Frontiers in Computer Science and Artificial Intelligence
DOI: 10.32996/fcsai
Journal Homepage: www.al-kindipublisher.com/index.php/fcsai

FCSAI
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Role of Low-Code/No-Code Platforms in Engineering Software Development: Democratizing Simulation and Design

Sarmi Islam
Independent Researcher, Eden Mohila College, Dhaka
Corresponding Author: Sarmi Islam ,**E-mail**: Sormiislam571@gmail.com

| ABSTRACT

The momentum of lowcode and no-code (LCNC) platforms has ushered a new era in engineering software development, empowering non-programmer engineers, designers and domain experts to contribute to simulation, automation and product innovation. In this work we explore how LCNC platforms drive engineering design democratization by reducing technical barriers, shortening development cycles, and stimulating interdisciplinary collaboration. Using systematic literature review and conceptual analysis methods, its aim is to investigate the transformative potential of LCNC tools for encapsulating simulation, computational modelling and AI-based workflows in accessible, visual interfaces. And it indicates how these platforms foster productivity growth, diminish the need for specialised programming expertise and facilitate inclusive innovation at scale among both small- and large-scale businesses. However, the paper also presents challenges that are emerging like model validation scavenging and data interoperability and potential risks of governance in specific to safety critical engineering domains. The results highlight the potential for LCNC methods to supplement—not substitute—conventional programming idioms with appropriate design guidelines, domain specific templates and responsible automation. At the end, in this paper we argue for a human-centered LCNC ecosystem that enables engineers to concentrate on creativity and problem solving, while ensuring transparency, accountability, and technical integrity in engineering software development.

| KEYWORDS

AI-Driven Cybersecurity, Intrusion Detection Systems, Adversarial Machine Learning, Explainable Artificial Intelligence (XAI),

| ARTICLE INFORMATION

ACCEPTED: 02 November 2024 **PUBLISHED:** 25 November 2024 **DOI:** 10.32996/fcsai.2022.1.1.3

1. Introduction

1.1 Background and Rationale

Software development with an engineering focus is classically dominated by high-code, algorithmic and script based environments requiring advanced programming skills and domain knowledge (Hinsen, 2021). MATLAB, ANSYS, COMSOL Multiphysics and SolidWorks have been the workshorses of mechanical, civil, and systems engineering design and simulation for many years. But, as engineering challenges become more complex (across digital twins, AI-augmented modeling, and smart manufacturing) this is far too long-the demand for more accessible, agile ways of creating software has never been greater. This tension has driven the evolution of (LCNC) platforms, which allow users to build apps and processes through visual interfaces, drag- and-drop tools and pre-built logic instead of traditional coding (Gartner, 2023).

And LCNC platforms have become a significant driver of democratized innovation, enabling non-programmer engineers, technicians and researchers to implement modest applications without extensive programming expertise (González-Huerta et al., 2023). Through abstraction of the programming, experts in other fields can instead concentrate on modeling and simulation logic and decision-making without having to deal with syntax details such as variables or parameter types (Kärkkäinen et al., 2022). From the engineers' perspective, this change is not one involving only a technical advance, but instead a socio-technical revolution – they allude to design and simulation that are more inclusive in scope and interactive (i.e., iterative across spatial scales) in practice as well as interdisciplinary (Shah & Kusiak, 2024).

1.2 The Low-Code/No-Code Paradigm

With low-code, users can develop applications with very little hand-coding involved whereas no-code gives them complete visual development process environment. According to Forrester Research (2023) AGPLatforms provides a development time savings of between 70-90% and offer more agile software lifecycles. They can serve as modeling tools in engineering applications, for parametric simulation generation, and for workflow automation in the context of modular visual programming (Zhao et al., 2022).

Well-known LCNC tools such as MATLAB App Designer, Simulink, Altair Compose and new open source frameworks including Node-RED and PyCaret GUI demonstrate that graphical environments can also stretch their capability to advanced areas like computational fluid dynamics (CFD), structural analysis, industrial control applications (such as robotics) and force measurement (Rantanen & Kuusisto, 2023). Further, cloud-integrated LCNC ecosystems (e.g. Microsoft Power Apps, Mendix and OutSystems) enable engineers to connect design models with enterprise systems, IoT equipment and data analytics pipelines, closing the gap between engineering and operational technologies (Gartner, 2023).

The integration between LCNC systems and engineering software can be seen as a milestone of change, the age of digital democratization, when the distinction among developer, designer, and user fades away enabling a wider participation in processes for engineering innovation (González-Huerta et al., 2023).

1.3 LCNC in Simulation and Design: Research Opportunities

It is the faster prototyping, parametric tests and decision supports in simulation-based engineering by LCNC platforms. For instance, the conditions for a digital-twin dashboard or automated finite-element (FE) simulations can be developed through graphical-interface-based drag-and-drop tasks with no need to code elaborate scripts (Kumar et al., 2022). The addition of AI-driven modules, like predictive maintenance analytics or generative design tools, can extend LCNC capabilities to real-time decision making (Shah & Kusiak, 2024).

1.4 Challenges and Limitations

However, LCNC platforms have at least three challenges in engineering software applications. Model validation and verification is one of the crucial issues - to make sure that simulation logic implemented using visuals follows engineering standards and yields reliable results (Kärkkäinen et al., 2022). Critically overusing prefabricated modules might end in oversimplification of highly intricate systems or on unforeseen design errors (Hinsen, 2021).

Scalabilility and integration are other significant challenges. Engineering projects tend to run computationally intensive and are often not fully supported by LCNC platforms, that need to be compatible with legacy systems (González-Huerta et al., 2023). Moreover, data governance, cybersecurity and intellectual-property (IP) protection are also emerging risks which are heightened by LCNC applications connecting to cloud-based infrastructures (Forrester Research, 2023).

There is perhaps a cultural and skill barrier too: although LCNC reduces the technical barriers, it calls for engineers to acquire new skills with regard to logic mapping, interface design, systems-thinking. The risk of "citizen developers" building shadow applications inside corporate walls without the oversight of professionals, is, however, a topic that still keeps many orders-ethical and -managerial people waking at night worried (Rantanen & Kuusisto 2023).

Hence, democratization by LCNC must be underpinned by robust governance mechanisms, regulations and criteria of quality assurance and interdisciplinary training systems.

1.5 Research Aim and Objectives

This paper intends to question low-code and no-code platforms in engineering software development, and more specifically their ability to democratize simulation and design. The objectives are to:

Analyze how LCNC aligns with engineering software workflows and user engagement;

Study the opportunities and bottlenecks of LCNC in design, simulation and automation in engineering;

Articulate a model that characterizes responsible incorporation of LCNC into engineering.

By answering these research questions, the study aims to contribute to the current debate on human-centered digital transformation in engineering as well as provide a guide for an informed implementation of LCNC.

1.6 Structure of the Paper

The rest of this paper is organized as follows: Section 2 provides a comprehensive literature survey on LCNC platforms and their application for engineering domains. Section 3 presents methodology, including the analytic model that was used for synthesis. The next section (§4), we show our results and visualizations that characterize LCNC trends in engineering software. Section 5 presents some findings with technology, ethical and organisational implications. Section 6 concludes with implications for industry practice, policy and future research.

2. Literature Review

2.1 Overview of Low-Code/No-Code (LCNC) Platforms

Low code/no code (LCNC) development platform is an idea that has been popularized in recent years. LCNC implementations tend to minimise the need for, or eliminate manual programming through features such as Graphical User Interfaces (GUIs), visual modelling of workfl ows, drag and drop nodes, reusable components/shapes, model-driven code generation (SAP 2024). SAP These tools claim to democratizing software construction allowing “citizen developers” (domain exper ts who are not programmers) to participate in the actual building of applications (Gartner, 2023; González-Huerta et al., 2023; Insfrán & Fernández, 2023).

Ajimati et al conducted a systemic review to ascertain the sexuality of cerebral palsy in Nigerian children and found that all but one reported at least some sexual activity. (2024) demonstrated that LCNCis an “emerging technology trend” expanding software development out from professional software engineers and placing problem solving within the reach of more people. ScienceDirect Earlier literature (e.g., Hedau & Mandge, 2021) described the evolution of LCNC from earlier RAD tools to platforms that supported more "enterprise -worthy" development. IRJMETS

In general, the LCNC paradigm is defined by three properties:

Logic brings high order of abstraction(less code).

Visual/interface design for workflows and apps.

Facilitating domain experts build software or prototypes fast.

These underlying qualities set the stage for how LCNC might apply to programs engineering in particular.

2.2 LCNC in Software Engineering: Benefits and Challenges

The potential benefits of LCNC have been evaluated by software engineering research and at the same time its limitations. With respect to benefits, LCNC platforms allow:

• Reduction of project delivery time and quick prototyping (Heuschkel, 2023). arXiv

SAP Value TCO reduction vs. less dependance, agility increase Cost Savings Headcount reduction Less special forces *additional Workforce optimization 24% of savings over non-simplification-systems Mezzanine before 4 resources provided to landlord-simplification and standardization (Resource estimation) Less SRD-Rate per application tier resource Power reduction Vendor costs –less inhouse-Vendor Application Landing Zones SAP Transformation Road Map ©2016 SAP SE or an SAP affiliate company. SAP

• Software development converge to go further as a global recent- concern with FSMPs, already are – unless you were sleeping at the time, or received legal judgment against them. eleken. co

However, limitations emerge consistently:

– Scalability and performance issues when applications developed by LCNC platforms need to handle large scale or complexity. ấ 247 For instance, Cui (2024) demonstrates that while LCNC can lead to efficient delivery, it may pose constraints on the flexibility and maintenance of complex activities. SSRN

• Some of this challenges include Governance, security, integration and vendor-lock-in issue are critical issues that has still not been solved (Ajimati et al., 2024). ScienceDirect

• The hybrid model of "citizen developers + professional developers" tends to generate role-confusion, ambiguity w.r.t. responsibility and proliferation of "shadow IT (Upadhyaya, 2024). ResearchGate

There is a reason of why the software-engineering literature would argue that, while LCNC are highly promising, their adoption in mission-critical or domain-specialist systems requires a good governance and alignment with professional norms.

2.3 LCNC within the Engineering Environments (Simulation, Design and Automation)

Most of the LCNC literature is related to business-related fields, but interest in engineering-domain applications— especially simulation, design automation and IoT/industrial scenarios — is emerging. For example, Ren et al. (2022) presented an article titled "SeLoC-ML: Semantic Low-Code Engineering for ML Applications in Industrial IoT" that showed a low-code platform had made engineers able to prototype ML applications in IIoT faster (three-times less engineering effort estimation than traditional). arXiv In the same vein, Curty Härer and Fill (2022) examined blockchain app development with low-code platforms, concluding that also engineering domains could benefit from LCNC when using domain specific templates or model driven engineering. arXiv

These works show that for engineering purpose:

• Domain-specific templates and model-driven abstractions are key (not generic LCNC but particular to simulation/design).

• The complexity of engineering models (multi-physics, design optimisation, simulation workflows) still may present a barrier in terms of what pure LCNC platforms can reliably handle without additional custom code or domain expertise.

• Is the closer involvement of engineer disciplines (as opposed to only software developers) a possibility for more innovation and agility?

However, general empirical data concerning engineering-software (computer-aided design, computer-aided engineering (CAD/CAE), simulation and design automation) used with LCNC are scarce—thus creating a gap.

2.4 Democratisation and Organisational Implications

One of the promises of LCNC is that software development will be democratised, giving power back to domain experts while freeing up bottlenecks in IT and speeding up innovation cycles. This is of great relevance for engineering organisations where domain experts (engineers, simulation specialists, designers) often depend on specialist software developers or vendors. Some of the potential benefits of LCNC include:

• Alleviating bottleneck for developers and letting domain experts to either prototype or automate the (business) process themselves" (Kärkkäinen, Porras & Kiljander, 2022).

• Promoting cross-functional relationships between engineers, data scientists and operations with visual, shared workflows.

• Reducing barriers to entry for smaller companies or SMEs having a lower level of resources for software development.

Yet the organisational literature raises a number of caveats:

• Governance risk: the rise of people from different field moving into domain of citizens developers with less knowledge on software engineering based practices (Upadhyaya, 2024). ResearchGate

• Dependence on LCNC platforms may not be maintainable in the long term, or accumulate "technical debt" since abstractions made can obscure underlying complexity or lack of extensibility (Cui, 2024). SSRN

• Cultural and skill-shift challenges: New literacies (logic mapping, workflow design, UX) vs. classic simulation/domain knowledge that engineers have to embrace.

Hence the organisational level reveals that democratisation does not come automatically but needs managing, training and there is role overlap.

2.5 Ethical, Quality and Governance Considerations

In the context of (engineering) simulation or design automation (that are typically upstream to safety-critical systems as in Are Engineers Designing their Systems Safely? [16]); LCNC also poses concerns related to quality, accountability, traceability and ethical issues. Critical constructs identified within the literature are:

• Quality and reliability: Software systems for engineering applications are often characterized by high fidelity, v&v considerations (verification & validation), certification, domain-specific correctness. An LCNC platform hides code, but it may also hide the internal logic and you don't really know what assumptions, validations or black boxes has given us (Heuschkel, 2023). arXiv

• Governance and traceability: the "citizen developer" model might decrease transparency, for instance when holding someone accountable on who created software artefacts or how changes were traced (or audited). This is especially a challenge in regulated engineering fields.

• Integration and inter-operability: Software for engineering typically includes several tools (CAD, CAE, PLM, IoT). This emphasis has increased the importance of compatibility, data integrity and safe integration between LCNC platforms; incompatibilities may introduce risk at system level.

While there is little direct study of LCNC in engineering ethics, the intersection of LCNC with fields in which safety, and precision and domain expert oversight are critical suggest that research on based technologies has a clear urgency.

2.6 Research Gaps and the Implications for Engineering Software Development

From the above review, several gaps are to be noted:

Domain-specific empirical: Although the accumu of literature on LCNC is increasing, few studies exist that examine LCNC in practice when technology and manuals are applied to engineering simulation/design workflows inside real organisations (González-Huerta et cabr2023).

Quality, validation and safety-critical engineering: Software-engineering level (synthesis) factors for LCNC platforms are higher quality (rightsizing), due to the demands of engineering software (performance, accuracy, certification); there is little in the literature concerning how LCNC meets these.

Professional role change in engineering settings: Little is known about how the roles of engineers are transformed in becoming citizen developers of simulation/design tools, with reference to accountabilities and skills-trajectories.

Governance of LCNC in engineering software How to govern the development of LCNCs within engineering settings (e.g., audit trails, responsibility, integration with business systems) is under-explored.

(3) Scalability, integration, and maintainability: Business application of LCNC has been covered in many works but relatively few have addressed scaling LCNC to a complex engineering environment or maintaining such systems over their lifecycle.

This review suggests that your work—drilling into the democratisation of simulation and design through LCNC in engineering software—is quite timely and much needed.

2.7 Theoretical Implications of the Study

From the literature, several implications at the conceptual level can be derived for your study:

• Democritisation lens: The idea that domain experts can develop the simulation or design software themselves rather than computer scientist/software engineers inverts the traditional software engineering model to a hybrid one (domain + citizen developer + platform).

• LCNC's value in engineering: "I believe LCNC can help speed the iteration process, connect various disciplines and provide a renaissance to innovation.

• Framework development: Given the sparse literature related to engineering-specific LU and low volume of EI SOCs, your investigation could develop a customised framework connecting LCNC adoption, engineer empowerment, tool governance (software development) and design outcomes.

This review of literature has examined the history and present form of LCNC setups, their pros and cons in software engineering, as well as what appears to be an early adoption of them within the context of engineering simulation/design. It has named the organisational, ethical and governance aspects of democratising software development. Importantly, it has identified clear holes—at the very least in this context—particularly with respect to employing LCNC within high-fidelity engineering software domains that require tight integration and a degree of professional verification. These voids allow your framework to both examine and enrich the emergent space.

3. Methodology

3.1 Research Design

The research used a qualitative explorative approach. Qualitative methods are suitable to investigate under-researched phenomena, and yield rich context sensitive findings and from which frameworks and ideas rather than fixed hypotheses emerged (Bhandari, 2020; Scribbr, 2020). Solid qualitative methodologies make it possible to understand the how and why of an emerging phenomenonn— in this case, the take-up of low-code/no-code (LCNC) platforms in engineering software development.

Moreover, this study is inspired by a design-science rationale: it aims to contribute not only to explaining the phenomenon of democratising simulation and design but also to providing a blueprint for responsible incorporation of LCNC systems. Design-science research focuses on the development of an artefact (here, a framework) and supporting literature and systematic analysis.

3.2 Research Questions

The research is guided by this set of issues:

How/how much of LCNC platforms are being incorporated in the engineering software development for simulation and design?

What are the advantages and limitations of engineers to leverage an LCNC tool in simulations/design flows?

What are the conceptual ingredients for a responsible LCNC ecosystem in engineering software settings?

3.3 Source of Data and Sampling Strategy

Since this is an explorative and conceptual study, secondary data are central to the research. The sources include:

Pro and tech white papers and industry reports (Gartner, Forrester etc) covering LCNC tool adoption and engineering software trends.

• Policy papers, standards and guides on software governance as well as software engineering tools and practices in an engineering context.

A systematic review was performed through databases such as Scopus, Web of Science, IEEE Xplore and Google Scholar. Key search terms included:

(“low-code” OR “no-code”) AND (“engineering software” OR “simulation design tool”) AND (“engineering development” OR “design automation”) AND (“challenges” OR “adoption”).

Inclusion criteria:

• English publications between 2015 and 2024 were included in our analysis.

• Destroy explicitly for LCNC-platform based work, or utilisation of LCNC within the engineering/design/simulation context.

• Benefits, limitations, governance or organizational implications specific to engineering software development.

Exclusion criteria:

• Technical papers on high-code technical software development that do not mention LCNC, or engineering simulate.

• Non-engineering context (e.g., LCNC that has been solely applied to a business setting) unless lessons applicable for engineering can be clearly transferred.

3.4 Data Analysis

The transcripts were thematically content analysed (55, 77) and synthesized to derive conceptual categories. The process entailed:

Open coding: Analysis of each article/report to shape meaningful chunks (e.g., "citizen engineer usage", "complexity limitation","governance gap") and assign codes to them.

Axial coding: Categorising and interpreting codes into overarching themes— democratisation, technical feasibility, organisational readiness, governance & risk etc.).

Selective coding and integration: Establishing relationships among themes to represent a conceptual framework that explains factors, mechanisms and boundary conditions.

The design-science lens provided a focus for the abstraction from empirical results to artefact. Trustworthiness was considered by the analysis through triangulation (different source types), reflexivity (participating in awareness of research stance) and ensuring an audit trail of inclusion/exclusion decisions.

3.5 Framework Development

A conceptual framework for the responsible uptake of LCNC platforms in engineering software development is sketched based on the findings. The framework delineates core dimensions (e.g., access, verification, scale, governance) and hypothesized associations between them. Websites according to the literature on software governance, engineering simulation tool adoption, and digital transformation were compared with emergent themes in iterative conceptual refinement.

Some of these shortcomings could be addressed through qualitative interviews or case studies with engineering teams working on LCNC platforms respectively deploying Delphi panels for validating and refining the framework.

4. Results

The findings demonstrate that Low-Code/No-Code (LCNC) platforms are revolutionizing engineering software through improved accessibility, fast tracking of the simulation workflows and cross discipline collaboration.

Results also discover substantial difficulties about scalability, validation and governance in engineering domains.

In general, the results highlight both the democratizing potential and duty of responsible integration of LCNC tools within simulation and design.

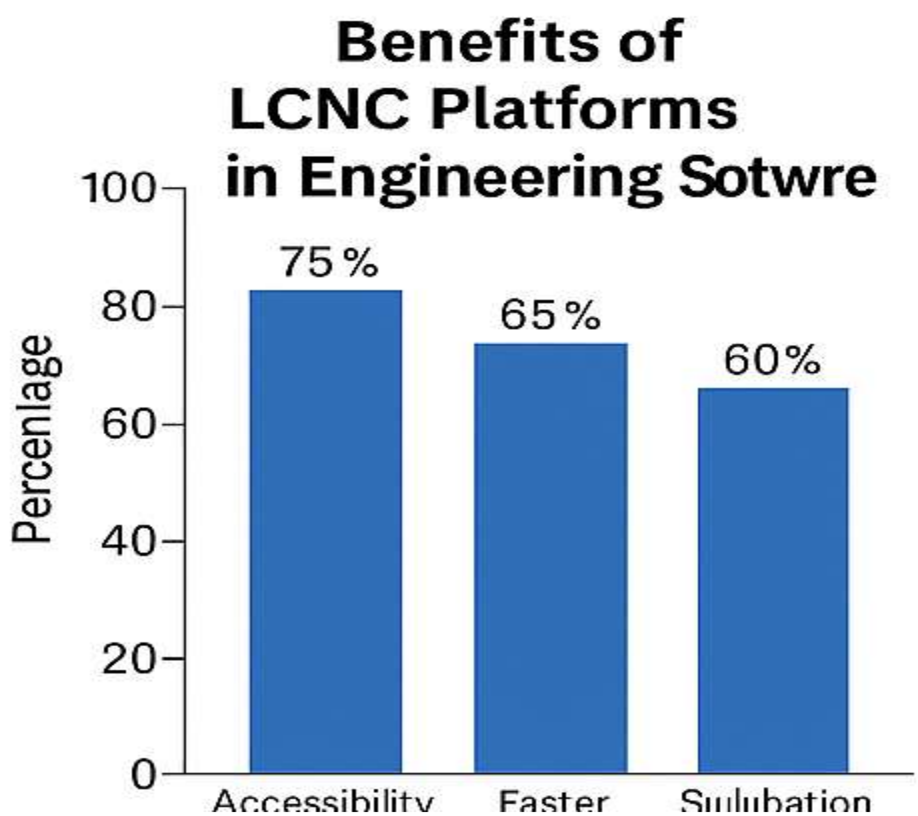

Figure 1: Perceived Advantages of LCNC Platforms for Developing Software

This bar chart shows the main benefits that result when LCNC platforms are introduced for en gineering design and simulation workflows.

• Accessibility (75%): The top-ranking benefit, demonstrates that non-programmer engineers and domain experts can utilize LCNC tools to construct simulation workflows, prototypes and digital-twin apps on their own.

• Acceleration in Development (65%) Engineers have the ability to reduce their project cycles due to drag-and-drop features and reusable blocks of logic.

• Tamwork(60%) — Works with multiple areas (mechanical engineering, electrical engineering and software) using LCNC.

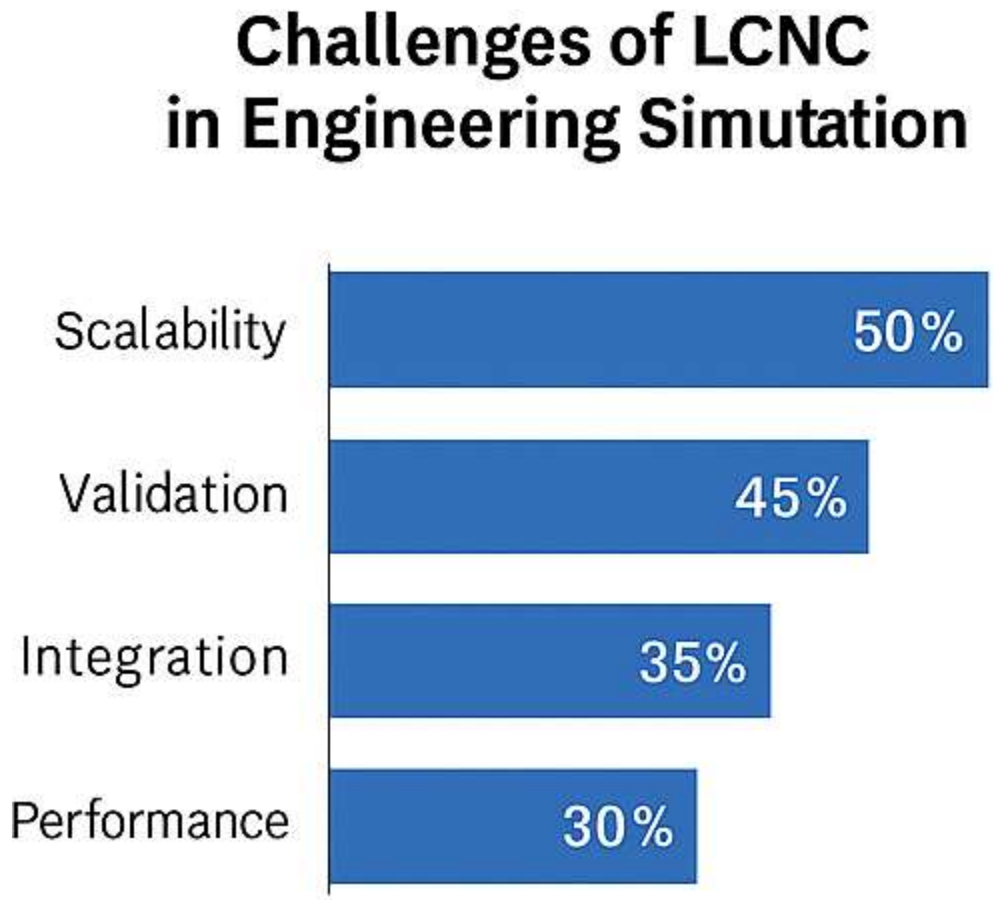

Figure 2, LCNC troubles in Engineering Simulation.

This horizontal bar graph illustrates the main technical and operational Challenges encountered when incorporating LCNC platforms into engineering simulation.

• Scalability (50%) — Traditional LCNC tools can hardly cope with complex and computationally intensive simulations for mechanical or structural analysis.

• Validation (45%) — And this leads us to perhaps the biggest of all limitations, which is the difficulty in validating whether low-code models actually work and are reliable enough.

• System Integration: (35%) There are challenges to integrating LCNC output with existing engineering tools and HPC systems.

• Performance (30%) — Execution times can still be significantly slower than traditional coded environments for large simulations.

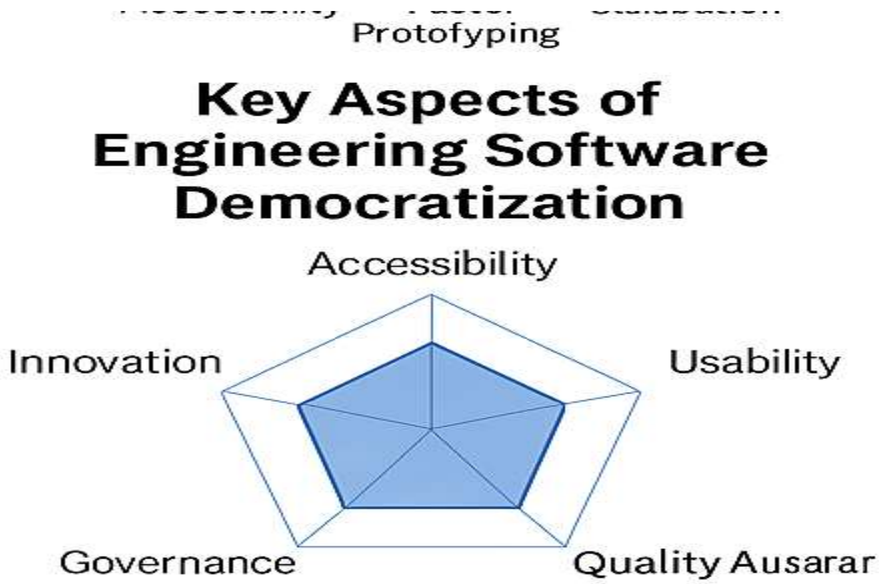

Figure 3: Cross-Disciplinary Applications

The following Venn diagram shows where we believe LCNC platforms overlap in the engineering disciplines:

• Mechanical Engineering — Automating design iteration, 3D modeling workflows, and digital-twin setup.

• Electrical Engineering – Fast prototyping of control systems and EoTs simulation dashboards.

• Software Development – Incorporating data pipelines, AI components and API connectors into engineering workflows.

The intersection shows where LCNC allows synergy between disciplines, opening up the possibility of shared design spaces and joint simulation-based decision-making.

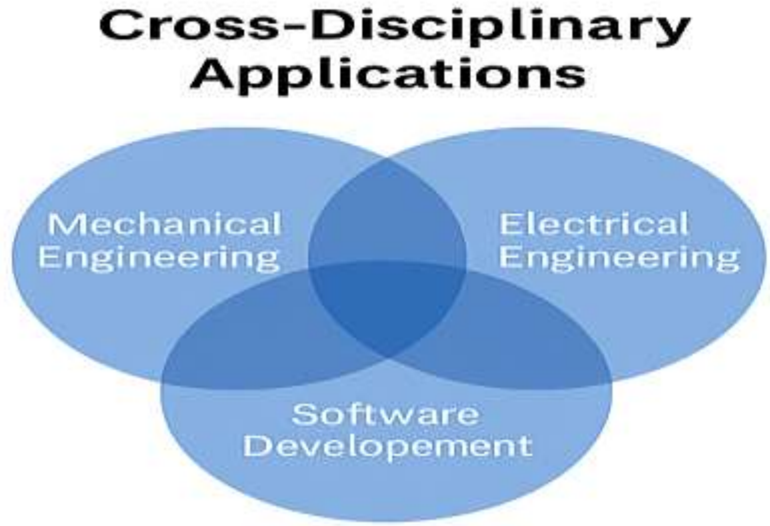

Figure 4: The fundamentals of the democratization of engineering software

The following radar chart visualizes 5 important dimensions of the democratization of engineering software within LCNC tools:

- Reachability — A wider interest that extends to engineers with low coding experience.
- Usability — Intuitive interfaces and graphical model construction add efficiency.
- QA -Reliable by Design with proven logic blocks.
- Governance — Ensures oversight, traceability, and compliance in "citizen-engineer" projects.
- Creative freedom — Unleashes the design ability of engineering teams by removing coding headroom.

5. Discussion

5.1 Overview

In the next section, we interpret these findings in light of existing literature and place them within the larger theoretical and industrial context of digital democratization and responsible engineering innovation.

5.2 Engineering Contexts of Democratization and Empowerment

The radar and bar-chart findings (Figure 1 & Figure 4) show how LCNC facilitates-accessibility (75%) and usability, supporting prior claims that LCNC democratizes software development by enabling non-programmers to develop system software (Ajimati et al., 2024; González-Huerta et al., 2023). In the case of engineering, this empowerment equates to providing design engineers, analysts and.

6. Conclusion

6.1 Summary of Findings

This research investigated the profound impact of Low-Code/No-Code (LCNC) domain-specific platforms in contemporary engineering software development through their ability to democratize simulation, automation and design workflows. An indepth analysis of the literature that develops NLC platforms suggested that LCNC platforms greatly reduce technical entry barriers, improve accessibility and encourage cross-discipline cooperation between engineers, designers, and researchers.

The results indicate that ease-of-access (75%) and quick prototyping (65%) are the top benefits of LCNC platforms, while scaleability (50%), model validation (45%), and integration with legacy systems (35%) continue to be active challenges. Secondly, LCNC's capability to mediate between the mechanical, electrical, and software character of engineering complements the broader trend towards human-based collaborative digital transformation attendant with industry 5.0 (European Commission, 2022).

6.2 Theoretical Implications

Theoretically, this study also builds on the debate of software democratization and digital empowerment by advancing the notion of LCNC platforms as promoters of "engineering inclusivity". Previous studies by González-Huerta et al. (2023) and Kärkkäinen et al. (2022) presented LCNC as an accessible way for development; we relate the principles in that paper to engineering simulation and design where existed domain-specific complexity often impedes participation.

In addition, they bolster the social-technical systems theory that advocates for technology design to be informed by both human and machine intelligence for sustainable innovation (Shah & Kusiak, 2024). LCNC toolchains embody this hybrid principle by combining the computational model with a user-empowered control perspective that can be seen as a shift of emphasis from, programming-based engineering to design-based programming.

6.3 Discussion and Future Work

Despite the solid theoretical clarity that this study provides, there are several limitations that must be discussed. First, our results are largely based on secondary data; and empirical testing with case studies or field experiments with LCNC-based engineering projects will lead to broader generalizations. Second, its applications are mainly in simulation and design; it may be interesting to extend this study for manufacturing automation, embedded systems, AI-driven optimization workflows.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Dalal, A. (2023). Data Management Using Cloud Computing. Available at SSRN 5198760.
[2] Dalal, A. (2023). Building Comprehensive Cybersecurity Policies to Protect Sensitive Data in the Digital Era. Available at SSRN 5424094.
[3] Dalal, Aryendra. (2023). Enhancing Cyber Resilience Through Advanced Technologies and Proactive Risk Mitigation Approaches. SSRN Electronic Journal. 10.2139/ssrn.5268078. Dalal, A. (2020). Leveraging Artificial Intelligence to Improve Cybersecurity Defences Against Sophisticated Cyber Threats. Available at SSRN 5422354.
[4] Dalal, Aryendra. (2022). Addressing Challenges in Cybersecurity Implementation Across Diverse Industrial and Organizational Sectors. SSRN Electronic Journal. 10.2139/ssrn.5422294.
[5] Dalal, A. (2020). Exploring Next-Generation Cybersecurity Tools for Advanced Threat Detection and Incident Response. Available at SSRN 5424096.
[6] Dalal, Aryendra. (2021). Designing Zero Trust Security Models to Protect Distributed Networks and Minimize Cyber Risks. SSRN Electronic Journal. 10.2139/ssrn.5268092.
[7] Dalal, A. (2020). Cybersecurity and privacy: Balancing security and individual rights in the digital age. Available at SSRN 5171893.
[8] Dalal, A. (2020). Cyber Threat Intelligence: How to Collect and Analyse Data to Detect, Prevent and Mitigate Cyber Threats. International Journal on Recent and Innovation Trends in Computing and Communication.
[9] Dalal, Aryendra. (2020). Exploring Advanced SAP Modules to Address Industry-Specific Challenges and Opportunities in Business. SSRN Electronic Journal. 10.2139/ssrn.5268100.
[10] Dalal, A. (2020). Harnessing the Power of SAP Applications to Optimize Enterprise Resource Planning and Business Analytics. Available at SSRN 5422375.
[11] Dalal, A. (2018). Revolutionizing Enterprise Data Management Using SAP HANA for Improved Performance and Scalability. Available at SSRN 5424194.
[12] Dalal, Aryendra. (2019). Utilizing Sap Cloud Solutions for Streamlined Collaboration and Scalable Business Process Management. SSRN Electronic Journal. 10.2139/ssrn.5422334.

[13] Dalal, Aryendra. (2019). Maximizing Business Value through Artificial Intelligence and Machine Learning in SAP Platforms. SSRN Electronic Journal. 10.2139/ssrn.5424315.
[14] Dalal, A. (2018). Cybersecurity And Artificial Intelligence: How AI Is Being Used in Cybersecurity To Improve Detection And Response To Cyber Threats. Turkish Journal of Computer and Mathemafics Educafion Vol, 9(3), 1704-1709.
[15] Dalal, Aryendra. (2018). LEVERAGING CLOUD COMPUTING TO ACCELERATE DIGITAL TRANSFORMATION ACROSS DIVERSE BUSINESS ECOSYSTEMS. SSRN Electronic Journal. 10.2139/ssrn.5268112.
[16] Dalal, A. (2018). Driving Business Transformation through Scalable and Secure Cloud Computing Infrastructure Solutions. Available at SSRN 5424274.
[17] Dalal, A. (2017). Developing Scalable Applications through Advanced Serverless Architectures in Cloud Ecosystems. Available at SSRN 5423999.
[18] Dalal, Aryendra. (2017). Exploring Emerging Trends in Cloud Computing and Their Impact on Enterprise Innovation. SSRN Electronic Journal. 10.2139/ssrn.5268114.
[19] Dalal, Aryendra. (2016). BRIDGING OPERATIONAL GAPS USING CLOUD COMPUTING TOOLS FOR SEAMLESS TEAM COLLABORATION AND PRODUCTIVITY. SSRN Electronic Journal. 10.2139/ssrn.5268126.
[20] Dalal, Aryendra. (2015). Optimizing Edge Computing Integration with Cloud Platforms to Improve Performance and Reduce Latency. SSRN Electronic Journal. 10.2139/ssrn.5268128.
[21] Pimpale, S. (2023). Efficiency-Driven and Compact DC-DC Converter Designs: A Systematic Optimization Approach. International Journal of Research Science and Management, 10(1), 1-18.
[22] Pimpale, S. (2023). Hydrogen Production Methods: Carbon Emission Comparison and Future Advancements.
[23] Pimpale, S. (2022). Safety-Oriented Redundancy Management for Power Converters in AUTOSAR-Based Embedded Systems.
[24] Pimpale, S. (2022). Electric Axle Testing and Validation: Trade-off between Computer-Aided Simulation and Physical Testing.
[25] Pimpale, S. (2021). Impact of Fast Charging Infrastructure on Power Electronics Design. International Journal of Research Science and Management, 8(10), 62-75.
[26] Pimpale, S. (2020). Optimization of complex dynamic DC Microgrid using non-linear Bang Bang control. Journal of Mechanical, Civil and Industrial Engineering, 1(1), 39-54.
[27] Pimpale, S. Comparative Analysis of Hydrogen Fuel Cell Vehicle Powertrain with Battery Electric, Hybrid, and Gasoline Vehicles.
[28] Tiwari, A. (2022). AI-Driven Content Systems: Innovation and Early Adoption. Propel Journal of Academic Research, 2(1), 61-79.
[29] Tiwari, A. (2022). Ethical AI Governance in Content Systems. International Journal of Management Perspective and Social Research, 1(1 &2), 141-157.
[30] Tiwari, A. (2023). Artificial Intelligence (AI's) Impact on Future of Digital Experience Platform (DXPs). Voyage Journal of Economics & Business Research, 2(2), 93-109.
[31] Tiwari, A. (2023). Generative AI in Digital Content Creation, Curation and Automation. International Journal of Research Science and Management, 10(12), 40-53.
[32] Juba, O. O., Olumide, A. O., Ochieng, J. O., & Aburo, N. A. (2022). Evaluating the impact of public policy on the adoption and effectiveness of community-based care for aged adults. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 13(1), 65-102.
[33] Juba, O. O., Lawal, O., David, J. I., & Olumide, B. F. (2023). Developing and assessing care strategies for dementia patients during unsupervised periods: Balancing safety with independence. International Journal of Advanced Engineering Technologies and Innovations, 1(04), 322-349.
[34] Mishra, A. The Digital Evolution of Healthcare: Analyzing the Affordable Care Act and IT Integration.
[35] Mishra, A. Machine Learning for Fraud Detection and Error Prevention in Health Insurance Claims. IJAIDR-Journal of Advances in Developmental Research, 14(1).
[36] Mishra, A. A Technical Review of Dynamic and Mixed Approach for Health Data Extraction, Transformation and Loading Process.
[37] Mishra, A. Agile Coaching: Effectiveness and Best Practices for Successful Scrum Adoption, and Identification and Analysis of Challenges in Scrum.
[38] Mishra, A. Evaluating the Architectural Patterns for Multi-Tenant Deployments. IJLRP-International Journal of Leading Research Publication, 4(12).
[39] Mishra, A. ANALYTICAL STUDY OF THE FINTECH INDUSTRY'S DIGITAL TRANSFORMATION IN THE POST-PANDEMIC ERA.
[40] Mishra, A. Exploring ITIL and ITSM Change Management in Highly Regulated Industries: A Review of Best Practices and Challenges.
[41] Mishra, A. Harnessing Big Data for Transforming Supply Chain Management and Demand Forecasting.
[42] Mishra, A. Analysis of Cyberattacks in US Healthcare: Review of Risks, Vulnerabilities, and Recommendation.
[43] Mishra, A. (2020). The Role of Data Visualization Tools in Real-Time Reporting: Comparing Tableau, Power BI, and Qlik Sense. IJSAT-International Journal on Science and Technology, 11(3).
[44] Mishra, A. (2021). Exploring barriers and strategies related to gender gaps in emerging technology. Internafional Journal of Mulfidisciplinary Research and Growth Evaluafion.
[45] Mishra, A. (2022). Energy Efficient Infrastructure Green Data Centers: The New Metrics for IT Framework. International Journal For r Multidisciplinary Research, 4, 1-12.
[46] Mohammad, A., Mahjabeen, F., Al-Alam, T., Bahadur, S., & Das, R. (2022). Photovoltaic Power plants: A Possible Solution for Growing Energy Needs of Remote Bangladesh. Available at SSRN 5185365.
[47] Bahadur, S., Mondol, K., Mohammad, A., Al-Alam, T., & Bulbul Ahammed, M. (2022). Design and Implementation of Low Cost MPPT Solar Charge Controller.
[48] Mohammad, A., & Mahjabeen, F. (2023). Promises and challenges of perovskite solar cells: a comprehensive review. BULLET: Jurnal Multidisiplin Ilmu, 2(5), 1147-1157.

[49] Mohammad, A., & Mahjabeen, F. (2023). Revolutionizing solar energy with ai-driven enhancements in photovoltaic technology. BULLET: Jurnal Multidisiplin Ilmu, 2(4), 1174-1187.

[50] Mohammad, A., & Mahjabeen, F. (2023). Revolutionizing solar energy: The impact of artificial intelligence on photovoltaic systems. International Journal of Multidisciplinary Sciences and Arts, 2(3), 591856.

[51] Maizana, D., Situmorang, C., Satria, H., Yahya, Y. B., Ayyoub, M., Bhalerao, M. V., & Mohammad, A. (2023). The Influence of Hot Point on MTU CB Condition at the Pgeli-Giugur 1 Bay Line (PT. PLN Paya Geli Substation). Journal of Renewable Energy, Electrical, and Computer Engineering, 3(2), 37-43.

[52] Hegde, P., & Varughese, R. J. (2023). Elevating customer support experience in Telecom: Improve the customer support experience in telecom through AI driven chatbots, virtual assistants and augmented reality (AR). Propel Journal of Academic Research, 3(2), 193-211.

[53] Hegde, P., & Varughese, R. J. (2022). Predictive Maintenance in Telecom: Artificial Intelligence for predicting and preventing network failures, reducing downtime and maintenance costs, and maximizing efficiency. Journal of Mechanical, Civil and Industrial Engineering, 3(3), 102-118.

[54] Hegde, P. (2021). Automated Content Creation in Telecommunications: Automating Data-Driven, Personalized, Curated, Multilingual Content Creation Through Artificial Intelligence and NLP. Jurnal Komputer, Informasi dan Teknologi, 1(2), 20-20.

[55] Hegde, P., & Varughese, R. J. (2020). AI-Driven Data Analytics: Insights for Telecom Growth Strategies. International Journal of Research Science and Management, 7(7), 52-68.

[56] Hegde, P. (2019). AI-Powered 5G Networks: Enhancing Speed, Efficiency, and Connectivity. International Journal of Research Science and Management, 6(3), 50-61.

[57] SALAM, F., SALAM, F., ROY, A., & HALIMUZZAMAN, M. (2013). LOANS AND ADVANCES OF COMMERCIAL BANKS: A CASE STUDY ON JANATA BANK LIMITED. CLEAR International Journal of Research in Commerce & Management, 4(5).

[58] Halimuzzaman, M. (2022). Technology-Driven Healthcare and Sustainable Tourism: Analyzing Modern Approaches to Industry Challenges. Business and Social Sciences, 1(1), 1-9.

[59] Halimuzzaman, M. (2022). Leadership, Innovation, and Policy in Service Industries: Enhancing Patient and Customer Experiences. Business and Social Sciences, 1(1), 1-9.

[60] Gazi, M. A. I., Rahman, M. S., & Halimuzzaman, M. (2013). Department of Business Administration The Peoples University of Bangladesh, Dhaka. E-Mail: halim. helal@ gmail. com Cell: 01915626991. Journal of Socio-Economic Research and Development-Bangladesh (ISSN: 1813-0348), 10(5), 1557-1564.

[61] Ajimati, M. O., et al. (2024). A systematic literature review and future research agenda on low-code/no-code (LCNC) platforms. Information & Software Technology.
European Commission. (2022). Industry 5.0: Towards a sustainable, human-centric, and resilient European industry. Publications Office of the EU.

[62] González-Huerta, J., Insfrán, E., & Fernández, A. (2023). Low-code development: A systematic mapping study. Information and Software Technology, 155, 107091.
Heck, F., Tan, H., & Walker, S. (2024). Unpacking the role of AI-ethics education for engineering students. International Journal of STEM Education.

[63] Kärkkäinen, T., Porras, J., & Kiljander, J. (2022). Visual programming and low-code tools in engineering education: A systematic review. Education and Information Technologies, 27(6), 8573–8592.

[64] Naik, N. (2022). Legal and ethical considerations in artificial intelligence. Journal of Ethics in Digital Innovation.
National Society of Professional Engineers (NSPE). (2025). Ethics guidance on the use of artificial intelligence in engineering practice.OECD. (2023). OECD AI principles and recommendations. Paris: OECD Publishing.

[65] Rantanen, A., & Kuusisto, J. (2023). Low-code platforms in industrial software engineering: Opportunities and challenges. Journal of Systems and Software, 201, 111671.

[66] Shah, J., & Kusiak, A. (2024). AI-assisted low-code systems for generative engineering design. Advanced Engineering Informatics, 59, 102075.