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 6, Issue 11, May 2026
Impact Factor: 8.2

Low-Code and No-Code Platforms and Generative AI: Opportunities, Risks and Governance

Divya Bhogawade
Student, MCA, Sadhu Vaswani Institute of Management Studies, Pune, India

Abstract: *Organizations need software faster than before, but many IT teams cannot keep up with every request. Because of this, low-code and no-code platforms are becoming popular in business, education, and service sectors. These tools help users build apps, automate tasks, and manage workflows with less manual coding.*

At the same time, generative AI tools such as ChatGPT and GitHub Copilot and Claude are changing software work by helping users write code, explain logic, and prepare drafts of technical content. This creates new opportunities for citizen developers, who are business users building digital solutions without being full-time programmers.

This paper studies the role of low-code/no-code platforms, citizen development, and generative AI in software development. It explains the major benefits, common risks, and the need for governance. The paper argues that these technologies are useful, but they work best when organizations support users with training, review, and clear rules.

Keywords: Low-code, No-code, Microsoft Power Apps, Citizen Development

I. INTRODUCTION

Software is now part of almost every business process. However, many organizations still struggle to deliver internal apps, workflow tools, and digital services quickly because IT teams are often overloaded with large and urgent projects. This gap has increased the use of low-code and no-code platforms. These platforms allow users to build applications through visual tools, prebuilt templates, and data connectors instead of writing everything from scratch.

Generative AI has made this shift stronger. Tools such as ChatGPT and GitHub Copilot can help users generate sample code, explain programming steps, and support software development work in a faster and more interactive way.

Together, these technologies support citizen development. In this model, business users or domain experts take part in building apps for their own teams, often with some support from Information Technology.

II. PROBLEM STATEMENT AND OBJECTIVES

Low-code tools are often advertised as easy ways for anyone to build software. However, this is only partly true. Many organizations still struggle with issues like weak security, poor integration with legacy systems, lack of maintenance, and confusion over app ownership created by business users.

Generative AI adds another layer of concern. While it can save time, it may also produce incorrect or insecure outputs if users rely on it without proper verification.

The main issue discussed in this paper is how organizations can use low-code/no-code platforms and generative AI to enhance software delivery while ensuring quality and security.

The goals of this paper are:

- To explain citizen development, low-code/no-code platforms, and generative AI.
- To outline the main benefits of these technologies.
- To pinpoint common risks and limitations in their use.
- To recommend simple governance practices for safer adoption.

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 6, Issue 11, May 2026
Impact Factor: 8.2

III. RESEARCH METHOD

This paper relies on secondary research. It draws on guidance from platforms, security resources, and discussions surrounding low-code/no-code development, citizen development, and generative AI in software projects.

The material was reviewed to find recurring themes, which were grouped into three main areas: benefits, risks, and governance practices. The intent is to provide a straightforward and practical understanding of the topic rather than conduct a statistical or experimental study.

IV. LITERATURE REVIEW

Citizen Development

Citizen development involves business users creating apps or process solutions for their work using approved digital platforms. This method helps organizations address local problems more quickly, but it still requires guidance, ownership, and review. Citizen developers are generally not full-time software engineers. They may be HR staff, office managers, finance executives, teachers, or operations personnel who know the process well and want to enhance it using digital tools.

Low-Code and No-Code Platforms

Low-code and no-code platforms are systems that enable users to create apps via drag-and-drop interfaces, prebuilt modules, workflow tools, and connectors. Microsoft points out that these platforms reduce the need for manual coding, while IBM describes them as a quicker way to create business applications for common uses.

Examples include Microsoft Power Apps, Mendix, OutSystems, and Salesforce Lightning. These platforms are commonly used for internal forms, leave approval apps, inventory tracking, customer service flows, and mobile access to business data.

Generative AI in Software Development

Generative AI refers to AI systems that can produce content such as text, code, suggestions, or explanations. In software development, these tools can assist with coding, problem-solving, testing support, and documentation.

GitHub Copilot is well-known as an AI coding assistant, while ChatGPT is frequently used to clarify programming concepts, generate sample code, and facilitate learning. These tools can help beginners get started more quickly, but their output should always be verified by humans.

V. PLATFORM EXAMPLES

Platform examples clarify and make the topic more practical.

- Microsoft Power Apps helps build internal business applications with a visual interface and strong Microsoft ecosystem support.
- Power Automate enables workflow automation like approvals, reminders, and office task automation within organizations.
- Mendix supports rapid application development and model-based enterprise apps.
- OutSystems is used for web and mobile business applications with faster delivery cycles.
- Salesforce Lightning aids in app building and workflow customization related to CRM and customer-facing processes.
- GitHub Copilot assists in writing and suggesting code during development work.
- ChatGPT is commonly used for code explanations, first drafts, and logic support.

For example, a college office can use Power Apps and Power Automate to create a student leave or document request app. A retail company may use Mendix or OutSystems to develop a stock-checking or approval app faster than traditional coding methods.

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 6, Issue 11, May 2026
Impact Factor: 8.2

VI. BENEFITS AND OPPORTUNITIES

Faster Development

One major advantage of low-code/no-code platforms is speed. Visual tools and reusable components enable teams to build common business applications more quickly than old development methods.

Better Fit with Business Needs

Business users typically understand their work processes very well. When they are involved in building digital tools, the final solution may better meet real needs and reduce repeated communication between IT and the business team.

Broader Involvement in Digital Work

Citizen development allows more employees to participate in solving problems through technology. This can promote innovation in departments that usually wait a long time for technical support.

Learning Support Through AI

Generative AI tools can aid learning by explaining concepts, suggesting code, and providing examples. This can help new developers or students grasp app logic more quickly, especially in the early stages of development.

VII. RISKS AND CHALLENGES

Security Risks

Security is a primary concern associated with citizen development. OWASP highlights important risks such as weak access control, insecure data handling, unsafe integrations, and poor testing in applications built outside of robust security review processes.

Weak Governance and Shadow IT

Without rules and oversight, organizations can end up with many small apps that lack proper ownership and monitoring. Microsoft emphasizes that governance is crucial to regulate who can create, share, and deploy low-code solutions.

Software Quality Issues

Applications built quickly may work temporarily but can become difficult to maintain over time. Inadequate documentation, repetitive logic, weak testing, and unclear ownership can compromise long-term quality.

Integration with Legacy Systems

Low-code tools are beneficial, but they do not eliminate the challenges associated with older systems, poor data quality, and complex business rules. Integration remains a significant hurdle for many organizations.

Trust in AI Output

Generative AI can save time, but it may also produce incorrect, insecure, or incomplete content. Therefore, AI-generated code and suggestions should always undergo review before implementation in real systems.

VIII. GOVERNANCE PRACTICES

To use these tools safely, organizations should implement a straightforward governance model.

- Approve only selected platforms for official use.
- Define what business users can create independently and what should remain under IT control.
- Provide basic training in platform use, app design, and security awareness.
- Use standard templates and approved connectors whenever possible.
- Review citizen-built solutions before deployment, especially those that manage sensitive data.
- Verify AI-generated code and content through human review before production use.

A balanced model is typically best. Simple departmental tools can be created by trained users, while critical systems and sensitive processes should stay under experienced IT teams.

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 6, Issue 11, May 2026
Impact Factor: 8.2

IX. RELEVANCE FOR INDIAN ORGANIZATIONS AND ACADEMIA

Low-code/no-code platforms can be beneficial in Indian colleges, startups, offices, and service organizations that need digital tools but lack technical resources. They can assist with attendance systems, request forms, dashboards, helpdesk solutions, and student projects.

In education, these tools can aid students in understanding how software supports real business processes. At the same time, colleges should emphasize that rapid development does not eliminate the need for testing, security checks, and clear ownership.

X. CONCLUSION

Low-code or no-code platforms and generative AI are making software development more accessible and faster for many companies. They support quicker app creation, wider participation, and practical digital problem solving.

However, these benefits come with risks. Security gaps, weak governance, poor maintenance, and blind trust in AI outputs can create serious problems if these tools are used carelessly.

The best approach is balanced adoption. We should use these technologies for suitable tasks, train users properly, and keep IT involved where complexity or risk is high.

REFERENCES

[1]. Microsoft Power Apps, "What is Low-Code Governance," Microsoft, available at: https://www.microsoft.com/en/power-platform/products/power-apps/topics/low-code-no-code/what-is-low-code-governance-and-why-it-matters.
[2]. OWASP Foundation, "OWASP Citizen Development Top 10," OWASP, available at: https://owasp.org/www-project-citizen-development-top10-security-risks/.
[3]. Salesforce, "What is Citizen Development?" Salesforce, available at: https://www.salesforce.com/platform/citizen-development/.
[4]. ServiceNow, "What is Citizen Developer?" ServiceNow, available at: https://www.servicenow.com/workflows/creator-workflows/what-is-a-citizen-developer.html.
[5]. IBM, "Low-Code vs. No-Code: What's the Difference?" IBM, available at: https://www.ibm.com/think/topics/low-code-vs-no-code.
[6]. GitHub, "GitHub Copilot," GitHub, available at: https://github.com/features/copilot/ai-code-editor.
[7]. Oracle, "7 Ways GenAI Can Help Improve Software Development," Oracle, available at: https://www.oracle.com/in/artificial-intelligence/generative-ai/generative-ai-software-development/.
[8]. Mendix, "Low-Code Examples and Use Cases," Mendix, available at: https://www.mendix.com/low-code-guide/low-code-use-cases/.
[9]. Microsoft Power Apps, "Low-Code vs. No-Code App Development," Microsoft, available at: https://www.microsoft.com/en-us/power-platform/products/power-apps/topics/low-code-no-code/low-code-no-code-development-platforms.
[10]. Research study on generative AI and developer workflows using GitHub Copilot and ChatGPT.