

## LOW-CODE AND NO-CODE PLATFORMS: DEMOCRATIZING APPLICATION DEVELOPMENT AND EMPOWERING NON-TECHNICAL USERS

Swamy Prasadarao Velaga

Sr. Technical Programmer Analyst, Department of Information Technology

### ABSTRACT

This paper will discuss the development of low code and no code application and the technologies that empower their application. The effects of these platforms on software development, particularly to PD and CD, will be discussed in this paper. The paper will identify specific trends, difficulties, and issues that have not been solved in this field. Low code and no code platforms are trending in today's world of software development [1]. These platforms are dissimilar to the orthodox software development paradigms inasmuch as they allow the creation of application using graphic user interfaces and configurations with a little amount of hand-made, conventional programming code. Thus, more customers with low programming abilities are associated with the creation of software. This results in shorter development cycles and costs, as well as; increased variety of applications. Over the past decade, certain transformations occurred in the sphere of modern software development, primarily due to the development of low code and no-code environments. These platforms allow software construction through GUI and settings combined with a relatively small portion of hand-made, traditional, imperative code [1,2]. Consequently, other than the experienced personnel who code, several more stakeholders participate in the development of software. This results in shorter times and lower costs for product development as well as in a diversification of application portfolios. Gartner defined that a no-code development platform is a tool that allows the creation of application software by using elaborate graphical user interfaces and configurations. This environment isolates developers from other intricate aspects of software development, which includes actual code or the substructure on which the software is built. A low code development platform is somewhat similar but offers somewhere for the coder to code a little and provide for moderately complicated decision making. Due to the complexity of the topics and the limitation of space for this paper, the terms low-code platforms and low-code tools shall be taken to mean one and the same throughout this paper.

**Keywords—** *Low-code development, No-code platforms, Citizen developers, Application democratization, Rapid application development, Visual programming, Drag-and-drop interface Workflow automation, automation, non-technical users, App development, Software solutions, Digital transformation, Enterprise application development, Agile development*

### INTRODUCTION

Low-code and no-code solutions have become increasingly integral as a means of application development that has become an effective and efficient way of creating an application. These platforms enable the creation of applications as well as the continuous and fast prototyping of applications, even to the non-technical 'citizen' developer, enabling the development of complex solutions and speedy deployment of applications to meet business requirements, by IT teams. It can therefore be said that the Information Age has almost entirely depended on software development. Due to the skillful development of software, professional developers and development teams have contributed to the growth of industries, business, and commerce at a much higher level in almost all fields. However, the ways to provide for powerful and efficient implementation of a particular type of generality are not limited merely by the actions of professional computer programmers. End users, and their developing intelligence, creativity, and initiative in the utilization of software represent the growing potential force in the development of software [2,3]. As a result, the categorical dichotomy that has long characterized the relationship between fully developed software professionals and consumers/end-users is gradually closing with the help of numerous technologies and approaches to application software development. Therefore, end users and software professionals remain interlinked as two companies hence establishing the ever enhancing relationship which is revolutionizing the practice of software engineering.

User feedback, ideas, and their preferences has now emerged as one of the most significant paradigms in current software development practices, due to the need for designing easy-to-use, effective and customer-oriented applications that would satisfy customers' needs[3]. This change in the social dynamics in software development is enabling creativity and expanding the possibilities of what can be done through the use of digital technology. With the end-user as a co-creator of the software, the effect they have on defining the course and features of the environment is substantially immense thereby creating a new epoch of user-friendly and participation-driven generation that involves the end-users as well as the professionals.

Thus, low-code and no-code platforms are increasing the rate of this evolution. They allow organizations to produce a vast number of applications at a much faster rate than was previously thought possible. Gartner is also of the opinion that by 2024 more than 65% of application development will be done through low code application development, whilst Gartner also predicts that by 2024 at least four low code tools will be in used in at least three fourths of larger enterprises for both IT and business users[3]. This chapter provides an overview on the evolution of low-code and no-code platforms, their market, advantages, and disadvantages. We also describe their role in contemporary software development and possibly possible future tendencies. Still, from a recent phenomenon, low-code and no-code are quickly redefining the application development landscape within organizations. These platforms allow for building solutions quickly, iterating through them, and putting out applications in a format that enables the administrative teams and the IT departments, as well as citizen developers to build complex applications with little or no coding skills at all[4]. That is why we observe increased uses of low-code and no-code platforms in organizations, which, in turn, leads to important transformations in the field of software development. Their simplicity, versatility and the ways, they help to optimize the processes of development are the key benefits in the contemporary environment. When looking more closely into how low-code/ no-code platforms are positioning themselves as the key tools to define the future of application development, it only becomes apparent that these platforms will be instrumental in driving innovation and therefore, shortening time-to-market for a variety of applications[5]. "The future of low code platforms as a core aspect of business maintenance and undertaking various activities in the business setting is expected to improve by 2024, whereby  $\frac{3}{4}$  of the large organisations will be using at least four low-code development tools in both IT application development, and citizen development." The following chapter is aimed at discussing the advancement of low-code and no-code, the market of the two, the merits, and either the demerit of We also consider their applicability to contemporary software development as well as probable trends in the progress.

## RESEARCH PROBLEM

The main research problem in this study is to establish how low code and no code platforms enhance application development for non technical users and at the same time determining the advantages, difficulties, and prospectuses for low code, no code technologies in different sectors. Low-code and no-code development have come to light as powerful resources in the context of the IT sector and empower people with limited or even no programming skills to build applications. This democratization of app creation presents chances for a lot of people and companies to come up with and address issues with the help of applying and creating apps without a multitude of specialized knowledge[6]. The danger of these platforms is that they could act as the entry point for those people who otherwise would have been left out of the digital transformation process and could bring innovation and, therefore, improve productivity across various industries.

Nevertheless, there are some drawbacks that have been reported or can be expected when implementing low-code and no-code applications. Indeed, it is revealed that there are issues and drawbacks relative to these

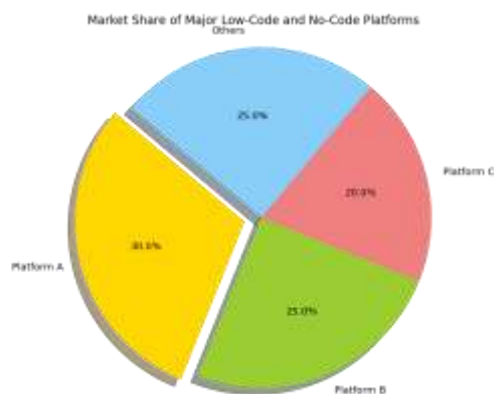
increasingly popular tools that are widely used in development. Challenges include; scalability, security, integration with current systems is a big challenge that businesses face. Also, owing to the adopted platforms these applications can be developed with poor designs due to the ease of use provided by these platforms. The purpose of this research is to examine such issues in detail while presenting an objective analysis of the state of affairs in the relevant field[7]. Moreover, it will consider possible future trends and innovations that would be useful in enhancing the low-code and no-code movement further, what new directions these platforms would be able to develop and contribute to the further evolution of the application development field. Currently, these platforms address a predefined set of capabilities. As the enterprise requirements evolve, the gap of what can be achieved with low-code and no-code platforms and the services provided by them will wide[8]. This research aims to explore this widening gap, the challenges and opportunities along the widening gap, implications on and the future of modern software development. With the insights gained from the exploration, this research will move towards deriving patterns for such platforms.

## LITERATURE REVIEW

### INTRODUCTION TO LOW-CODE AND NO-CODE PLATFORMS

There is increased interest in low-code platforms because of inefficient traditional application development. More than two-thirds of development shops complain about the fact that they experience ‘Severe backlog issues,’; new prioritized projects flow in so quickly that the backlog can never be reduced, while budgets expand. Furthermore, projects are becoming more difficult; 80% of new custom applications being delivered to end users that include experience as a formal specification – this often transmits a signal that the project will be lengthy and miserably complex. Platforms and tools ensure that through visual, model-driven approaches the hardness of tough projects is reduced tenfold. This minimizes the hand-off and rework process that messes up the regular software delivery and allows AD&R pros to manage more of them[8]. With Low-code platforms, the pros can develop the critical applications on their organizations’ needs much faster as compared to hand-coding and, On the other hand, No-code platforms facilitate the business experts or so-called citizen developers to develop the certain applications by them without help from the too busy AD&D pros. Thus, low-code and no-code platforms are oriented at making development as agile and efficient as possible and allowing developers and business professionals to innovate and drive advanced digital transformation processes. Low-code and no-code solutions are the way for improving the effectiveness of development and defining new directions in the software creation process that can help organizations adapt to the constant growth of the digital economy’s demands.

Low-code and no-code have become innovative tools in today’s world because they made software development more accessible and efficient. Low code platforms are developed for solving the requirements of the professional developers by saving time for the creation of the more complex applications and using more of the model based development. Citizen developers use no-code platforms – individuals who are not IT professionals but build business applications for their top teams to use with limited programming (usually restricted to scripting) or no programming at all but with drag-and-drop and ready-made templates[8]. Also known as Rapid Application Development platforms, they lower the time and cost of software development generally. This is best exemplified by those platforms that I have not considered compliant with the no-code setups since they are very much inclined towards enhancing the software delivery stream for the pro developers.



**Fig. 1** Market Share of Major Low-Code and No-Code Platforms

RAD tools define several customized, sophisticated Application Development Life Cycle that offer the facility of integration, testing, and deployment of applications without hindrance. These applications' main purpose is to optimize the processes of application development, to integrate Dev and business teams, and, as a result, to unlock faster time-to-market for new applications. The nature of RAD platforms is most effective in supporting application development that is implemented on an agile model to help the teams meet the changing demands of consumers and the market. Also, RAD frameworks make it possible to develop relative prototypes, and MVPs (Minimum Viable Products) to launch new concepts and ideas without high-cost experimentation as a standard strategy in conventional software engineering[9]. Thus, RAD platforms are critical for accelerating innovation and achieving corporate digitization processes with the help of empowering business users, citizen developers, and professional developers comprehensively. Low-code and no-code platforms are the critical factors in contemporary software development, providing specialists with the possibility to implement new complex and advanced applications much faster — not only for professional developers but also for citizen developers. Working with RAD platforms cuts the total time and total cost that would otherwise be taken in developing applications and also promote the integration of development and business personnel. It fosters the ability to work in the agile ways and master in creating prototypes, it defines the pace of the transformation and the role of digital in organizations[9,10].

#### **THE RISE OF CITIZEN DEVELOPERS: EMPOWERING NON-TECHNICAL USERS**

Low-code platforms are igniting the imaginations of professional developers as well as so-called "citizen developers" – business users who can build and deploy applications with minimal coding or development experience. If professional developers are Leonardo da Vinci – endlessly curious polymaths who use their wide-ranging knowledge to constantly push the edges of what is possible – citizen developers are the shopkeepers and artisans of the Renaissance city. Business users, or citizen developers, know the domain they are working in very well. They understand the problems and the processes. What they lack is traditional software development skills. The rise of the citizen developer is the most significant trend in technology that no one is talking about. Encouraging and enabling citizen developers – within certain parameters – will revolutionize the way software is built[11]. Line of Business IT using low-code tools to accelerate the development of business process applications; and business users applying no-code tools to create departmental and personal productivity apps.

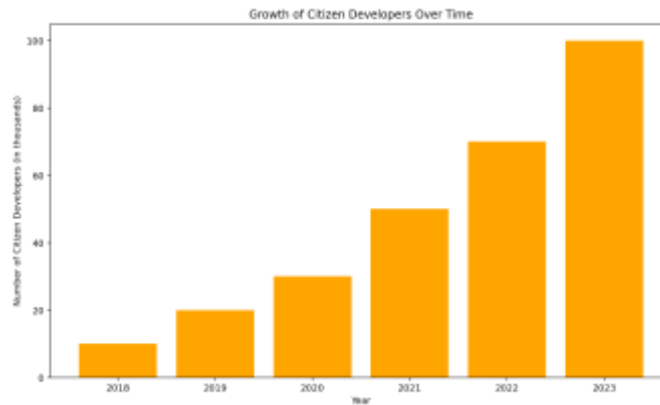


Fig. 2 Growth of Citizen Developers Over Time

As low-code and no-code platforms have become more powerful and easier to use, they have enabled a new breed of "citizen developers" – business users who can build and deploy applications with minimal coding or development experience. Business users, or "citizen developers," know the domain they are working in very well. They understand the problems and the processes[11]. What they lack is traditional software development skills. The rise of the citizen developer is the most significant trend in technology that no one is talking about. Encouraging and enabling citizen developers – within certain parameters – will revolutionize the way software is built. No longer will central IT be exclusively responsible for software projects. We will see a move, in fact we are seeing it now, to a three-tiered approach: Central IT supporting the development of complex, mission-critical apps; Line of Business IT using low-code tools to accelerate the development of business process applications; and business users applying no-code tools to create departmental and personal productivity apps[12].

#### KEY FEATURES OF LOW-CODE PLATFORMS

Low-code platforms enable professional developers to deliver applications faster by reducing or completely removing the need for manual hand-coding; conversely, no-code platforms empower a range of people to create applications and automate workflows with little to no coding. The code that can be produced often sits at the 'shallow' end of the spectrum, with low-code and no-code solutions typically skirting more difficult or subtle programming tasks, providing 'layers' that are easier to work with[13]. Handling the complex, fiddly, or risky aspects of programming work automatically, or hiding them away, is a key feature of such platforms. Designing and implementing such features demands considerable programming skill, leading to further specialization of an already diverse profession of software developers. Furthermore, it allows the platform to widen the group of people who can make practical use of computers[13].

A very simple application for creating, modifying and updating data warehouses; used for mass data combining, query and report generation with friendly graphical user interface and scheduling of automated data update. An RIF replaces detailed tasks traditionally performed by numerous specialists in organizations, thus allowing organizations to create and implement information systems and data analysis solutions swiftly and inexpensively. RIFs incorporate no-code, low-code, and more conventional code Development approaches and can cater for pretty much anything from simple Data analytics and interrogation tools to highly sophisticated Systems to support business decisions and operations. RIF or Rapid Information Factory works as a help to the companies who want to build a good environment for managing their organization's data and want to get good analysing tools for decision making[14]. By being capable of integrating information from various sources with minimal effort while also offering a straightforward system of query and report writing, Business Objects' RIF generally

transforms the management of data in organizations. It overrides reliance on labor and skilled workforce by providing a privy platform for designing and implementing information systems and data analysis solutions. Moreover, RIFs integrate a vast expanse of development methodologies encompassing, no code, low code, and even full code, which makes it appropriate for a wide scope of data interpretative endeavors. In conclusion, the Rapid Information Factory is an invaluable tool for organizations that want to improve their capabilities regarding data and improve their business[14].

#### UNDERSTANDING NO-CODE PLATFORMS: SIMPLIFYING APP DEVELOPMENT

A significant part of the digital transformation in an enterprise context is moving towards development processes that are more accessible, intuitive, and visual in nature to a broader range of business, data, and citizen developers. No-code platforms enable just that, more simplistic application development where writing of code becomes mostly or entirely redundant. A new breed of developers is called forth, ones who understand business needs, workflow logic, and can create meaningful, dynamic outcomes from ready-made building blocks[16]. No-code platforms pioneer in bringing enterprise applications and workflows to life by hiding or, in most cases, completely removing the task complexities that stand in the way of businesses addressing application backlogs. The delivery acceleration that no-code platforms enable results in lower cost, reduced risk hybrid applications which are developed at a fraction of the time. In the race to innovation, companies are increasingly turning to no-code platforms to reduce the application backlog and enable business-IT collaboration. In fact, the no-code development platform market is predicted to top \$52.3 billion by 2024. Today's no-code platforms allow for the building of robust applications with advanced capabilities far beyond simple form-filling applications. The essence of no-code lies in providing a visual development environment which is accompanied by simple data, logic, and workflow designs along with drag and drop, customizable elements. The user's sole focus is on creating meaningful outcomes and business applications without worrying about backend code[16]. By completely removing the need for code, many complex application development roadblocks can be avoided. The resulting applications are generally more secure, maintainable, and are created at a fraction of the time of traditional development processes.

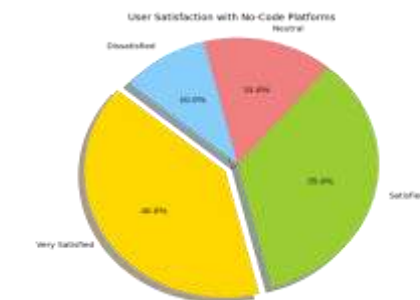
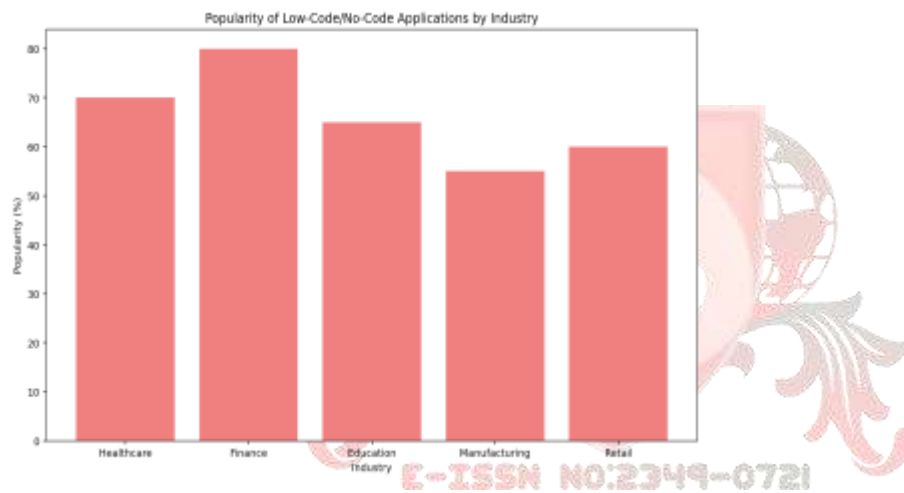


Fig. 3 User Satisfaction with No-Code Platforms

#### USE CASES AND APPLICATIONS ACROSS INDUSTRIES

Low-code and no-code platforms provide a standard mechanism to build custom applications, which can cater to specific needs. With the ease of development, these platforms are gaining a lot of developer attraction. Low-code and no-code platforms are being used to address various use cases across industries. Some of the common

use cases include citizen development, process automation, in-house development, app modernization, customer portals, workflow applications, and operational management applications. Independent Software Vendors are also making use of the low-code and no-code platforms to create and offer smart analytics as part of their product with minimal development effort. Low-code and no-code platforms are helping various organizations create business intelligence dashboards to visualize the data and extract meaningful insights. In the banking industry, low-code and no-code platforms are helping banks in automating Know Your Customer (KYC) processes, loan origination systems, and customer onboarding processes[17]. In the insurance industry, these platforms are being used to create claims processing applications, agent portals, and risk assessment applications. In the manufacturing sector, low-code and no-code platforms are helping teams in creating operations management applications, tracking applications, and quality and compliance applications. In the education sector, these platforms are helping in creating student information systems, portals for parents and students, and applications for admissions and academics. In the public sector, low-code and no-code platforms are being used to create solutions for issue tracking, citizen engagement, and human resource onboarding. Independent Software Vendors are using low-code and no-code platforms to create and offer custom software solutions to small and medium-sized businesses.



**FIG. 4 POPULARITY OF LOW-CODE/NO-CODE APPLICATIONS BY INDUSTRY**

#### LIMITATIONS OF LOW-CODE AND NO-CODE SOLUTIONS

As promising as low-code and no-code development may be, they don't come without their drawbacks. Existing low-code and no-code solutions limit the complexity of applications that can be developed and are focused on providing the fastest solution path. As a result, several potential issues and risks can be introduced. Firstly, the additional abstraction layer introduces performance inefficiencies and restricts invoking underlying low-level functionalities of the target platform with most of the rich applications capable of outgrowing the low-code or no-code platform[17]. Secondly, the generated code or application assets are often required to be hosted on the vendor's platform, which can unexpectedly increase long term maintenance costs or introduce vendor lock-in risks. Even without these limitations, the overall immature nature of both types of platforms introduces uncertainties in design, architecture, and development, in addition to the lack of established best practices to build mission-critical applications. Furthermore, the fastest solution path often reinforces the least performing and least secure coding practices, without educating the developers how to address the root cause of the most common security issues. Most low-code platforms also have very limited support for modern DevOps, continuous integration, and automated delivery practices, introducing manual and error-prone processes when integrating low-code or no-code solutions into the enterprise delivery pipelines [17]. With the positive trajectory that the low-

code and no-code space is on, many of these issues and risks are being promptly addressed. However, careful evaluations and selections of the platform are still required at this time. With the right solution and approach, both types of platforms have the potential to reshape the modern enterprise software development processes and delivery landscapes

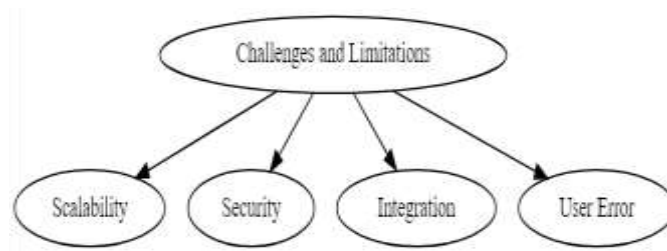


Fig. 5 Challenges and Limitations

### CONTRIBUTIONS

My contribution in this study is to assess in detail the use of low-code and no-code platforms for application development for non-technical users. This entails an analysis of the possibilities that these reflect and opportunities that they present, coupled to the threats and difficulties that are associated with these. Using case analysis and collecting data on how these technologies are being used within several industries, I hope to demonstrate how sectors implement it for Innovation and Competitive advantage. However, for the purpose of my study, I shall investigate best approaches to adopting low-code as well as no-code tools, which, in turn, will provide real-world recommendations to organizations and users in regard to these tools. Establishing some general best practices for implementing LCNC platforms can help organizations avoid some common pitfalls and also help maximize the potential benefits of these platforms. These best practices are largely technology neutral – they can be applied no matter which LCNC platforms are being used. Organizations should also ensure that they have sufficient training and support in place to enable users to make the most of the platforms. Additionally, regular assessments and updates should be scheduled to ensure that the platforms are operating at their full potential. As a radical innovation in low-code development, NFTs - unique blockchain tokens that represent ownership of an item - are now possible to build without any prior coding knowledge with the help of a newborn no-code platform, Unique. In the nearest future, this move can democratize software development in the NFT area at least, allowing a broader group of less technical creators to come up with new ideas.

### SIGNIFICANCE AND BENEFITS

Low-code development platforms hold the promise of developing application software faster and more collaboratively than the traditional methods of coding. This paper explores the current facilitation of low-code and no-code platforms, locates them in the historical evolution of software development tools, and explores their future trajectory and potential transformational impact[18]. The recent prominence and success of low-code platforms has elevated the issue of the sufficiency of the term "low-code" to properly identify and distinguish these platforms in the broader field of software development. It is sometimes said that the only truly new thing is the history that you don't know. Hiring of low-code and no-code platforms helps to optimize the processes of creating applications, and their dependence on coding languages decreases. This brings in more decentralization of software development and can encourage the customers to participate in the software development activities in a much bigger way[18].

People cannot remain indifferent to the modern advancements in technology as the sphere of software development is experiencing constant changes. This has created some nice possibilities of numerous innovative

methods of software development, particularly, the low-code and no-code. Although there is a relatively well-defined term, called “low-code”, it seems that it does not capture all the possibilities and the extent of transformation it provides. The further evolution of software development is accompanied by the integration of new development tasks and further technological achievements, thus creating new routes to evolution and revolutionary changes. It is always fascinating to see how the history of software development is pretty much about new challenges and new capabilities sprouting new forms of solutions, and the case of low-code platforms is no exception here[19]. The increasing interest in the possible advantages of low-code and no-code platforms and the presence of major low-code suppliers changes the already mentioned previous waves of automation in coding tasks deeper than before. Starting from the aspects potentially capable of diversification and democratization of industry representation and shortening the development cycles, the real options in the interest of rapid development, explorative development, and supporting digital innovation as well as business transformation intensifies the enterprise business cases[19]. By developing the real options perspective it is possible to propose that these front-loaded advantages contribute to the further diffusion of these platforms and the realization of the even longer term positive impacts upon enterprise adaptability and inherent ability to reply to the surrounding business environments.

## CONCLUSION

This paper presented an overview of low-code and no-code platforms’ development history and outlined the features of their usage in contemporary software development. Because of the dynamic nature they offer especially to the entire stakeholder community that is involved in the process of making and maintaining software, these platforms experience a rising influence in organizations as well as in different business sectors. They do come with some challenges that, at the moment, are being dealt with by approaches falling within the gamut ranging from correlation to start-ups and academic institutions. As one can infer from the previous sections that the LCPs and the NCPs become even more potent when supplemented with AI techniques. We anticipate that this dynamic area will still spur new types and significant shifts in the very definition of software. In addition, the expansion of the use of these applications in various sectors defines how different companies exist and evolve. Therefore, it is evident that LC/NC platforms are now part of the market and their role will only increase when it comes to application development. Moreover, with the use of these platforms cloud computing and IoT integration is opening new possibilities for the advancements in software solutions and services. Low and no-code platforms are showing promise in the field of software development ranging from the reduction of development time to the increase of application collaboration among development teams and many more. The findings on low-code and no-code platforms in the current software development practices are presented. This way, due to the speed and the flexibility granted to the entire stakeholder community engaged in software creation and maintenance, such platforms are steadily assuming a higher importance in organizations and across different business fields. They do have some limitations but today they are responded to by activities ranging from large business, ventures, and academia. Thus while separately, LCPs and NCPs can be quite effective, when combined with AI techniques the effectiveness is many fold. These reasons reduce the costs of running a business while increasing the gains from efficiency, shopping and productivity; thus the adoption of these platforms. Low-code and no-code platforms have started gaining popularity in today’s world mostly for enterprise applications and as these organizations continue to adopt these platforms, the future path of software development seems like it will continue to be having low-code and no-code based platforms.

**REFERENCES**

- [1] J. Martin, Application Development Without Programmers. Prentice Hall, 1982.
- [2] T. H. Davenport, Big data at work : dispelling the myths, uncovering the opportunities. Boston, Massachusetts: Harvard Business Review Press, 2014.
- [3] B. Van, D. Broeders, and E. Schrijvers, Exploring the Boundaries of Big Data. Amsterdam: Amsterdam University Press ; The Hague, 2014.
- [4] W. R. Vaughn, ADO. NET and ADO examples and best practices for VB programmers. Editorial: Berkeley, Ca: Apress ; New York, 2002.
- [5] S. Spielman, JSTL : practical guide for JSP programmers. Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2004.
- [6] B. Hook, Write portable code : an introduction to developing software for multiple platforms. San Francisco: No Starch Press, 2005.
- [7] G. T. Goodnight and G. A. Willard, Critical problems in argumentation : selected papers from the 13th Biennial Conference on Argumentation : sponsored by the American Forensics Association and National Communication Association August, 2003. Washington, Dc: National Communication Association, 2005.
- [8] J. Bosch, Continuous Software Engineering. Cham: Springer, 2014.
- [9] F. Bergenti, M. Gleizes, and F. Zambonelli, Methodologies and software engineering for agent systems : the agent-oriented software engineering handbook. Boston Mass. ; London: Kluwer Academic, 2004.
- [10] S. Schackow, Professional ASP.NET 2.0 security, membership, and role management. Indianapolis, Ind.: Wiley, 2006.
- [11] R. Hyde, Write Great Code Vol-2: Thinking Low-Level Writing High-Level. 2006.
- [12] P. Group, Embracing and de-risking digital transformation. Probrand, 2015.
- [13] Rajagopal, The Butterfly Effect in Competitive Markets. Springer, 2015.
- [14] R. P. Ramachandran and R. Mammone, Modern Methods of Speech Processing. Springer Science & Business Media, 2012.
- [15] T. Ganchev, Contemporary Methods for Speech Parameterization. Springer Science & Business Media, 2011.
- [16] D. Schatsky, Signals for strategists : sensing emerging trends in business and technology. San Francisco: RosettaBooks, 2015.
- [17] L. M. Camarinha-Matos and F. For, Emerging trends in technological innovation proceedings. Berlin Heidelberg New York, Ny Springer, 2010.
- [18] L. Mann and J. B. L. Chan, Creativity and innovation in business and beyond : social science perspectives and policy implications. New York: Routledge, 2011.
- [19] B. Kahin and H. R. Varian, Internet publishing and beyond : the economics of digital information and intellectual property. Cambridge, Mass.: Mit Press, 2000.
- [20] C. Rautenstrauch, R. Seelmann-Eggebert, and K. Turowski, Moving into mass customization : information systems and management principles. Berlin ; New York: Springer, 2002.