Data and Metadata. 2025; 4:1218
doi: 10.56294/dm20251218

SYSTEMATICC REVIEW

Low-Code and No-Code Development in the Era of Artificial Intelligence: A Systematic Review

Desarrollo Low-Code y No-Code en la Era de la Inteligencia Artificial: Una Revisión Sistemática

Gabriel Luis L. Liwanag[1], Ryan A. Ebardo[1], Danny C. Cheng[1]

[1]De La Salle University, College of Computer Studies. Manila, Philippines.

Cite as: Liwanag GLL, Ebardo R, Cheng D. Low-Code and No-Code Development in the Era of Artificial Intelligence: A Systematic Review. Data and Metadata. 2025; 4:1218. https://doi.org/10.56294/dm20251218

Submitted: 12-06-2025 **Revised:** 28-08-2025 **Accepted:** 26-10-2025 **Published:** 27-10-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding author: Gabriel Luis L. Liwanag

ABSTRACT

Low-code, no-code has transformed software development by enabling non-programmers to build applications through visual interfaces. With the growing integration of artificial intelligence (AI), these platforms are evolving into intelligent environments that accelerate innovation and bridge the global shortage of skilled developers. This significance of this research is it clarifies how AI enhances LCNC development, a field increasingly central to digital transformation and automation. Using the systematic literature review method, the study gathered 1,582 initial records from Scopus, EBSCO, and AISeL, applying inclusion and exclusion criteria with PRISMA to arrive at 60 peer-reviewed journals published between 2020 and 2025. Each article was analyzed in terms of theoretical framework, industry context, development phase AI was integrated, challenges, and opportunities, with data thematically synthesized to trace emerging trends. The research found out that AI's role in LCNC development is expanding and acting as co-developer increasing software reliability and adaptability. Results show that AI is primarily utilized in the development and maintenance phases of LCNC, functioning as a co-developer that automates code generation, debugging, and optimization. Most studies focus on applications in healthcare, manufacturing, logistics, and education, and finance, while theoretical engagement and diversity across industries remain limited. The study concluded that the convergence of LCNC and AI marks a shift toward intelligent, citizen-driven software development, underscoring the need for future research on theory, governance, and sector-specific best practices.

Keywords: Low-Code; No-Code; Development; Innovation; Artificial Intelligence; Citizen Development.

RESUMEN

El desarrollo low-code y no-code ha transformado la creación de software al permitir que personas sin conocimientos avanzados de programación construyan aplicaciones mediante interfaces visuales. Con la creciente integración de la inteligencia artificial (IA), estas plataformas están evolucionando hacia entornos inteligentes que aceleran la innovación y reducen la escasez global de desarrolladores especializados. La importancia de esta investigación radica en que aclara cómo la IA potencia el desarrollo LCNC, un campo cada vez más central en la transformación digital y la automatización. Mediante el método de revisión sistemática de la literatura (SLR), el estudio recopiló 1 582 registros iniciales de Scopus, EBSCO y AISeL, aplicando criterios de inclusión y exclusión bajo el enfoque PRISMA hasta llegar a 60 artículos revisados por pares publicados entre 2020 y 2025. Cada artículo fue analizado según su marco teórico, contexto industrial, fase de desarrollo en la que se integró la IA, desafíos y oportunidades, sintetizando los datos temáticamente para identificar tendencias emergentes. La investigación reveló que el papel de la IA en el desarrollo LCNC se está expandiendo y actuando como co-desarrollador, aumentando la confiabilidad y adaptabilidad del software.

Los resultados muestran que la IA se utiliza principalmente en las fases de desarrollo y mantenimiento, automatizando la generación de código, la depuración y la optimización. La mayoría de los estudios se centran en aplicaciones en salud, manufactura, logística, educación y finanzas, mientras que la fundamentación teórica y la diversidad sectorial siguen siendo limitadas. El estudio concluye que la convergencia entre LCNC e IA marca una transición hacia un desarrollo de software inteligente y dirigido por los ciudadanos, subrayando la necesidad de futuras investigaciones sobre teoría, gobernanza y mejores prácticas específicas por sector.

Palabras clave: Low-Code; No-Code; Desarrollo; Innovación; Inteligencia Artificial; Desarrollo Ciudadano.

INTRODUCTION

The emergence of low-code, no-code (LCNC) development platforms creates a revolutionary approach to software creation through democratizing application development by enabling individuals with limited programming expertise to build functional software solutions through visual interfaces and pre-built components. The evolution of software development has been marked by progressive ideas aimed at improving developer productivity and accessibility. From the complexity of assembly language, through the structured approaches of C, to the more intuitive syntax of JavaScript and Python, each advancement has sought to make programming more accessible. However, even these higher-level languages still require substantial knowledge of computer science concepts.(1) Traditional programming, however, remains resource-intensive and dependent on highly skilled developers, contributing to the persistent global shortage of software talent. The emergence of LCNC development thus represents not merely a technical innovation, but a socio-technical response to this structural problem in the software industry.

LCNC platforms represent the next significant step in this evolution, abstracting complex programming concepts behind visual interfaces and drag-and-drop functionality. The rise of LCNC platforms is advantageous given the current global shortage of professional developers. Organizations face pressure to digitize their operations and create custom software solutions, yet the traditional pool of skilled developers cannot meet this exponential demand. LCNC platforms address this gap by enabling ‘citizen developers’, who are business users with domain expertise but limited programming knowledge, to create applications that would require professional development teams.(2)

In recent years, the capabilities of LCNC platforms have been enhanced by the integration of artificial intelligence (AI) technologies. This convergence represents a new frontier in software development, where AI-powered features such as natural language processing for requirements gathering, component suggestions, and automated testing are revolutionizing how applications are built. The addition of AI capabilities to LCNC platforms not only accelerates development processes but also introduces new possibilities for intelligent automation and decision support within applications. At this stage, artificial intelligence becomes integral to LCNC development as its incorporation allows platforms to go beyond visual programming by embedding generative and analytical intelligence into the development process. Through AI-powered features such as automated code synthesis, context-aware debugging, and predictive component recommendation, LCNC systems evolve from static builders into adaptive, learning environments that respond dynamically to user input.

While numerous grey literature sources discuss the viability, usability, and efficiency of LCNC development platforms, there is a growing need to understand how the integration of AI is reshaping this field. This systematic literature review (SLR) aims to provide a comprehensive analysis of peer-reviewed studies on LCNC development and its intersection with AI technologies, examining current trends, challenges, and opportunities. By synthesizing research in this rapidly evolving area, the review seeks to clarify the implications for software development practices and identify promising directions for future inquiry. To address the objectives of this SLR, the following research questions will guide the investigation:

- Which phase in LCNC development is AI being utilized?
- What industries are leveraging AI on LCNC development?
- What are the challenges in utilizing AI to LCNC development?
- What are the opportunities in utilizing AI to LCNC development?
- What are the theoretical approaches in studying LCNC and AI Integration?

To support a comprehensive analysis, this study will examine and synthesize existing peer-reviewed studies on LCNC development platforms and their integration with AI technologies. The review aims to trace the evolution of LCNC platforms, from basic visual programming tools to AI-enhanced development environments. Through this analysis, the study seeks to identify key patterns, challenges, and opportunities within this rapidly evolving domain. Despite the growing popularity of these tools, there remains limited scholarly synthesis that maps the current state of research, identifies emerging trends, and highlights the theoretical and practical implications of this integration. Addressing this gap is essential for both academics and practitioners seeking to understand how AI-driven LCNC systems contribute to the democratization of software development and to the broader goals of digital transformation

This research contributes to the existing body of knowledge through several novel aspects. First, it provides an up-to-date analysis of LCNC platforms in the context of emerging AI capabilities as existing reviews have focused primarily on LCNC alone.(3,4,5,6) For example, the review of Ajimati et al.(3) emphasizes the perspective of citizen development (CD), suggesting that future research should develop empirical, quantitative case studies to measure real-world performance outcomes of LCNC/CD adoption, formulate governance and compliance frameworks for managing citizen development effectively, and explore strategies to bridge the cultural gap between IT and business units for successful LCNC integration. The review by Khalajzadeh et al.(7) explored accessibility by assessing whether LCNC platforms that rely on visual interfaces accommodate users with visual impairments. Martinez et al.(6), on the other hand, examined LCNC usage in the manufacturing industry, highlighting synergies with lean construction, internet of things (IoT), industry 4.0, and building information modelling. Mosquera et al.(5) reviewed model-driven engineering (MDE), which differs from LCNC as it relies on formalized models to generate code and artifacts.

Building on these foundations, this study explores how recent technological advances in AI are transforming LCNC platforms by enhancing their capabilities and expanding their potential applications. It also aims to advance the discussion on LCNC development by providing a structured analysis of current research, emerging trends, and future directions, with a focus on how AI technologies are reshaping the capabilities and applications these platforms where the findings will inform both academic research and practical implementation strategies. Finally, it identifies research gaps and proposes directions for theory-informed inquiry and practical deployment making it, to the authors' knowledge, the first systematic review to comprehensively map the intersection of LCNC and AI.

METHOD

SLR is a research methodology that aims to analyze, identify, and collect research studies through a set of defined procedures, resulting in a close view of critical points in current knowledge and an examination of areas for further study.(8) The study employed a systematic literature review methodology guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework. This approach ensured a structured and transparent process across the four phases — identification, screening, eligibility, and inclusion which allows the research team to document each decision and maintain methodological rigor. Utilizing a combination of keywords stated below, the researchers extracted literature from Scopus by Elsevier, Elton B. Stephens Company (EBSCO), and the Association for Information Systems Electronic Library (AISeL). Scopus, EBSCO, and AISeL are reputable research databases with distinct specializations. Scopus is one of the largest citation databases, tracking research across various disciplines. EBSCO provides full-text access to peer-reviewed journals, e-books, and industry reports, covering fields such as business, healthcare, and education. AISeL (AIS eLibrary) specializes in information systems, digital transformation, and IT research. The keyword combination ((“low-code” OR “no-code” OR “low code” OR “no code”) AND (“artificial intelligence” OR “AI” OR “machine learning” OR “ML” OR “GenAI” OR “Generative AI” OR “Generative” OR “ChatGPT” OR “OpenAI” OR “LLM” OR “LLMs” OR “large language model”)) was utilized to extract published data from Scopus, EBSCO, and AISeL, which resulted in an initial set of 1 582 articles.

To minimize selection bias in the journal and conference proceedings review, all co-authors evaluated each paper for eligibility, cross-checking decisions during a consensus meeting. This collaborative approach ensured that no single author's preference disproportionately influenced the dataset and that the final selection represented a balanced, field-wide perspective. The authors checked each article's main contribution against the goals of the SLR—to be specific, how AI is integrated into LCNC platforms. Methodological quality was also considered, where applicable, to ensure clear input from the selected articles.

The initial extract yielded 1582 articles from SCOPUS, EBSCO and AISeL where initial review was conducted to assess eligibility based on the following exclusion criteria:

- Papers from the last five years were selected, as ChatGPT entered the mainstream in 2021.
- Research-in-progress was excluded to ensure that observations and syntheses were complete.
- Journals were limited to those published in English, given the researchers' current language skills (aside from Filipino).
- Reviews and books were excluded, with a focus placed on journal articles and published research.
- Only peer-reviewed articles were considered to ensure the highest quality of studies.
- Duplicate records were removed to avoid repetition of ideas and potential bias in the dataset.

The preliminary exclusion resulted in 376 articles considered as eligible articles for detailed screening to which the researchers reviewed the titles and abstracts of the articles to assess alignment with the subject and research questions. The review of title and abstract narrowed down to 135 articles. This was subsequently refined further through full-text manual review for relevance to the research questions and objectives. This process resulted in 60 peer-reviewed articles considered eligible which formed the final corpus for synthesis on the study.

For the final pool of articles selected, the researchers extracted key data required for the synthesis, including publication details (year, authors, journal or conference name), industry, phase of LCNC where AI was integrated, challenges, opportunities, theoretical framework, and key findings. This multi-phase process, supported by transparent inclusion and exclusion criteria, provides an open and reproducible method for identifying and examining existing research on AI and LCNC platform integration and evolution.

RESULTS

The results of this systematic literature review are presented in two parts. First, is the descriptive results below present the distribution and characteristics of these studies, followed by a thematic synthesis which provide a synthesized overview of the major areas of focus in LCNC–AI research

Descriptive Findings

A total of 60 peer-reviewed articles were included in the final analysis after the systematic screening and eligibility process described in the methodology. The PRISMA flow diagram (figure 1) summarizes the number of records identified, screened, excluded, and retained from the initial 1 582 articles retrieved from Scopus, EBSCO, and AISeL.

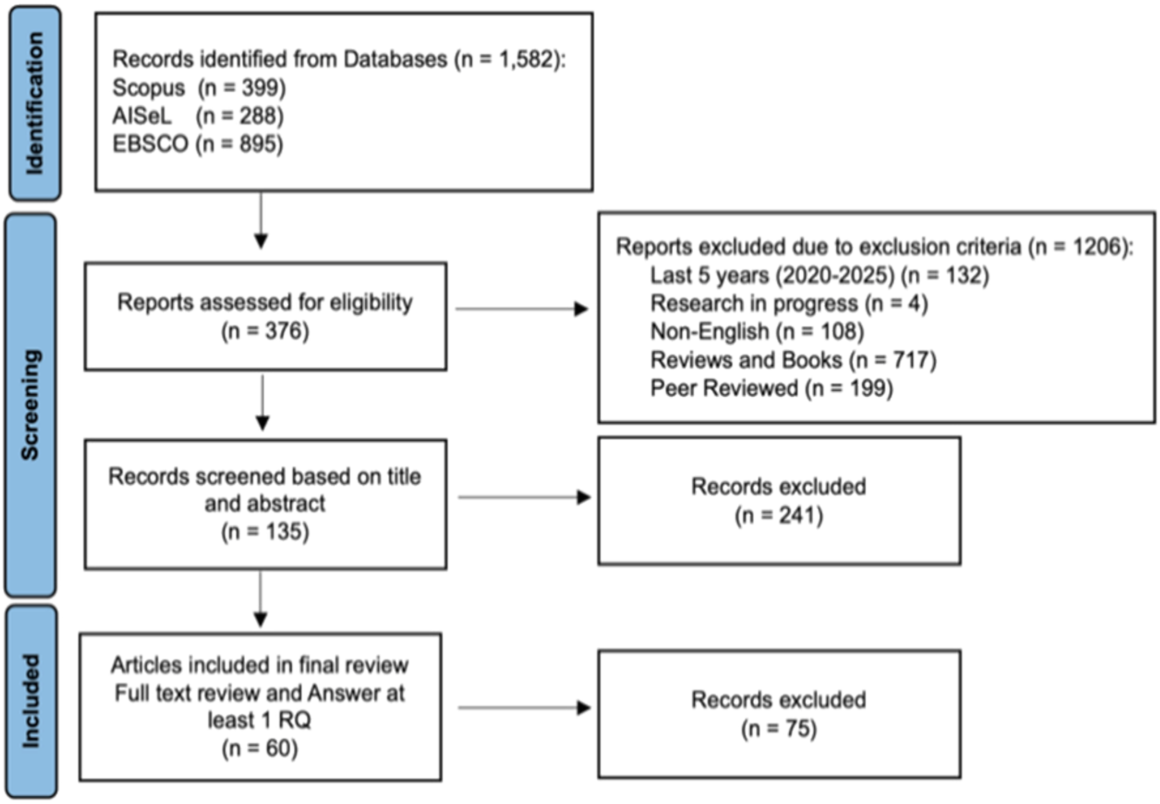

Figure 1. Literature search flow

These publications represent research conducted between 2020 and 2025 showing an increase in research interest in LCNC development and AI within the academic field. In 2021, two journal articles were published, but this number steadily increased over the years and peaked in 2024 with thirty-nine journal publications (figure 2). The year 2020 was included in the review; however, no journal articles were extracted for that year, as ChatGPT entered mainstream awareness in 2021. According to the study by Bisi et al.[9], following the release of ChatGPT v3.5 in November 2022, a significant increase in AI usage was detected in the journal Orthopaedics & Traumatology: Surgery & Research, using ZeroGPT as supporting evidence. This trend was further supported by the study of Ding et al.[10], which observed a rapid growth pattern in GenAI publications beginning around 2022.

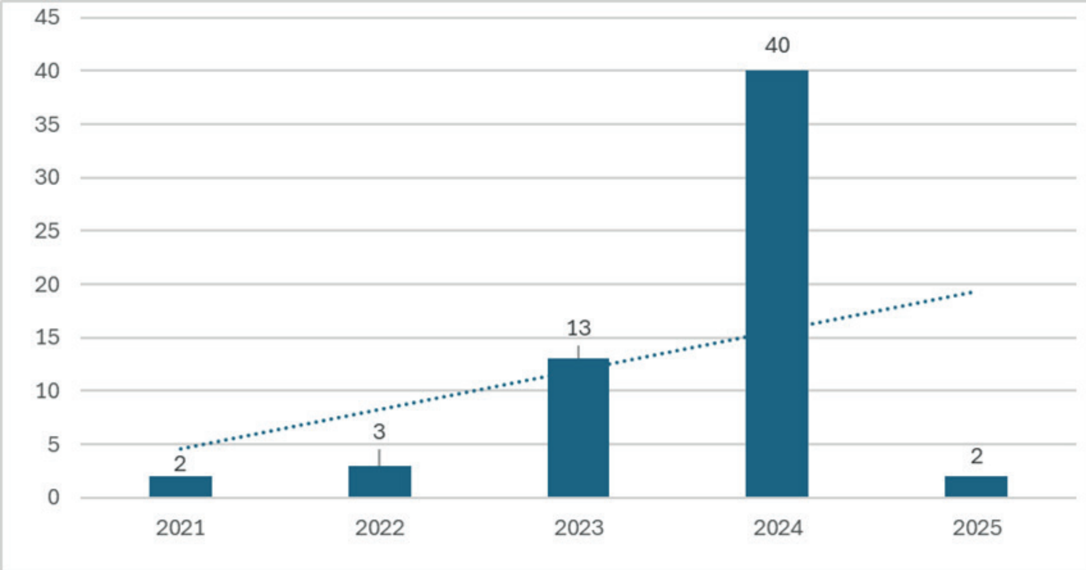

Figure 2. Number of publications per year

The descriptive results indicate a growing interest across disciplines, with studies appearing in journals related to information systems, computer science, and digital transformation. This upward trend demonstrates that the intersection of LCNC and AI is emerging as a significant field of inquiry in both academia and industry.

Thematic Findings

The Role of AI in different phases of LCNC development

LCNC platforms are increasingly integrating AI throughout the software development lifecycle. AI is being applied in requirements gathering and user interface prototyping, where generative AI assists in creating and arranging front-end elements with minimal manual effort.(11,12) Code generation and debugging are also supported by large language models (LLMs) that translate natural language instructions into underlying models or code, reducing the complexity of traditional development.(13,14) Many platforms incorporate automated quality checks and MLOps features, allowing non-technical users to deploy ML workflows while maintaining reliability and manageability.(15,16) These studies demonstrate how AI can optimize every phase of LCNC development, from ideation to deployment.

Numerous articles address more than one phase of the software development lifecycle. As such, when counting multiple mentions of different phases (meaning the mention percentages do not sum to 100 percent), Table 1 shows that most, if not all, studies include some form of AI and LCNC usage. Based on the data, it can be inferred that AI is primarily leveraged during the development phase. This suggests a growing trend where users are increasingly assisted by AI in building LCNC solutions. AI-enabled LCNC platforms also support natural language-to-code translation, allowing users to provide instructions in plain language, which the system then converts into functional backend logic and workflows.(16)

Table 1. Sdlc phase, activities ai was leveraged and mentioned

Phase	Activities	Mentions
Planning	Feasibility Analysis(19,20,21) Planning(18,22)	5 out of 60 (8 %)
Requirements Analysis	Requirements Gathering(13,17,18)	3 out of 60 (5 %)
Design	UI/UX Conceptualization(11,23) Architecture(13,24,25) Data Modeling(12,16,26)	7 out of 60 (12 %)
Implementation (Development & Coding)	Component Assembly(15,27,28) Data Integration(26,29,30) Code Generation(12,24,33) Debugging(16,28) Code Completion(12,16) Component Recommendation(23,34)	17 out of 60 (28 %)
Testing	Testing(14,31) Quality Assurance(14,32)	4 out of 60 (7 %)
Deployment	Deployment(14,15,35)	3 out of 60 (5 %)

Maintenance / Operations	Monitoring[15,27,36] Updates[18,28,37] Security Patches[20,38] Ongoing Security and Governance[29,38,39] Performance Optimization[33,34,36] User Support[18,32,40] Continuous Improvement[16,37,41]	21 out of 60 (35 %)

Industries leveraging AI for LCNC development

A variety of industries are tapping into AI-driven LCNC solutions, underscoring the broad applicability and versatility of these platforms. In healthcare, multiple studies highlight improved clinical decision support, workflow automation, and patient monitoring systems enabled through AI-enabled LCNC tools.[33,44] Within manufacturing, researchers demonstrate how AI-integrated LCNC platforms facilitate predictive maintenance, real-time monitoring, and streamlined production status assessment.[15,45] The logistics sector harnesses AI and LCNC for automating complex processes such as shipment tracking, warehouse management, and supply chain optimization, thereby reducing operational costs and errors.[18]

Beyond these domains, AI-enabled LCNC is also used in education, where it helps the creation of interactive learning modules, personalized assessment tools, and data analytics dashboards for improved decision-making at educational institutions.[11,46] In the legal sphere, AI-enhanced LCNC solutions support large-scale text analytics, contract review, and legal precedent analysis, contributing to faster research cycles and more consistent compliance checks.[47,48] Finance organizations benefit similarly by accelerating the development of risk analysis systems, customer-facing chatbots, and fraud detection pipelines with minimal coding.[26] The sports and fitness industry demonstrates how AI and LCNC can track performance metrics, automate data logging, and provide real-time feedback for athletes and coaches.[27]

These varied applications across healthcare, manufacturing, logistics, education, legal, finance, and sports/fitness illustrate the adaptability of LCNC platforms when integrated with AI. By reducing developmental overhead and empowering domain experts to rapidly prototype digital solutions, AI-driven LCNC environments have proven to be a robust choice for organizations seeking increased agility and cost-effectiveness across an array of sectors.[16,22]

Opportunities in implementing AI to LCNC development

The integration of AI into LCNC platforms presents various opportunities for organizations looking to optimize their business processes, increase agility, and reduce development overhead. First, business requirements can be captured more quickly and intuitively, thanks to AI-driven interfaces that transform natural language specifications directly into code or system workflows.[12,16] This capability, in turn, reduces manual documentation, since automatically generated outlines, diagrams, and specifications serve as living documentation throughout the development lifecycle.[23] Early validation of user flows is also more efficient with AI-enhanced prototypes, allowing stakeholders to see how an application might function before committing extensive resources to coding.[18]

AI also accelerates development cycles by providing real-time code suggestions and prompts based on best practices, reducing coding errors through automated checks and refactoring recommendations.[14,28] This guided assistance helps citizen developers gain confidence in building more complex applications.[32] Such systems support proactive issue resolution by identifying potential workflow bottlenecks, security vulnerabilities, or performance concerns before they escalate.[20,26] Moreover, AI enables continuous optimization of applications without requiring extensive manual intervention, allowing businesses to respond quickly to evolving needs and market demands.[15,43] Over time, these adaptive updates extend system lifespans and minimize downtime, ensuring consistent service delivery and maintaining competitive advantage.[27,44]

Challenges in implementing AI to LCNC development

Although AI integration in LCNC platforms offers numerous advantages, it also introduces significant complexities for both business and technical stakeholders. One key issue is that many LCNC users lack formal software development training, and the addition of AI—particularly large language models—can further complicate workflows.[16,18] For example, non-coders may struggle to interpret AI-generated responses accurately or formulate effective prompts, often leading to misaligned expectations.[37] The same research identifies a "lack of common vocabulary or technical understanding" as a factor that can hinder successful adoption.

AI outputs can be incomplete, biased, or outright incorrect (hallucinations), requiring careful oversight from domain experts to maintain accuracy and reliability.[20,34] The limited explainability of AI-generated designs presents additional risk, as end-users may not understand how or why certain system components were created.[32] Over-reliance on AI suggestions can also hinder developer skill growth and creativity, particularly if teams begin to depend on automated tools instead of learning fundamental coding principles.[14]

Data privacy emerges as a prominent concern when AI relies on cloud-based training or inference, raising questions about regulatory compliance and enterprise security standards.[29,38] Organizations often face integration complexities when aligning AI-driven LCNC solutions with existing backend systems or legacy software.[15] Robust governance is essential to ensure that automated updates comply with company policies and do not inadvertently introduce vulnerabilities.[39] The risk of AI-driven refactoring disrupting custom integrations underscores the need for a structured testing framework.[13,45] Finally, ongoing subscription and licensing costs for AI services, in addition to LCNC platforms, can present budgetary challenges, particularly for smaller firms without substantial IT financing,[26] as AI functions are typically offered as separate or add-on subscriptions.

LCNC development platforms leveraging AI

Several studies highlight how specific LCNC development platforms are integrating AI to expand their capabilities. For instance, Oracle APEX has been paired with large language models (LLMs) to let users modify fully functional web applications simply by chatting with an AI assistant.[37,42] This approach reduces the learning curve which enables "citizen developers" to focus on app logic rather than technical development. Meanwhile, Microsoft Power Platform provides AI-enabled modules for tasks such as sentiment analysis, anomaly detection, and process automation, allowing small businesses or non-technical staff to build advanced analytics solutions with minimal coding.[26] Similarly, OutSystems is used in contexts like sports performance management, where integrated AI functions handle data extraction and facilitate real-time feedback loops.[27]

Beyond traditional LCNC offerings, there is a growing trend toward "AI as a no-code platform" itself.[16,28] Newly emerging frameworks aim to handle entire multi-agent workflows or generate application code from plain English prompts, effectively blurring the boundary between coding and natural language input.[12] Researchers argue this shift can democratize AI development and accelerate digital transformation by empowering a larger pool of users with minimal programming background.[13,18] As these AI-enabled LCNC platforms evolve, it promises to streamline the software lifecycle from rapid prototyping to post-deployment maintenance, further driving adoption across industries seeking quicker, more adaptive solutions.

Theoretical Approaches in studying LCNC and AI integration

Only 2 of the 60 studies, or 3,33 %, explicitly mention a formal theoretical framework. Cuthbert and Pearse[48] adopt a contingent resource-based perspective to analyze how organizations strategically use low-code platforms, while Truss and Schmitt[49] employ a design science research (DSR) approach in developing a conceptual framework for human-centered AI product prototyping with no-code AutoML. The majority of papers focus on prototype development, engineering solutions, or case implementations without grounding their work in an established theoretical lens. While most studies emphasize practical demonstrations or system evaluations, a smaller subset incorporates theories to frame their research questions, methodologies, or findings.

DISCUSSION

Building on the descriptive results presented above, this section interprets the synthesized evidence in relation to the study's objectives and previous literature.

Integration of AI Across the LCNC Lifecycle

The reviewed literature highlights that AI integration across the LCNC lifecycle is expanding rapidly. AI assists in requirements gathering, user interface design, code generation, testing, deployment, and maintenance, effectively functioning as a co-developer. This convergence transforms LCNC platforms from visual programming tools into intelligent systems capable of learning and adaptation. As a result, software development has become more accessible and efficient, particularly for non-technical users. However, this growing reliance on AI also introduces challenges in interpretability and transparency which emphasizes the need for explainable systems and responsible AI design.

Industry Patterns and Theoretical Gaps

The synthesis reveals that AI-LCNC adoption is concentrated in industries with strong data infrastructures where automation and real-time analytics are essential. Yet, most studies approach the topic from an engineering or case-based perspective rather than a theoretical one. Only a few employ an established framework such as design science research [49] or the resource-based view.[48] This indicates a methodological gap that limits understanding of user motivation, organizational strategy, and socio-technical transformation. Future studies must integrate theoretical perspectives to connect LCNC adoption with innovation theory, digital transformation models, and human-AI collaboration frameworks.

Methodological and Governance Challenges

The integration of AI in LCNC platforms also presents governance and ethical challenges. Key issues include limited AI literacy among users, data privacy concerns, and the difficulty of monitoring AI-assisted workflows. Overreliance on automated recommendations may reduce creativity and critical thinking, while unclear accountability can complicate software validation. Furthermore, integrating AI modules—often developed by third parties—raises security and compliance issues. These challenges underscore the need for governance frameworks and educational programs that enhance digital literacy and promote responsible AI-LCNC development.

Remarks and Implications for Future Research

Summary

The synthesis of studies on LCNC development and AI integration reveals broad adoption across multiple industries, including healthcare, manufacturing, education, logistics, and the legal sector.[15,26] Researchers highlight how LCNC platforms integrate with AI to support various software development phases, from early UI prototyping and requirements gathering to code generation, testing, and post-deployment monitoring.[12,16] While formal theoretical frameworks such as design science research [49] and resource-based perspectives[48] are present in some studies, many articles adopt a more engineering-driven or case study approach rather than theory-oriented research.

Multiple benefits emerge from this integration of AI and LCNC such as faster development cycles, enhanced citizen-developer empowerment, and reduced coding errors through automated checks.[14,18] Platforms such as Oracle APEX, Microsoft Power Platform, and OutSystems incorporate AI modules to facilitate data analysis, user interface enhancements, and process automation.[26,27,47] It comes with its own challenges, however, including a general lack of AI literacy among non-technical users, potential data privacy issues, and governance complexities for organizations aiming to manage AI-driven changes.[34,38] The literature emphasizes the promise of AI-enabled LCNC platforms in accelerating innovation, reducing development backlogs, and expanding software creation to a wider range of stakeholders, yet also underscores the importance of robust oversight to ensure successful, sustainable implementations.[11,13]

Limitations

The scholarly work was primarily derived from SCOPUS, EBSCO, and AISeL, and was limited to completed, English-written journal articles and conference proceedings. As a result, non-English data, non-indexed journals, and in-progress work were excluded from the study, potentially omitting valuable perspectives from other contexts. Grey literature was also not considered, despite the abundance of works on LCNC and AI in such publications. This exclusion was due to the researchers' focus on peer-reviewed journals, which may limit the observation of emerging or practice-focused innovations in LCNC and AI. The studies collected were further limited to a five-year scope, reflecting the relevance of recent developments in LCNC and AI integration, as AI gained mainstream attention following the launch of ChatGPT by OpenAI in 2022;[9,10] while AI and LLMs were mentioned in 2021, no relevant studies were found for 2020.

Gaps and Agenda for Future Research

Several studies have addressed how AI influences LCNC development, but empirical validation of AI's effectiveness across pre-, during, and post-development phases remains scarce.[14,16] Clarifying the actual benefits of AI integration is crucial to determine whether industries genuinely reap productivity and cost-savings gains or if they are merely following a trend.[13] Future research could apply controlled experiments or longitudinal studies to pinpoint the phase(s) where AI delivers the most measurable impact, thereby guiding practitioners on where to concentrate AI resources.[12,37]

Theoretical underpinning also emerges as a clear gap based on the observation that, despite the large number of journals gathered, only a few authors explicitly applied theoretical frameworks.[48,49] This suggests opportunities for further research using established lenses such as the resource-based view, self-determination theory, or technology acceptance models to investigate AI-integrated LCNC adoption, motivation, barriers, and outcomes. A more systematic application of theory could yield deeper insights into user motivation, governance, and strategic alignment, as noted by Binzer et al.[29] There is also a noticeable lack of empirical evidence[3] on the benefits and structured governance models necessary for effectively implementing LCNC.

Industry-specific explorations also remain limited, particularly in the banking sector, which is frequently identified as a leading user of LCNC platforms. Future research could examine risk assessment, trading, and fraud detection solutions built on LCNC platforms with AI capabilities, with a focus on sector-specific best practices.[26] Further studies may also investigate how ongoing AI integration is reshaping the barrier to entry for citizen developers—potentially lowering skill requirements while introducing new governance and compliance challenges.[38,39] Addressing these gaps could lead to a richer, more theory-informed understanding of how AI-enabled LCNC influences organizational innovation.[50]

CONCLUSION

This study aimed to systematically analyze how AI integrates into LCNC development platforms and to identify their collective implications for software development. The findings indicate a structural transformation in software development rather than a temporary technological trend. It signifies a shift from human-exclusive programming to collaborative intelligence, wherein AI supports creativity, problem-solving, and accessibility in digital production. Beyond technical advantages, this integration reshapes how organizations approach innovation, skill development, and governance. AI-enabled LCNC platforms blur the boundaries between developers and end-users, encouraging inclusiveness in the development process but as a result also requires new ethical and institutional safeguards. Understanding this convergence not only advances the academic discourse on AI-driven development but also informs policies and practices for a sustainable digital future.

REFERENCES

1. Elbatanony A, Succi G. The pareto distribution of software features and no-code. In: Proceedings of the 1st ACM SIGPLAN International Workshop on Beyond Code: No Code; Oct 2021. p. 18-22.

2. Lichtenthäler R, Böhm S, Manner J, Winzinger S. A use case-based investigation of low-code development platforms. In: Proceedings of ZEUS; Feb 2022. p. 76-83.

3. Ajimati MO, Carroll N, Maher M. Adoption of low-code and no-code development: A systematic literature review and future research agenda. J Syst Softw. 2025;222:112300.

4. Sufi F. Algorithms in low-code-no-code for research applications: A practical review. Algorithms. 2023;16(2):108.

5. Mosquera D, Ruiz M, Pastor O, Spielberger J. Understanding the landscape of software modelling assistants for MDSE tools: A systematic mapping. Inf Softw Technol. 2024;173:107492.

6. Martinez E, Pfister L. Benefits and limitations of using low-code development to support digitalization in the construction industry. Autom Constr. 2023;152:104909.

7. Khalajzadeh H, Grundy J. Accessibility of low-code approaches: A systematic literature review. Inf Softw Technol. 2025;177:107570.

8. Carrera-Rivera A, Ochoa W, Larrinaga F, Lasa G. How-to conduct a systematic literature review: A quick guide for computer science research. MethodsX. 2022;9:101895.

9. Bisi T, Risser A, Clavert P, Migaud H, Dartus J. What is the rate of text generated by artificial intelligence over a year of publication in Orthopedics & Traumatology: Surgery & Research? Analysis of 425 articles before versus after the launch of ChatGPT in November 2022. Orthop Traumatol Surg Res. 2023;109(8):103694.

10. Ding L, Lawson C, Shapira P. Rise of generative artificial intelligence in science. arXiv preprint. 2024. Available from: https://arxiv.org/abs/2412.20960

11. Shan Q. Design and application of a web front-end development course training platform based on generative artificial intelligence and low code development. In: Proceedings of the 2nd International Conference on Education, Knowledge and Informatization; July 2024. p. 287-91.

12. Hagel N, Hili N, Schwab D. Turning low-code development platforms into true no-code with LLMs. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems; Sept 2024. p. 876-85.

13. Noble J. Automatic programming vs. artificial intelligence. In: Proceedings of the 1st ACM International Conference on AI-Powered Software; July 2024. p. 144-6.

14. Choudhury D, Gupta D. Investigation on integration of machine learning techniques into LC/NC platforms for code review, quality assessment, and error detection automation. In: Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)

(ICRITO); Mar 2024. p. 1-7. IEEE.

15. Grilo A, Figueiras P, Rêga B, Lourenço L, Khodamoradi A, Costa R, et al. Data analytics environment: Combining visual programming and MLOps for AI workflow creation. In: Proceedings of the 2024 IEEE International Conference on Engineering, Technology, Innovation (ICE/ITMC); June 2024. p. 1-9. IEEE.

16. Rao N, Tsay J, Kate K, Hellendoorn V, Hirzel M. AI for low-code for AI. In: Proceedings of the 29th International Conference on Intelligent User Interfaces; Mar 2024. p. 837-52.

17. Rogozov YI, Lapshin VS, Borovskaya MA. Approach to the implementation of intelligent low-code platforms. In: Proceedings of the International Conference on Intelligent Information Technologies for Industry; Oct 2022. p. 42-50. Cham: Springer International Publishing.

18. Martins J, Branco F, Mamede H. Combining low-code development with ChatGPT to novel no-code approaches: A focus-group study. Intell Syst Appl. 2023;20:200289.

19. Kedziora D, Aunimo L, Kortesalmi H. Between no-code, low-code, custom-code, and AI. Hyperautomation strategies at Nordic enterprises. In: Proceedings of the Machine Intelligence and Digital Interaction Conference; Dec 2023. p. 172-9. Cham: Springer Nature Switzerland.

20. Kandaurova M, Skog DA, Bosch-Sijtsema PM. The promise and perils of low-code AI platforms. MIS Q Exec. 2024;23(3):275-89.

21. Carroll N, Holmström J, Matook S. Special issue editorial: Transforming business with low-code and no-code. MIS Q Exec. 2024;23(3):v-xiii.

22. Novales A, Mancha R. Fueling digital transformation with citizen developers and low-code development. MIS Q Exec. 2023;22(3):221-34.

23. Choi J, Park S, Koh G, Kim Y, Jeong WK. Intuitive design of deep learning models through visual feedback. In: Proceedings of the 2024 IEEE Visualization and Visual Analytics (VIS); Oct 2024. p. 111-5. IEEE.

24. Dong Y, Kong L, Zhang L, Wang S, Liu S, Chen M. A search-and-fill strategy to code generation for complex software requirements. Inf Softw Technol. 2025;177:107584.

25. Ike, BarefootDev. Shackled: A 3D rendering engine programmed entirely in Ethereum smart contracts. In: Proceedings of the International Conference on Blockchain; Dec 2022. p. 131-43. Cham: Springer Nature Switzerland.

26. Kok CL, Tan HR, Ho CK, Lee C, Teo TH, Tang H. A comparative study of AI and low-code platforms for SMEs: Insights into Microsoft Power Platform, Google AutoML and Amazon SageMaker. In: Proceedings of the 2024 IEEE 17th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC); Dec 2024. p. 50-3. IEEE.

27. Oliveira J, Nicola S, Graça P, Martins S, Gafeira T. Application for the management of sports performance in CrossFit supported by an artificial intelligence cognitive service. In: Proceedings of the International Conference on Innovation in Bio-Inspired Computing and Applications; Dec 2021. p. 590-8. Cham: Springer International Publishing.

28. Dibia V, Chen J, Bansal G, Syed S, Fourney A, Zhu E, et al. AUTOGEN STUDIO: A no-code developer tool for building and debugging multi-agent systems. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2024; Miami, FL, USA. p. 72-9. Association for Computational Linguistics.

29. Binzer B, Fürstenau D, Till T, Winkler T. Establishing a low-code/no-code-enabled citizen development strategy. MIS Q Exec. 2024;23(3):253-73.

30. Liu C, Downing C. Teaching tip using text analytics AI insights in Microsoft Power BI Desktop to score sentiments, extract key phrases, and discover unstructured data patterns. J Inf Syst Educ. 2024;35(1):48-55.

31. Martínez-Lasaca F, Díez P, Guerra E, de Lara J. LowcoBot: Towards chatting with low-code platforms. In: Proceedings of the First Workshop on Large Language Models for Model Driven Engineering (LLM4MDE 2024); 2024. p. 66-76, vol. 3727.

32. Versino S, Turchi T, Malizia A. Democratizing AI: Expert-tested VPL-based prototype to foster participation. In: Proceedings of the 2024 International Conference on Advanced Visual Interfaces; June 2024. p. 1-3.

33. Ashraf AR, Somogyi-Végh A, Merczel NS, Gyimesi N, Fittler A. Leveraging code-free deep learning for pill recognition in clinical settings: A multicenter, real-world study of performance across multiple platforms. Artif Intell Med. 2024;150:102844.

34. Demir MA, Ölmez E, Yıldız U, Konyar MZ, Akbulut O. Node recommendation system utilizing NLP models on no-code platforms. In: Proceedings of the 2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom); June 2024. p. 296-9. IEEE.

35. Phoo-In J, Leelasantitham A, Sukamongkol Y. Development of low-code platform for electrical power trading using blockchain and artificial intelligence. In: Proceedings of the 2024 5th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON); June 2024. p. 1-5. IEEE.

36. Anzai TK, Furtado PHT, de Brito GM, Santos JS, Moreira PCM, Diehl FC, et al. Catching failures in 10 minutes: An approach to no code, fast track, AI-based real time process monitoring. Presented at the Offshore Technology Conference Brasil; Oct 2023; Rio de Janeiro, Brazil.

37. Gorissen SC, Sauer S, Beckmann WG. Supporting the development of Oracle APEX low-code applications with large language models. In: Proceedings of the International Conference on Product-Focused Software Process Improvement; Nov 2024. p. 221-37. Cham: Springer Nature Switzerland.

38. Viljoen A, Radic M, Hein A, Nguyen J, Krcmar H. Governing citizen development to address low-code platform challenges. MIS Q Exec. 2024;23.

39. Biedova O, Ives B, Male D, Moore M. Strategies for managing citizen developers and no-code tools. MIS Q Exec. 2024;23(2):165-83.

40. Grashoff I, Recker J. How GuideCom used the Cognigy.AI low-code platform to develop an AI-based smart assistant. MIS Q Exec. 2024;23(3):325-44.

41. Paliwal G, Donvir A, Gujar P, Panyam S. Low-code/no-code meets GenAI: A new era in product development. In: Proceedings of the 2024 IEEE 8th Ecuador Technical Chapters Meeting (ETCM); Oct 2024. p. 1-9. IEEE.

42. Gorissen SC, Sauer S, Beckmann WG. A survey of natural language-based editing of low-code applications using large language models. In: Proceedings of the International Conference on Human-Centred Software Engineering; July 2024. p. 243-54. Cham: Springer Nature Switzerland.

43. Liu JW. A guide to network meta-analysis using generative AI and no-code tools. Hu Li Za Zhi. 2024;71(5):29-35.

44. Gonzalez C, Badr Z, Güngör HC, Han S, Hamdan MD. Identifying primary proximal caries lesions in pediatric patients from bitewing radiographs using artificial intelligence. Pediatr Dent. 2024;46(5):332-6.

45. Schenkenfelder B, Salomon C, Schwandtner M, Zefferer R, Derfler M, Wimmer M. Leveraging the power of LLMs to transform robot programs into low-code. In: Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA); Sept 2024. p. 1-4. IEEE.

46. Pastierik I, Kvet M. Exploring Oracle APEX for the university data analysis. In: Proceedings of the 2023 21st International Conference on Emerging eLearning Technologies and Applications (ICETA); Oct 2023. p. 395-402. IEEE.

47. Cho K, Han S, Choi YR, Hwang W. NESTLE: A no-code tool for statistical analysis of legal corpus. arXiv

https://doi.org/10.56294/dm20251218 ISSN: 2953-4917

preprint. 2023. Available from: https://arxiv.org/abs/2309.04146

48. Cuthbert CE, Pearse NJ. Strategic use of low code platforms. In: Proceedings of the International Conference on Education and Information Systems, Technologies and Applications (EISTA); vol. 1, 2021. p. 978-1.

49. Truss M, Schmitt M. Human-centered AI product prototyping with no-code AutoML: Conceptual framework, potentials and limitations. Int J Hum-Comput Interact. 2024;1-16.

50. Alt R, Zimmermann HD. Towards AI application marketplaces—An interview with Dorian Selz. Electron Markets. 2022;32(1):139-43.

FINANCING
The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST
The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION
Conceptualization: Gabriel Luis L. Liwanag.
Data curation: Gabriel Luis L. Liwanag.
Formal analysis: Gabriel Luis L. Liwanag.
Research: Gabriel Luis L. Liwanag.
Methodology: Gabriel Luis L. Liwanag.
Project management: Gabriel Luis L. Liwanag.
Resources: Gabriel Luis L. Liwanag.
Software: Gabriel Luis L. Liwanag.
Supervision: Ryan A. Ebardo, Danny C. Cheng.
Validation: Ryan A. Ebardo,Danny C. Cheng.
Display: Gabriel Luis L. Liwanag.
Drafting - original draft: Gabriel Luis L. Liwanag.
Writing - proofreading and editing: Ryan A. Ebardo, Dr. Danny C. Cheng.