Using LowCode and NoCode tools in DevOps: A Multivocal Literature Review

Muhammad Waqas[1], Zohaib Ali[1], Mary Sánchez-Gordón[1], Monica Kristiansen[1],

[1] Østfold University College, Department of Computer Science and Communication, 1757 Halden, Norway
{muhammad.waqas, zohaib.ali, mary.sanchez-gordon, monica.kristiansen}@hiof.no

Abstract. DevOps and Low-code/No-code tools are two trends that are gaining interest in both academic and industry environments. This study analyzed the usage of low-code and no-code tools in DevOps. It examined the available tools, their applications, and the associated concerns or limitations. To do so, a Multivocal Literature Review (MVLR) was conducted to include academic and grey literature. The results reveal the utilization of a range of tools across multiple aspects of DevOps, including application delivery, workflow automation, process management, continuous integration, deployment, monitoring and logging. Despite their potential benefits, concerns and limitations such as script standardization, security risks, limited customization, vendor lock-in, and scalability issues persist. This review, alongside identified concerns, emphasizes the increasing adoption and benefits of Low-code/No-code tools in DevOps.

Keywords: Low-code, No-Code, DevOps Tools, Multivocal Literature Review

1 Introduction

DevOps integrates cultural values, practices, and technologies to improve an organization's capacity for producing applications and services quickly [1]. Practitioners consider DevOps to be a significant advantage over traditional software development methodologies since it reduces communication barriers over the development cycle [2].

The DevOps Research and Assessment (DORA) report states that high-performing organizations can deploy code up to 46 times more frequently than their low-performing counterparts [3]. Furthermore, a recent report conducted by Puppet points out that organizations that use DevOps practices experience a 63% decrease in the time needed to deploy new software [4]. Moreover, the International Data Corporation (IDC) survey found that organizations using DevOps practices experience a 17% improvement in team collaboration, which results in better alignment, quicker problem resolution, and better decision-making [5]. Thus, companies can enhance overall performance in their operations and software development processes by using DevOps techniques [6]. Despite the benefits, transforming a traditional product organization to a DevOps model may be expensive and time-consuming [7].

DevOps transformation requires more than just cultural adjustment and tools, as new governance frameworks, resources, and processes are also necessary [2, 8]. Challenges include unclear definitions, insufficient communication, established company cultures, and global distribution [9, 10]. Integrating various DevOps tools can be difficult due to compatibility issues and different skill requirements [11]. Short timelines in DevOps necessitate automation to speed up processes and reduce errors [11–13]. However, managing DevOps systems can be challenging without experienced engineers [14].

In this context, Low-code and No-code (LC and NC) tools offer promising solutions for quick and cost-effective application development and deployment [15]. These tools address various phases of the software development life cycle, facilitating collaboration and reducing anxiety related to cultural changes [2, 14]. LC and NC platforms provide intuitive interfaces, drag-and-drop functionality, visual modeling, and shared workflows, improving team collaboration [16]. Platforms like OutSystems and Mendix enable centralized collaboration and automate build and deployment processes [14, 16, 17]. Integrating DevOps with LC or NC approaches accelerates software deployment and helps meet continuous integration, deployment, and testing requirements.

In a recent analysis, Gartner predicted that the market for LC application development will increase at a compound annual growth rate (CAGR) of 23% between 2021 and 2026, reaching a market size of $45.5 billion by that year [18]. According to the global newswire, the growth of the LC industry has the potential to reach a global market size of $187 billion by 2032 [19]. The need for more projects involving digital transformation and for quicker software development and deployment are the primary factors of this rise. According to the same report from Gartner, the number of users of LC development tools will increase from 60% in 2021 to at least 80% by 2026 among developers outside of recognized IT departments.

Furthermore, compared to developers who only use high code, a survey by Appian [20] in 2020 revealed that LC developers are, on average, happy with their careers and the kinds of projects they get to work on. Because LC is simpler, quicker, and easier to use, 87% of developers with LC abilities say they enjoy their work with it. It enables developers to concentrate on more engaging tasks. These figures suggest that LC and NC tools are significantly influencing the DevOps market, enabling quicker and more effective software development while encouraging cooperation between development and operations teams [20].

Although LC/NC tools can play a significant role in the DevOps transformation, to the best of our knowledge there is no previous review on the available LC or NC tools in the context of DevOps. There are some previous studies about DevOps tools and the perception of practitioners about LC/NC tools, e.g., [14] (see Related Work section). Therefore, this study conducts a multivocal literature review (MVLR) to explore the various tools available in the DevOps market and identify which of these tools are reported as LC or NC. A MVLR allows us to include grey literature that ensures the collection of the avant-garde tools available in the DevOps market. As the emergence of LC and NC DevOps, tools can disrupt the market, our MVLR provides professionals and researchers with a comprehensive characterization of the available tools, the DevOps aspects that these tools address and their reported impact on the DevOps market.

The rest of this paper is structured as follows. In section 2, related work is presented. Section 3 describes the research methodology. Then, Section 4 presents the results and discusses them. Finally, conclusions and future work are presented in section 5.

2 Related Work

As mentioned before, LC and NC tools are gaining significant attention in the DevOps market due to their ability to improve the software development process. This section summarizes some of the relevant studies that have advanced our knowledge of the LC and NC landscape with respect to DevOps. The growing body of literature on LC and NC has explored the various kinds of tools that are available and how they impact the software development process.

One study conducted by Rafi et al. [14] suggests that combining DevOps with LC methodologies benefits software organizations by streamlining development and addressing talent shortages. The study highlights three advantages: accelerated application development, consistency maintenance, and bridging the skills gap. These authors highlighted that such an approach can lead to quality products being developed quickly and cost-effectively. Bock and Frank [21] analyzed the trend of LC platforms (LCPs) and their implications for software development. While LCPs show promise, little conceptualization exists about their differences from current tools. This study addresses concerns and concludes that LCPs primarily improve productivity by reducing effort in low to moderate complexity projects.

Kaess [22] presents a research model that integrates social, technical, and environmental considerations for the adoption of LCDPs (Low-code Development Platforms). The model combines socio-technical systems theory and the Technology-Environment-Organization (TOE) model, emphasizing business-driven processes for non-professional developers. Frank, Maier, and Bock [23] analyze LC platforms, focusing on their components and impact on software development. These authors argue that LC platforms integrate existing system design components to streamline business application development. They highlight the importance of conceptual modeling and suggest research opportunities related to evaluation, economic analysis, cognitive fit, and organizational behaviour.

To sum up, LC and NC tools are gaining significant attention in the DevOps market due to their ability to improve the software development process. While studies have explored the various kinds of tools available and how they impact software development, there is currently no comprehensive review that specifically examines the landscape of these tools in the context of DevOps. Our review paper aims to fill this gap in the literature by synthesizing existing research and providing a holistic view of the role and potential impact of LC and NC tools in DevOps and contributing positively to the ongoing discussion on the use of these tools in software development, providing valuable insights for both researchers and practitioners in the field.

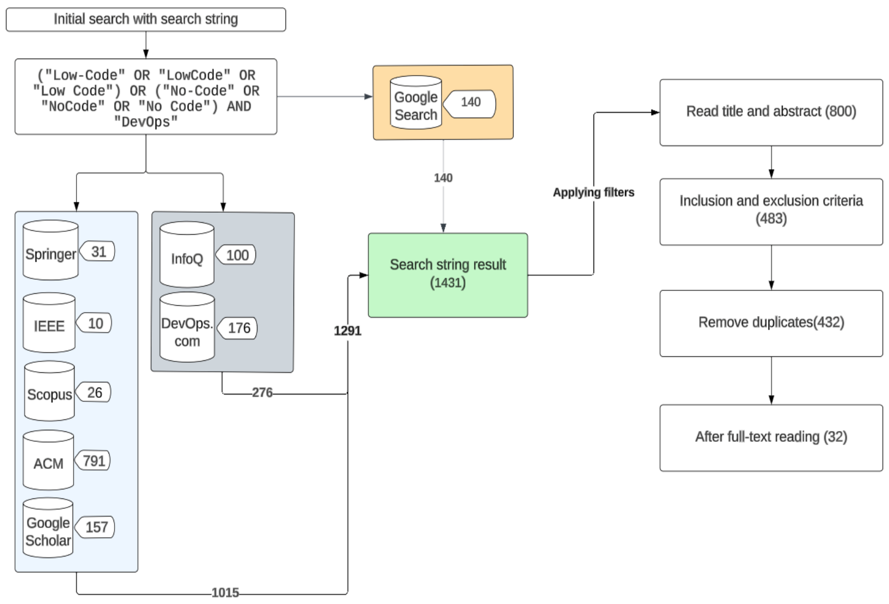
3 Research Methodology

3.1 Multivocal Literature Review

This paper utilized a Multivocal Literature Review (MVLR), which incorporates grey literature alongside published academic literature [24]. MVLRs in software engineering have included various information sources, such as blogs, videos, white papers, and web pages [25]. The inclusion of grey literature helps identify a comprehensive list of DevOps tools and provides multiple perspectives to support conclusions [24]. Moreover, conducting a MLVR on this emerging topic can help to identify areas where there is a lack of research and understanding [26]. This review focuses on both white and grey literature concerning LC and NC tools in the DevOps market. The MVLR planning phase consists of two phases [27]:

1. Describing the need to conduct a MVLR for a chosen topic.
2. Defining MVLR goals and research question.

We have already defined the purpose and motivation of conducting MVLR for the topic LC and NC available tools in the DevOps market. Figure 1 depicts an overview of the research process (see details in Section 3.7). Then, we define the research questions based on the research objectives.



Initial search with search string
("Low-Code" OR "LowCode" OR "Low Code") OR ("No-Code" OR "NoCode" OR "No Code") AND "DevOps"
Google Search
140
140
Springer
31
IEEE
10
Scopus
26
ACM
791
Google Scholar
157
InfoQ
100
DevOps.com
176
276
1015
1291
Search string result (1431)
Applying filters
Read title and abstract (800)
Inclusion and exclusion criteria (483)
Remove duplicates(432)
After full-text reading (32)

Fig. 1. Overview of the research process. Google's PageRank algorithm was used as a relevance ranking approach in Google Search and Google Scholar.

3.2 Research Questions

This literature review aims to understand the question, how can LC and NC tools support software development activities in the context of DevOps? This high-level research question is broken down into the following research questions (RQs).

- **RQ1:** What available LC and NC tools have been adopted/developed to support DevOps?
- **RQ2:** How have LC and NC tools impacted DevOps?
- **RQ3:** What are the concerns, or the limitations reported when using LC/NC tools in the context of DevOps?

3.3 Search strategy

To gather the sources included in this MVLR, an extensive screening of scientific and grey literature that pertained to the topic was conducted. The search period was not restricted, and information was gathered from relevant sources without a specific time limit to ensure a comprehensive search. The results of this research, which centered on the LC and NC tools in DevOps, were derived in large part from grey literature, scholarly journals and publications that were relevant to the topic under investigation.

3.4 Search string

```
("Low-Code" OR "LowCode" OR "Low Code") OR ("No-Code" OR
"NoCode" OR "No Code") AND "DevOps"
```

3.5 Databases

According to the guidelines [24, 25], different databases focus on specific areas of interest, catering to the specific needs and requirements within those domains. As we are interested in both grey and white literature, we include IEEE, Scopus, ACM, Springer and Google Scholar for white literature and DevOps.com, InfoQ, and Google Search for grey literature.

3.6 Selection criteria and process

To select the most relevant sources for our review, we applied the following inclusion and exclusion criteria to filter out irrelevant sources.

Inclusion Criteria

- Sources that discuss the LC and NC tools for DevOps.
- Sources that discuss concerns about using LC and NC tools in the context of DevOps.

Exclusion Criteria

- Sources that are not in the English language.
- Sources not available in full text.

- Advertisement or Job Post.

For grey literature, our search yielded a total of 176 references from DevOps.com and 100 from InfoQ, which provided valuable sources for our review. To restrict the search space in Google Search and Google Scholar, we utilized Google's PageRank algorithm as a relevance ranking approach. It means that relevant results usually only appear in the first few pages. Therefore, we checked only the first several pages (at least 10) and only continued further if needed so that we proceeded to the (n + 1)th page only if the results in the nth page still looked relevant. As a result, we analyzed 19 pages of Google search results, resulting in 140 potentially relevant sources. For white literature, we searched 20 pages of Google Scholar results, resulting in 157 relevant papers. Moreover, we included IEEE, Scopus, ACM and Springer and found a further 858 papers.

3.7 Data Extraction Process

The initial search results from eight different databases were 1431 sources. Papers with relevant titles and abstracts were added to our Zotero library, narrowing the selection to 483 papers. Duplicates were removed, and unique sources were subjected to full-text reading. Then, we applied our inclusion/exclusion criteria, and 32 sources were selected. Zotero was used to manage references, add notes and tags, and facilitate analysis. Relevant data from each source was extracted and recorded in a spreadsheet for further analysis. Moreover, a unique identifier was formulated and assigned to the 32 selected sources [PS01] - [PS32]. Consequently, the sources are referred to in that form throughout the rest of this paper (see the bibliographic details for selected sources in the Appendix).

Figure 2 shows that most of our selected sources were grey literature followed by white papers. This means that we relied heavily on grey literature to address our research question, emphasizing the importance of exploring and incorporating this type of literature in future research. It also suggests that further research studies are needed.

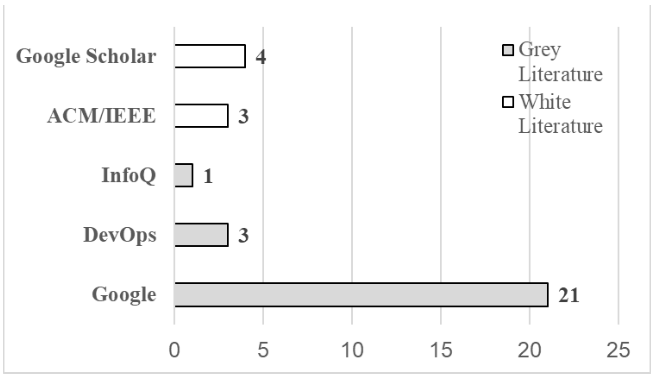


Fig. 2. Number of selected sources by type of literature.

Table 1 lists the research questions proposed in this review along with the ID and number of sources that answer each research question.

Table 1. Selected sources grouped by research question.

RQs	Source ID	# sources
RQ1	[PS1], [PS2], [PS3], [PS4], [PS5], [PS6], [PS7], [PS8], [PS9], [PS10], [PS11], [PS12]	12
RQ2	[PS1], [PS3], [PS9], [PS13], [PS14], [PS15], [PS16], [PS17], [PS18], [PS19], [PS20], [PS21], [PS22], [PS23], [PS24], [PS25], [PS26], [PS27], [PS28]	19
RQ3	[PS28], [PS1], [PS9], [PS14], [PS16], [PS20], [PS22], [PS24], [PS29], [PS30], [PS31], [PS32]	13

4 Results

The section presents the results of our MVLR on LC and NC tools in the context of DevOps. The review aimed to identify and analyze the available tools that have been adopted or developed to support software development activities in the DevOps context. Additionally, the review aimed to investigate the various DevOps aspects in which LC and NC tools have been used and to explore the concerns or limitations that have been reported when using these tools. Findings provide insights into the current state of LC and NC tools in DevOps, highlighting their potential benefits and drawbacks, and offering suggestions for future research and practice in this area.

4.1 What available LC and NC tools have been adopted/developed to support DevOps?

LC and NC platforms have gained significant traction in recent years as they allow businesses to create applications without the need for extensive coding experience. These tools can also be used to support various DevOps practices. In [PS1], Nguyen discusses the rising popularity of LC and NC development platforms, which utilize visual or drag-and-drop user interfaces to facilitate faster application development. This study also provides some examples of LC development use cases including creating e-commerce services, scheduling appointments, crafting customer relationship management (CRM) apps, building digital commerce apps and services, updating legacy systems, and designing telehealth services.

When introducing new tools to a DevOps ecosystem, it is important to consider their compatibility with existing automation and development stacks [PS2]. Bitton [PS2] suggests evaluating whether the tool supports the current development stack, can be used across multiple projects, and is compatible with different deployment configurations. Utilizing LC methods in DevOps automation can streamline the process by providing a visual representation and emphasizing configuration over extensive coding [PS3].

By reviewing the literature, ([PS2], [PS3], [PS4], [PS5], [PS6], [PS7], [PS8], [PS9], [PS10], [PS11], [PS12]) we identify the available LC and NC tools for DevOps:

- **Microsoft Power Platform:** This LCP includes Power Apps, Power Automate, and Power BI ([PS2], [PS3], [PS4], [PS6]). Power Apps offers features such as the Test Studio service for managing test cases and EasyRepro for UI (User Interface) testing. The monitoring feature allows developers to debug applications and improve performance ([PS7], [PS8], [PS9]).
- **Nintex**: This LCP is for process management and workflow automation [PS3], [PS5].
- **Appian:** This LCP enables quick deployment of enterprise applications using drag-and-drop tools, templates, and reusable components ([PS4], [PS5], [PS6], [PS10]).
- **Joget DX:** This platform blends LC application development with business process automation and workflow customization [PS11].
- **Stackstorm:** This platform focuses on event-driven automation with sensors, triggers, actions, rules, workflows, and audits [PS11].
- **OpenCatapult:** This platform is an open-source LC DevOps automation platform for managing routine tasks [PS11].
- **Mendix** is a key player in the LC development field, offering features like cloud-native development, mobile development, and process automation for LC DevOps ([PS2], [PS7], [PS9], [PS10]).
- **Amazon Web Services (AWS):** Provides LC and NC tools like AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy [PS13].
- **Microsoft Azure:** Offers LC and NC support through tools like Azure DevOps and Azure Functions [PS12].
- **Google Cloud Platform (GCP):** Provides DevOps tools including Cloud Build, Cloud Functions, and Cloud Deployment Manager for automated development, testing, and deployment [PS12].

To sum up, various LC and NC tools have been developed and adopted to support DevOps practices in response to the growing demand for efficient and accelerated application development. These tools, including those from Microsoft Power Platform, Nintex, Appian, Joget DX, Stackstorm, OpenCatapult, and Mendix, as well as those from cloud providers like AWS, Microsoft Azure, and Google Cloud Platform, enable companies to develop applications with little to no coding knowledge, streamline operations, and automate workflows. Organizations may improve teamwork, increase productivity, and speed up the time it takes to launch digital products by incorporating these tools into the DevOps ecosystem. Further automation and development are predicted to be introduced as the LC and NC environment continues to develop, enabling even better efficiency and agility in the DevOps space.

4.2 How have LC and NC tools impacted DevOps?

The increasing demand for accelerated software delivery has contributed to the growing popularity of LC and NC tools in the DevOps landscape [PS13]. These tools have been employed across a wide range of DevOps areas, offering numerous benefits and streamlining various processes. This section will delve into the specific DevOps areas where

LC and NC tools have found application and the reported advantages they provide in each area. By reviewing [PS3], we identified the following four broad areas:

1. **Application delivery:** LC and NC platforms have automated the development process, significantly reducing the time required for building and deploying software applications.
2. **Workflow automation:** DevOps emphasizes workflow automation to save time and resources by automating repetitive tasks and processes.
3. **Process management:** NC platforms offer customizable components and templates, simplifying the management of complex processes and improving operational efficiency.
4. **Composable enterprise:** LC and NC tools break down silos, foster collaboration, and enable organizations to respond rapidly to market changes and exploit new opportunities.

Additionally, LC and NC tools are employed in various DevOps tasks, including workflow development, configuration management, security automation, and runtime security [PS14]. According to [PS1], LC and NC tools have proven valuable in reducing time-to-market, minimizing resource consumption, promoting collaboration, and facilitating hyper-automation.

These types of tools are popular in application development, process automation, user experience improvement, and system upgrades ([PS15], [PS16]). They play a crucial role in ongoing maintenance by scanning code for risky design patterns, protecting against vulnerabilities, and ensuring secure and optimal application performance [PS17]. LCPs accelerate application development, deployment, and maintenance while promoting a DevOps culture [PS18]. They provide visual development environments, lifecycle management, transparency, accountability, and compliance [PS15].

LC and NC platforms have evolved to support continuous integration, deployment, monitoring, compliance, security, governance, and infrastructure independence [PS19]. They bridge the gap between development and operations, allowing users to deploy applications without relying on DevOps or engineering teams, [PS20]. These platforms supplement the lack of DevOps talent, enable rapid application creation and delivery, and reduce workloads through automation [PS18-45].

DevOps teams benefit from LC and NC tools in application development, automation, security, and infrastructure management ([PS1], [PS6], [PS9], [PS14], [PS28]). LCPs simplify the development process, allowing developers of different skill levels to work on applications throughout their lifecycle [PS21]. These tools increase deployment success rates, reduce errors, and offer consistency, reusability, and faster application development ([PS23], [PS24]). LCPs simplify scaling, and maintenance, and offer user-friendly interfaces, empowering business users to tackle operational challenges [PS25]. Moreover, according to [PS25], the use of LC and NC tools in DevOps leads to cost reduction, increased efficiency, and faster time-to-market for digital products.

Recently, AI-driven natural language processing tools like ChatGPT have emerged as a technology that impacts software development. According to [PS27], the integration of ChatGPT with LCPs has the potential to revolutionize the development process by enhancing usability and creating a new class of intelligent developer technology.

Although ChatGPT's code generation capabilities have limitations, its integration with LCPs can lower barriers to adoption, empower users with limited coding knowledge, and accelerate development cycles [PS27].

While challenges such as task formulation difficulties and the need for professional developers to validate and fix code exist, generative AI holds promise for the future of software development [PS26]. This integration can provide natural language processing capabilities for app development, offering shortcuts, recommendations, and auto code completion [PS27]. Generative AI has the potential to drive market growth and attract acquisitions in the intelligent developer technology market [PS59]. Therefore, despite its current limitations, ChatGPT, when integrated with LCPs, has the potential to significantly impact and improve software development processes [PS27].

In summary, LC and NC tools have been employed across various aspects of DevOps, demonstrating their versatility and potential in optimizing and streamlining development processes. These tools have been applied in application delivery, workflow automation, process management, composable enterprise, continuous integration and deployment, monitoring and logging, and infrastructure management, among others. By leveraging LC and NC platforms, organizations can accelerate application development, enhance collaboration, improve security and compliance, and ultimately achieve greater efficiency and reduced time-to-market for their digital products. The adoption of these tools in DevOps enables businesses to adapt to rapidly changing market conditions and fosters a culture of innovation and agility.

4.3 What are the concerns, or the limitations reported when using LC/NC tools in the context of DevOps?

Using LC and NC tools for DevOps automation has several concerns and limitations, as identified by multiple sources, e.g., ([PS28], [PS14]):

- **Standardization:** Standardizing scripts can be challenging due to the complexity of tasks and the time-consuming nature of fragile scripts.
- **Maintenance Difficulty:** LC and NC tools require maintenance skills, which may require a scripting mindset and skill set.
- **Quality Control:** Quality control is a concern, as these tools may lack rigour and testing compared to traditionally developed applications.
- **Learning Curve:** Adoption of LC and NC tools may require additional training and resources due to the learning curve.
- **Security and Governance:** Implications of using LC and NC tools may lead to less control in security and governance compared to traditional development approaches.
- **Vendor Lock-In:** LC and NC tools may be tied to specific platforms or technology stacks, leading to potential vendor lock-in.
- **Limited Customization:** LC and NC tools may have restricted customization options, potentially limiting their flexibility in meeting specific business requirements.

Given that citizen developers increasingly use NC and LC development tools to create applications and automate processes, security risks become a concern. Citizen developers are people who develop or modify software applications for either personal or business purposes, without any formal education or training in software development [28]. [PS29] emphasizes the importance of integrating security throughout the application lifecycle and adopting DevSecOps to mitigate vulnerabilities in modern applications. Concerns about scalability, limited customization opportunities, and potential vendor lock-in have also been raised ([PS1], [PS16]). Some argue that LC approaches may be unsuitable for complex applications, and NC automation has been criticized for its limited flexibility and customizability in cybersecurity ([PS1], [PS30]). Security risks associated with proprietary libraries, lack of security training for citizen developers, and potential risks of API integrations have been identified ([PS16], [PS31]).

Mitigation strategies include static code analysis, auditing proprietary libraries, providing security training, and evolving security approaches to account for risks introduced by LC and NC tools ([PS16], [PS31], [PS32]). LC and NC tools have limitations such as limited customization, scalability issues, control over the development process, performance challenges with large data volumes, maintenance complexities, tool obscurity, compliance risks, security vulnerabilities, difficulties integrating with legacy systems, technical debt, and limited support for complex business logic and workflows ([PS10], [PS13], [PS21], [PS23], [PS25], [PS17]). Careful evaluation of these capabilities and limitations is necessary before adopting LC and NC tools in a DevOps context [PS32].

In conclusion, while LC and NC tools have shown significant potential in the context of DevOps, several concerns and limitations have been reported. These include standardizing scripts, time-consuming complex tasks, security risks, limited customization opportunities, vendor lock-in, scalability issues, and the need for a scripting mindset and skill set. Additionally, concerns regarding quality control, learning curve, security and governance, limited customization, and vendor lock-in have been raised. Despite these limitations, the ongoing innovation in the LC and NC platform market is addressing some of these concerns, with some platforms now offering technologies to integrate with DevOps processes. Organizations should carefully evaluate the capabilities and limitations of these tools before adopting them in a DevOps context, ensuring that they remain agile and responsive while mitigating potential risks.

5 Implications

5.1 Implications for Practitioners

The implications of our research extend to practitioners who are working at the intersection of DevOps, LC, and NC. As LC and NC tools are demonstrating the ability to expedite application development and deployment, practitioners can leverage these tools to address a range of DevOps aspects, including application delivery, workflow automation, and continuous integration. However, despite their potential benefits, practitioners should be aware of the reported concerns and limitations such as script standardization, security risks, limited customization, vendor lock-in, and scalability issues. Adopting these tools should be done with a clear understanding of the related challenges and strategies for managing them. Furthermore, recognizing the potential for market disruption and significant growth in LC and NC tools, practitioners should stay updated with current trends and make informed decisions about tool adoption to remain competitive in their respective domains.

5.2 Implications for Researchers

The rapid growth and increasing adoption of LC and NC tools within the DevOps domain highlight an important area for future research. Our study reveals that there is a need for more research on this topic since a large part of the literature used in our review is grey literature. Researchers should continue exploring this intersection to understand the critical success factors and evaluate the potential impacts of these tools on software development and deployment practices. Furthermore, they should also focus on addressing the reported concerns and limitations of using these tools. Additionally, studies that offer a more nuanced understanding of the interaction between these tools and DevOps practices, their impacts on productivity, and their potential role in facilitating digital transformation would be particularly beneficial. Overall, researchers have a crucial role in further exploring, conceptualizing and assessing the role and potential impact of LC and NC tools in DevOps.

6 Conclusions and Future Work

This review examines LC and NC tools in the context of DevOps through a MVLR. Various tools in the DevOps market are identified, along with their applications and reported concerns or limitations. The adoption of LC and NC tools in DevOps shows their potential to streamline development processes, improve collaboration, and bridge talent shortages. However, concerns remain regarding standardizing scripts, security risks, limited customization, vendor lock-in, and scalability issues. Continuous innovation aims to address some of these concerns, requiring careful evaluation before adoption.

There are limitations to this research, including potential bias in source selection, English language sources only, evolving nature of LC/NC tools, lack of practical implementation insights, subjective data interpretation, and limited generalizability. To mitigate these threats, we formulated a protocol to conduct this review and more than one author was involved in each step. Future research can address these limitations by including non-English sources, longitudinal and case studies. Overall, this review contributes to understanding LC and NC tools in the DevOps context, providing valuable insights for professionals and researchers. Further investigation into LC/NC tools is crucial to maximizing their potential and addressing identified concerns and limitations as DevOps evolves.

References

1. What is DevOps? - Amazon Web Services (AWS), https://aws.amazon.com/what-is-devops/, last accessed 2023/03/02.
2. Nahla, D.: The Future of DevOps is No-Code, https://www.infoq.com/articles/devops-future-no-code/, last accessed 2023/01/24.
3. 2022 State of DevOps Report, https://cloud.google.com/devops/state-of-devops, last accessed 2023/03/02.
4. 2020 State of DevOps Report: Presented by Puppet and CircleCI, https://circleci.com/resources/state-of-devops-report-2020/, last accessed 2023/03/02.
5. Elliot, S.: DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified.
6. Azad, N., Hyrynsalmi, S.: DevOps critical success factors — A systematic literature review. Inf. Softw. Technol. 157, 107150 (2023). https://doi.org/10.1016/j.infsof.2023.107150.
7. Senapathi, M., Buchan, J., Osman, H.: DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. In: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018. pp. 57–67. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3210459.3210465.
8. Lwakatare, L.E., Kuvaja, P., Oivo, M.: Dimensions of DevOps. In: Lassenius, C., Dingsøyr, T., and Paasivaara, M. (eds.) Agile Processes in Software Engineering and Extreme Programming. pp. 212–217. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_19.
9. Smeds, J., Nybom, K., Porres, I.: DevOps: A Definition and Perceived Adoption Impediments. In: Lassenius, C., Dingsøyr, T., and Paasivaara, M. (eds.) Agile Processes in Software Engineering and Extreme Programming. pp. 166–177. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_14.
10. Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L.E., Tiihonen, J., Männistö, T.: DevOps Adoption Benefits and Challenges in Practice: A Case Study. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., and Mikkonen, T. (eds.) Product-Focused Software Process Improvement. pp. 590–597. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-49094-6_44.
11. Gill, A.: DevOps: Concepts, Practices, Tools, Benefits and Challenges. In: the Pacific Asia Conference on Information Systems (PACIS). pp. 1–12. AIS Electronic Library (2017). http://aisel.aisnet.org/pacis2017/96.
12. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: A Survey of DevOps Concepts and Challenges. ACM Comput. Surv. 52, 127:1-127:35 (2019). https://doi.org/10.1145/3359981.
13. Khan, A.A., Shameem, M.: Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process. J. Softw. Evol. Process. 32, e2263 (2020). https://doi.org/10.1002/smr.2263.
14. Rafi, S., Akbar, M.A., Sánchez-Gordón, M., Colomo-Palacios, R.: DevOps Practitioners 2019; Perceptions of the Low-code Trend. In: Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 301–306. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3544902.3546635.
15. Talesra, K., G. S., N.: Low-Code Platform for Application Development. Int. J. Appl. Eng. Res. 16, 346 (2021). https://doi.org/10.37622/IJAER/16.5.2021.346-351.
16. How to Do DevOps With Low-Code, Pros and Cons of Low-Code Development, last accessed 2023/03/03.

17. DevOps, https://www.mendix.com/evaluation-guide/app-lifecycle/devops/, last accessed 2023/03/03.
18. Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 20% in 2023, https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023, last accessed 2023/02/21.
19. FACT.MR: Low Code Development Industry is Projected to Achieve a Global Market Size of US$ 187 Bn by 2032, Currently US Accounts For the Largest Market Share in the World, https://www.globenewswire.com/news-release/2022/03/09/2400432/0/en/Low-Code-Development-Industry-is-Projected-to-Achieve-a-Global-Market-Size-of-US-187-Bn-by-2032-Currently-US-Accounts-For-the-Largest-Market-Share-in-the-World.html, last accessed 2023/02/21.
20. Appian: Low-Code Developer Survey Report: State of Low-Code for Developers. (2021).
21. Bock, A., Frank, U.: Low-Code Platform. Bus. Inf. Syst. Eng. 63, (2021). https://doi.org/10.1007/s12599-021-00726-8.
22. Kaess, S.: Low Code Development Platform Adoption: A Research Model. ACIS 2022 Proc. (2022).
23. Frank, U., Maier, P., Bock, A.: Low code platforms: Promises, concepts and prospects. A comparative study of ten systems. ICB-Research Report (2021). https://doi.org/10.17185/duepublico/75244.
24. Kitchenham, B.A., Dybå, T., Jørgensen, M.: Evidence-based software engineering. Presented at the Proceedings - International Conference on Software Engineering (2004).
25. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf. Softw. Technol. 106, 101–121 (2019). https://doi.org/10.1016/j.infsof.2018.09.006.
26. Garousi, V., Felderer, M., Mäntylä, M.V.: The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering. pp. 1–6. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2915970.2916008.
27. Ogawa, R.T., Malen, B.: Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method. Rev. Educ. Res. 61, 265–286 (1991). https://doi.org/10.3102/00346543061003265.
28. Lebens, M., J Finnegan, R., C Sorsen, S., Shah, J.: Rise of the Citizen Developer. Muma Bus. Rev. 5, 101–111 (2021). https://doi.org/10.28945/4885.

Appendix: Selected Sources

ID	Year	Authors. *Title*	Database
PS1	2023	Nguyen, Spencer. *Pros and Cons of Low-Code Development*	Google
PS2	2022	Bitton, David. *Modern methods for Low-code DevOps*	Google
PS3	2023	Codemotion. *Will Low-Code Take Over the World in 2023?*	Google
PS4	2021	Walker, James. *What Is Low-Code and No-Code Development?*	Google
PS5	2022	Christian, Bluein. *Top 10 Low Code Development Platforms in 2022-23*	Google
PS6	2022	Ankit Sahu.Ten Best Low Code Platforms to Use in 2023	Google
PS7	2020	Khorram, F.; Mottu, J.-M.; Sunyé, G.. *Challenges & opportunities in low-code testing*	ACM/IEEE
PS8	2020	Ramel, David. *Microsoft Details Low-Code DevOps*	Google

PS9	2021	Saffa, Sahr. *Challenges to Overcome When Implementing Low-Code Development | Sauce Labs*	Google
PS10	2023	Elshan, Edona; Dickhaut, Ernestine; Ebel, Philipp. *An Investigation of Why Low Code Platforms Provide Answers and New Challenges*	Google Scholar
PS11	2021	Mousa, Hamza.*Top 18 Open-source Free Low- and No-Code platforms for enterprise*	Google
PS12	2020	Sanchis, Raquel; García-Perales, Óscar; Fraile, Francisco; Poler, Raul. *Low-Code as Enabler of Digital Transformation in Manufacturing Industry*	Google Scholar
PS13	2020	Layne, Skyler. *No-Code Continuous Integration and Delivery : A Developer Perspective*	Google
PS14	2022	Jonathan Roquelaure.The Role of No-Code/Low-Code Tooling in DevOps Automation - Blink	Google
PS15	2017	Easter, Dave. *Low-Code Development Platform in DevOps*	Google
PS16	2022	Arul, Akashdeep. *IT professionals and DevOps say no to low-code*	Google
PS17	2021	Alamin, A. A.; Malakar, S.; Uddin, G.; Afroz, S,; Haider, T. B.; Iqbal, A. *An Empirical Study of Developer Discussions on Low-Code Software Development Challenges*	IEEE
PS18	2022	Josephson. Mike. *How to Do DevOps With Low-Code*	Google
PS19	2020	Duensing, Mike. *How low-code/no-code platforms may reinvent DevOps*	Google
PS20	2021	Alyousef, Zaher. *Challenges Development Teams Face In Low-code Development Process*	Google Scholar
PS21	2021	Manby, Andrew. *How Low-Code Makes DevOps Stronger*	DevOps
PS22	2021	eWEEK EDITORS. *DevOps, Low-Code and RPA: Pros and Cons*	Google
PS23	2022	Van Ash, Tim. *How DevOps Automation Solves Low-Code Security Issues*	Google
PS24	2021	Fashakin, Alexander. *How No-Code/Low-Code Impacts DevOps*	Google
PS25	2022	Arunachalam, Arunkumar. *IT professionals and DevOps say no to low-code*	InfoQ
PS26	2023	Vladimirov, Vadym. *Will Chat GPT make low-code obsolete?*	Google
PS27	2023	Ghoshal, Anirban. *Why generative AI will turbocharge low-code and no-code development*	Google
PS28	2022	Rafi, S.; Akbar, M. A.; Sánchez-Gordón, M.; Colomo-Palacios, R.. *DevOps Practitioners' Perceptions of the Low-code Trend*	ACM
PS29	2021	Vistola, Louis. *Security Risks With No-Code/Low-Code Tools*	DevOps
PS30	2022	Forte, Dario. *No-code vs. low-code and near-no-code security automation*	Google
PS31	2021	Doerrfeld, Bill. *How to Mitigate Low-Code Security Risks*	DevOps
PS32	2020	Woo, Marcus.*The Rise of No/Low Code Software Development—No Experience Needed?*	Google Scholar