CHALLENGES OF LOW-CODE/NO-CODE PLATFORMS IN SUPPORTING ORGANISATIONAL INFORMATION PROCESSES: A LITERATURE REVIEW AND CASE STUDY EVIDENCE

1 Lazar Raković, University of Novi Sad, Faculty of Economics in Subotica, Subotica, Serbia
2 Lena Đorđević Milutinović, University of Belgrade, Faculty of Organizational Sciences, Belgrade, Serbia
3 Paweł Lula, Krakow University of Economics, Kraków, Poland
4 Bogdan Ubiparipović, University of Banja Luka, Faculty of Economics, Bosnia and Herzegovina
5 Marek Dziura, Krakow University of Economics, Kraków, Poland
6 Tomasz Rojek, Krakow University of Economics, Kraków, Poland
*Corresponding author's e-mail: lazar.rakovic@ef.uns.ac.rs
1 ORCID ID: 0000-0002-1465-588X
2 ORCID ID: 0000-0002-3758-0147
3 ORCID ID: 0000-0003-2057-7299
4 ORCID ID: 0000-0002-2462-610X
5 ORCID ID: 0000-0002-4889-2883
6 ORCID ID: 0000-0002-2977-4312

ARTICLE INFO

Review Scientific Paper
Received: 15.07.2025
Revised: 20.11.2025
Accepted: 03.12.2025

DOI 10.63356/ace.2025.013

UDK 338.434:339.972(4-672EU)

COBISS.RS-ID 143723521

Keywords: *low code/no code development; end-user development; inhibitors*

JEL Classification: M15, O33

ABSTRACT

Digital transformation increasingly positions business informatics at the core of organisational competitiveness, driving companies to digitalise and automate their business processes. In this context, low-code/no-code (LCNC) platforms have emerged as a promising solution within business information systems, enabling rapid development of process-oriented applications with minimal or no programming. By empowering employees without formal IT backgrounds to participate in system development, LCNC platforms address the shortage of IT professionals and help bridge the gap between technical and domain-specific business knowledge. Although vendors emphasise advantages such as ease of use, accelerated development cycles, reduced costs, lower IT dependency and enhanced process innovation, they often overlook the organisational, technological and governance challenges associated with LCNC adoption. This paper systematically identifies key inhibitors and LCNC implementation through a comprehensive literature review, followed by an assessment of their significance across multiple case studies with LCNC users. Seven major inhibitors are identified: vendor lock-in, security and compliance risks, integration challenges, limited scalability, insufficient documentation, limited testing support, and lack of flexibility. Case study findings indicate that lack of flexibility and customisation, vendor lock-in and insufficient testing support represent the most critical barriers. The paper presents preliminary insights from an ongoing investigation into LCNC development.

1. INTRODUCTION

Digital transformation represents a process that has affected many organisations. The workplace has become increasingly digital, with numerous applications accessed by employees daily. An increasing number of business processes are being partially or fully automated. To enable this, it is necessary to develop applications that will automate these business processes. Given that the number of professional programmers is insufficient to meet the ever-growing demand, one of the driving forces behind digital transformation (Viljoen, Radić, et al., 2024), large number of software vendors are turning to the development of tools that allow end-users to create their own applications, automate business processes and build customised business solutions tailored to specific organisational needs (Kok et al., 2024). These end-users are often referred to as citizen developers or end-user developers, while the tools they use are known as low-code and no-code tools/platforms, and the development approach itself as low-code and no-code development (LCNC). It is important to emphasise the observation by Viljoen et al. (2024) that citizen developers do not create complex software, but rather smaller applications aimed at solving specific problems within their business units, thus extending development beyond the boundaries of traditional IT departments (Matook Wang & Axelsen, 2025). Viljoen et al. (2025) emphasise that LCNC platforms are increasingly recognised as essential for advancing digital transformation. When referring to LCNC platforms, the most mentioned examples include ServiceNow, Microsoft Power Platform, Mendix, Google AutoML, Amazon SageMaker and similar solutions (Kok et al., 2024). These platforms may play a pivotal role in accelerating digital transformation by democratising software development (Viljoen, Radić, et al., 2024).

The terms low-code and no-code are frequently used interchangeably (Elshan et al., 2024). However, it is important to distinguish between them. No-code tools enable application development without any need for coding, whereas low-code tools require some degree of code customisation or manual code integration. As noted by Elshan et al. (2024), the low-code paradigm is broader in scope, with no-code representing a subset within it.

Numerous benefits are often associated with LCNC tools, such as rapid, simple and cost-effective development of business applications (Raković, Djordjevic Milutinovic, et al., 2025; Raković, Đorđević, et al., 2025). Four out of five organisations believe that LCNC accelerates digital transformation (Gonçalves et al., as cited in Viljoen et al., 2025). Furthermore, Hurlburt et al. (2025) report forecasts suggesting that LCNC will account for as much as 65% of application development, with development time potentially reduced by up to 90%. Viljoen et al. (2024) also cite Gartner research predicting that, in the future, as many as 80% of applications will be developed by individuals without traditional IT roles. However, what LCNC platform vendors often fail to disclose are the numerous challenges or inhibitors that accompany LCNC development. Viljoen et al. (2025) emphasise the need to balance the flexibility and risk associated with LCNC platforms, warning that efforts to reduce risk through excessive standardisation may compromise the fundamental purpose of LCNC tools, to empower users through autonomy and flexibility.

As the inhibitors of LCNC platforms and LCNC development are often insufficiently explored and largely overlooked, the following research questions have been formulated:

RQ1: What are the inhibitors of LCNC development?
RQ2: How important are the identified inhibitors of LCNC development?

To address the first research question, a systematic literature review was conducted. Subsequently, a multiple case study, based on interviews with users of LCNC platforms, was carried out to evaluate the importance of the previously identified inhibitors of LCNC development.

Chapter two presents the phases of the systematic literature review. The same chapter also presents the inhibitors of low-code/no-code development. The third chapter is reserved for the design and results of the conducted empirical research. The paper concludes with final considerations.

2. LITERATURE REVIEW

The systematic literature review was conducted following the methodology proposed by Xiao and Watson (2019), with additional guidance from the works of Kitchenham et al. (Brereton et al., 2007; Kitchenham, 2004; Kitchenham et al., 2010, 2013; Turner et al., 2008). To identify relevant inhibitors, two major citation databases SCOPUS and Web of Science (WoS) were searched using the keywords "Low Code" and "No Code". The database search was carried out in May 2025. Table 1 presents the number of hits retrieved from each database. In the first iteration, titles, keywords and abstracts were screened. The inclusion criterion for proceeding to the next phase was the potential relevance of a study to the identification of LCNC development inhibitors. The third column of Table 1 shows the number of studies that met this criterion and were included in the subsequent analysis. Next, the results from WoS and Scopus were merged, and duplicates were removed, resulting in 641 unique records. These remaining studies were then analysed in detail. Ultimately, 25 studies were selected for the final analysis focused on identifying inhibitors of LCNC development. The full list of included studies is provided in Table 2.

Table 1: Searched databases

Citation Database	Number of Hits in Initial Search	Number of Studies Included in the Next Phase
Scopus	2146	537
WoS	1236	329

Source: Authors' calculation

Table 2: Identified studies

Authors and Year	Title	Type of Source
Ajimati, Carroll & Maher (2025)	Adoption of low-code and no-code development: A systematic literature review and future research agenda	Journal
Beranic, Rek & Hericko (2020)	Adoption and Usability of Low-Code/ No-Code Development Tools	Conference proceedings
Biedova et al. (2024)	Strategies for Managing Citizen Developers and No-Code Tools	Journal
Binzer & Winkler (2024)	'To Code, or Not to Code', Unpacking the Understanding and Difficulties of Citizen Development Programs	Conference proceedings
Binzer et al. (2024)	Establishing a Low-Code/No-Code-Enabled Citizen Development Strategy	Journal
De Silva, Shangavie & Ranathunga (2024)	Role of Quality Assurance in Low-Code/No-Code Projects	Conference proceedings
Di Ruscio et al. (2022)	Low-code development and model-driven engineering: Two sides of the same coin?	Journal

Authors and Year	Title	Type of Source
Elshan et al. (2024)	Unveiling Challenges and Opportunities in Low Code Development Platforms: A StackOverflow Analysis	Conference proceedings
Gomes & Brito (2022)	Low-Code Development Platforms: A Descriptive Study	Conference proceedings
Hintsch et al. (2021)	Low-code development platform usage: Towards bringing citizen development and enterprise IT into harmony	Conference proceedings
Kaess (2022)	Low Code Development Platform Adoption: A Research Model	Conference proceedings
Käss et al. (2023a)	A Multiple Mini Case Study on the Adoption of Low Code Development Platforms in Work Systems	Journal
Käss et al. (2023b)	Practitioners' Perceptions on the Adoption of Low Code Development Platforms	Journal
Kass et al. (2022)	Drivers and Inhibitors of Low Code Development Platform Adoption	Conference proceedings
Luo et al. (2021)	Characteristics and Challenges of Low-Code Development	Conference proceedings
Martinez, Pfister& Stauch (2024)	Developing a novel application to digitalize and optimize construction operations using low-code technology	Conference proceedings
Martins, Branco & Mamede (2023)	Combining low-code development with ChatGPT to novel no-code approaches: A focus-group study	Journal
Mottu & Sunyé (2024)	Emerging New Roles for Low-Code Software Development Platforms	Conference proceedings
Pańkowska (2024)	Low Code Development Cycle Investigation	Conference proceedings
Pinho, Aguiar & Amaral (2023)	What about the usability in low-code platforms? A systematic literature review	Journal
Rokis & Kirikova (2023)	Exploring Low-Code Development: A Comprehensive Literature Review	Journal
Rokis & Kirikova (2022)	Challenges of Low-Code/No-Code Software Development: A Literature Review	Conference proceedings
Sahay et al. (2020)	Supporting the understanding and comparison of low-code development platforms	Conference proceedings
Viljoen, Radić, et al. (2024)	Governing Citizen Development to Address Low-Code Platform Challenges	Journal
Viljoen et al. (2025)	Navigating Flexibility and Standardisation in Low-Code/No-Code Development	Journal

Source: Authors' calculation

2.1 Low code/No code inhibitors

Viljoen et al. (2024) identify the primary risks of LCNC platforms as the quality of the software produced, the emergence of shadow IT, which poses a significant threat, and the accumulation of so-called technical debt. Similarly, Zaheri et al. (2024) emphasise that, despite the advantages of rapid development, LCNC developed applications that frequently exhibit inconsistencies in functionality, often resulting in data loss. Dependence on platform vendors for software updates, security and long-term support is cited as one of the main inhibitors of LCNC platforms. Ajimati, Carroll & Maher (2025) argue that such dependency may lead to increased costs in cases requiring maintenance or platform replacement. These authors further recommend that organisations carefully select their LCNC platforms to minimise this inhibitor.

Additionally, Gomes and Brito (2022) highlight the lack of freedom and choice available to users of LCNC platforms compared to those who have access to source code. Increased risks related to security, compliance and privacy, such as data breaches, cyberattacks, shadow IT and non-compliance issues, are noted by Viljoen et al. (2024) as stemming from a lack of consistency in corporate identity across applications. According to Ajimati Carroll & Maher (2025), security risks also arise from the creation of unauthorised shadow IT applications by users who often lack awareness of cybersecurity threats and network security.

Integration and interoperability challenges, such as the difficulty in integrating LCNC tools with other enterprise systems, are cited as significant barriers to the adoption of LCNC platforms (Kass et al., 2022). Naqvi et al. (2025) mention "challenges integrating with digital ecosystems and difficulties migrating between platforms".

Kass et al. (2022) emphasise that although LCNC platforms are frequently associated with easy scalability in the literature, research indicates that developing scalable LCNC software requires advanced knowledge, which often lacks among citizen developers. Limited scalability is also associated with restricted support for large-scale computations or flexible cloud deployment. Insufficient documentation is linked, on one hand, to the lack of clear guidelines, user manuals, or best practices (Beranic, Rek & Hericko, 2020), and on the other, to the fact that citizen developers do not document the applications they develop. Documentation of the developed software is essential both for its usage and maintenance, and the absence of such documentation hinders subsequent maintenance of LCNC software (Hintsch et al., 2021).

Another commonly cited inhibitor is the inadequate support for testing and analytics. To ensure software quality, testing must be properly conducted;

however, citizen developers often lack training in testing, debugging or performance analysis (Hintsch et al., 2021). The lack of testing may result in reliability issues with LCNC developed software (Viljoen et al., 2025). Although LCNC platforms are associated with the flexibility of citizen developers to create their own applications (Viljoen et al., 2025), it is often pointed out that LCNC platforms have a restricted ability to implement complex or highly tailored functionalities, representing the inhibitor of lack of flexibility and customisation. Table 3 presents the most significant inhibitors of LCNC platforms.

Table 3: Low Code/No Code inhibitors

Inhibitors	Sources
Vendor or third-party lock-in	Ajimati, Carroll & Maher 2025; Biedova et al., 2024; Binzer et al., 2024; Binzer & Winkler, 2024; Di Ruscio et al., 2022; Kaess, 2022; Kass et al., 2022; Käss et al., 2023a, 2023b; Luo et al., 2021; Rokis & Kirikova, 2022, 2023; Sahay et al., 2020
Increased risks related to security, compliance and privacy	Ajimati, Carroll & Maher, 2025; Biedova et al., 2024; Hintsch et al., 2021; Kaess, 2022; Naqvi et al., 2025; Viljoen, Altın, et al., 2024; Viljoen et al., 2025; Viljoen, Radić, et al., 2024; Zaheri, Famelis & Syriani, 2024
Integration and interoperability challenges	Elshan et al., 2024; Kass et al., 2022; Naqvi et al., 2025; Rokis & Kirikova, 2022; Sahay et al., 2020
Limited scalability	Ajimati, Carroll & Maher, 2025; Käss et al., 2023b; Rokis & Kirikova, 2022, 2023; Sahay et al., 2020
Insufficient documentation	Biedova et al., 2024; Hintsch et al., 2021; Kass et al., 2022; Käss et al., 2023b; Martins, Branco & Mamede, 2023
Limited testing and analytics support	De Silva, Shangavie & Ranathunga, 2024; Hintsch et al., 2021; Rokis & Kirikova, 2022, 2023
Lack of flexibility and customisation	(De Silva, Shangavie & Ranathunga, 2024., 2024; Kass et al., 2022; Käss et al., 2023b; Luo et al., 2021; Mottu & Sunyé, 2024; Naqvi et al., 2025; Rokis & Kirikova, 2022; Viljoen, Radić, et al., 2024)

Source: Authors' calculation.

3. RESEARCH DESIGN

To address the second research question, a qualitative study was conducted, using the multiple case study methodology (Runeson & Höst, 2009; Yin, 2009). Organisations utilising LCNC tools were selected, and the participants were users who employ LCNC platforms daily. A semi-structured interview was conducted with the participants via the MS Teams online platform. Following the initial questions, participants were asked to evaluate the relevance of previously identified inhibitors. The semi-structured interview was conducted in accordance with the guidelines provided by Ristić (2016) and Runeson & Höst (2009). The research was carried out during June and July of 2025 in organisations from Bosnia and Herzegovina. The inclusion criterion for participants was familiarity with LCNC tools and active use of them in their professional duties.

The case study was conducted in nine organisations, with one participant from each. Table 4 presents the structure of the participants according to the size and type of organisation, the department in which they are employed, their current position and years of experience. The majority of participants work in organisations with up to or over one thousand employees, primarily within the banking and private sectors. Furthermore, most participants are employed in IT departments and possess more than twenty years of experience.

Table 4: Case study organisations

Participant code	Size of organisation	Type of organisation	Department	Current position	Years of experience
Rsp. 1	201–1000	Financial services / Banking / Insurance	Information Technology (IT)	Manager / Team leader	20+
Rsp. 2	1000+	International organisation	Administration / General Management	Manager / Team leader	20+
Rsp. 3	1000+	Private company (for-profit)	Information Technology (IT)	Software developer / IT specialist	6-10
Rsp. 4	201–1000	Financial services / Banking / Insurance	Research & Development (R&D)	Manager / Team leader	20+
Rsp. 5	1000+	Telco company	Information Technology (IT)	Manager / Team leader	20+
Rsp. 6	201–1000	Financial services / Banking / Insurance	Information Technology (IT)	Business analyst	11-20
Rsp. 7	1–10	IT/Software services provider	Information Technology (IT)	Software developer / IT specialist	3-5
Rsp. 8	1000+	Financial services / Banking / Insurance	Information Technology (IT)	Vice president	20+
Rsp. 9	201–1000	Private company (for-profit)	Information Technology (IT)	Software developer / IT specialist	11-20

Source: Authors' calculation.

3.1 Results

Participants evaluated the impact of each inhibitor on LCNC software development using a scale from 0 to 5 (0 – no impact, 5 – strong impact). Table 5 presents the average scores assigned by the participants for each identified inhibitor. Based on the data, it can be concluded that participants consider the most significant inhibitor of LCNC development to be the lack of flexibility and customisation, followed by vendor or third-party lock-in and limited testing and analytics support. The least significant inhibitor identified by participants is insufficient documentation.

Table 5: Low Code/No Code inhibitors – Average scores

Inhibitors	Average score
Vendor or third-party lock-in	4.4
Increased risks related to security, compliance and privacy	4
Integration and interoperability challenges	4
Limited scalability	3.4
Insufficient documentation	3.3
Limited testing and analytics support	4.4
Lack of flexibility and customisation	4.6

Source: Authors' calculation.

The results of the conducted research indicate that users of LCNC tools express the greatest concern regarding the limited adaptability of the systems and dependence on external platforms or vendors. This may suggest a need for greater modularity and the adoption of open standards within LCNC solutions. High ratings for inhibitors related to testing, analytics, security and interoperability suggest that, although LCNC platforms offer rapid development and accessibility, their technical maturity and ability to integrate into complex IT environments still raises skepticism among users. On the other hand, the relatively low rating for "insufficient documentation" may indicate that users, while recognising shortcomings in this area, do not consider it a significant limitation to tool usage, or that they rely on alternative sources of knowledge (e.g., communities, tutorials, support).

4. CONCLUSIONS

LCNC platforms enable rapid development of business applications and the automation of numerous business processes. End-users who utilise LCNC platforms and develop their own applications are referred to as citizen developers or end-user developers. LCNC development remains a relatively new phenomenon, and according to Viljoen et al. (2024), it is still unclear what organisations should do to strike a balance between speed and decentralisation and high software quality. The same authors emphasise the need for governance of LCNC development, stating that without proper management, issues such as poor software quality, shadow IT and technical debt will only escalate.

Furthermore, Matook Wang & Axelsen, l. (2025) argue that the training of new generations of developers (citizen developers) is the responsibility of higher education institutions, while Naqvi et al. (2025) highlight that organisations must develop comprehensive programs that demonstrate both the capabilities and limitations of LCNC platforms. A similar claim is made by Takahashi et al. (2024), who state that merely introducing LCNC platforms is insufficient; management must provide support for their use by fostering a cooperative environment through workshops and other forms of assistance.

This study aimed to identify the key inhibitors of LCNC solution development and to assess their significance from the perspective of end-users. To answer the first research question, a systematic literature review was conducted, through which a range of inhibitors to LCNC software development was identified. These inhibitors were derived from an analysis of 25 relevant papers sourced from the SCOPUS and Web of Science databases.

The answer to the second research question was established based on a multiple case study involving end-users of LCNC tools. Based on the collected data, users highlighted lack of flexibility and customisation, vendor lock-in, and limited support for testing and analytics as the most significant inhibitors. In contrast, insufficient documentation was rated as the least restrictive factor. The results indicate that users perceive technical and infrastructural limitations as the primary barriers to broader and more effective use of LCNC platforms.

The research conducted confirms that the inhibitors to LCNC solution development are real and multifaceted, and that ignoring them may significantly diminish the benefits that the LCNC approach can offer to users. Future research should focus on developing concrete guidelines and strategies for overcoming the identified obstacles, as well as on longitudinal studies that track changes in user perception and the maturity of LCNC technologies over time.

This paper presents only preliminary findings from ongoing research aimed at examining the drivers and inhibitors of LCNC software development. Current results indicate that the most significant inhibitors to LCNC development are the lack of flexibility and customisation, vendor or third-party lock-in, and limited testing and analytics support. Full findings will be presented in future publication.

Conflict of interests

The authors declare there is no conflict of interest.

REFERENCES

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

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.

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

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

Binzer, B., & Winkler, T. J. (2024). *'To Code, or Not to Code', Unpacking the Understanding and Difficulties of Citizen Development Programs*. In M. Klein, D. Krupka, C. Winter, M. Gergeleit, & L. Martin (Eds*.), INFORMATIK 2024,* Lecture Notes in Informatics (LNI), Gesellschaft für Informatik (pp. 1587–1596).

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software, 80*(4), 571–583. https://doi.org/10.1016/j.jss.2006.07.009

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

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

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.

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

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.

Hurlburt, G. F., Thiruvathukal, G. K., Kshetri, N., & Ahmad, N. (2025). Low Code/No Code Meets the Metaverse. *Computer, 58*(3), 22–28. https://doi.org/10.1109/MC.2024.3520883

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

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

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

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

Kitchenham, B. (2004). *Procedures for performing systematic reviews*. Keele University. https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2013). Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology, 51*(1), 7–15.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering-A tertiary study. *Information and Software Technology, 52*(8), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

Kok, C. L., Tan, H. R., Ho, C. K., Lee, C., Teo, T. H., & Tang, H. (2024). *A Comparative Study of AI and Low-Code Platforms for SMEs: Insights into Microsoft Power Platform, Google AutoML and Amazon SageMaker*. 2024 IEEE 17th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSoC), 50–53. https://doi.org/10.1109/MCSoC64144.2024.00018

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

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

Martins, J., Branco, F., & Mamede, H. (2023). Combining low-code development with ChatGPT to novel no-code approaches: A focus-group study. *Intelligent Systems with Applications*, 20, 200289. https://doi.org/10.1016/j.iswa.2023.200289

Matook, S., Wang, Y. M., & Axelsen, M. (2025). Experiential Learning for Citizen Developers. *Business & Information Systems Engineering, 67*(1), 7–30. https://doi.org/10.1007/s12599-024-00921-3

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

Naqvi, B., Kedziora, D., Zhang, L., & Oyedeji, S. (2025). Quality of Low-Code/No-Code Development Platforms Through the Lens of ISO 25010:2023. *Computer, 58*(3), 30–40. https://doi.org/10.1109/MC.2024.3493192

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

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

Raković, L., Djordjevic Milutinovic, L., Vukovic, V., & Ubiparipovic, B. (2025). Low code/no code development drivers assessed by experts. *XIII INTERNATIONAL SCIENTIFIC CONFERENCE EKONBIZ 2025*, 26–32.

Raković, L., Đorđević, M., Lula, P., Vuković, V., Dziura, M., & Rojek, T. (2025). Low code/no code software development: Definition, drivers, and inhibitors. *Anali Ekonomskog Fakulteta u Subotici*, *54*, 147-164. https://doi.org/10.5937/AnEkSub2500007R

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

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

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

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

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

Takahashi, N., Javed, A., & Kohda, Y. (2024). How Low-Code Tools Contribute to Diversity, Equity, and Inclusion (DEI) in the Workplace: A Case Study of a Large Japanese Corporation. *Sustainability, 16*(13), 5327. https://doi.org/10.3390/su16135327

Turner, M., Kitchenham, B., Budgen, D., & Brereton, O. P. (2008). Lessons Learnt Undertaking a Large-Scale Systematic Literature Review. *EASE'08 Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, 110–118.

Viljoen, A., Altın, E. N., Hein, A., & Krcmar, H. (2024). Beyond Citizen Development: Exploring Low-Code Platform Adoption by Professional Software Developers. *AMCIS 2024 Proceedings - Adoption and Diffusion of IT (SIG ADIT).*

Viljoen, A., Radić, M., Hein, A., Nguyen, J., & Krcmar, H. (2024). Governing Citizen Development to Address Low-Code Platform Challenges. *MIS Quarterly Executive, 23*(3).

Viljoen, A., Stelzl, B., Yang, M., Nguyen, J., Hein, A., Elshan, E., & Krcmar, H. (2025). Navigating Flexibility and Standardisation in Low-Code/No-Code Development. *Information Systems Journal.* https://doi.org/10.1111/isj.70001

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

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

Zaheri, M., Famelis, M., & Syriani, E. (2024). Catch Me If You Can: Detecting Model-Data Inconsistencies in Low-Code Applications. *The Journal of Object Technology, 23*(1). https://doi.org/10.5381/jot.2024.23.1.a5

ИЗАЗОВИ ПЛАТФОРМИ СА МАЛО КОДИРАЊА ИЛИ БЕЗ КОДИРАЊА У ПОДРШЦИ ОРГАНИЗАЦИОНИМ ИНФОРМАЦИОНИМ ПРОЦЕСИМА: ПРЕГЛЕД ЛИТЕРАТУРЕ И ДОКАЗИ ИЗ СТУДИЈЕ СЛУЧАЈА

1 Лазар Раковић, Универзитет у Новом Саду, Економски факултет у Суботици, Суботица, Србија
2 Лена Ђорђевић Милутиновић, Универзитет у Београду, Факултет организационих наука, Београд, Србија
3 Павел Лула, Економски факултет у Кракову, Краков, Пољска
4 Богдан Убипариповић, Универзитет у Бањој Луци, Економски факултет, Босна и Херцеговина
5 Марек Ћјура, Економски факултет у Кракову, Краков, Пољска
6 Томаш Ројек, Економски факултет у Кракову, Краков, Пољска

САЖЕТАК

Дигитална трансформација све више позиционира пословну информатику у средиште организационе конкурентности, подстичући компаније да дигитализују и аутоматизују своје пословне процесе. У овом контексту, low-code/no-code (LCNC) платформе су се појавиле као обећавајуће рјешење у оквиру пословних информационих система, омогућавајући брз развој процесно оријентисаних апликација уз минимално програмирање или без њега. Оснаживањем запослених без формалног ИТ образовања да учествују у развоју система, LCNC платформе рјешавају проблем недостатка ИТ стручњака и помажу у премошћавању јаза између техничког и пословног знања из одређене области. Иако креатори LCNC платформи наглашавају предности као што су једноставност употребе, убрзани развојни циклуси, смањени трошкови, нижа зависност од ИТ-а и унапређена иновативност процеса, они често занемарују организационе, технолошке и управљачке изазове повезане са усвајањем LCNC-а. Овај рад систематски идентификује кључне инхибиторе и имплементацију LCNC-а кроз свеобухватан преглед литературе, након чега слиједи процјена њиховог значаја кроз вишеструке студије случаја са LCNC корисницима. Идентификовано је седам главних инхибитора: везаност за добављача, ризици безбједности и усаглашености, изазови интеграције, ограничена скалабилност, недовољна документација, ограничена подршка тестирању и недостатак флексибилности. Налази из студија случаја указују да недостатак флексибилности и прилагођавања, везаност за добављача и недовољна подршка тестирању представљају најкритичније баријере. Рад представља прелиминарне увиде из истраживања LCNC развоја које је у току.

Кључне речи: *развој са мало кодирања или без кодирања; развој од стране крајњих корисника; инхибитори.*