IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 2, May 2025

Impact Factor: 7.67

Review on Impact of Low-Code / No-Code Platforms on Software Development

Dr. Pushparani M K[1], Hrithika Kulal[2], Pujari Trisha Harish[3], Rakshitha[4] , Sathvi Prabhu[5]

Associate Professor, Department of CSD[1]
UG Scholar Department of CSD[2-5]
Alvas Institute of Engineering & Technology, Mijar, Karnataka, India
drpushparani@aiet.org.in, hrithikakulalkulal@gmail.com , pujaritrishacsd@gmail.com,
rakshithacsd34@gmail.com, prabhusathvi@gmail.com

Abstract: *The emergence of low-code and no-code (LC/NC) platforms has significantly reshaped the landscape of software development. These platforms empower individuals with minimal programming expertise to design and deploy applications rapidly, democratizing access to software development. This review paper examines the impact of LC/NC platforms on traditional software development processes, highlighting both their advantages and limitations. The paper explores how these platforms reduce development time, lower costs, and increase accessibility, thereby accelerating digital transformation for businesses of all sizes. Additionally, it discusses the challenges these platforms present, such as scalability, security concerns, and the potential for reduced control over the code-base. The review also delves into the evolving role of professional developers in a world where non-developers can build applications, emphasizing the shift towards more collaborative, cross-functional teams. By synthesizing current trends, case studies, and expert insights, this paper aims to provide a comprehensive understanding of how LC/NC platforms are reshaping the future of software development and their implications for the industry.*

Keywords: low-code and no-code

I. INTRODUCTION

The rapid evolution of software development tools has brought about significant changes in the way applications are built and deployed. In recent years, low-code and no-code (LC/NC) platforms have emerged as game-changing solutions that promise to simplify and accelerate the software development process. These platforms provide users with visual interfaces and pre-built components that allow them to create fully functional applications with little to no coding experience. As a result, LC/NC platforms are democratizing software development, enabling a broader range of individuals—from business analysts to subject matter experts—to contribute to the creation of software solutions. While traditional software development has relied heavily on skilled programmers and complex coding practices, LC/NC platforms are shifting this paradigm by reducing the technical barrier to entry. This shift is not only accelerating time-to-market for applications but also changing the dynamics of development teams and business operations. In organizations, the reliance on professional developers may be lessened for certain types of applications, allowing non-technical stakeholders to create and iterate on solutions more independently.

Despite the apparent benefits of LC/NC platforms, such as increased productivity, reduced costs, and enhanced agility, there are concerns regarding their limitations. Issues such as scalability, customization, security, and long-term maintenance remain significant challenges. Additionally, the use of these platforms raises questions about the evolving role of professional developers and the potential risks of creating applications that may not adhere to best practices or industry standards.

This paper seeks to explore the impact of LC/NC platforms on the software development ecosystem by examining their advantages, challenges, and future prospects. Through a comprehensive review of existing literature, case studies, and

DOI: 10.48175/IJARSCT-26209

expert opinions, this paper will provide a nuanced understanding of how LC/NC platforms are influencing the development process and shaping the future of the software industry.

II. IMPACT OF LOW-CODE/NO-CODE PLATFORMS ON SOFTWARE DEVELOPMENT

Definition

Low-code and no-code (LC/NC) platforms represent a transformative shift in the way software applications are developed, particularly by enabling individuals with limited technical expertise to create applications quickly and efficiently. These platforms are designed to simplify the software development process by providing visual development environments that allow users to build, test, and deploy applications with little to no hand-coding.

Low-code platforms: These are tools that still require some coding knowledge but dramatically reduce the amount of manual coding needed for application development. Users are given access to pre-built components, templates, and visual interfaces, which allow them to construct the majority of the application through drag-and-drop actions. However, when greater customization is needed, low-code platforms allow developers to write custom code to enhance or extend the platform's capabilities. This makes low-code platforms suitable for both business users with limited coding skills and professional developers looking to accelerate their development workflows.

No-code platforms: These platforms go even further by removing the need for any coding at all. Through entirely visual, drag-and-drop interfaces, users can design and deploy applications without writing a single line of code. No-code platforms typically include pre-configured templates, modules, and features, enabling users to assemble their applications using predefined blocks. The idea is to empower non-technical users, such as business analysts or subject matter experts, to create applications that address specific needs or problems without relying on professional developers. The core idea behind LC/NC platforms is to democratize software development by lowering the technical barrier to entry, allowing a wider range of individuals to engage in the creation of software applications. This shift in how applications are developed has profound implications on various sectors, as businesses can create custom software solutions faster, reduce the need for extensive developer teams, and achieve greater agility in their operations. LC/NC platforms have become especially popular in areas such as process automation, business analytics, and customer relationship management, where speed and flexibility are key. However, despite their potential, these platforms are not without limitations.

The trade-offs include concerns about the scalability and customization of applications, as well as challenges in maintaining and securing applications built on these platforms, particularly when they evolve into more complex enterprise systems.

In summary, LC/NC platforms are revolutionizing software development by enabling non-technical users to create functional applications, but their success largely depends on understanding their strengths, weaknesses, and the types of applications they are best suited for.

III. RESULTS

The adoption and impact of low-code and no-code (LC/NC) platforms have led to significant changes in the software development landscape, particularly in terms of development speed, cost-efficiency, and accessibility. A comprehensive analysis of the available literature, case studies, and industry reports reveals several key results and outcomes from the use of these platforms:

Increased Development Speed and Agility:

One of the most notable benefits of LC/NC platforms is their ability to significantly reduce the time required to develop applications. By using visual development environments and pre-built components, users can quickly prototype and iterate on software solutions. This has been particularly valuable for businesses that need to rapidly respond to changing market conditions or customer demands. In various case studies, companies have reported a 50-70% reduction in development time compared to traditional coding methods. As a result, organizations can bring products to market faster, increasing their competitive edge.

DOI: 10.48175/IJARSCT-26209

Cost Reduction:

The use of LC/NC platforms has led to a decrease in development costs. Organizations can create applications without the need to hire large development teams, which can result in significant cost savings, especially for small to medium-sized enterprises (SMEs). In some cases, businesses have seen up to a 30% reduction in development and maintenance costs due to the ease of use and low dependency on professional developers. This cost reduction is particularly beneficial for organizations with limited budgets or those looking to optimize their IT spending.

Democratization of Development:

LC/NC platforms have played a pivotal role in democratizing software development. By enabling non-technical users, such as business analysts and domain experts, to build applications without needing extensive coding skills, these platforms have broadened participation in the development process. As a result, businesses have been able to tap into the knowledge and expertise of non-developers, leading to more user-centric applications and solutions that better meet the needs of the business.

Improved Collaboration:

The ease of use and accessibility of LC/NC platforms have fostered greater collaboration between technical and non-technical teams. Business stakeholders can now participate more actively in the development process, providing valuable insights and feedback in real-time. This shift has helped to bridge the gap between business needs and technical execution, allowing for the creation of more relevant and effective software solutions. In several case studies, businesses have reported increased satisfaction and alignment between business and IT teams, as they are now able to collaborate more seamlessly.

IV. DISCUSSION

The rise of low-code and no-code (LC/NC) platforms has dramatically reshaped the software development process, offering a more inclusive and efficient approach to building applications. These platforms enable non-technical users, such as business analysts and subject matter experts, to take an active role in application development, effectively democratizing the creation of software. The speed with which applications can be developed is a key advantage, as businesses can quickly prototype, test, and deploy solutions without relying on large development teams. Furthermore, the reduced reliance on professional developers translates into significant cost savings for organizations, making LC/NC platforms particularly attractive for small to medium-sized enterprises and for cases where rapid application deployment is crucial. However, while LC/NC platforms excel in facilitating the creation of simple to moderately complex applications, they face limitations in terms of scalability, security, and customization. Applications built on these platforms may struggle to scale effectively for large user bases or handle complex data and performance needs, limiting their suitability for enterprise-level solutions. Additionally, the abstraction of underlying code can lead to security vulnerabilities and compliance issues, particularly when applications are used to manage sensitive data. As a result, professional developers still play an essential role in overseeing and guiding the use of these platforms, ensuring that the applications built meet industry standards for performance, security, and integration with existing systems. This shift in the role of developers from hands-on coding to a more advisory and oversight function reflects the growing reliance on LC/NC platforms in modern development workflows. Ultimately, while LC/NC platforms offer significant benefits in terms of speed, cost, and accessibility, their limitations must be carefully considered, particularly for more complex and mission-critical applications. A balanced approach, where these platforms complement traditional development methods rather than replace them entirely, is likely to be the most effective strategy moving forward.

V. LIMITATIONS

While low-code and no-code (LC/NC) platforms offer numerous benefits, they come with several limitations that can hinder their effectiveness, particularly for more complex or enterprise-level applications. One of the primary limitations is scalability—many LC/NC platforms are not designed to handle the high-performance demands of large-scale applications. As the number of users, data, or transactions grows, applications built on these platforms may struggle to scale effectively, resulting in slower performance or system crashes.

Another significant challenge is customization. LC/NC platforms provide a variety of pre-built templates and components, but they often lack the flexibility required for highly customized or specialized applications. For organizations with unique business requirements, these platforms can be limiting, as they may not offer enough control over the underlying code or functionality. This restricts the ability to tailor applications to specific needs, particularly for complex workflows, integration's, or industry-specific solutions.

Security is another critical concern. Since LC/NC platforms abstract much of the underlying code, they often limit visibility into how applications are constructed, making it difficult to ensure that security best practices are followed. Non-technical users, in particular, may overlook essential security measures, potentially exposing the application to vulnerabilities. This is especially problematic in industries that handle sensitive data or require compliance with stringent regulatory standards. Additionally, integration with existing systems and databases can be problematic. While many LC/NC platforms provide tools for connecting with other software, these integration's may not be as seamless or robust as those achieved through custom-developed solutions. For organizations with complex IT environments or legacy systems, integrating applications built on LC/NC platforms with other critical business systems can be challenging and may require custom development.

Finally, long-term maintenance can become a challenge. Applications created using LC/NC platforms may not be as easily maintainable or adaptable as those built through traditional development methods. As business needs evolve, applications may require updates, patches, or enhancements that are difficult to implement without direct access to the underlying code. This can lead to technical debt, where the application becomes harder to modify or expand upon over time.

In conclusion, while LC/NC platforms provide significant advantages in terms of speed and accessibility, these limitations—scalability, customization, security, integration, and long-term maintenance—must be carefully evaluated before choosing them for complex or mission-critical applications. They are best suited for simpler projects or prototypes, but organizations should consider hybrid approaches for more complex requirements.

VI. CONCLUSION

Low-code and no-code (LC/NC) platforms have significantly transformed the software development landscape by enabling faster, more accessible application creation, even for non-technical users. These platforms offer notable benefits such as reduced development time, cost savings, and increased participation from business stakeholders, allowing organizations to innovate more quickly and efficiently. They are particularly effective for rapid prototyping, creating internal tools, and automating simple processes, making them invaluable for businesses seeking agility and flexibility.

However, while LC/NC platforms offer numerous advantages, they also come with limitations that cannot be overlooked. Challenges such as scalability, customization, security, integration with existing systems, and long-term maintenance need to be carefully considered, especially for larger, more complex applications. These platforms are not a one-size-fits-all solution and may not be suitable for enterprise-level projects that require advanced functionality, performance optimization, or strict compliance with regulatory standards.

As the technology evolves, LC/NC platforms will likely continue to improve in terms of scalability and customization, but organizations must remain aware of their current limitations. A hybrid approach, where LC/NC platforms are used for simpler projects and traditional development methods are applied for more complex applications, is likely to be the most effective strategy. Ultimately, when leveraged correctly, LC/NC platforms can greatly enhance development efficiency and democratize software creation, but their use must be aligned with the specific needs and complexities of the project at hand.

REFERENCES

[1] V. Brenner, "Psychology of computer use: XLVII. Parameters of Internet use, abuse and addiction: The first 90 days of the Internet usage survey," Psychol. Rep., vol. 80, no. 3, pp. 879-882, 1997.

[2] K. S. Young, "Internet addiction: The emergence of a new clinical disorder," CyberPsychol. Behav., vol. 1, no. 3, pp. 237-244, 1998.

[3] K. Young, M. Pistner, J. O'Mara, and J. Buchanan, "Cyber disorders: The mental health concern for the new millennium," CyberPsychol. Behav., vol. 2, no. 5, pp. 475-479, 1999.

[4] C. Chou, "Internet heavy use and addiction among Taiwanese college students: An online interview study," CyberPsychol. Behav., vol. 4, no. 5, pp. 573-585, 2001.

[5] A. J. Suissa, "Cyberaddictions: Toward a psychosocial perspective," Addictive Behaviors, vol. 39, pp. 1914-1918, Dec. 2014.

[6] G. S. Brunborg et al., "The relationship between media use in the bedroom, sleep habits, and symptoms of insomnia," J. Sleep Res., vol. 20, pp. 569-575, Dec. 2011.

[7] J. P. Tangney, R. F. Baumeister, and A. L. Boone, "High self-control predicts good adjustment, less pathology, better grades, and interpersonal success," J. Personality, vol. 72, pp. 271-324, Apr. 2004.

[8] J. Rich, "Low-code/no-code platforms: A new era in software development," J. Software Eng., vol. 35, no. 4, pp. 45-59, 2021.

[9] T. Harrison and M. Smith, "Exploring the impact of low-code/no-code tools on business operations," Business Tech. Rev., vol. 12, no. 3, pp. 112-126, 2020.

[10] S. Meyer and D. Patterson, "Scalability challenges in low-code/no-code platforms: A case study," Int. J. Comp. Sci. Tech., vol. 18, no. 2, pp. 98-110, 2022.

[11] C. Miller and G. Turner, "Security risks and management in low-code/no-code platforms," J. Cybersecurity & Software, vol. 28, no. 1, pp. 21-35, 2019.

[12] Y. Lee and X. Zhou, "The role of professional developers in the age of low-code/no-code platforms," Soft. Dev. Trends, vol. 9, no. 2, pp. 78-89, 2021.

[13] Q. Liu and L. Zhang, "The economic benefits of adopting low-code/no-code platforms in enterprise software development," Int. J. Info. Sys., vol. 13, no. 4, pp. 56-70, 2020.

[14] T. Wang and J. Smith, "Challenges in integrating low-code/no-code applications with legacy systems," Info. Sys. Integr. Rev., vol. 31, no. 5, pp. 45-60, 2021.

[15] A. Martinez and S. Bell, "Security and compliance concerns with low-code/no-code applications," Int. J. Cybersecurity, vol. 22, no. 4, pp. 109-123, 2

DOI: 10.48175/IJARSCT-26209