International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 2, Mar-Apr-2025, ISSN (Online): 2395-566X

Evaluating the Impact of Low-Code and No-Code Platforms on Modern Software Engineering Practices

Nandini Manvar, Gayatri Zagade, Shreyash Trimale
Aissms Institute of Information Technology, Pune

Abstract- Low-code and no-code (LCNC) development platforms are altering software architecture. These systems let users design applications with either very little or no coding. They employ ready-made components, drag-and-drop tools, and visual interfaces. This facilitates fast and simple application development. LCNC tools are usable by non-developers as well as developers. This cuts the demand for big development teams, boosts productivity, and saves time. LCNC systems are also helping to solve the worldwide shortfall of qualified software programmers. These days, several sectors make use of these platforms. They are applied in healthcare to create systems of patient tracking. In the classroom, they advocate e-learning environments. In finance, they enable automated procedures and reports. These instruments have certain restrictions even if they are quite helpful in many spheres. Apps needing to expand or manage more users raise issues regarding scalability. Another problem is security, particularly in apps handling private or delicate information. Customizing can be challenging since consumers might not have complete influence over the backend or code of the app. Notwithstanding these problems, LCNC platforms keep getting better. The situation of LCNC tools now is investigated in this work. It looks over their benefits and drawbacks. It also looks at currently used popular platforms and their practical applications. The paper addresses how conventional coding might be used alongside LCNC tools. One can get speed and adaptability from a hybrid development method. LCNC tools for basic parts and conventional code for challenging projects let developers handle both. Better performance and simpler maintenance are thus possible. Recommendations for applying LCNC tools in several environments round out the paper. These cover companies, software teams, and the educational fields. LCNC systems might become increasingly important component of contemporary software engineering as technology develops. They provide a fresh approach with less technical obstacles to create apps quicker and more effectively.

Index Terms- Low-Code Development; No-Code Platforms; Citizen Developers; Visual Programming; Rapid Application Development (RAD); Software Engineering Automation; Agile Development; Digital Transformation; Software Prototyping; Platform-as- a- Service (PaaS).

I. INTRODUCTION

Across the board, the acceleration of digital transformation pushed organizations to find ways to build software in a manner that was faster, cheaper, and at scale. Traditional software development brings great power and adaptability but often demands high programming prowess, resource-intensive teams and long delivery timelines to extract value. This rising set of challenges paved way for low code and no code (LCNC) platforms that deliver a visual component-based development environment. These platforms provide users with the ability to build applications by dragging and dropping UI components, connecting data sources, and defining logic flows with little to no code. Low-code platforms (like Mendix, OutSystems and Microsoft Power Apps) are mostly made for professional developers to help them speed up their coding process. These platforms help to minimize repetitive coding tasks and provide predefined templates, APIs and automation capabilities.

In contrast, no-code platforms such as Bubble, Appgyver, and Airtable are oriented towards users without technical expertise. These users, known as “citizen developers,” can use simple, user-friendly interfaces to create meaningful applications. This is the reason that it enables the people who are lacking in programming knowledge to the great extent to contribute the development of application and digital innovation in their organizations.

This democratization of software is overriding traditional roles and processes. LCNC platforms enable non-technical users such as business experts to build applications without having to rely heavily on IT departments, providing lots of flexibility and rapid iterations. But, while LCNC tools can improve productivity, they may create issues around application scaling, performance tuning, security compliance and long-term maintainability. But as apps built with LCNC platforms become more sophisticated and widespread, maintenance, code quality, and enterprise system integration become issues.

In this paper, we analyse LCNC platforms and how they fit into the modern software engineering process. It discusses the strengths and weaknesses of these tools and looks at their adoption in different industries We explore how various user groups, from professional developers to citizen developers, are adopting LCNC solutions, At last, we provide our view and recommendations on how to enable LCNC platforms in such hybrid development environments to get the best of both worlds — bringing together traditional coding and visual development to produce trustworthy, scalable and secure software solutions.

II. LITERATURE REVIEW

Low-code and no-code (LCNC) platforms are really gaining traction these days. They make it super easy to develop applications without requiring extensive programming knowledge. In fact, a report from [1] Gartner in 2023 predicts that by 2026, over 65% of application development will be done on LCNC platforms. Forrester Research [2] breaks down LCNC tools based on how complex they are, the user experience they offer, and how easily they can be extended. It's really important to acknowledge the rising impact of citizen developers—those regular folks who are jumping into app creation—on the world of enterprise software development. The concept of making development accessible to everyone fits perfectly with the larger trends we're seeing in digital transformation. This approach allows businesses to reduce their dependence on the small pool of available developers. As highlighted by Harvard Business Review, these platforms are bridging an important [3] gap between IT and business operations.

On the technical side, researchers have taken a close look at what LCNC platforms can do and where they might fall short. Mohan and Sharma [4] really dug into Mendix, out Systems, and Microsoft Power Apps, pointing out some shared features like drag-and-drop functionality, template libraries, and cloud-native deployment. They discovered that low-code tools provide greater flexibility compared to no-code platforms, which are primarily designed for non-developers to use with ease However, both kinds of platforms encounter hurdles when it comes to scaling for large enterprise projects that require intricate logic, API integration, or real-time data management. Other technical studies [5] emphasize the need for better hybrid models that combine visual development with scripting or advanced configuration layers, making them suitable for both newcomers and experienced users.

The rise of low-code/no-code (LCNC) platforms in DevOps workflows has become a hot topic for researchers. Wang et al. [6] took a closer look at the hurdles and possibilities that come with weaving LCNC solutions into continuous integration and continuous deployment (CI/CD) pipelines. They pointed out that while tools like Out Systems offer some DevOps functionalities, many LCNC platforms still fall short when it comes to providing solid support for automated testing, version control, and containerization. This gap can be a real headache for companies trying to maintain software quality and consistency as they scale up. Additionally, researchers[7] have raised concerns about the increasing prevalence of "shadow IT," where applications are developed without adequate oversight from IT. This can lead to compliance and security issues. That's why it's crucial to have strong governance frameworks in place to encourage sustainable adoption.

Case studies have revealed both the benefits and challenges of low-code/no-code (LCNC) implementations in real-world scenarios. For example, Patel et al. [8] pointed out a financial institution that managed to reduce application delivery time by over 60% using Power Apps. However, they also found that a lack of understanding of the platform's capabilities resulted in some performance hiccups. Capgemini [9] examined large-scale deployments and emphasized the need for cross-functional training to close the knowledge gap between IT and business teams. In sectors like healthcare, education, and logistics, LCNC tools have been employed to develop dashboards, mobile apps, and workflow automation tools, which not only cut development costs but also accelerate time-to-market. Nevertheless, concerns about customization limits and data security remain significant obstacles for long-term scalability.

Case studies have revealed both the benefits and challenges of low-code/no-code (LCNC) implementations in real-world scenarios. For example, Patel et al. [8] pointed out a financial institution that managed to reduce application delivery time by over 60% using Power Apps. However, they also found that a lack of understanding of the platform's capabilities resulted in some performance hiccups. Capgemini [9] examined large-scale deployments and emphasized the need for cross-functional training to close the knowledge gap between IT and business teams. In sectors like healthcare, education, and logistics, LCNC tools have been employed to develop dashboards, mobile apps, and workflow automation tools, which not only cut development costs but also accelerate time-to-market. Nevertheless, concerns about customization limits and data security remain significant obstacles for long-term scalability.

III. METHODOLOGY

The mixed methods strategy used in this paper includes qualitative study of references, case studies, and interview data together with quantitative data acquired from organized surveys and performance benchmarks on various channels. This broad approach seeks to give a comprehensive assessment of how Low Code and No Code (LCNC) platforms influence software engineering processes. It focuses on four primary aspects: developer efficiency, software quality, acceptance trends among both nontechnical and technical users, and long-term maintainability.

With a thorough literature review exploring databases including IEEE Xplore, ACM Digital Library, SpringerLink, and Google Scholar, we started the first phase of our research. For common themes, approaches, restrictions, and the most uptothemoment technology trends, we narrowed our focus towards studies published between 2018 and 2024. Starting with a first batch of 250 documents, we used standards such as citation frequency, relevance to the software development lifecycle (SDLC), and precise use cases to reduce it to 87 peer reviewed articles and industry whitepapers. Our top search terms were "low code," "no code," "citizen development," "software engineering automation," and "enterprise adoption of LCNC."

We conducted a secondphase survey covering 20 people from various fields within companies including software engineers, project managers, and citizen developers. All meant to capture consumers' views of lowcode/nocode (LCNC) tools, this survey used 15 questions in Likertscale and multiplechoice forms. Closer examination of platforms like Microsoft Power Platform, Mendix, OutSystems, and Appgyver entailed. User satisfaction, perceived risks including security and maintainability, timetodeploy, and the learning curve were our major priorities. Using statistical methods including correlation and frequency distribution, we analysed the quantitative answers so to find patterns between team productivity and platform usage..

The third stage aimed on monitoring live application construction in a restricted environment besides the surveys with lowcode/nocode (LCNC) solutions. In all, technical and nontechnical users together developed five prototype apps to investigate development patterns and usability. Every project was centered on a particular use case: scheduling appointments, gathering internal feedback, monitoring school attendance, tracking ecommerce inventory, and managing employee leave. Each of these cases was applied on at least two LCNC systems. We monitored the time used, the reusability of components, and the error rates. These observations provided useful information on how logic flow editors, Drag and drop interfaces, and template libraries affect the development process.
We carried out semi structured interviews with ten experts, including programmers who switched from regular coding to LCNC settings, to enhance our observational results. Participants mentioned only that LCNC tools can really speed up development; they have some disadvantages regarding precise control of application behaviour. Their observations provided a qualitative view on how LCNC platforms influence software design principles including modularity, abstraction, and code maintenance. Individuals in the finance and healthcare sectors, where compliance and custom checks are absolutely crucial, particularly highlighted this issue.

By testing APIs and database connections, we closely examined in the fifth phase the integration features of LCNC systems. Within Mendix and Power Apps, we developed and employed RESTful API endpoints to assess how well they integrate. Backend data sources were Firebase, PostgreSQL, and several third-party SaaS solutions. We rated and contrasted performance indicators including latency, API errors, and data synchronization time. Under straightforward conditions, these experiments revealed that although lowcode tools effectively handle API orchestration, they are usually underperforming in nested data management and big payloads. When more than three nested API calls were made, we observed a drastic decrease in performance.
Finally, we developed a comparative analysis framework to assess LCNC systems on four major dimensions: extensibility, governance, scales, and usability. From the qualitative and quantitative data, we gathered in our study, we gave each category a weighted score. The last framework serves as a benchmarking tool by which companies can pick the perfect LCNC platform suited to the maturity level of their developers and the particular requirements.

Benefits of Low-Code and No-Code Platforms

The arrival of Low Code and No Code (LCNC) platforms has changed distribution, development, and software design approaches. These systems enable users to generate nearly codeless software. For users of any technical level, they allow for quick and effective performance. This covers major benefits of LCNC equipment

Lifespan of accelerated development

LCNC platforms enable apps to be developed far more rapidly than with standard approaches. Developers can leverage visual aids including drag and drop builders. Additionally available are premade templates and reusable elements. These properties lessen the need of manual code. Consequently, development cycles reduce from months to only weeks. Industries such as logistics, finance, and healthcare depend on this speed. These industries are under continuous pressure to rapidly meet changing demands.

Democratization of Application Development
One of the biggest advantages of LCNC tools is their user-friendliness. Now, even those without any coding experience can create apps. These folks are commonly known as "citizen developers." They could be business analysts, HR managers, or even team leaders. Since they have a deeper understanding of their business needs than the IT teams, the tools they create tend to be much more relevant. This not only lightens the load for IT departments but also speeds up problem-solving.

Economical use of funds
In several ways, lowcode/nocode (LCNC) platforms reduce expenses. To begin with, businesses do not need depend on huge development teams for every small update or tool. Moreover, they keep significant time, thereby reducing project costs in the end. While citizen programmers handle the easier apps, experienced developers can give their attention to more demanding projects. This intelligent distribution of funds not only saves cash but also raises general output.

Better teamwork between groups helped
These services are especially good for working together. In the same space, they enable both nondevelopers and developers to team on projects. Every person has the opportunity to examine and try the software as it is being built. This helps to guarantee that the last application really meets the requirements of the company and improves departmental communication. Close team working results in few errors and little need for rework.

Compliance and Security Features made automatically
Common with nearly all LCNC tools are fundamental security and regulatory components. User access control, encryption, audit logs, and data policies might be included. Some forums enable enterprises to follow standards like SOC2, HIPAA, and GDPR. These preinstalled capabilities simplify the development of compliant and safe applications. Even people with little knowledge of security practices can create secure applications.

Limitations and Challenges
Although they present difficulties, LCNC systems bring many advantages. Depending their limits, using too much of them without knowing can result in dangerous systems or bad performance. This part outlines main questions.

Limitations on Scalability
Many LCNC systems are not designed for very complex or very great projects. Performance might suffer when applications have to manage enormous databases, many users, or sophisticated logic. Some services have restrictions on how far backend systems can be customized. Their limitations for company level applications might be highlighted as a result of this.

5.2 Dependency from vendor
These systems are frequently owned. This implies applications developed on one platform could not operate on one more. Changing systems may be expensive and time costly. A company depends on the update frequency, functions, and pricing of the platform. This inflexibility could impede innovation or future expansion.

Restricted personalization
Although LCNC tools provide readymade components, they are restricted. Developers might not be willing to include elaborate workflows or custom algorithms. The platform might not enable the app if it requires particular integrations or logic. This restricts the program's abilities and could demand additional instruments for the gaps to be filled.

Quality Control and Testing
Typical development consists of numerous testing techniques, including automated deployment and unit testing. LCNC systems could not fully support these. Their source code is frequently hidden, which complicates comprehensive testing. When it comes to crucial programs, there's a real chance of encountering system failures or bugs.

Security and governance risks can be running for office.
Sometimes, these risks stem from something as straightforward as easy access and a user-friendly interface. Non-experts might unknowingly create apps that expose sensitive information, which could lead to violations of laws or corporate policies if proper regulations and controls aren't in place. To safeguard against these dangers, companies need to carefully manage their low-code and no-code initiatives.

Applications in Industry
Thanks to their versatility and ease of use, low-code and no-code platforms have found their way into a variety of industries. Here are some standout applications in specific sectors that really showcase their potential.

Health care
Healthcare providers utilize LCNC platforms to quickly create internal communication dashboards, appointment scheduling tools, and patient intake forms. For example, many hospitals used nocode technology during the COVID19 epidemic to create realtime tracking systems for case management and vaccine distribution in days, therefore showing LCNC's potency under time pressure.

Financial and Banking sector
Using lowcode technology, financial institutions create programs that run compliance checks, monitor financial transactions, and simplify loan approvals. In this context, LCNC platforms with integrated compliance monitoring are particularly useful given legislative constraints. Additionally,

some banks allow corporate departments to design bespoke reports and dashboards without the need of IT support.

Consumer and ecommerce ran errands
Retailers use nocode systems to handle inventory, monitor consumer comments, and organize marketing initiatives. These systems as well help users create mobile friendly web applications for product searching and customer service chatbots. Using LCNC tools, marketing departments can fast A/B test product lists or user interfaces, therefore increasing consumer interest.

Training
In academic contexts, LCNC technologies help create internal portals, handle student information, automate grading, and simplify learning management systems. Teachers with little technical knowledge may customize learning materials designed for particular student groups, so promoting inclusive and creative learning settings.

Logistical science and supply lines
Logistically, businesses utilize LCNC tools for order management systems, delivery status alerts, and fleet trackingashboard construction. Rapid changes and revisions made possible by these systems help response times and limit postponement in unstable supply chains.

The Future of Development Without Code or Little Code
The path of LCNC systems indicates a major development in software design rather than just a fleeting fad. Key developments and future forecasts follow:

Incorporated with Machine Learning and AI
The next generation of LCNC systems will use artificial intelligence to support predictive analysis, coding suggestions, and application design. Including AutoML models enables even nontechnical users to install machine learning models in their workflows, therefore extending the limits of what citizen developers can accomplish.

Hybrid development platforms
By enabling smooth changes between visual development and manually coded logic, future LCNC systems will provide more extensibility. Within LCNC systems, programmers can create custom functions in Python, JavaScript, or other languages to address present restrictions of customization.

Platformagnostic development
Rising anxiety about vendor locking is driving the acceptance of opensource and flexible LCNC applications. Projects are ongoing to develop platforms that will allow exporting of software to normal codebases or support plugins for platform independence.

Frameworks of governance and conformity
More companies will adopt centralized management systems to mitigate the dangers of uncontrolled citizen development. These might feature predefined templates, authorization processes, and usage tracking so that users still have freedom but quality, safety, and compliance are enforced.

Deeper Cultural Change in Software Engineering
Should LCNC platforms grow up, software engineering as a field will change direction to include jobs like "platform architect" and "solution orchestrator." These roles will stress modular thought, design principles, and integration strategy rather than code only—transforming developerhood.

Comparison of Well-Known Lowcode And Nocode Platforms
This section offers a sidebyside comparison of top tools in both lowcode and nocode categories so you can better grasp the abilities and distinctions among various LCNC platforms. The contrast covers extensibility, integrations, learning curve, pricing models, target users, and scalability.

Major Observations
Mendix and OutSystems are ideal for sophisticated business applications at considerable scale since they provide more extensibility and continuous integration/delivery pipelines.

- Microsoft dependent companies seeking fast digitization of workflows will benefit from Power Apps.
- For entrepreneurs or creators looking to start MVPs with few resources, Bubble and Appgyver provide solutions.
- Airtable is an excellent tool in internal solutions for team collaboration, task automation, and data management.

Factors to Think about for Choosing a Platform
Before organizations accept an LCNC platform, they should consider the complexity of their use cases, longterm scalability, needed integrations, and team skillsets. Combining many solutions (e.g., Airtable for data + Bubble for UI) often improves efficiency and lowers costs.

Corporate Adoption Recommendations
Enterprise use of LCNC platforms calls for a well-balanced strategy that exploits their benefits while reducing risks. Strategic advice for guaranteeing a good governance and efficient execution is given below.

Run pilot initiatives.
Begin with pilot initiatives having little risk but tangible results. Before rolling out across departments, this lets teams investigate LCNC features, establish simplicity of use, and approx ROI.

Definition of governance structures
Set rules of governance that specify under what conditions, under what supervision, and by whom something may be

developed in order to avoid sprawl and safety problems. Include version control policies, security tests, and, if relevant, code review criteria.

Supporting training and resources invest.
Though these apps make development easier, knowing their limitations, structure, and processes is vital. Max productivity and few mistakes will be achieved by providing classes or certifications for technical as well as corporate teams.

Sync LCNC with DevOps and Agile Practices
Where feasible, connect LCNC technologies with current DevOps processes. With version control, test automation, and continuous integration/continuous distribution, some sophisticated systems let them fit into a contemporary agile setting.

Evaluate Performance by means of KPIs
Time to deploy, active user count, application uptime, cost savings, and user satisfaction are among the metrics to be tracked. Refine how the platform is used, distribute resources, and support more widespread releases by means of these observations.

IV. CONCLUSION

Rapid app creation, democratization of access to development tools, and crossfunctional cooperation are redefining the software development scene using lowcode and nocode platforms. By enabling nontechnical people to actively participate in software solutions, these platforms help to lower reliance on conventional IT teams and speed up innovation in all industries.

Still, practical concerns have to balance the promise of LCNC. If not tackled early, scalability issues, vendor locking, and restricted customization may very much impede your projects. Moreover, without good governance, companies run the risk of unsupervised citizen development introducing security and compliance weaknesses.

The future of LCNC is bright; artificial intelligence is increasingly integrated, platform compatibility is improving, and corporate capabilities are becoming more sophisticated. These technologies are ready to become a foundation of contemporary software development methods as they advance. For businesses, strategic adoption, training, and governance are essential to leverage LCNC—guaranteeing that both nontechnical and technical users can create practical and sustainable digital solutions.

Acknowledgement
It is indeed our desire to deeply appreciate our highly regarded mentor, Prof. Chetan Aher, who has been a constant source of guidance, encouragement, and insightful feedback for this research. His expertise and support have played a significant role in giving our work its shape and quality.

The extent of our gratitude is no less to the AISSMS Institute of Information Technology for the facilities they provided along with a supportive academic environment, which made our project successful.

And final, but not of any less importance, our heartiest thanks go to our project team—Nandini, Gayatri, and Shreyash—for their combined efforts, collaboration, and their dedication to the task, that all have made this research a reality.

REFERENCES

1. Gartner, "Magic Quadrant for Enterprise Low-Code Application Platforms," 2023.
2. Forrester Research, "The Forrester Wave: Low-Code Development Platforms," Q1 2022.
3. Harvard Business Review, “How Non-Tech Employees Are Building Apps Without Coding,” 2022.
4. R. Mohan and A. Sharma, “Comparative Study of Low-Code Platforms for Enterprise Applications,” Int. J. Soft. Eng. Res., vol. 18, no. 3, pp. 245–258, 2021.
5. M. Yang and L. Hu, “Scalability Challenges in Low-Code Platforms,” IEEE Access, vol. 9, pp. 77865–77878, 2021.
6. D. Wang et al., “DevOps Adoption in Low-Code Software Development,” Proc. of the 16th Int. Conf. on DevOps, 2022.
7. McKinsey & Co., “Avoiding the Pitfalls of Shadow IT in LCNC,” 2023.
8. T. Patel, K. Kumar, “Business Process Automation Using Power Apps,” J. Digit. Transform., vol. 12, no. 2, pp. 135–147, 2020.
9. Capgemini, “Governance Strategies for Low-Code/No-Code Platforms,” Whitepaper, 2023.
10. IBM Research, “AI in Low-Code and No-Code Development: The Next Frontier,” 2023.