070007) 1-4). https://doi.org/10.1063/5.0042213

[28] Richardson, C., & Rymer, J. R. (2016). *Vendor landscape: The fractured, fertile terrain of low-code application platforms*. Forrester. On-line: https://www.forrester.com/report/Vendor-Landscape-The-Fractured-Fertile-Terrain-Of-LowCode-Application-Platforms/RES122549

[29] Outsystems. (2019). *The State of Application Development Is IT Ready for Disruption?* On-line: https://rb.gy/stqzxh

[30] Alamin, M. A. 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 *IEEE/ACM 18th International Conference On Mining Software Repositories (MSR)* (pp. 46–57). https://doi.org/10.1109/MSR52588.2021.00018

[31] Tisi, M., Mottu, J.-M., Kolovos, D., Lara, J. D., Guerra, E., Ruscio, D. D., … Wimmer, M. (2019). Lowcomote: Training the next generation of experts in scalable low-code engineering platforms. In *STAF Co-Located Events Joint Proceedings*. https://doi.org/10.5281/zenodo.4314656

[32] Martinez, E., Ezzeddine, A., & García de Soto, B. (2022). Integrating project delivery and Information Technology: challenges and opportunities. In V. González, F. Hamzeh, & L. F. Alarcón (Eds.), *Lean Construction 4.0: Driving a Digital Revolution of Production Management in the AEC Industry* (1st ed., pp. 275–287). Routledge.

[33] Martinez, E., & Cisterna, D. (2023). Using Low-Code and Artificial Intelligence to Support Continuous Improvement in the Construction Industry. In *Proceedings of the 31st Annual Conference of the International Group for Lean Construction (IGLC31)* (pp. 197–207). Presented at the 31st Annual Conference of the International Group for Lean Construction (IGLC 31), Lille, France. https://doi.org/10.24928/2023/0236

[34] Yin, R. K. (2014). *Case study research: design and methods* (5th Edition.). Los Angeles, CA, USA: SAGE Publications.

[35] Microsoft. (2022). Microsoft Power Platform. Retrieved August 15, 2022, On-line: https://powerplatform.microsoft.com/en-us/

[36] Microsoft. (2022). What is Power Apps. Retrieved August 11, 2022, On-line: https://docs.microsoft.com/en-us/powerapps/powerapps-overview

[37] Elshan, E., Germann, D., Dickhaut, E., & Li, M. (2023). Faster, Cheaper, Better? Analyzing How Low Code Development Platforms Drive Bottom-Up Innovation. *ECIS 2023 Research-in-Progress Papers*. On-line: https://aisel.aisnet.org/ecis2023_rip/82

[38] Peansupap, V., & Walker, D. H. T. (2006). Information communication technology (ICT) implementation constraints. *Engineering, Construction and Architectural Management*, *13*(4), 364–379. https://doi.org/10.1108/09699980610680171

[39] Adrian, B., Hinrichsen, S., & Nikolenko, A. (2020). App Development via Low-Code Programming as Part of Modern Industrial Engineering Education. In I. L. Nunes (Ed.), *Advances in Human Factors and Systems Interaction* (pp. 45–51). https://doi.org/10.1007/978-3-030-51369-6_7

[40] Boehm, B., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches — A survey. *Annals of Software Engineering*, *10*(1), 177–205. https://doi.org/10.1023/A:1018991717352

[41] Butler, R. R. (2020). *Estimating Effort for Low-code Applications* (PhD Thesis). *ProQuest Dissertations and Theses*. On-line: https://www.proquest.com/dissertations-theses/