

# Low-Code/No-Code Development: Democratizing Software Engineering for the Digital Era

## OPEN ACCESS

Volume: 13

Special Issue: 3

Month: February

Year: 2026

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Citation:

Kavitha, B., et al.

“Low-Code/No-Code Development: Democratizing Software Engineering for the Digital Era.” *Shanlax International Journal of Arts, Science and Humanities*, vol. 13, no. 3, 2026, pp. 363–65.

DOI:

<https://doi.org/10.34293/sijash.v13iS3-i2-Feb.10306>

**Mrs. B. Kavitha**

*Assistant professor, Department of B.Sc.(CS)  
JHA Agarsen College, Madhavaram, Chennai*

**Mrs. M. Vanitha**

*Assistant professor cum Coordinator  
MMM College of Health Sciences, Mogappair East, Chennai*

**Dr. R. Sujatha**

*HOD, Department of Computer Applications  
JHA Agarsen College, Madhavaram, Chennai*

## Abstract

*Low-Code/No-Code (LCNC) development platforms are rapidly transforming the software development landscape by enabling non-technical users, or “citizen developers,” to create applications with minimal or no programming knowledge. This paper explores the evolution, benefits, limitations, and future trajectory of LCNC development. Through a systematic literature review and case study analysis, the research highlights the role of LCNC in accelerating digital transformation, enhancing organizational agility, and mitigating developer shortages. The study finds that while LCNC platforms offer significant efficiency gains, they also raise concerns regarding scalability, security, and maintainability. Recommendations are proposed for organizations aiming to adopt LCNC solutions strategically.*

## Introduction

In today’s digital economy, the demand for rapid application development and continuous innovation has increased. Traditional software development processes often struggle to keep pace with business needs due to limited developer resources and lengthy development cycles. Low-Code/No-Code (LCNC) platforms have emerged as a disruptive solution, enabling faster application delivery with less reliance on professional software engineers. These platforms use graphical user interfaces (GUIs), pre-built templates, and logic-based workflows to simplify programming tasks.

This paper investigates the impact of LCNC development on software engineering practices, workforce dynamics, and organizational strategy. It also addresses the challenges associated with LCNC adoption and outlines potential future developments.

## **Literature Review**

### **Definition and Classification**

Low-code platforms require some coding but reduce complexity through visual modeling tools. No-code platforms, by contrast, aim to eliminate coding entirely. Major platforms include Microsoft Power Apps, OutSystems, Mendix, and Appgyver.

### **Historical Background**

The roots of LCNC development date back to 4GLs (fourth-generation languages) and rapid application development (RAD) tools in the 1990s. Modern LCNC platforms extend these ideas using cloud computing, APIs, and AI-enhanced interfaces.

### **Current Applications**

LCNC tools are widely used in

- Workflow automation
- Internal business applications
- Customer relationship management (CRM)
- Data dashboards
- Mobile app development

### **Benefits**

- **Speed:** Drastically reduces time to deployment.
- **Accessibility:** Empowers non-developers to solve their own problems.
- **Cost-efficiency:** Reduces reliance on expensive software development teams.
- **Agility:** Facilitates rapid iteration and prototyping.

### **Criticisms and Challenges**

- **Scalability:** Limited customization for complex applications.
- **Security:** Vulnerabilities due to lack of rigorous development standards.
- **Governance:** Risk of “shadow IT” where employees build apps without oversight.
- **Lock-in:** Difficulty migrating away from proprietary LCNC ecosystems.

### **Methodology**

This study combines a qualitative literature review with two illustrative case studies. Sources include peer-reviewed journals, industry whitepapers, and vendor documentation. The case studies analyze:

- A financial services company using Power Platform
- A healthcare startup utilizing Bubble.io

Each case examines implementation strategy, outcomes, and challenges encountered.

## **Results**

### **Case Study 1: Financial Services Firm**

The firm deployed a no-code platform to automate client onboarding and compliance workflows.

Key outcomes:

- 60% reduction in manual processing time
- Positive feedback from internal teams
- Some scalability limitations encountered after initial rollout

## Case Study 2: Healthcare Startup

The startup built a mobile patient engagement app using a no-code tool. Key findings:

- App launched in under 6 weeks
- Total development cost under \$10,000
- Required later transition to custom codebase due to HIPAA compliance issues

## Discussion

For internal or non-critical applications, LCNC platforms have a strong value proposition, particularly for SMBs and departments with constrained IT resources. They are not a universally applicable solution, though. For mission-critical systems, hybrid approaches—combining LCNC technologies with traditional development—are frequently required due to scalability, compliance, and-technical-constraints.

In order to effectively manage citizen development and ensure compliance with organizational standards and security rules, organizations must also invest in governance frameworks.

## Conclusion

Platforms for low-code/no-code development are democratizing software development and facilitating quicker, more inclusive innovation. Although they have drawbacks, when used strategically, they can reduce developer bottlenecks and hasten digital transformation. More AI integration, more security features, and hybrid development models that combine low-code tools with conventional coding techniques are probably in store for the future.

## References

1. Yan, Z. (2021). The Effects of Low-Code/No-Code Development on Software Development and Digital Transformation. arXiv:2112.14073
2. N. Carroll (2025). A Comprehensive Review of the Literature on the Adoption of Low-Code and No-Code Development. Elsevier, Information and Software Technology.
3. Sufi, F. (2023). A Practical Review of Algorithms in Low-Code-No-Code for Research Applications. Systems and Software Modeling.
4. Magic Quadrant for Enterprise Low-Code Application Platforms, Gartner, 2024.
5. The 2023 Mendix. Low-Code Development Status Report.