International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 13 Issue 4 ǁ April 2025 ǁ PP. 208-212

Impact of Low-Code / No-Code Platforms

Santhosh Kusuma Kumar Parimi
Senior IEEE Member Software Engineering, Researcher & Machine Learning Specialist Austin, Texas, USA

Abstract*—Low-code and no-code platforms have revolutionized software development by leaps and bounds. These tools have enabled individuals with minimal coding experience to create and deploy applications quickly and efficiently. These platforms offer ready to use plugins to accelerate development along with options such as drag-and-drop components, pre-built templates to ease accessing various libraries. The tools help software developers to create software programs at speed unlike traditional methods of manually writing every piece of code.*

This paper outlines the impact of low-code/no-code platforms, including their function in speeding innovation and reducing development costs, as well as addressing the growing demands of digital transformation. At the same time, it looks at the challenges and obstacles from scalability through security to customization that organizations need to consider when implementing these platforms. Furthermore, the potential future of low-code/no-code development in forging collaboration between technical and non-technical teams in software development. Ultimately, these platforms will bridge the gap between company needs and IT solutions much faster. However, their integration into enterprise settings must be carefully managed so as to ensure sustainability as well as efficacy.

***Index Terms*—** *Low-code, No-code platforms, Software development, Digital transformation, Citizen developers, Innovation*

Date of Submission: 12-04-2025 Date of acceptance: 23-04-2025

I. INTRODUCTION

Low-code and no-code platforms are enabling a paradigm shift in the software development space by democratizing application development. These platforms offer intuitive interfaces allowing people with little to no coding experience to create, build and deploy software solutions. They empower businesses to solve their business problems independently, using visual tools, drag-and-drops, and prebuilt templates to meet the needs of technical and non-technical users, without having to depend entirely on an IT department or professional developer.

These platforms have much more reaching implication beyond ease of development. With this capability, organizations are now capable of speeding up their time-to- market, enhancing agility, and cutting the cost of traditional development. Furthermore, low-code and no-code platforms also drive innovation by empowering employees in different departments such as marketing, finance, and operations to create customized solutions that solve their unique challenges.

Enabling "citizen developers" has become a major catalyst of digital transformation for organizations to remain competitive in the fast-evolving landscape.

Nevertheless, there are some issues with how widespread low-code and no-code platforms have become. Feature set and functionality questions remain, and largely around scalability and data security and long-term maintainability specifically in enterprise environments. Nonetheless, their ability to increase productivity, automate workflows, and democratize technology is a catalyst for modern development, and they will likely continue to remain a mainstay of development. Not only are these being used more and more, they are going to change the way organizations innovate based on changing business demands.

Since the birth of the first computer, ENIAC (electronic numerical integrator and computer) in 1943, the way of writing algorithms has evolved from machine code towards low-code-no-code (LCNC). As seen from Fig. 1, in 1943, machine codes were written in binary which could be comprehended by early generation computers [1]. In 1949, the birth of assembly languages simplified machine code. Then, in 1952, Autocode enabled the programmers to write algorithms in low-level computer programming language, which could then be translated into machine codes [2]. Writing algorithms became much simpler with fewer lines of code, with the birth of high-level procedural languages (e.g., Fortran, Algol, COBOL) in 1957 [2,3]. Then, from 1965 onwards, the way of writing algorithms veered to be in object-oriented languages with the birth of Simula and Smalltalk. In 2001, Outsystems was born with the vision of faster delivery of digital transformation [4].

II. RESEARCH METHODOLOGY

Purpose: To identify and understand existing research on the area of LCNC (Low Code No Code platforms)

Approach: Systematic or narrative review of academic articles, industry reports, white papers, and case studies.

Expected Outcome: Expected outcome of this research is thorough understanding of LCNC ((Low Code No Code platforms), its evolution. Its impact in Software life Cycle and future scope.

A. Research Gap

Although low-code and no-code platforms are being rapidly adopted and growing in importance, there are still several research gaps remaining to be filled. One key area is in the long-term scalability and sustainability of applications that these platforms build. They may work well enough for simple to moderately complicated applications but whether they can be scaled to enterprise-level performance, reliability and maintainability with the passage of time is an open question. These platforms need to be extensively investigated on the basis of the scale and nature of their deployments, the complexity of integrations between various participation points, and the changing landscape of business demand without rendering too much technical debt in the architecture.

A related avenue that has been not scrutinized enough is the security and compliance implications of the widespread adoption of low-code/no-code. With these platforms enabling non-technical users to build software, the likelihood of misconfigurations, vulnerabilities, and a lack of industry standard compliance also rises. Built-in security refer to the security features provided by application development tools and frameworks, including security controls or guidelines available to application developers to foster secure development practices; This leads researchers to conduct the in-depth and extensive analysis to evaluate the robustness of these built-in security measures, the effectiveness and use of governance frameworks and best practices that lead to security-aware and compliant application development processes.

Finally, the need to carry out further research on the socio- economic effects of these platforms. Though they enable access to technology and innovation, the broader impact of those changes in job roles, workforce impacts and the digital divide are less clearer. Unanswered questions include: how such platforms will affect the market for professional developers, change priorities around education, and stifle or enable innovation among humans with fewer resources. Filling these gaps will contribute to our understanding of low-code/no-code platforms and enable their responsible and effective adoption.

B. Addressing the Gap

A theoretical and practical blend can answer the research voids in low-code/no-code platforms. When it comes to scalability and sustainability challenges, researchers could perform longitudinal studies and offer empirical real-life case studies to assess the performance of applications built over time using these platforms. Using comparative benchmarking to spot the strengths, limits, and optimization opportunities of traditional development methods compared to low-code/no- code solutions as it pertains to enterprise-grade workload handling.

Creating robust frameworks and tools is foundational for security and compliance issues. Governance models that keep the development practice secure and regulatory compliant can be designed in collaboration between researchers and experts from the industry. At the same time, we need research into the most common and critical vulnerabilities inherent to low- code/no-code applications, mitigations (e.g., through better user training and automated security checks built-into platforms themselves).

In the socio-economic sphere, interdisciplinary research could address the implications of these platforms for labor markets, skill sets and workforce composition. That may also involve examining how they impact the roles of traditional developers and citizen developers, and how much the technology is used, and by whom, to support greater access and inclusion for underrepresented populations. Further, integrating feedback from educators could also help us in designing the training programmes and curriculums that can prepare every individual to use these platforms efficiently.

This technology can achieve its potential while avoiding risk and harm but not without addressing the gaps through co- creation between academia, industry, and policymakers. Such an approach will guarantee their responsible and sustainable usage, worldwide by countless organizations.

III. EVOLUTION OF LOW-CODE / NO – CODE PLATFORMS

Low-code and no-code platforms date back to the conceptual origins of software development at a time when the requirement for faster and more rapid application development paved the way for such platforms. Introducing visual development tools such as Microsoft Access and Visual Basic during the 1980s and 1990s

prompted a fundamental change in the way software was developed, moving away from heavy code commentary into the graphical domain. Developers had been building simple applications with little to no code using these tools, and they paved the way for more advanced platforms.

With the growing trend of cloud computing and Software as a Service (SaaS) in the 2000s, the evolution of low-code/no- code platforms were given another push. Salesforce, and Zoho are just two examples of companies that started introducing low-code capabilities that allow users to personalize workflows and extend functionality within their ecosystems and without having to develop the entire workflow. Then, came with drag-and-drop tooling and prebuilt components which made it easier for non-engineers to build applications.

These platforms emerged throughout the 2010s as the combination of technological advances, including AI, APIs, and mobile-first became prevalent. New vendors like Mendix, OutSystems, and Bubble came up with these capable, scalable solutions for building anything from simple to complex apps. As businesses looked for speedier time-to-market, more agility and lesser dependency on IT teams, low-code and no-code platforms found increasing traction in various sectors. These platforms are now crucial in digital transformation initiatives, allowing businesses to change their fortunes, innovate and transform in a fast-paced environment. In the future, AI and machine learning integration will continue to lead the charge, enabling more automated, personalized, and intelligent applications that adapt to individual needs and preferences.

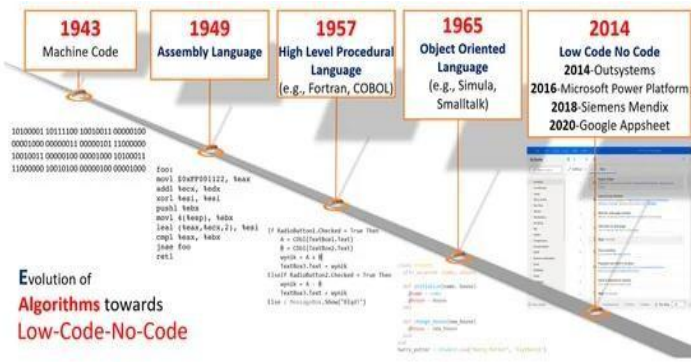

Fig. 1. Evolution of algorithms from machine code to low- code-no-code.

LCNC platforms have emerged as a compelling solution to address these challenges, providing visual interfaces and pre- built components that reduce the reliance on manual coding. By enabling users to build applications through drag-and-drop functionality and modular components, LCNC platforms lower the barrier to entry for software development and make it accessible to a wider audience [6]. Low-code-no-code platforms allow the business leaders and business analysts with in-depth domain knowledge to transform their business vision into the IT landscape. Indeed, LCNC allows the business to realize their vision without relying on IT experts. Hence, LCNC delivers business–IT alignment [7]. By providing accessible tools for non-coders, LCNC platforms empower employees across departments to create solutions that address specific needs without extensive technical intervention [8].

IV. IMPACT OF SOFTWARE DEVELOPMENT LIFECYCLE

A whole new dimension of efficiencies as well as workflows has been introduced to the Software Development Life Cycle (SDLC) by the onset of low-code and no-code platforms. Such platforms simplify the stages of the traditional SDLC helping organizations to bring down the time it takes to develop while also enabling seamless collaboration between the technical and non-technical stakeholders of the organization. But they also bring new requirements to planning, executing and maintaining projects that use them.

A. Requirement Gathering and Planning

Low-code/no-code platforms ease the part of gathering requirements in the development cycle as it allows non-technical users to directly be a part of the development process. Business users prototype applications on their own, which cuts down on the time spent translating requirements to developers. By working so closely together, they promote understanding of the project goals and they get a good idea of what the final product should look like to correspond to the user expectations.

B. Development

The development process is where most change will happen. Visual interfaces, drag-and-drop functionality, and prebuilt components speed up and simplify the process of application creation. Further, LCNC help to automate tasks that are repetitive such as UI design and data integration, to enable teams to focus on customization and innovation.

C. Testing

Automated testing tools, never one of the strong suits of most development environments, are built-in to these low- code/no-code platforms and speed testing considerably. Such platforms usually come with preconfigured testing environments and debug options out of box so you can identify and resolve the issue faster. But for more complex applications, testing for edge cases, or verifying performance under scale will still require some manual effort, or the use of external tools.

D. Deployment

It allows you to deploy with a single click, which is something almost every platform offers nowadays. They simplify the environmental and infrastructure management. This phase is further simplified by vendor-based cloud hosting, from which organizations can also roll out applications in no-time, and scale them effectively.

E. Maintenance and Updates

By using centralized dashboards and even enabling real- time editing, low-code/no-code platforms are designed to simplify updating and maintenance. The platform may ultimately lead to easier development but long-term maintainability can be difficult especially if the platform makes opt-in backwards compatibility burdensome or if the project outgrows the platform. Other factors to consider during this stage are vendor lock-in risks and dependency on proprietary technology.

Low-code-no-code platforms allow the business leaders and business analysts with in-depth domain knowledge to transform their business vision into the IT landscape. Indeed, LCNC allows the business to realize their vision without relying on IT experts. Hence, LCNC delivers business–IT alignment [5]. Low-code development platforms (LCDPs) are easy to use visual environments that are being increasingly introduced and promoted by major IT players to permit citizen developers to build their software systems even if they lack a programming background [9].

Current LCNC platforms, such as Outsystems, Mendix, Microsoft Power Platform, and others, provide interactive data visualization capabilities to non-programmer researchers. Hence, studies could easily portray the research findings. [10- 22]

V. FUTURE SCOPE

Low-code and no-code platforms are not only here to stay but are set to grow exponentially as they are integrated into digital transformation across every industry. Excelling in the most promising, emerging capabilities, such as the embedding of artificial intelligence (AI) and machine learning (ML) functionality in these platforms.

Enhanced AI-driven features, including automatic code generation, predictive analytics, and intelligent suggestions, will make it even easier to build applications and guide better choices. The evolution of platforms is anticipated to drive a broader set of use cases even more sophisticated ones, such as AI-driven chatbots, IoT applications, and advanced workflow automation that can be pushed into the hands of citizens developers and professional teams alike.

Further growth path is related to enterprise-grade features expansion. Vendors focusing on the scalability, security, and compliance needs as more businesses adopt low-code/no-code platforms for mission-critical applications. These consist of strong governance capabilities, multi-cloud/hybrid support, and integrations with ERP and CRM systems.

Finally, as interoperability progress further, they will be integrated into existing IT ecosystems so organizations can use these platforms without having to replace any already established infrastructure.

Low-code/no-code platforms are also having an important social impact by democratizing technology. These platforms are bridging the digital divide by enabling non-technical users to develop applications and empowering innovation at the grassroots. Their full potential is expected to be in education, empowering students and educators to create tech solutions without advanced programming proficiency. Additionally, the global increase in digital transformation demand will enable startups, small businesses, and individuals to build scalable and competitive solutions, promoting economic development and innovation around the world.

VI. CONCLUSION

Low-code and no-code platforms are turning the software development landscape upside down by making application creation accessible to diverse teams and accelerating innovation across the enterprise. These platforms are bridging business and technical execution, by making complex development processes simple and giving power to non- technical users. They have demonstrated critical importance in minimizing development timeframes, enabling collaboration, and promoting digital transformation in various sectors. However, issues relating to scalability, security, and vendor lock-in development are still important topics needing solutions to ensure the continued, safe adoption of these technologies.

Forward-looking, the convergence of upcoming technologies like AI, ML, and IoT with these platforms will create new space for innovation, thus making them vital to the future of tech-driven businesses. The technology landscape will continue to see their expanded use in education, small business, and large-scale enterprise solutions no doubt changing industries and societies. Largely driven by research, sound governance, and efforts to scale and democratize the platforms, low-code and no-code platforms are poised to fundamentally reshape the way that technology is created and consumed for the foreseeable future.

REFERENCES

[1] Computer History Museum, Birth of Computer—ENIAC, CHM. [Online]. Available online:https://www.computerhistory.org/revolution/birth-of-the- computer/4/78 (accessed on 12 November 2022).

[2] HP, Computer History: A Timeline of Computer Programming Languages. 15 October 2018. [Online]. Available online: https://www.hp.com/us- en/shop/tech-takes/computer-history-programming-languages (accessed on 12 November 2022).

[3] Rizwan, O. A Snapshot of Programming Language History, Increment. April 2018. [Online]. Available online:https://increment.com/programming- languages/language-history/ (accessed on 12 November 2022).

[4] Outsystems. It Began with a Vision. 2001. [Online]. Available online: https://www.outsystems.com/evaluation-guide/it-began-with-a-vision/ (accessed on 9 November 2022).

[5] Sufi, F. (2023). "Algorithms in low-code-no-code for research applications: a practical review. Algorithms", 16(2), 108.

[6] Gartner Research, "Low-Code Development Technologies Evaluation and Trends," Gartner, Inc., 2021. [Online]. Available: https://www.gartner.com

[7] McKendrick, J. Low-Code No-Code Market Keeps Growing, and that Means Sifts in Technology Roles. ZDNET. 3 November 2022.

[8] IDC Research, "Low-Code and No-Code Platforms in the Age of Digital Transformation," IDC Report, 2020. [Online].

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

[10] Sahinaslan, E.; Sahinaslan, O.; Sabancıoglu, M. Low-code application platform in meeting increasing software demands quickly: SetXRM. In Fourth International Conference of Mathematical Sciences (ICMS 2020), AIP Conference Proceedings; AIP Publishing LLC.: Istanbul, Turkey, 2021; Volume 2334.

[11] Sufi, F.K. Automatic identification and explanation of root causes on COVID-19 index anomalies. MethodsX 2023, 10, 101960.

[12] Sanchis, R.; García-Perales, Ó.; Fraile, F.; Poler, R. Low-Code as Enabler of Digital Transformation in Manufacturing Industry. Appl. Sci. 2020, 10, 12.

[13] Waszkowsk, R. Low-code platform for automating business processes in manufacturing. IFAC-PapersOnLine 2019, 52, 376–381.

[14] Sufi, F.; Khalil, I. Automated Disaster Monitoring from Social Media Posts using AI based Location Intelligence and Sentiment Analysis. IEEE Trans. Comput. Soc. Syst. 2022, 1–11, in press.

[15] Sufi, F. A decision support system for extracting artificial intelligence- driven insights from live twitter feeds on natural disasters. Decis. Anal. J. 2022, 5, 100130.

[16] Sufi, F.K.; Alsulami, M. Automated Multidimensional Analysis of Global Events With Entity Detection, Sentiment Analysis and Anomaly Detection. IEEE Access 2021, 9, 152449–152460

[17] Sufi, F.K. Identifying the drivers of negative news with sentiment, entity and regression analysis. Int. J. Inf. Manag. Data Insights 2022, 2, 100074.

[18] Sufi, F.K.; Alsulami, M.; Gutub, A. Automating Global Threat-Maps Generation via Advancements of News Sensors and AI. Arab. J. Sci. Eng. 2022, 48, 2455–2472. [Google]

[19] Sufi, F.; Razzak, I.; Khalil, I. Tracking Anti-Vax Social Movement Using AI based Social Media Monitoring. IEEE Trans. Technol. Soc. 2022, 3, 290– 299.

[20] Sufi, F.; Alsulami, M. AI-based Automated Extraction of Location- Oriented COVID-19 Sentiments. Comput. Mater. Contin. (CMC) 2022, 72, 3631–3649.

[21] Alsulami, M. Knowledge Discovery of Global Landslides Using Automated Machine Learning Algorithms. IEEE Access 2021, 9, 131400–131419.

[22] Sufi, F.; Alam, E.; Alsulami, M. A new interactive system for analyzing historical records of tornedoes in Bangladesh. Sustainability 2022, 14, 6303.

[23] Chhor, J.; Fischer, V.; Kröppel, F.; Schmitt, R.H. Rule-based Decision Support for No-Code Digitalized Processes. Procedia CIRP 2022, 107, 258– 263.

[24] Ramdurai, B., & Adhithya, P. (2023). The impact, advancements and applications of generative AI. International Journal of Computer Science and Engineering, 10(6), 1-8.

[25] Brandon, C., & Margaria, T. (2023). Low-Code/No-Code Artificial Intelligence Platforms for the Health Informatics Domain. Electronic Communications of the EASST, 82.