Proceedings of EKONBIZ 2025
ISBN: 978-99955-45-50-5
UDK: 316.344.7:338.124.4

Submitted: 04.04.2025.
Accepted: 05.05.2025.
Original Article

LOW CODE/NO CODE DEVELOPMENT DRIVERS ASSESSED BY EXPERTS

Lazar Raković
Faculty of Economics in Subotica, University of Novi Sad Subotica, Serbia
lazar.rakovic@ef.uns.ac.rs
ORCID: 0000-0002-1465-588X

Lena Đorđević Milutinović
Faculty of Organizational Sciences, University of Belgrade, Serbia
lena.djordjevic.milutinovic@fon.bg.ac.rs
ORCID: 0000-0002-3758-0147

Vuk Vuković
Faculty of Economics in Subotica, University of Novi Sad, Subotica, Serbia
vuk.vukovic@ef.uns.ac.rs
ORCID: 0000-0002-0040-9936

Bogdan Ubiparipović
Faculty of Economics, University of Banja Luka, Republic of Srpska, BiH
bogdan.ubiparipovic@ef.unibl.org
ORCID: 0000-0002-2462-610X

Abstract: *Low-code/no-code is a relatively new concept that has been gaining significance recently. The concept encompasses software development using tools designed for non-IT professionals, often referred to as citizen developers. Depending on whether a certain level of coding knowledge is required, it is classified as low-code or no-code development. The manuscript provides specific definitions of the aforementioned software development concept and identifies the most common advantages of low-code/no-code development based on the existing literature. Furthermore, a case study conducted in a multinational company validates the identified advantages, accompanied by expert observations. This paper can serve as a starting point for further research in this field.*

Key words: *low-code, no-code development, software development, citizen development.*

JEL classification*: O30, C89*

1. INTRODUCTION

Digital transformation of organizations has intensified over the last decade. A growing number of applications are needed to digitize numerous business processes. However, software developers are in short supply, which makes it difficult to meet demand.

Therefore, many software companies have developed platforms that enable end users to create business-specific applications with little or no coding required. These platforms and this method of software development are called *Low Code/No code platforms/development* (LC/NC) (Hagel et al., 2024; Rokis & Kirikova, 2022).

End users, who typically have no software development experience, are often called *Citizen Developers* (CDs) (Binzer et al., 2024; McHugh et al., 2024). Nonetheless, IT professionals can also use these platforms to accelerate the development of applications.

Major software companies such as *Microsoft*, *Google* and *Amazon* offer platforms for *Low Code/No Code development (Binzer et al., 2024; Di Ruscio et al., 2022)*. Numerous benefits or drivers are often highlighted in advertisements of software companies that present their LC/NC platforms.

There appears to be a need not only to identify, but also to validate specific drivers of LC/NC development and the use of LC/NC platforms. From the above, the following research questions can be distinguished:

- RQ1: Identifying the most important drivers of LC/NC development.
- RQ2: Assessing the importance of the identified drivers of LC/NC.

A systematic literature review was conducted to answer the first research question posed. Thus, the most frequently mentioned drivers of LC/NC development were identified. A case study was conducted in a multinational company that uses LC/NC development to validate and evaluate the identified drivers.

2. LITERATURE REVIEW - LOW CODE/NO CODE DRIVERS

According to Xiao and Watson's methodology (Xiao & Watson, 2019), a systematic literature review was conducted in order to answer the first research question. Firstly, the two largest citation databases, WoS and Scopus, were searched using the keywords "Low Code" and "No Code". Table 1 presents the hits from the scientific papers databases mentioned above. Following an analysis of the titles, abstracts, and keywords, 17 WoS papers and 43 Scopus papers were included in the analysis.

A total of 49 papers remained after eliminating duplicates, and they were downloaded and analyzed according to the inclusion criteria. To be included, a paper had to address, at least partially, the advantages of Low Code/No Code development and platforms. There were 23 papers in the final set.

Table 1. Number of results after databases were searched by keywords

Citation database	Number of hits based on the keywords "Low code" or "No code" development	Number of papers included in second phase
Web of Knowledge	49	17
Scopus	279	43
	Application of criteria	Number of papers
Web of Knowledge and Scopus	After removing duplicates	49
	After applying the inclusion criteria	17

Source: Authors

One of the most frequently mentioned drivers of Low Code/No Code platforms is cheaper software development. Essentially, this means end users, citizen developers, can develop their own applications. As well as being one of the cheaper approaches to developing software, Low Code/No Code platforms are praised for their speed and ease of use. As a result, users turn to this type of software development because they can easily and quickly create the applications they need.

A slow response from the IT department often results in business units being more dependent on the IT department. This explains the proliferation of LCNC development. Furthermore, tools are often used to automate business processes. Also, CDs often develop applications in order to solve their own needs, which leads to innovation. Last but not least, easier integration with other software is also identified as a driver of LCNC software development. Drivers and their references are shown in Table 2.

There is no doubt that LCNC platforms offer a simpler, faster, and more cost-effective alternative to traditional software development (Gomes & Brito, 2022). Moreover, the adoption of LCNC platforms reduces organizational dependence on IT departments (Binzer et al., 2024). These platforms facilitate the automation of business processes while promoting collaboration and fostering a culture of innovation within organizations (Ajimati et al., 2025). According to Pańkowska (2024), the use of LCNC platforms encourages end users to engage in more creative activities. This perspective is further supported by Biedova et al. (2024), who emphasize that the use of LCNC platforms enables end users to acquire new skills.

Table 2. Low Code/No Code drivers

Driver	Source
Cheaper software development	Ajimati et al., 2025; Biedova et al., 2024; Kaess, 2022; Kass et al., 2022; Käss et al., 2023a, 2023b; Martinez et al., 2024; Pańkowska, 2024; Rokis & Kirikova, 2022
Faster software development	Ajimati et al., 2025; Beranic et al., 2020; Biedova et al., 2024; De Silva et al., 2024; Elshan et al., 2024; Gomes & Brito, 2022; Hintsch et al., 2021; Luo et al., 2021; Martinez et al., 2024; Mottu & Sunyé, 2024; Overeem & Jansen, 2021; Phalake et al., 2024; Rafiq et al., 2022; Rokis & Kirikova, 2022, 2023

Easier software development	Ajimati et al., 2025; Biedova et al., 2024; Di Ruscio et al., 2022; Gomes & Brito, 2022; Kass et al., 2022; Käss et al., 2023a, 2023b; Luo et al., 2021; Sahay et al., 2020
Reducing dependence on the IT department and delays in application development	Biedova et al., 2024; Binzer et al., 2024; Kass et al., 2022; Käss et al., 2023b
Automation of business processes	Ajimati et al., 2025; Biedova et al., 2024; Luo et al., 2021
Promotion of digital innovations, development of ideas	Ajimati et al., 2025; Rokis & Kirikova, 2022
Easier software integration	De Silva et al., 2024; Käss et al., 2023b

Source: Authors

3. RESEARCH DESIGN

Empirical research in the form of qualitative research was conducted using case study methodology Runeson (Runeson & Höst, 2009; Yin, 2009). A multinational company using Microsoft 365 is the criterion for choosing the company.

The reason that Microsoft 365 was chosen is that it contains LC/NC platforms, which are frequently used by citizen developers. A multinational corporation with more than 150 organizations was selected. This selected company uses the Microsoft Dynamics ERP solution as well as the Microsoft 365 package that includes LC/NC development tools.

As a research instrument, a semi-structured interview was used in accordance with Ristić (2016). The interviews were conducted in the period from February to March 2025. Inclusion criteria included employees with some IT experience who also frequently use LC/NC platforms. Although the aforementioned platforms are often used by end-user developers without adequate IT knowledge, the aim of this paper was to study the opinions of experts about the drivers of LC/NC development, so IT-savvy users were chosen.

4. RESULTS

The following sections will present the structure of the respondents and the research results.

4.1. STRUCTURE OF RESPONDENTs

The following Table 3 gives data on respondents, their job title, years of experience, and the LC/NC platforms they use.

Table 3. Data on respondents

	Respondent 1 (R1)
Job title	Business Applications Manager
Years of experience	13
LowCode/No Code platforms	Power BI, Power Automate, SharePoint, UiPath
	Respondent 2 (R2)
Job title	Dynamics NAV Analyst
Years of experience	2
LowCode/No Code platforms	Power Automate, Power Apps, SharePoint
	Respondent 3 (R3)
Job title	Dynamics NAV Analyst
Years of experience	11
LowCode/No Code platforms	Power BI, Power apps, power automate, Microsoft fabric, windsor ai
	Respondent 4 (R4)
Job title	Dynamics NAV Analyst
Years of experience	7
LowCode/No Code platforms	Power BI, SharePoint, Power Automate
	Respondent 5 (R5)
Job title	Dynamics NAV Analyst
Years of experience	6
Low code /No code platforms	Power Automate, Power Apps, SharePoint, PowerBI

Source: Authors

4.2. RESULTS ANALYSIS

According to the respondents, the identified drivers of LC/NC development were ranked between 1 - no impact and 6 - very strong impact. As a means of avoiding responding with midpoints, the respondents were given a scale from 1 to 6. Respondents were asked to comment on each driver's rating. Table 4 shows the average scores for individual drivers.

The speed of software development and the automation of business processes are the most significant drivers, as shown in the table. They are followed by cheaper and easier software development and promotion of digital innovations, development of ideas. The driver regarding easier software integration received the lowest rating of all the listed drivers.

Table 4. Low Code/No Code drivers – experts' ratings

Driver	Average score
Cheaper software development	5.00
Faster software development	5.60
Easier software development	5.00
Reducing dependence on the IT department and delays in application development	3.40
Automation of business processes	5.40
Promotion of digital innovations, development of ideas	4.80
Easier software integration	3.40

Source: Authors

4.2.1. Cheaper software development

According to Biedova et al.(2024), LC/NC platforms are popular among citizen developers due to their low price. According to Respondent 1 (R1), this driver is not critical. However, R2 points out the following: "A cheaper development process can have a significant impact on the applicability of the LC/NC approach, resulting in significant savings of resources, time, and money".

A similar approach is taken by R3: "The cost of LC/NC services is much lower than the cost of a full development with all basic tools". Furthermore, R4 points out: "LCNC enables cheaper development thanks to savings in resources (time, employees) and required tools". The following is stated by R5: "The Microsoft 365 tools are bundled, so this is a cheaper way to develop software".

4.2.2. Faster software development

Gomes & Brito (2022) mention faster development as one of the advantages of using LC/NC platforms, and therefore the LC/NC driver. Respondent 2 (R2) claims "Faster software development is one of the key drivers, as it enables significantly faster software development compared to traditional development", and a similar claim was made by Respondent 4 (R4 states the following: "Compared to traditional methodologies and tools, development is significantly faster"). R3 confirms the previous statement: "Now, development takes a few days instead of weeks or months".

4.2.3. Easier software development

Development on LC/NC platforms is generally easier than traditional programming, but it requires some learning and some time to become familiar with the tool (Luo et al., 2021).

Easier software development is "one of the most important drivers, as LC/NC significantly simplifies software development, enables the creation of applications even without advanced knowledge of programming languages" (R2). "It is easier to build and maintain software" (R2), that is "A person without an IT background can create basic applications by following quick tutorials. A great impact on people's development from business to business informatics" (R3).

R4 points out that LC/NC development "enables domain experts, end users, without IT knowledge to develop the necessary software", while R5 states that "Expertise is not required, but an IT background is a plus".

4.2.4. Reducing dependence on the IT department and delays in application development

Binzer et al.(2024) state that LC/NC platforms allow citizen developers to develop custom solutions with minimal or no dependence on the IT department. R1 contrasts with the previous authors, stating that "we rarely see any dependence on IT departments". Similarly, R2 says this driver "has a lower impact, as IT still plays a key role in the development and maintenance of LC/NC applications".

It is further stated that "IT is often part of LC/NC" (R3), and that "IT is still very important" (R5). Unlike the previous respondents, R4 points out the following: "IT support is necessary in some aspects of development, but the development of applications without IT's direct involvement is one of the main drivers of LC/NC implementation".

4.2.5. Automation of business processes

The LC/NC platforms are often used to automate certain business processes and perform work more efficiently (Rokis & Kirikova, 2022). Experts agree with the previous statement. LC/NC "is mostly used for automation".

Furthermore, automation is "one of the key drivers, a large number of processes are automated with LC/NC" (R2). However, one should not ignore the claim of R5 that "it is easy to automate simpler, more trivial processes".

4.2.6. *Promotion of digital innovations, development of ideas*

According to Ajimati et al. (2025), LC/NC platforms encourage problem-solving and innovation in organizations. This is confirmed by R2, saying that "the promotion of digital innovations has a significant impact on the development of LC/NC". Similarly, R4 points out that LC/NC "enables employees to test and implement their ideas and domain expertise faster and easier, thereby supporting digital innovation at various levels".

4.2.7. *Easier software integration*

Authors Käss et al. (2023b) and De Silva et al. (2024) emphasize the good integration of LC/NC-developed applications with existing systems. R2 points out that this driver has a moderate impact while R3 claims that integration can be fast but is not a decisive driver. Furthermore, software integration is also easier in Microsoft environment (R4), which is also confirmed by R5's statement that everything depends on the software used for integration.

CONCLUSION

It would be useful to begin by explaining the general attitudes of respondents R2 and R1 regarding LC/NC development.

Respondent 2 stated: "*Within our organization, LC/NC has a substantial impact on the automation of business processes and the development of innovative applications. This approach is applied across sectors, including HR, EHS, IT, sales and many others. LCNC makes application development and process automation faster, which is one of the key drivers. IT department continues to play a key role in the development and implementation of LC/NC solutions*" (R2).

Respondent 1 claimed *"Since LC/NC is gaining popularity in recent years, we use it whenever possible to automate processes. Although some things still require coding, the global trend is to use LC/NC tools to handle these requirements. Furthermore, modules in ERP systems are being developed to provide solutions that are applicable as low code - this still requires an IT department or someone who is technically savvy" (R1).*

The proliferation of LC/NC development in many organizations is undeniable. According to experts, rapid software development is the main driver of LC/NC development. The software development time has been significantly shortened compared to standard software development by IT experts. The next driver highlighted by experts is the automation of business processes, as the digital transformation of processes requires automation. However, it is important not to ignore the fact that such automation is often limited to simpler business processes. The next two drivers are cheaper and easier software development. LC/NC is cheaper because it often includes other tools purchased separately (e.g. Microsoft 365) and easier since it allows people without IT knowledge to create business applications. Furthermore, LC/NC supports innovation in organizations. Easier software integration has the least impact among the drivers identified in a systematic literature review, mainly due to the use of different business software in organizations.

A main limitation of this paper is that the drivers were identified based on studies from only two citation databases. Moreover, the research was conducted only in one organization and included only users with IT background. The above findings can be used to guide future research. Therefore, future research could focus on researching additional literature as well as multi-case study research involving citizen developers.

REFERENCES

[1] Ajimati, M. O., Carroll, N., & Maher, M. (2025). Adoption of low-code and no-code development: A systematic literature review and future research agenda. Journal of Systems and Software, 222, 112300. https://doi.org/10.1016/j.jss.2024.112300

[2] Beranic, T. , Rek, P., & Hericko, M. (2020). Adoption and Usability of Low-Code/ No-Code Development Tools. Proceedings of the Central European Conference on Information and Intelligent Systems, 97–103.

[3] Biedova, O., Ives, B., Male, D., & Moore, M. (2024). Strategies for Managing Citizen Developers and No-Code Tools. MIS Quarterly Executive, 23(2), 165–183. https://doi.org/10.17705/2msqe.00093

[4] Binzer, B., Elshan, E., Fürstenau, D., & Winkle, T. J. (2024). Establishing a Low-Code/No-Code-Enabled Citizen Development Strategy. MIS Quarterly

Executive, 23(3), 253–273. https://doi.org/10.17705/2msqe.00097

[5] De Silva, D. I., Shangavie, R., & Ranathunga, R. A. A. L. (2024). Role of Quality Assurance in Low-Code/No-Code Projects. 2024 International Conference on Information Networking (ICOIN), 789–794. https://doi.org/10.1109/ICOIN59985.2024.10572203

[6] Di Ruscio, D., Kolovos, D., de Lara, J., Pierantonio, A., Tisi, M., & Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin? Software and Systems Modeling, 21(2), 437–446. https://doi.org/10.1007/s10270-021-00970-2

[7] Elshan, E., Siemon, D., Bruhin, O., Kedziora, D., & Schmidt, N. (2024). Unveiling Challenges and Opportunities in Low Code Development Platforms: A StackOverflow Analysis. Hawaii International Conference on System Sciences 2024, 7269–7279.

[8] Gomes, P. M., & Brito, M. A. (2022). Low-Code Development Platforms: A Descriptive Study. 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 1–4. https://doi.org/10.23919/CISTI54924.2022.9820354

[9] Hagel, N., Hili, N., & Schwab, D. (2024). Turning Low-Code Development Platforms into True No-Code with LLMs. Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 876–885. https://doi.org/10.1145/3652620.3688334

[10] Hintsch, J., Staegemann, D., Volk, M., & Turowski, K. (2021). Low-code development platform usage: Towards bringing citizen development and enterprise IT into harmony. Australasian Conference on Information Systems, ACIS 2021.

[11] Kaess, S. (2022). Low Code Development Platform Adoption: A Research Model. ACIS 2022 Proceedings, Australasian Conference on Information Systems.

[12] Kass, S., Strahringer, S., & Westner, M. (2022). Drivers and Inhibitors of Low Code Development Platform Adoption. 2022 IEEE 24th Conference on Business Informatics (CBI), 196–205. https://doi.org/10.1109/CBI54897.2022.00028

[13] Käss, S., Strahringer, S., & Westner, M. (2023a). A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems. IEEE Access, 11, 118762–118786. https://doi.org/10.1109/ACCESS.2023.3325092

[14] Käss, S., Strahringer, S., & Westner, M. (2023b). Practitioners' Perceptions on the Adoption of Low Code Development Platforms. IEEE Access, 11, 29009–29034. https://doi.org/10.1109/ACCESS.2023.3258539

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

[16] Martinez, E., Pfister, L., & Stauch, F. (2024). Developing a novel application to digitalize and optimize construction operations using low-code technology. 41st International Symposium on Automation and Robotics in Construction (ISARC 2024), 283–290. https://doi.org/10.22260/ISARC2024/0038

[17] McHugh, S., Carroll, N., & Connolly, C. (2024). Low-Code and No-Code in Secondary Education—Empowering Teachers to Embed Citizen Development in Schools. Computers in the Schools, 41(4), 399–424. https://doi.org/10.1080/07380569.2023.2256729

[18] Mottu, J.-M., & Sunyé, G. (2024). Emerging New Roles for Low-Code Software Development Platforms. Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 905–914. https://doi.org/10.1145/3652620.3688337

[19] Overeem, M., & Jansen, S. (2021). Proposing a Framework for Impact Analysis for Low-Code Development Platforms. 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 88–97.

https://doi.org/10.1109/MODELS-C53483.2021.00020

[20] Pańkowska, M. (2024). Low Code Development Cycle Investigation. In X.-S. Yang, S. Sherratt, N. Dey, & A. Joshi (Eds.), Lecture Notes in Networks and Systems 1011 (pp. 265–275). Springer. https://doi.org/10.1007/978-981-97-4581-4_19

[21] Phalake, V. S., Joshi, S. D., Rade, K. A., Phalke, V. S., & Molawade, M. (2024). Optimization for achieving sustainability in low code development platform. International Journal on Interactive Design and Manufacturing (IJIDeM), 18(8), 5717–5724. https://doi.org/10.1007/s12008-023-01338-0

[22] Rafiq, U., Filippo, C., & Wang, X. (2022). Understanding Low-Code or No-Code Adoption in Software Startups: Preliminary Results from a Comparative Case Study. In D. Taibi, M. Kuhrmann, T. Mikkonen, J. Klünder, & P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. PROFES 2022. Lecture Notes in Computer Science, vol 13709 (pp. 390–398). Springer. https://doi.org/10.1007/978-3-031-21388-5_27

[23] Ristić, Ž. (2016). Objedinjavanje kvantitativnih i kvalitativnih istraživanja. Evropski centar za mir i razvoj (ECPD).

[24] Rokis, K., & Kirikova, M. (2022). Challenges of Low-Code/No-Code Software Development: A Literature Review. In Ē. Nazaruka, K. Sandkuhl, & U. Seigerroth (Eds.), Perspectives in Business Informatics Research. BIR 2022. Lecture Notes in Business Information Processing (Vol. 462, pp. 3–17). Springer. https://doi.org/10.1007/978-3-031-16947-2_1

[25] Rokis, K., & Kirikova, M. (2023). Exploring Low-Code Development: A Comprehensive Literature Review . Complex Systems Informatics and Modeling Quarterly, 36, 6886. https://doi.org/10.7250/csimq.2023-36.04

[26] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14(2), 131–164. https://doi.org/10.1007/s10664-008-9102-8

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

[28] Xiao, Y., & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. Journal of Planning Education and Research, 39(1), 93–112. https://doi.org/10.1177/0739456X17723971

[29] Yin, R. (2009). Case study research: design and methods. SAGE

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License