

Security Implications and Governance Risk of Citizen Development in Low-Code/No-Code Platforms: A Systematic Review

Durojaye E. Olatunji, Nwachukwu Peter, Omolaja Daniel, Olagbenro Marvellous, Okorie Christopher & Ewulana Idris

ARTICLE INFORMATION

Article history:

Published: August 2026

Keywords:

Citizen Development, Low-Code Platforms, No-Code Platforms, LCNC, Security Implications, Governance Risks, Shadow IT, Cybersecurity, IT Governance, Data Privacy, Compliance, Systematic Review, Citizen Developers, Application Security

ABSTRACT

The rapid proliferation of low-code/no-code (LCNC) platforms has empowered citizen developers—non-professional users with limited technical expertise—to create and deploy business applications at unprecedented speed. While this democratization of software development offers significant productivity gains and agility, it introduces substantial security implications and governance risks that remain underexplored. This systematic review synthesizes existing literature on the security vulnerabilities and governance challenges associated with citizen development in LCNC environments. The study highlights the tension between innovation velocity and risk management, proposing a balanced framework that integrates security-by-design principles, automated governance controls, and hybrid oversight models. Implications for IT leaders, policymakers, and platform vendors are discussed, emphasizing the need for proactive risk mitigation strategies to harness the benefits of LCNC platforms without compromising enterprise security and governance standards. Future research directions are suggested to address evolving threats in increasingly complex digital ecosystems.

1. Introduction

Low-code/no-code (LCNC) platforms have emerged as transformative tools that enable users to design, develop, and deploy applications and automated workflows through visual interfaces, drag-and-drop components, and configuration settings, with minimal or no traditional coding required. By significantly lowering technical barriers, these platforms have democratized application development, allowing business users and domain experts to rapidly create tailored solutions without heavy reliance on professional IT teams. This shift has given rise to the phenomenon of citizen development, whereby non-IT employees—often referred to as citizen developers—actively build and maintain software applications. The adoption of LCNC platforms has accelerated across diverse industries, fueling a substantial increase in citizen-driven development initiatives. As noted by Ajimati et al. (2025) and ToolJet (2026), the proliferation of these platforms is fundamentally reshaping software development practices and expanding the scale of citizen developer activity.

However, this rapid empowerment of non-technical users introduces critical challenges. While LCNC platforms deliver speed and agility, they also create significant security vulnerabilities and governance risks that organizations must address. This systematic review examines the security implications and governance risks associated with citizen development in low-code/no-code environments, synthesizing current evidence to highlight key challenges and propose pathways for responsible adoption.

1.1 What are the Security Implications and Governance Risks of Citizen Development in Low-Code/No-Code Platforms?

Citizen development in low-code/no-code (LCNC) platforms enable business users to rapidly build applications and automations with minimal coding.

Security Implications:

- Data Exposure: Frequent misconfigurations lead to unintentional public sharing of sensitive data.
- Weak Access Controls: Overly broad permissions and authorization misuse expose confidential information.
- Injection Vulnerabilities: Failures in input validation enable SQL/script injection attacks.
- Insecure Integrations: Unvetted third-party connectors and hardcoded secrets increase breach risks.
- Authentication Weaknesses: Poor session management and account impersonation risks.
- Shadow IT Vulnerabilities: Unmonitored apps expand the attack surface.

Governance Risks:

- Application Sprawl & Shadow IT: Proliferation of unmanaged applications with unclear ownership.
- Compliance Violations: Inconsistent data handling and missing audit trails breach regulations.
- Governance Debt: Lack of lifecycle management, documentation, and maintenance.
- Skill Gaps: Citizen developers often bypass security reviews and best practices.
- Accountability Issues: Ambiguous responsibility between business users and IT.

1.2 Why It Is Important

The study of security implications and governance risks in citizen development using low-code/no-code (LCNC) platforms is critical for the following reasons:

- **Rapid Adoption Outpacing Controls:** LCNC platforms are being widely adopted across industries, with citizen developers expected to outnumber professional developers in many organizations. Without proper safeguards, this creates widespread security vulnerabilities and governance gaps.
- **Significant Business and Financial Risks:** Misconfigurations and shadow IT have already caused major data breaches, compliance violations (e.g., GDPR, HIPAA), and financial losses. The average cost of a data breach continues to rise, with shadow IT-linked incidents being notably more expensive.
- **Expanding Attack Surface:** Citizen-built applications often bypass traditional IT security reviews, increasing exposure to data leaks, unauthorized access, and cyberattacks in an era of sophisticated threats.
- **Regulatory and Compliance Pressure:** Organizations face stricter regulations around data privacy and cybersecurity. Uncontrolled LCNC development heightens the risk of hefty fines and reputational damage.
- **Balancing Innovation and Risk:** While LCNC delivers speed, agility, and digital transformation, unchecked citizen development can lead to "governance debt," application sprawl, and long-term technical liabilities that undermine the very benefits these platforms promise.

1.3 Core Components

- **Citizen Developers:** Non-IT business users building applications.
- **LCNC Platforms:** Visual, drag-and-drop tools with minimal coding (e.g., Power Platform, ServiceNow, Salesforce).
- **Developed Artifacts:** Apps, workflows, automations, and AI agents.
- **Security Components:** Data protection, access controls, input validation, integrations, and configurations.
- **Governance Components:** Policies, ownership, visibility/monitoring, training, and lifecycle management.

1.4 Current Trends and Innovations (2020–2026)

Between 2020 and 2026, several important trends have shaped the security implications and governance risks of citizen development in low-code/no-code (LCNC) platforms:

Explosive Growth of Citizen Development: Adoption of LCNC platforms accelerated rapidly, with citizen developers increasingly outnumbering professional developers in many organizations. This expansion significantly increased the volume of unmanaged applications and shadow IT (Ajimati et al., 2025; ToolJet, 2026).

Rise of AI-Assisted Development: From 2023 onward, generative AI features became standard in major LCNC platforms (e.g., Microsoft Power Platform, Salesforce, ServiceNow). While this improved productivity, it also introduced new risks such as AI-generated insecure logic, hardcoded credentials, and reduced visibility into application behavior (ToolJet, 2026).

Formalization of Risk Frameworks: The OWASP Citizen Development Top 10 (and earlier Low-Code/No-Code Top 10) emerged as a key reference for identifying platform-specific security risks, including data leakage, authorization misuse, and injection vulnerabilities (OWASP Foundation, 2025).

Shift Toward Hybrid Governance Models: Organizations began moving away from pure restriction policies toward hybrid approaches that combine citizen empowerment with Centers of Excellence (CoE), automated policy enforcement, and risk-based approval processes (Ajimati et al., 2025).

Increased Focus on Governance Debt: Research and industry reports highlighted the accumulation of long-term risks from undocumented applications, unclear ownership, and weak lifecycle management (Nokod Security, 2024).

Platform-Level Security Enhancements: Vendors introduced more built-in controls such as environment isolation, data loss prevention (DLP), and automated scanning, although these features remain underutilized by many citizen developers (OWASP Foundation, 2025; Nokod Security, 2024).

Overall, the period 2020–2026 shows that while LCNC platforms matured technically, security and governance practices have struggled to keep pace with the speed of adoption and AI integration.

1.5 Evaluation

The evaluation of security implications and governance risks in citizen development using low-code/no-code (LCNC) platforms reveal a clear imbalance: the benefits of speed and innovation are significant, but the risks are substantial and often underestimated.

Key Findings

- **High Prevalence of Risks:** Misconfigurations, weak access controls, and data exposure are widespread due to limited security knowledge among citizen developers.
- **Governance Gaps:** Most organizations lack mature policies, resulting in shadow IT, application sprawl, and compliance vulnerabilities.
- **Severity:** Risks are amplified by rapid adoption, AI integration, and the ease of platform use, which creates a false sense of security.
- **Impact Level:** Medium to High — potential for data breaches, regulatory fines, reputational damage, and increased technical debt.

1.6 Overall Assessment

While LCNC platforms empower business agility, uncontrolled citizen development introduces systemic risks that can outweigh the benefits if not properly managed. Current governance practices are largely insufficient, highlighting an urgent need for structured frameworks, training, and automated controls.

Conclusion: Proactive evaluation and mitigation are essential to safely realize the value of citizen development.

1.7 Goals

The primary goals of this systematic review are:

- To identify and analyze the key security implications associated with citizen development in low-code/no-code (LCNC) platforms.
 - To examine the major governance risks, including shadow IT, compliance issues, and accountability gaps.
 - To evaluate the current state of research and practices regarding these risks.
 - To highlight the interplay between innovation benefits and potential vulnerabilities.
 - To propose actionable recommendations for balancing rapid development with security and governance controls.
- This review aims to provide organizations with a clear understanding of the risks and support the development of effective mitigation strategies for safe LCNC adoption.

2. Literature Review

The literature on security implications and governance risks of citizen development in low-code/no-code (LCNC) platforms has grown significantly in recent years. Early studies primarily emphasized the productivity and innovation benefits of LCNC platforms (Ajimati et al., 2025; ToolJet, 2026). However, more recent works have shifted focus toward the associated risks.

Key security concerns documented across studies include data leakage through misconfigurations, weak access controls, injection vulnerabilities, insecure third-party integrations, and authentication weaknesses (OWASP, 2025; Nokod Security, 2024). The OWASP Citizen Development Top 10 has become a foundational framework for identifying platform-specific risks.

On the governance side, researchers highlight shadow IT proliferation, application sprawl, unclear ownership, compliance gaps, and the emergence of governance debt as major challenges.

Several empirical reports note that rapid adoption often outpaces organizational policies and IT oversight, amplifying risks in regulated industries such as finance and healthcare.

2.1 Definition and Characteristics

Citizen Development refers to the practice where non-IT employees (business users and domain experts) create, modify, and maintain business applications and automations using low-code/no-code tools.

Low-Code/No-Code (LCNC) Platforms are visual development environments that allow users to build applications through drag-and-drop interfaces, pre-built components, and configuration rather than traditional programming.

Key Characteristics:

Ease of Use: Minimal or no coding required.

Speed: Rapid application development and deployment.

Democratization: Empowers non-technical users to solve business problems.

Integration: Easy connectivity with existing systems and data sources.

Accessibility: Cloud-based with templates and AI assistance.

2.2 Historical Development

Security and governance concerns in citizen development emerged around 2015–2018 with the growth of low-code platforms and early shadow IT issues. These risks intensified between 2020 and 2022 due to rapid digital transformation (Ajimati et al., 2025).

From 2023 onward, AI-assisted development and large-scale LCNC adoption further amplified risks such as data leakage, application sprawl, and governance debt, leading to formal frameworks like the OWASP Citizen Development Top 10 (OWASP Foundation, 2025; ToolJet, 2026).

2.3 Impact on Organizations

Citizen development in low-code/no-code platforms have profound effects on organizations:

Positive Impacts:

- Accelerated innovation and faster time-to-market for business solutions.
- Reduced IT backlog and dependency on professional developers.
- Increased business agility and employee empowerment.
- Significant cost savings and productivity gains.

Negative Impacts:

- Heightened security risks, including data breaches and misconfigurations.
- Governance challenges such as shadow IT, application sprawl, and compliance violations.
- Increased technical and governance debt over time.
- Potential reputational damage and regulatory fines.

Overall, while LCNC platforms drive digital transformation, they require strong governance to maximize benefits and minimize risks. Organizations that implement proper controls can achieve sustainable competitive advantage; those that do not face growing operational and security vulnerabilities.

2.4 Adoption Across Industries

Low-code/no-code platforms and citizen development have gained significant traction across multiple sectors. Finance and banking use them for compliance, customer onboarding, and workflow automation (Ajimati et al., 2025). Healthcare adopts them for patient portals, reporting tools, and operational efficiency.

Manufacturing, logistics, and retail leverage these platforms for supply chain management and personalized customer experiences (ToolJet, 2026). Government and insurance sectors apply them to digitize services and streamline claims processing.

2.5 Challenges and Limitations

Citizen development in low-code/no-code platforms face several key challenges and limitations:

- Security Vulnerabilities: High risk of data leaks, misconfigurations, and weak access controls due to limited expertise of citizen developers.
- Governance Gaps: Shadow IT, application sprawl, unclear ownership, and compliance issues.
- Skill and Quality Issues: Inconsistent application quality, lack of proper testing, and technical debt accumulation.
- Scalability Constraints: Many citizen-built apps struggle with complex enterprise requirements and long-term maintenance.
- Organizational Resistance: Tension between business agility and central IT control.
- Regulatory and Compliance Risks: Difficulty meeting strict standards in regulated industries.

2.6 Future Directions

Future research and practice should focus on developing integrated security-by-design frameworks for low-code/no-code platforms (OWASP, 2025), advancing AI-assisted governance and automated risk detection tools, creating scalable training programs for citizen developers (Ajimati et al., 2025), and establishing hybrid governance models. Organizations will also benefit from more empirical studies on real-world mitigation strategies and the long-term impact of AI-enhanced citizen development (ToolJet, 2026).

2.7 Core Concepts

Citizen development refers to non-IT business users building applications using low-code/no-code tools. Low-code/no-code platforms are visual development environments that minimize traditional coding through drag-and-drop interfaces and configurations. Security implications include risks such as data leakage, misconfigurations, injection vulnerabilities, and insecure integrations (OWASP, 2025). Governance risks involve challenges including shadow IT, compliance gaps, application sprawl, and unclear ownership. A balanced approach combines innovation speed with security-by-design and structured oversight (Ajimati et al., 2025).

2.7.1 Visual Development Environment

A visual development environment is the core interface of low-code/no-code platforms. It enables users to build applications using drag-and-drop components, pre-built templates, visual workflows, and configuration settings instead of writing extensive code (Ajimati et al., 2025).

2.7.2 Pre-Built Templates and Components

Pre-built templates and components are ready-made elements provided by low-code/no-code platforms, such as forms, dashboards, connectors, and workflow modules. They allow citizen developers to quickly assemble applications without building everything from scratch.

2.7.3 Automation and Workflow Design

Automation and workflow design in low-code/no-code platforms enable citizen developers to create business processes, integrations, and repetitive task automations through visual flow builders. These tools connect applications, data sources, and external systems with minimal coding.

2.7.4 Scalability and Reusability

Scalability and reusability refer to the ability of low-code/no-code applications and components to handle growing user loads and be repurposed across different business needs. Platforms support modular design, shared libraries, and easy updates to promote reuse.

2.7.5 Built-In Security and Compliance

Built-in security and compliance features in low-code/no-code platforms include role-based access controls, data encryption options, audit logging, and pre-configured regulatory templates. These aim to support citizen developers in meeting security and compliance requirements more easily.

2.7.6 Collaboration and Team work

Collaboration and teamwork features in low-code/no-code platforms allow multiple citizen developers and IT professionals to work together on the same applications through shared workspaces, version control, commenting, and real-time co-editing.

2.7.7 Application Lifecycle Management

Application lifecycle management (ALM) in low-code/no-code environments encompass the processes of planning, designing, developing, testing, deploying, monitoring, and retiring citizen-developed applications. Effective ALM ensures applications remain secure, compliant, and maintainable throughout their lifespan. However, many LCNC initiatives suffer from weak ALM practices, resulting in undocumented apps, unpatched vulnerabilities, and accumulating technical debt.

Strong governance and platform-supported ALM tools are essential for sustainable citizen development.

2.7.8 Integration Capabilities

Low-code/no-code platforms enable rapid connections to databases, APIs, and enterprise systems through pre-built connectors (Ajimati et al., 2025).

However, these capabilities often lead to over-permissioned access, unvetted third-party integrations, and insecure data flows, increasing the risk of data leakage and unauthorized access (OWASP Foundation, 2025; Nokod Security, 2024).

2.8 Challenges and Limitations

Citizen development in low-code/no-code platforms present several notable challenges and limitations:

- Security vulnerabilities from misconfigurations, weak access controls, and insecure integrations by untrained users.
- Governance gaps, including shadow IT, application sprawl, and unclear ownership.
- Quality and maintainability issues, leading to technical debt and scalability problems.

- Compliance risks in regulated industries due to inconsistent standards.
- Skill gaps and organizational resistance between business and IT teams.
- Addressing these challenges requires robust training, governance frameworks, and platform-level controls to ensure sustainable adoption.

2.9 Applications of Security Implications and Governance Risks of Citizen Development In Low-Code/No-Code Platforms

The security implications and governance risks of citizen development manifest in various real-world applications across organizations:

- Data Exposure in Business Apps: Citizen-built reporting tools or customer portals often expose sensitive data due to misconfigured sharing settings.
- Workflow Automations: Insecure Power Automate or similar flows can lead to unintended data leaks or malware-like behavior through unmonitored connectors.
- Shadow IT in Departments: HR, finance, and operations teams create unapproved apps that bypass central security reviews, increasing breach risks.
- Compliance Violations: Citizen-developed apps handling personal or financial data frequently fail regulatory requirements (GDPR, HIPAA) due to poor audit trails.
- Enterprise Integration Risks: Direct connections to core systems by citizen developers expand the attack surface through over-permissioned APIs.

2.10 Technical Insights

Low-code/no-code platforms abstract much of the underlying complexity, but several technical aspects remain critical for security and governance:

- Abstraction Layers: Visual interfaces hide code, making it harder for citizen developers to identify vulnerabilities such as injection points or insecure data flows.
- Connector Ecosystem: Easy integration with APIs and third-party services expands the attack surface if connectors are over-permissioned or unvetted.
- Runtime Environment: Many platforms run citizen apps with broad permissions by default, increasing the impact of misconfigurations.
- AI-Assisted Development: Generative AI features can produce insecure code or logic if prompts lack security context.
- Logging and Monitoring: Limited visibility into application behavior often results in undetected security incidents.
- Component Reuse Risks: Reliance on marketplace templates and shared components can propagate vulnerabilities across multiple applications.

These technical characteristics explain why security implications often materialize quickly and why governance is difficult to enforce at scale. Addressing them requires a combination of platform-level controls, automated scanning, and developer guardrails.

2.11 Case Study

Microsoft Power Platform Misconfigurations: Multiple organizations (including large enterprises) experienced significant data exposures due to citizen-developed Power Apps being shared publicly or with overly broad permissions. Nokod Security reports have documented cases where simple citizen-built apps led to salary data leaks and external exposures.

Power Automate as Malware Vector: Security researchers demonstrated how citizen-developed flows could be weaponized for data exfiltration and ransomware-like behavior using legitimate platform features.

Shadow IT in Finance and Healthcare: Several banks and hospitals reported compliance violations and data breaches from unregulated citizen-developed apps handling sensitive customer or patient data.

Enterprise-Wide Sprawl: Companies adopting ServiceNow, Salesforce, and Mendix at scale have faced challenges with thousands of unmanaged citizen apps, leading to governance debt and security blind spots.

3. Methodology

This study follows a systematic review approach to examine the security implications and governance risks of citizen development in low-code/no-code (LCNC) platforms (Ajimati et al., 2025).

Relevant literature published between 2018 and 2026 was identified through academic databases and industry sources. Key search terms included “citizen development,” “low-code,” “no-code,” “security risks,” “governance,” and “shadow IT” (OWASP Foundation, 2025).

Studies were selected based on relevance to security vulnerabilities, governance challenges, and organizational impacts of LCNC platforms. Both peer-reviewed articles and credible industry reports were included to provide a comprehensive view (ToolJet, 2026; Nokod Security, 2024).

The selected materials were analyzed and synthesized to identify common risks, trends, and gaps in current research and practice.

4. Findings

The systematic review reveals the following key findings:

Citizen development in LCNC platforms significantly increases the risk of data exposure through misconfigurations and overly broad sharing settings.

Weak access controls and over-permissioned connectors are among the most common security vulnerabilities.

Shadow IT and application sprawl create major governance challenges, with many organizations lacking clear ownership and visibility.

Compliance risks are high, especially in regulated industries, due to missing audit trails and inconsistent data handling. Current governance practices are generally insufficient to match the speed of LCNC adoption and AI-assisted development. Without structured controls, the security and governance risks can outweigh the productivity benefits of citizen development.

5. Conclusion and Recommendation

5.1 Conclusion

This systematic review shows that citizen development in low-code/no-code platforms deliver strong productivity and agility benefits, but it also creates substantial security and governance risks such as data exposure, weak access controls, shadow IT, and compliance gaps (Gartner, 2024; OWASP, 2021; NIST, 2024).

5.2 Recommendations

- Organizations should establish a Center of Excellence (CoE) to coordinate and govern citizen development initiatives (Microsoft, 2024).
- Automated security testing, least-privilege access controls, and data loss prevention mechanisms should be integrated into LCNC platforms (NIST, 2024; ISO, 2022).
- Citizen developers should receive continuous security awareness and governance training to reduce human-related risks (World Economic Forum, 2023).
- Every application should have a designated owner and follow a structured application lifecycle management process (OutSystems, 2023; Microsoft, 2024).
- Organizations should adopt hybrid governance frameworks that balance innovation with regulatory compliance and security (Gartner, 2024; Salesforce, 2023).

References

- [1] Ajimati, A., et al. (2025). Citizen development and low-code platforms in modern enterprises. *Journal of Digital Transformation*.
- [2] Gartner. (2024). Magic Quadrant for enterprise low-code application platforms. Gartner.
- [3] Microsoft. (2024). Microsoft Power Platform Center of Excellence (CoE) Starter Kit. Microsoft Learn. <https://learn.microsoft.com/power-platform/guidance/coe/>
- [4] Microsoft. (2024). Power Platform Well-Architected Framework. Microsoft Learn. <https://learn.microsoft.com/power-platform/well-architected/>
- [5] Open Worldwide Application Security Project. (2021). OWASP low-code/no-code top 10 security risks. <https://owasp.org/www-project-top-10-low-code-no-code-security-risks/>
- [6] OutSystems. (2023). Application lifecycle management in low-code development. <https://www.outsystems.com/>
- [7] Salesforce. (2023). Low-code governance: Best practices for enterprise application development. <https://www.salesforce.com/resources/articles/low-code/>
- [8] World Economic Forum. (2023). Global cybersecurity outlook 2023. <https://www.weforum.org/publications/global-cybersecurity-outlook-2023/>
- [9] Nokod Security. (2024). Power Platform security risks and data leakage in low-code environments.
- [10] OWASP Foundation. (2025). OWASP Citizen Development Top 10 Security Risks.
- [11] ToolJet. (2026). The state of low-code/no-code adoption and citizen development report.