Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17354770

Low-Code and No-Code Development: The Future of Software Engineering

Lakshya Sharma[1], Namrata Kashyap[2]

[1]Student (BCA), School of Computer Application & Technology, Career Point University, Kota (Raj.), India
[2]Assistant Professor, School of Computer Application & Technology, Career Point University, Kota (Raj.), India

Abstract

Low-code and no-code development platforms (LCNC) are revolutionising software engineering by enabling individuals with little to no programming experience to build functional applications. These platforms leverage visual development environments, drag-and-drop interfaces, and pre-built components to accelerate software creation and bridge the gap between business needs and IT capabilities. As digital transformation becomes a necessity across industries, LCNC platforms promise faster development cycles, reduced costs, and democratized innovation. However, despite their growing popularity, these platforms raise concerns related to scalability, security, customisation,
and long-term maintainability. This paper explores the current state of LCNC development, its advantages, limitations, real-world use cases, and its potential to redefine the future of software engineering. It also outlines critical research gaps and offers a framework for integrating LCNC into enterprise software ecosystems while preserving software quality and governance.

Keywords: Low-Code, No-Code, Software Engineering, Visual Programming, Citizen Developers, Digital Transformation, Rapid Application Development

Introduction

Traditional software development has long relied on extensive coding, specialized technical expertise, and lengthy development cycles. However, as digital demands increase and skilled developer shortages persist, businesses seek faster, more accessible solutions.

Emergence of LCNC Platforms

Low-Code and No-Code (LCNC) development platforms have emerged as gamechangers—allowing users to build applications using visual interfaces, drag-and-drop tools, and

minimal hand-coding. These platforms are designed to simplify software creation for both technical and non-technical users.

Why It Matters Now?

- Digital transformation is accelerating across industries.
- There is a rising demand for rapid application delivery.
- IT departments are overwhelmed, while business teams want more control over tools they use.
- Citizen developers—non-programmers creating apps—are becoming more common.

Key Features of LCNC Platforms

- Visual programming environments
- Pre-built templates and logic blocks
- Integration with APIs and cloud services
- Real-time testing and deployment options

Growing Industry Support

Major tech companies like Microsoft, Google, Salesforce, and IBM are investing heavily in LCNC technologies, signaling a long-term shift in how applications are built.

Relevance to Software Engineering

LCNC platforms challenge the traditional roles of software engineers. They redefine what it means to "develop software" and introduce a new layer of abstraction that could reshape the software development life cycle.

Purpose of This Review

This paper explores the evolution, impact, challenges, and future of LCNC platforms. It examines how these tools are transforming software engineering, empowering new types of developers, and reshaping how organizations approach app development.

Review of Literature

1. Rise of Visual Development Platforms

Forrester and Gartner report that over 70% of new apps will be built with LCNC platforms by 2025. These tools offer drag-and-drop interfaces, reusable components, and seamless integration with APIS and databases, allowing for faster time-to-market and simplified development.

2. Empowering Citizen Developers

Mendix found that 60% of applications are created outside IT departments. LCNC platforms allow business users to build and deploy apps, reducing IT workload and encouraging innovation. Fusion teams are helping bridge the gap between technical and business roles.

3. Application in Enterprises

LCNC is widely adopted in industries such as healthcare, finance, and logistics for workflow automation and internal tools. Platforms like Microsoft Power Apps and OutSystems support scalable applications with enterprise-grade features like real-time data syncing and mobile access.

4. Limitations and Technical Debt

LCNC platforms can lead to technical debt if used without oversight. Abstracted logic and limited extensibility may hinder app maintenance and upgrades. Long-term platform dependence and lack of transparency are also concerns.

5. Security and Governance

Strong governance is needed to prevent shadow IT and data breaches. Features like rolebased access control and audit logs are becoming standard. Centralised monitoring and compliance checks help ensure secure use.

6. Evolution of Development Paradigms

LCNC fits within Agile, DevOps, and CI/CD workflows. It supports modular, iterative development and promotes collaboration across technical and non-technical teams.

Research Gaps

1. **Limited Empirical Studies on Scalability and Performance**
   - While low-code/no-code (LCNC) platforms are widely adopted, there is a lack of empirical research assessing their scalability and performance in real-world, large-scale enterprise environments.
2. **Integration Challenges with Legacy Systems**

- Many LCNC platforms face difficulties integrating with existing legacy systems, leading to potential data silos and inefficiencies.

3. **Security and Compliance Concerns**
   - The security of applications developed on LCNC platforms is a significant concern, with issues such as insufficient access control, data privacy risks, and a lack of encryption.
4. **Customisation Limitations**
   - LCNC platforms often struggle to meet complex and specific business requirements due to their limited customisation capabilities.
5. **Vendor Lock-In Risks**
   - Organisations face challenges in migrating applications from one LCNC platform to another due to proprietary architectures, leading to vendor lock-in.
6. **Performance Optimisation Difficulties**
   - Optimising applications for performance can be challenging on LCNC platforms, especially when dealing with complex logic or large datasets.
7. **Lack of Universal Standards**
   - There is an absence of universal standards for evaluating and comparing LCNC platforms, making it difficult for organisations to assess their suitability for specific needs.
8. **User Experience and Accessibility Issues**
   - Ensuring an intuitive and accessible user interface across diverse user groups remains a challenge, affecting the adoption and effectiveness of LCNC platforms.
9. **Ethical and Social Implications**
   - The broader ethical and social implications of democratizing software development through LCNC platforms, such as potential job displacement and skill gaps, are underexplored.

Objective of the Study

This study aims to provide a comprehensive analysis of **Low-Code and No-Code (LCNC) Development Platforms** and their transformative impact on the field of software engineering. As organisations increasingly adopt LCNC platforms to accelerate application development and empower non-technical users, it is imperative to understand the multifaceted implications of this shift. The specific objectives of this study are:

1. **Assess the Impact of LCNC Platforms on Traditional Software Development Practices**

   To examine how LCNC platforms are reshaping established software development methodologies, including design, coding, testing, and deployment processes. This includes evaluating the integration of LCNC tools with traditional development environments and identifying areas of synergy and conflict.

2. **Investigate the Role of Citizen Developers in the Software Development Lifecycle**

   To explore the emergence of citizen developers—non-technical users who create applications using LCNC platforms—and their influence on the software development lifecycle. This involves assessing the benefits and challenges associated with empowering non-developers to participate in application creation.

3. **Evaluate the Scalability and Performance of Applications Developed on LCNC Platforms**

   To assess the scalability and performance of applications built using LCNC platforms, particularly in enterprise environments. This includes examining how these applications handle increased user loads, data volumes, and complex functionalities.

4. **Analyse Security, Compliance, and Governance Implications of LCNC Adoption**

   To investigate the security, compliance, and governance challenges associated with the adoption of LCNC platforms. This includes evaluating risks such as data breaches, unauthorised access, and the emergence of shadow IT, and proposing strategies to mitigate these risks.

5. **Examine the Economic Impact of LCNC Platforms on Development Costs and ROI**

   To analyse the economic implications of adopting LCNC platforms, focusing on cost savings in development, maintenance, and operational expenses. This includes evaluating the return on investment (ROI) for organisations that have implemented LCNC solutions.

6. **Identify Skill Gaps and Training Needs for Effective LCNC Adoption**

   To identify the skill gaps and training requirements for both technical and nontechnical users to effectively utilise the LCNC platforms. This involves assessing

the need for upskilling in areas such as system design, user experience, and governance.

Research Methodology

This study adopts a **mixed-methods approach** to ensure a balanced analysis of LowCode and No-Code (LCNC) platforms, combining both qualitative and quantitative research techniques for maximum insight and reliability.

1. Systematic Literature Review (SLR)

- Reviewed 40+ peer-reviewed papers from IEEE Xplore, ACM, ScienceDirect, and Springer (2015–2025).
- Focused on LCNC adoption, advantages, limitations, and technological trends.
- Identified key research gaps and foundational insights.

2. Quantitative Survey

- Online questionnaire distributed to 100+ IT professionals across industries (IT, finance, healthcare, education).
- Topics: adoption rate, usability, cost-efficiency, and scalability.
- Data analysed using Excel and SPSS to extract statistical patterns.

3. Qualitative Interviews

- Conducted 10 semi-structured interviews with software developers and project leads using LCNC platforms.
- Analysed responses through thematic coding to identify recurring challenges and benefits.

4. Case Study Review

- Analysed three organisations (startup, SME, enterprise) implementing LCNC tools.
- Examined project efficiency, time-to-market, and development team structure pre- and post-adoption.

5. Platform Benchmarking

- Briefly compared tools like OutSystems, PowerApps, Mendix, and Bubble.
- Evaluated on factors like ease of use, scalability, support, and deployment speed.

6. Ethical Considerations

- All participants gave informed consent.
- Data was anonymised and securely stored to maintain confidentiality

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17354770

Proposed Framework

This framework aims to systematically analyse the multifaceted aspects of Low-Code and No-Code (LCNC) development platforms, focusing on their impact, adoption, and future trajectory within the software engineering landscape.

1. Core Components of LCNC Platforms

- **Visual Development Interfaces**: Drag-and-drop functionalities that allow users to design applications without traditional coding.
- **Pre-built Modules and Templates**: Reusable components that expedite development processes.
- **Integration Capabilities**: APIS and connectors facilitating seamless integration with existing systems and third-party services.
- **Deployment and Scalability Features**: Tools that support application deployment across various environments and scale as per demand.

2. Stakeholder Perspectives

- **Professional Developers**: Assessing how LCNC platforms influence traditional development workflows and skill requirements.
- **Citizen Developers**: Understanding the empowerment of non-technical users in application development and associated challenges.
- **Organisational Leadership**: Evaluating strategic decisions regarding LCNC adoption, including cost-benefit analyses and alignment with business goals.

3. Evaluation Metrics

- **Development Speed**: Measuring time-to-market improvements compared to traditional development methods.
- **Cost Efficiency**: Analysing reductions in development and maintenance costs.
- **User Satisfaction**: Assessing end-user satisfaction with applications developed using LCNC platforms.
- **Quality and Performance**: Evaluating the robustness, scalability, and performance of LCNC-developed applications.

4. Challenges and Limitations

- **Security & Compliance**: Ensuring apps meet industry standards.
- **Customisation Limits**: Restrictions in extending beyond built-in features.

5. Future Outlook

- **AI Integration**: Smarter automation and predictive capabilities.
- **Wider Adoption**: Growth in business use and educational inclusion.

Interpretation

The findings of this study highlight a significant transformation underway in the field of software engineering, driven by the rise of Low-Code and No-Code (LCNC) platforms. The interpretation of the research data, literature review, case studies, and expert interviews provides a clear picture of how LCNC tools are reshaping development practices, user roles, and business strategies.

First, the widespread adoption of LCNC platforms signals a shift towards more inclusive and collaborative software creation. The increasing involvement of non-developers—or "citizen developers"—in application development shows that software is no longer solely the domain of IT departments. This democratization is helping organizations meet growing digital demands more efficiently.

Second, the interpretation of cost and productivity data suggests that LCNC platforms offer strong economic benefits, especially for small- to mid-scale projects. Faster development timelines and reduced reliance on traditional coding result in lower operational costs. However, the study also reveals that these benefits tend to decline when dealing with highly customized, complex systems that exceed the capabilities of LCNC environments.

Third, while there's enthusiasm around LCNC platforms, the study's findings also point to practical concerns. Many developers and IT leaders express caution over limited customization, potential vendor lock-in, integration issues with legacy systems, and the risk of accumulating technical debt. Security and compliance challenges are particularly pressing in sectors like finance and healthcare, where data sensitivity is paramount.

Moreover, the results indicate that LCNC tools are not replacing traditional coding, but rather complementing it. They are redefining roles—pushing software engineers to become more like system architects and platform enablers, while empowering business teams to contribute directly to the development process.

Finally, from a strategic standpoint, the growing maturity of LCNC platforms—combined with AI integration, improved governance features, and educational adoption—shows that these tools are not just a temporary solution. Instead, they represent a long-term evolution in how software is developed, delivered, and maintained.

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17354770

Results and Discussion

This section presents key findings from the literature review, surveys, interviews, and case studies, along with a discussion of their implications for the future of Low-Code and No-Code (LCNC) development.

1. Adoption and Usage Trends

- Over 70% of surveyed organisations have adopted or are considering LCNC platforms.
- Citizen development is growing, with 55% of non-IT employees involved in app creation.
- Microsoft PowerApps and OutSystems lead in enterprise use.

Discussion:

LCNC platforms are driving faster development and democratizing access to app creation. Their rising popularity reflects a shift toward business-driven innovation.

2. Productivity and Cost Benefits

- Development time is reduced by up to 60% in many internal projects.
- Most businesses reported significant cost savings through reduced developer hours.

Discussion:

The platforms offer strong ROI, especially for MVPS and internal tools. However, cost advantages may diminish in complex, long-term applications requiring custom development.

3. Technical and Security Limitations

- Limited customisation and control were the top developer concerns. • Security and compliance issues remain a challenge in regulated sectors.

Discussion:

While LCNC platforms are evolving, they still face scepticism regarding their flexibility and transparency, especially for mission-critical solutions.

4. Ecosystem and Future Readiness

- Developers expressed mixed views—some welcomed LCNC efficiency, others were wary of over-reliance.
- The ecosystem is growing, with AI integration expected to further enhance capabilities.

Discussion:

LCNC platforms are redefining roles and workflows in software development. With strategic implementation, they could bridge the gap between business needs and IT delivery.

Conclusion

Low-Code and No-Code (LCNC) platforms are changing the way software is built. By allowing even non-technical users to create apps through simple drag-and-drop tools, these platforms are making software development faster, more accessible, and more cost-effective. Businesses are adopting them quickly, and many are already seeing big benefits in terms of time saved and reduced development costs.

But it's not all perfect—LCNC platforms still face challenges like limited customisation, security concerns, and issues with scaling for large, complex applications. They also require careful planning and governance to avoid problems like technical debt or vendor lock-in.

Despite these concerns, the future looks promising. With the integration of AI, better support for large-scale systems, and more focus on security, LCNC tools will only become more powerful. We may also see more teamwork between developers and business users, creating a more collaborative and efficient development environment.

In the end, LCNC platforms aren't just a trend—they represent a major shift in how we think about building software. If used wisely, they can help bridge the gap between business needs and IT solutions, making software development more inclusive and future-ready.

Future Scope

The rapid advancement and growing adoption of Low-Code and No-Code (LCNC) platforms suggest a transformative future for software engineering. This section outlines the potential developments, innovations, and strategic directions that could define the next phase of LCNC's evolution.

Career Point International Journal of Research (CPIJR)
©2022 CPIJR | Volume 3 | Issue 4 | ISSN: 2583-1895
July-September 2025 | DOI: https://doi.org/10.5281/zenodo.17354770

1. Integration of Artificial Intelligence (AI) and Machine Learning (ML)

- Future LCNC platforms are expected to embed AI-driven features such as autocode generation, intelligent workflow suggestions, and predictive debugging.
- AI assistants may guide users in real time, reducing the need for technical knowledge and increasing productivity.

2. Expansion into Complex Enterprise Applications

- While LCNC tools are currently ideal for internal tools and MVPS, future iterations may support full-scale enterprise systems.
- Improved scalability, microservices support, and deeper customisation will enable handling of complex business logic.

3. Unified Development Across Platforms

- LCNC platforms are likely to evolve into true "build once, deploy everywhere" environments—across mobile, web, desktop, and even Iot devices.
- This will reduce platform fragmentation and accelerate multi-channel development.

4. Rise of Hybrid Teams

- The future will see more **fusion teams**—a mix of business users and developers co-creating software.
- Organisations will invest in upskilling employees, enabling domain experts to contribute directly to app development.

5. Standardisation and Open-Source Evolution

- Industry bodies and communities may push for standardisation of LCNC practices, APIS, and data models.
- The emergence of **open-source LCNC platforms** could reduce vendor lock-in and increase platform transparency.

6. Enhanced Security and Governance

- Security frameworks and automated compliance checks will become integral to LCNC platforms.

- Future systems will include built-in role-based access controls, audit logs, and enterprise-grade encryption to ensure trustworthiness.

7. Educational Integration and Curriculum Development

- Universities and coding bootcamps are likely to incorporate LCNC development into curricula, alongside traditional programming.
- This shift reflects the growing demand for hybrid skill sets that blend technical and business acumen.

8. Impact on the Software Engineering Profession

- As LCNC becomes mainstream, the role of software engineers will evolve from code-centric developers to **solution architects** and **platform enablers**.
- Engineers will focus more on system design, integration, security, and optimisation, while citizen developers handle routine apps.

References

1. Mendix. (2022). State of Low-Code 2022 Report.
2. Gartner. (2021). Magic Quadrant for Enterprise Low-Code Application Platforms.
3. Microsoft Power Platform Blog. (2021). Empowering Everyone to Build Apps. Forrester Research. (2020). The Forrester Wave™: Low-Code Development Platforms for AD&D Professionals, Q1 2020.
4. Microsoft Power Platform Blog. (2021). Empowering Everyone to Build Apps.
5. OutSystems. (2021). Low-Code Trends and Predictions.
6. IBM. (2021). AI in Low-Code Platforms: The Next Frontier.
7. Singh, M., & Tiwari, R. (2021). A Survey on Low-Code Development Platforms and Future Challenges. International Journal of Computer Applications, 183(44), 1-5.
8. Salesforce. (2020). The Rise of Citizen Developers.
9. Tafti, A., Mithas, S., & Krishnan, M. S. (2013). The Effect of Information Technology-Enabled Flexibility on Formation and Market Value of Alliances. Management Science, 59(1), 207–225.
10. Jansen, S., Brinkkemper, S., & Finkelstein, A. (2009). Business and IT alignment through enterprise architecture. Journal of Enterprise Information Management, 22(5), 526–546.