Formosa Journal of Multidisciplinary Research (FJMR)
Vol. 4, No. 1, 2025: 141-150

Artificial Intelligence (AI) for Low-Code and No-Code Development: Making Non-Developers Developers in 2024

Goutham Kacheru[1]*, Nagaraju Arthan[2], Rohit Bajjuru[3]
[1]Software Engineer, Apexon, United States
[2]Information Technology, University of Cumberlands, United States
[3]Electrical Engineering, Southern Illinois University Edwardsville, United States
Corresponding Author: Goutham Kacheru Goutham.Kacheru@Apexon.com

A R T I C L E I N F O
Keywords: Artificial Intelligence (AI), Low-Code, No-Code Development,

Received : 16, December
Revised : 30, December
Accepted: 20, January

A B S T R A C T
Low-code and no-code development platforms are here to transform the software development landscape by allowing even non-technical users build applications without the need of their advanced programming skills. Artificial Intelligence (AI) is the most crucial player in this evolution; reinvigorating these platforms with intelligent automation, device responsive templates and user-friendly interfaces (2024). Users can design, build, and deploy applications easily with AI-powered features (e.g., Natural Language Processing (NLP), drag-and-drop functionality to-design application & code-generation tools). Democratization of Application Development These innovations democratize application development, thereby allowing businesses to innovate faster, lessen the reliance on professional developers, and meet the surging demand for digital solutions. In this paper, we look at how AI-assisted low-code and no-code platforms are changing the way new apps are being developed, empowering non-developers to participate in software creation and accelerating automation of app development. It also presents the challenges and future aspects that we have to deal with in order to use AI on these platforms, as well as their future role of building a bridge between technical expertise and creative problem-solving.

DOI: https://doi.org/10.55927/fjmr.v4i1.13369
ISSN-E: 2829-8896
https://journal.formosapublisher.org/index.php/fjmr

INTRODUCTION

The increased need for digital solutions in this world has brought about a mental revolution to the way that software applications get built. A writing powerful, scalable, customizable application typically needed extensive programming skills and was very time-consuming. Yet, with the advent of low-code development platforms and their seemingly trivial to no coding counterparts this paradigm has exploded in new directions that allow for nonprofessionals (and some wannabe professionals) to design, build and deploy solutions without any formal coding experience. These platforms are democratizing application development by increasing the accessibility to create applications for businesses and individuals to innovate faster and at a lower cost by lowering technical barriers of entry.

Not only that, but in 2024 Artificial Intelligence (AI) is also become one of the main regions of low-code and no code evolution of such platforms. Artificial Intelligence is taking these platforms to next level through intelligent automation, adaptive design recommendations and easy interfaces for no-code development. Features like natural language processing (NLP), machine-learning based code suggestions, and smart templates allow even the most non-technical of users to build sophisticated apps built to their specifications. Such innovations are enabling small businesses, entrepreneurs, and citizen developers to more nimbly solve for an immediate challenge without needing a large development team or years of technical training.

The proper understanding of human intent using natural language is one area where AI has been an important enabler for low-code and no-code platforms. Users simply articulate their needs in layman's and AI converts it to working app modules. For instance, a user may enter Create a customer feedback form with email and phone fields into the platform — only for the interface and background logic to automatically be created by the same platform. It serves as a bridge between techie skills and business solution creativity, allowing for faster prototyping and deployment of solutions.

Moreover, along with making application creation easier, low-code/no-code platforms powered by AI will increase productivity even more as repetitive works such as data integration, workflow creation and testing are automated. It allows users to create applications visually using templated solutions, with its drag-and-drop interfaces taking care of the visual aspect whilst AI configurations provide the best-optimized, error-free logic behind it. For organizations that lack extensive technical expertise, this is a boon because it helps them ride the wave of a fast-digitizing world.

As punishing as the upside of low-code and no-code tools enhanced by artificial intelligence may be, the associated bottom lines are also a double-edged sword. And AI solutions need to differentiate between a secure, scalable and industry-compliant solution or, simply put, an output require continued scrutiny and oversight. As with other AI tools, this shift comes whole a new set of ethical dilemmas that end-users will now need to navigate around potential biases in the logic produced by generative AI and over-reliance on automation without understanding how it works.

This is a research paper about the impact of AI on low-code and no code platform in enabling non-developers to be active contributors to application development. It covers why these platforms are accessible, the role of AI in simplifying complex tasks and what this means for businesses as well as individuals. It also goes into the obstacles to AI integration, such as ethical/security/scalability issues and identifies potential future trends in AI-assisted development. This research conveys a specific understanding of the influence of AI on software development and how it is changing the way various industries think about innovation through examining these facets.

LITERATURE REVIEW

Low-Code and No-Code Development

Low-code and no-code development platforms — Low-code and no-code development platform are great innovation in the field of software development, empowering individuals & enterprises to create applications with little or no programming skills. Using visual tools, drag-and-drop functionalities and recreated components, these platforms are demystifying the application development process to a wider range of users. This is incredibly useful in an age where the need for digital solutions far exceeds the number of qualified developers available.

What is Low-Code Development?

Low code development platforms are built to minimize the manual coding needed for application creation. They come with a variety of features like: Prebuilt Templates: Out of the box designs and components for common needs of an application such as forms, workflows, and dashboards. Drag-and-Drop Interfaces: Visual tools to build applications without writing code, by piecing together components. Configurable Code Blocks: Allowing custom code insertion for advanced functions in case you need further control (for developers). Anyone who has at least minimal familiarity with programming (or even professional developers looking to speed up application development) is an ideal user of a low-code platform.

What is No-Code Development?

No-code development expands on the ideas of low-code, aiming for users who have little to no knowledge of programming. These platforms offer fully visual development environments that enable applications to be created without writing even a line of code. Key features include: High Level Description in Natural Language: Users can describe a particular requirement in simple plain language and this platform will translate it to application logic. Automation Tools: Pre-configured automation workflows requiring zero setup. User-friendly Interfaces – Easy-to-use tools to help clicking and selecting, able to create the app. The no-code platforms are built for "citizen developers," non-technical employees in roles like marketing, operations, or sales with a desire to make applications suited to their needs.

Artificial Intelligence in Low-code and No-code Development

AI is evolving low-code and no-code platforms giving them much more wriggle room thus complementing the existing functionalities making it even simpler: Smart Code Recommendations: These AI-driven platforms offer suggestions for the best configurations, recommended workflows, or even components based on the user intent. AI can conduct automated testing that helps quickly identify errors or performance bottlenecks as soon they occur. Adaptive Design: Machine-learning algorithms examine user behavior and app data to suggest the best design elements or workflows. No Code with Natural Language: Users enter requirements in natural language, and AI generates the application piece you need.

Aspects of Low-Code and No-Code Platforms

These platforms provide a host of benefits to both organizations and individuals: Rapid Development: Applications are developed and shipped significantly faster compared to traditional coding. Affordability: The authorized version decreases the need for massive teams to build software, therefore reducing costs and opening access to small companies/start-ups. Enabling Non-Programmers: Applications can be built without requiring any technical background at all, raining the need of IT departments. Customization: Customized solutions can be developed as per the needs of a business with even non-developers working on it.

Challenges and Limitations

Yet for all their benefits, low-code and no-code platforms contend with some drawbacks: Scalability: Applications developed on these platforms may face performance and scalability issues as they expand. Customization Limitations: Although low-code platforms permit limited custom code implementations, no-code platforms are spared from various advanced functionalities. Security Issues: Applications may not adhere to strong security requirements leading to breaches without governance. Integration Complexities– Unless you have some tech experts on-board, integrating applications with underlying legacy systems or third-party tools can become complex.

Software Development Impact

No-code and low-code platforms are changing the software development landscape in the following ways: Development for the Masses: They narrow the technology gap between developers and laymen. Developer Shortage: Providing citizen developers tools that organizations can use to address a shortage of professional developer's skills. This enables rapid prototyping and deployment of new ideas, driving innovation for businesses.

Emerging Trends in Low-Code and No-Code App Development

The future of low-code and no code development is aligned with technological disruptions of future AI & emerging technologies. Trends to watch include: Intelligent Automation Capabilities: Apps are going to use AI features to save problems with complicated workflows and automate user experiences

across the board. Feature Expansion: Functionality will also continue to evolve in response to expansion of third-party data sources and integration with emerging technologies, such as IoT, blockchain, advanced analytics, etc. When these platforms mature, they will offer scalable and secure enterprise-grade solutions. AI-assisted low-code and no-code development has established itself as a trend that will only grow in prominence over time. These platforms are democratizing software development, empowering people and organizations to rapidly innovate in response to the new-found urgency for digital transformation.

METHODOLOGY

Change in Approach of Development

From changing needs of the market to a complete shift in technology and an advancement into the future, software development has come through a complete transformation over the years. Traditional Development to Low/No Code Platforms This transition from conventional development methods to low-code and no-code platforms signals a significant shift towards efficiency, accessibility, and inclusivity. This section discusses and traces evolution of development methodologies that are relevant to this study supported by the contribution of Artificial Intelligence (AI) in fast tracking the transformation.

Classic Development (Pre-2000s)

Early software development began with manual coding approaches to app building that involved developers writing every line of code from the ground up. This period is characterized by some key features like: Languages and Tools: Some low-level programming languages like C, C++, or more commonly Java were used for the architecture design process. While application development required a deep technical understanding and took an immense amount of time to build out. Isolated Development: Design and Development processes were relegated to within the confines of engineering teams with minimal interaction from other stakeholders outside. Challenges: Development teams are expensive. Hard to adjust with changing requirements because coding structure are not flexible.

Agile and Iterative Development (2000s)

From agile methodologies and iterative development approach: October 2023. Emphasis on Collaboration: Agile advocated collaborative and cross-functional working among team members, with a regular feedback loop involving stakeholders during the different stages of the development lifecycle. Incremental Delivery: Instead of building entire applications in one go, they were built through smaller incremental cycles enabling shorter time-to-market and frequent updates.

Tools and Technologies:

The 2010s: Frameworks and Libraries to the Rescue

The 2010s saw the rise of frameworks and libraries, such as React, Angular or Django that helped to simplify and speed up development:

Reusability: Developers could use prebuilt components that enabled reusing code for common functionalities. More productivity: Version controlling systems (Git, etc) and IDEs. Collaboration tools: GitHub allowed coding collaboration among teams who were working in distributed manner.

Challenges: It required technical expertise from developers end to integrate and customize the frameworks. The applications became bigger and started interacting with each other → increased complexity.

The Rise of Low-Code Platforms (Mid-2010s)

Low-code platforms were born to alleviate the appetite for speed of delivery when it comes to application development and open up development: Visual Development: More visual tools (including Low-Code and No-Code) that utilized prebuilt components, enabling most of a system to be designed versus manually coded. Target Professional Developers: Low-code platforms aimed developers who were looking to accelerate repetitive tasks (creation of forms, integration with data, etc.). Challenges: Faster development: but users needed tech skills for anything advanced. Low-code platform generated applications sometimes struggled with scalability and customization.

From Evolution to No-Code Platforms (End of 2010s – 2020s)

No-code platforms broadened the concepts of low-code development to non-technical users by allowing them to develop applications without writing a single line of code: Development for All: Marketers, ops folks citizen developers were able to build independent applications. AI-Enabled Capabilities: AI-enabled tools for natural language processing, workflow automation and template suggestions appeared within no-code platforms. Test cases: non-developers flocked to Bubble, Wix and Airtable portfolios.

AI-Driven Low-code and No-Code Development (2020s onwards)

With the adoption of Artificial Intelligence (AI), the low code and no code platforms have become more sophisticated and be able to tackle complex development tasks: Code from natural language: GPT and other types of AI models help users create application components in a ready-made programming language described in everyday human language. Automation of task: AI can automate workflows, testing and deployment processes, all of which was previously done manually. This technological progress highlights the paradigm shift from traditional programming to AI-assisted low-code and no-code platforms focused on inclusivity, efficiency, and innovation. These maturing technologies are erasing the line between technical and non-technical users, allowing businesses to meet demand in a fast-changing world where software solutions are both desired and needed. AI integration into these platforms is transforming engagement of applications and enabling organizations to rethink what could be done in the domain of software development.

RESEARCH RESULT AND DISCUSSION

The end result was a simple effective application that served its intended purpose as a contact manager. The application ended up being a guide that the

project requirements dictated, meaning it was functional yet simple and intuitive in design. It ran smoothly with adequate performance providing all expected features through a focus on usability and responsiveness.

Main features and functions

Designing the application focused on interactivity, on usability which has key aspects like;

a. User Interface Component < User interface was designed with lots of interactive elements — buttons, text fields, input fields and proficiency in using gallery to display contact data. Due to this, these components were contextually placed in order to make the application more user-friendly for an end-user.
b. CRUD: Users could execute the crud actions in a smooth fashion. This enabled you to manage and store multiple contact records, with ease.

Data Integration: A Microsoft Excel file was used to store and manage the contact details, which acted as the primary data source. This integration showed how adaptable PowerApps can be in accessing and manipulating non-PowerApps data.

Performance and Responsiveness

The application was responsive and interactive, it adapted to all devices and screen sizes. Users could smoothly navigate and operate the tool such that it would be able to run on various platforms without facing performance issue while using this tool.

Experience with Development

The development process was simple and quick; this would especially be true for an experienced developer who has worked in PowerApps & low-code platforms. Observations For Development Experience: Ease of Use: PowerApps offered a user-friendly interface that made building and configuring the application easier with minimal coding. Flexibility with Citizen Developers: The process of building these types of applications is rather simple for those who have had experience with software development before, but we suspect that citizen developers (those without formal coding experience) could easily build them as well. But they can take a little bit longer to learn the ropes of PowerApps and its capabilities. Flexibility: The app could be designed and function as desired, so the application would have customization possibilities.

Outcomes

This application was successfully designed and implemented to showcase the ease of development within a functional, interactive and user-friendly application using PowerApps. Key achievements include: Functional Requirement: The application worked as intended and covered functional requirements of managing contact information. Application interface is designed to be easy and intuitive, improving user engagement. Operational Efficiency: The CRUD features do what they are supposed to, offering a reliable alternative for data management.

What It Means for Citizen Developers

Although building the development was conceptually relatively simple, it became evident that experience developing apps on PowerApps depends on where a developer comes from. Minor coding experience is not required for citizen developers, but they may take additional time to gain insight into what the platform can do. The low-code and no-code approach makes PowerApps easy enough to be useful, but easy to build these applications that are functional and interactive with just a few hours of practice (for non-technical users). The fluid development process and successful delivery of this application shows how PowerApps can put the power in both technical and non-technical users' hands to create meaningful, engaging applications quickly. PowerApps serves as a low-code development platform that brings technical users and non-technical users together by simplifying the development process, but also allowing for integration with external data sources such as Microsoft Excel.

CONCLUSIONS AND RECOMMENDATIONS

This was a simple but realistic implementation of PowerApps, showcasing how the platform can enable users to develop interactive and functional work flow-based applications with little or no code expertise. The application was a success in that it achieved its purpose of enabling users to manage and store contact information effectively while provide an intuitive, user-friendly interface. CRUD operations, Microsoft Excel Data integration and responsive design made the app functional and accessible on different devices.

PowerApps: A robust low-code platform that streamlines application development The platform really instilled productivity and cut down development time in building a fully functional and responsive application for developers who had used no-code platforms before. While it required a lot more time and practice for citizen developers with no prior coding experience to capitalize on the platform's capabilities, it gave them an avenue to create meaningful apps.

This project showcases the power of PowerApps in making them create solutions which fill a gap between technical and no-technical users to democratize application development with independence for meaningful innovation & business upcycle speed. Given that PowerApps helps to empower Users with inclusivity and adopts the idea of rapid development, it is a perfect platform for building custom applications for specific business needs.

The effective development and implementation of the contact management application prove that low-code platforms such as PowerApps have good significance in modern rapid-paced technology environments. As it continues to evolve and be adopted by more organizations, they are going to take a larger role in the digital transformation equation and allow anyone, even those with little experience, to create applications that will matter.

ADVANCED RESEARCH

The implementation of PowerApps in developing the contact management application underscores the transformative potential of low-code platforms in driving digital innovation and operational efficiency. By integrating

responsive design, CRUD functionality, and seamless Microsoft Excel data integration, PowerApps enabled a user-friendly and scalable solution tailored to specific business needs. Its robust framework empowered developers to expedite the application development cycle significantly, while citizen developers, despite requiring additional practice, could leverage the platform to bridge the gap between technical and non-technical workflows. This project highlights PowerApps as a pivotal tool in democratizing application development, fostering inclusivity, and promoting rapid, iterative innovation. As organizations increasingly adopt such platforms, the paradigm of digital transformation is shifting towards empowering individuals at all skill levels to create impactful, customized solutions, accelerating business agility and competitiveness in the modern technological landscape.

REFERENCES

Böck, A. and Frank, U. 2021. Low-code platform. Business &Amp; Information Systems Engineering, 63(6), 733-740. Available at: https://doi.org/10.1007/s12599-021-00726-8 Accessed 10th April 2024.

Chrismanto, A. R., Santoso, H. B., Wibowo, A., Delima, R., & Kristiawan, R. A. 2019. Developing agriculture land mapping using rapid application development (rad): a case study from indonesia. International Journal of Advanced Computer Science and Applications, 10(10). Available at: https://doi.org/10.14569/ijacsa.2019.0101033 Accessed 19th January 2024.

Educative 2024. What is DevOps? A Complete Guide to DevOps Methodology, (Online) Available at: https://www.educative.io/answers/what-exactly-is-devops. Accessed 19th January 2024.

Juuti, J. 2021. Powerappsilla toteutettu lainausjärjestelmä. https://urn.fi/URN:NBN:fi:amk2021052611310 Accessed 10th April 2024.

Lehto, P. 2021. Power Appsin käytön aloittaminen. Available at: https://urn.fi/URN:NBN:fi:amk2021082017071. Accessed 19th January 2024.

Microsoft Power Apps in Brief, (Online) Available at: https://www.scnsoft.com/microsoft-power-apps. Accessed 19th January 2024.

Microsoft. 2022. PowerApps and Features of PowerApps! TechCommunity. (Online) Available at: https://techcommunity.microsoft.com/t5/nta-techies/powerapps-and-features-of-powerapps/bap/3672520 Accessed 19th January 2024.

Microsoft. 2023. What are model-driven apps in PowerApps? Available at: https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/model-driven-app-overview Accessed 19th January 2024.

Palmer, T., 2020. Microsoft PowerApps as an Alternative Solution to Business Application Development. Available at: https://urn.fi/URN:NBN:fi:amk-2020120325932. Accessed 19th January 2024.

Petersen, K., Wohlin, C., & Baca, D. 2009. The waterfall model in large-scale development. Lecture Notes in Business Information Processing, 386-400. Available at: https://doi.org/10.1007/978-3-642- 02152-7_29 Accessed 24th January 2024.

Rokis, K. and Kirikova, M. 2023. Exploring low-code development: a comprehensive literature review. Complex Systems Informatics and Modeling Quarterly, (36), 68-86. Available at: https://doi.org/10.7250/csimq.2023-36.04 Accessed 19th January 2024. ScienceSoft, 2024.

Simplilearn, 2023. Agile Development Methodologies, (Online) Available at: https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile. Accessed 19th January 2024.

Singh, A. 2023. 5 key features of Microsoft Power Apps that can help your business grow. LinkedIn. (Online) https://www.linkedin.com/pulse/5-key-features-microsoft-power-apps-can-help-your Accessed 19th January 2024.

Tollefson, K. 2022. Power Platform: Model-Driven vs. Canvas Apps vs Portal – What To Use When. (Online) Available at: https://www.withum.com/resources/power-platform-model-driven-vs-canvasapps-vs-portal-what-to-use-when/ Accessed 19th January 2