



NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE

Pampady, Thiruvilwamala, Thrissur, Kerala-680 567

**NAAC 'A' Accredited, Approved by AICTE-New Delhi
Affiliated to APJ Abdul Kalam Technological University, Kerala
An ISO 9001:2015 Certified Institution**



DEPARTMENT OF MCA

SEMINAR REPORT

ON

NO CODE/LOW CODE DEVELOPMENT

Submitted by

RESHMA K RAMESH

(NCE22MCA - 2039)

Under the guidance of

Ms. DIVYA P, MCA, MSc.Mathematics

Assistant Professor,

Department of MCA , NCERC, Pampady ,Thrissur

JULY 2024



SEMINAR REPORT

ON

NO CODE/LOW CODE DEVELOPMENT

Submitted in partial fulfillment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

OF THE

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted by

RESHMA K RAMESH

(NCE22MCA – 2039)

Under the guidance of

Ms. DIVYA P, MCA, MSc.Mathematics,

Assistant Professor



NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE,

PAMPADY, THIRUVILWAMALA, THRISSUR-680567

FEBRUARY 2024



NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMBADY

DEPARTMENT OF MCA

COLLEGE VISION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

COLLEGE MISSION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values. We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spirituality, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

DEPARTMENT VISION

To create a school of distinction for the PG students, prepare them to be industry-ready, and achieve Academic excellence by continuous endorsement of the faculty team in terms of Academics, Applications & Research.

DEPARTMENT MISSION

The Department of Computer Applications strives to provide quality and competency-based education and fine-tune the younger generation through Curricular, Co-Curricular and Extra-curricular activities so as to encounter the Professional and Personnel challenges ahead with Pragmatic skills & courage, thereby 'Creating the True Citizens'.

NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY



CERTIFICATE

This is to certify that the seminar entitled “**NO CODE/LOW CODE DEVELOPMENT**” submitted in partial fulfillment of the requirement for the award of the degree of Master of Computer Application of the University of KTU is a result of bonafide work carried out by “ Reshma K Ramesh ” of batch 2022-24 in the Department of MCA under the guidance of Ms.Divya P, Assistant Professor, Department of MCA, NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY under my supervision and guidance.

Guide

Head of the Department

Principal

External Examiner

DECLARATION

I hereby declare that the seminar report entitled “**NO CODE/LOW CODE DEVELOPMENT**” submitted to the **MCA DEPARTMENT** of **NCERC** in partial fulfillment of the requirement for the award of degree in **MASTER OF COMPUTER APPLICATION** from **KTU**, a record of original work done by me under the guidance of **Ms. Divya P**, Assistant Professor of MCA department, during Fourth Semester MCA course period.

Place

RESHMA K RAMESH

Date

ACKNOWLEDGMENT

First and most, I thank the **God Almighty** for showing me the path to the completion of seminar work. I thank **Prof. Dr. Karibasappa Kwadiki**, the principal of **NCERC**, for providing a good atmosphere for seminar completion and presentation. I thank **Dr. Sudheer S Marar**, Head of the MCA Department, and my seminar coordinator **Ms. DIVYA P**, Assistant professor, MCA Department for the guidance and support. I express my immense gratitude to all my friends, without whom I would have never been able to do my seminar well.

CONTENTS

SL.NO	TITLE	PAGE NO
1	INTRODUCTION	1
1.1	WHAT IS NO CODE/LOW CODE	1
1.2	NO CODE VS LOW CODE	2
1.3	TRADITIONAL CODING VS NO CODE/LOW CODE	2
1.4	USE CASES	3
1.5	DEVELOPMENT OF NO CODE/LOW CODE	4
2.	HISTORY OF NO CODE/LOW CODE	6
2.1	LIMITATIONS OF NO CODE/LOW CODE	6
2.2	COMMON FEATURES OF NO CODE/LOW CODE	7
3	WORKING OF NO CODE/LOW CODE PLATFORMS	9
4	SUMMARY	11
5.	REFERENCES	12
6.	PUBLICATION	13

1. INTRODUCTION

The No Code/Low Code paradigm represents a groundbreaking shift in software development, empowering individuals with limited coding experience to create sophisticated applications. This abstract provides a concise overview of the No Code Low C approach, exploring its benefits in terms of rapid prototyping, increased collaboration between technical and non-technical teams, and the democratization of app development. While highlighting its potential to accelerate innovation, the abstract acknowledges considerations such as security and governance, urging a balanced evaluation of No Code/Low Code tools for optimal integration into modern development workflows.

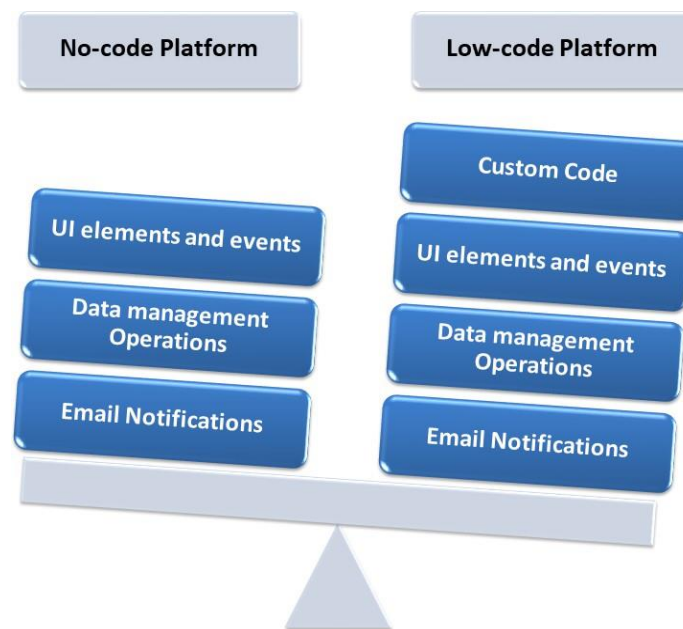
1.1 What is No Code/Low Code

No Code/Low Code signifies a revolutionary shift in the landscape of software development, simplifying the creation of applications for individuals with limited or no coding expertise. In a No Code environment, users can construct applications through intuitive visual interfaces, utilizing drag-and-drop functionalities for pre-built components without engaging in traditional coding practices. This approach aims to democratize the development process, enabling a diverse range of users, including business analysts and designers, to actively contribute to the application-building process. In the Low Code approach, a nuanced balance is struck between visual development tools and the integration of code. Users can harness visual interfaces for most tasks while retaining the flexibility to incorporate custom code when necessary. This feature caters to a broader spectrum of users, ranging from non-technical individuals to professional developers, fostering collaboration and efficiency within software development projects.

Both No Code and Low Code methodologies share the overarching objective of enhancing agility and lowering the barriers to entry in software development. These platforms have gained widespread acceptance due to their capacity to expedite development cycles, encourage collaboration between technical and non-technical teams, and empower organizations to swiftly adapt to evolving business requirements. However, careful consideration of challenges such as security implications and the appropriateness of No Code/Low Code for intricate projects is imperative to ensure a judicious and effective integration into the overall development workflow.

1.2 No Code vs Low Code

As it can be easily inferred from the names, low-code platforms have the provision to add the custom codes when required, whereas the no-code platforms don't have such provisions. These platforms provide pre-coded templates which can be customized according to the requirements and then can be published in different formats suited for desktops/laptops, tablets and mobile phones. Low-code platforms provide an additional facility to add custom logic by writing our own codes, and thus need some programming skills. The language used to code might be a proprietary language which is a wrapper on top of the existing programming languages and this saves time by writing code only once, which will automatically be converted into the multiple programming languages required to support different kinds of devices as mentioned earlier.



In a no-code platform, we can only choose from the in-built modules and events to create the application which limits us from customizing the app extensively - though many immediate purposes can be served. Both platforms have pre-coded visual elements like textboxes, buttons, drop-downs, checkboxes etc. which can be dragged and dropped into forms or pages. The data entered into the forms are automatically added into backend tables, which again saves time of creating a database, or connecting the front-end to the back-end. Most of the applications used in an enterprise like Human Resources, Helpdesk, Finance, Sales and Marketing rely on data. Low Code/ No Code platforms provide pre-built modules which can handle typical actions in a process flow, e.g. email notifications. These inbuilt capabilities enable the Low Code/No Code platforms to develop applications in a very short span as compared to the traditional coding.

1.3 Traditional Coding vs. Low Code/No Code

A survey conducted by a Low Code platform provider across six countries among 1209 IT leaders and 816 software developers says that Low Code projects reduce costs by 53%, can be developed 56% faster, and the customer revenue increased by an average of 58% compared to the traditionally built applications ^[15]. Following are some of the key differentiating factors between the traditional coding and Low Code/No Code:

1. Skill set:

To develop an application using the traditional coding method requires professional developers who are expert in different programming languages. There might be a need for different programmers who have expertise in developing the front-end for a web-based application and mobile-app, also another set of programmers who has expertise on handling the back-end, or simply extra- qualified '*full stack*' developers for any novel system being built. In contrast, in low-code/no-code platforms both 'citizen' developers with rudimentary skills and professional developers can build an application.

2. Duration:

As mentioned in section 3, automating the supply chain of a business could take up to nine to twelve months using the traditional coding method as it involves extensive coding , multiple code revisions , testing and bug fixes . It might appear to readers that 9-12 months is an over estimate , when in many Cos the attachment of a software developer to a project / product is for 1 quarter, ie.3months.

3. Cost:

Here we present a comparison claimed by a no-code platform. The estimated budget to develop a ride sharing app for Android or iOS platform in the traditional way is between \$ 55,000 to \$ 120,000^[16]. Whereas similar apps developed in this company's platform can be hosted in a custom domain and the charges start from \$25 per month ^[17].

4. Deployment Platform Support:

In traditional coding web applications and mobile apps need different sets of developers with different skill sets. Thus these are designed , developed and deployed separately .But the applications built using No Code/Low Code platforms can be deployed as web application , Android app and iOS app all at once.

1.4 Use cases

No-code/low-code development platforms can be used to design, build and deploy a variety of applications, here are some common use cases :

1.Rapid Prototyping: COO of a leading service based company from India says that the no-code/low-code platforms are very useful in demonstrating the prototypes of the applications to their clients ^[18]. Suggestions for Low Code/No Code use cases on similar lines are prototypes in a business school's product management class, where not all students might have a software development background.

2. Customer Engagement Apps: Low-code/no-code platform can be used to build the

customer engagement apps like chat bots , virtual assistants, feedback forms and surveys. There are customer engagement app templates provided by these platforms which can be easily customized ^[19]. Of these, mobile-based feedback forms and surveys - even to actively run business processes in a firm - are currently in vogue.

3. AI and Multi-experience apps: Low-code/no-code AI platforms can be used to create the multi-experience applications such as Virtual Reality and Augmented Reality. The applications like chat bots would come under this category as well because of the artificial intelligence requirements ^[19]. In both these use cases, there must be an existing platform where these add-on modules apply well. For example, without an e-commerce platform where past chats (or voice transcripts of customer calls) aren't available – chat bots may not be a success, esp. in dealing with customer voice or tone.

4. Operational Efficiency Apps: The apps which can be used by the employees of an organization to automate the manual or paper-based processes. These kinds of apps can be built easily with the help of No Code/Low Code platforms ^[19]. This can also include robotic process automation where the data from a legacy application has to be copied to the current application or vice-versa-or other repetitive routine tasks ^[20]. For instance, a low-code mobile phone app will provide employees with information on an organization's conveyance arrangement - incl. input from peripherals such as GPS.

1.5 Development of No code/low code

For decades, there have been only two routes for businesses to take on their way to application development: buy apps ready-made from an external vendor or build and customize them from scratch using skilled developers and coders. But today, we're seeing the rise , growing sophistication of No code/Low code development alternatives that bring the power of application development to users across the business.

These alternative app development methods leverage an intuitive and graphical interface to give users the power and freedom to quickly build applications and automate business processes without having to write code line by line. The adoption of Low Code/No Code tools comes with the promise of greater user accessibility, which in turn spurs more innovation and reduces strain on IT departments. The implementation of No Code/Low Code platforms is the next leap in making application development easy and accessible to all.

The evolution of No Code/Low Code platforms represents a transformative journey in response to the evolving landscape of software development. The concept of visual programming and simplified development interfaces began to take shape in the late 20th century, laying the groundwork for what would later become No Code/Low Code. The 1980 and 90s, the rise of the Rapid Application Development movement, emphasizing quick development cycles and prototyping. As the internet gained prominence in the 2000s, web development tools started incorporating visual elements, influencing the trajectory of No Code/Low Code. The emergence of Low Code platforms in the 2010s marked a pivotal stage, striking a balance between

minimal hand-coding and flexibility for developers. This was followed by the widespread adoption of pure No Code platforms in the 2010s and 20s, catering to users with limited coding experience. Over time, No Code/Low Code platforms integrated with DevOps practices, embraced agile methodologies, and expanded into enterprise solutions. The focus shifted towards collaboration and democratization, positioning No Code/Low Code as tools for enabling non-technical stakeholders to contribute to application creation. In the 2020s, the industry witnessed the incorporation of advanced features such as AI integration, reflecting an ongoing commitment to innovation and addressing the dynamic needs of organizations. Today, No Code/Low Code continues to grow, adapt, and shape the future of software development.

2. HISTORY OF NO CODE/LOW CODE

Low-code technology has its roots in the early 2000s, with rise of rapid application development platforms. These platforms aimed to provide a more visual, drag-and-drop approach to application development, allowing users to build applications faster and with less coding. By the mid-2010s, low-code had become mainstream technology, embraced by organizations of all sizes for its ability to speed up development and reduce costs.

To say that low-code development platforms have made an impact in the world of programming would be a monumental understatement; they've revolutionized the way companies, and developers work. Low-code application development platforms have made programming easier and more efficient than ever before. Instead of typing out hundreds of thousands of lines of code to create an application, these platforms have made it simple with visual workflows, drag-and-drop features, and more.

In 2014, Forrester coined the term “Low-code” to classify development platforms that focused on development simplicity and ease of use. These platforms allowed developers and users of all skill levels to code applications, without having the outright need to know coding. And the trend caught on like wildfire.

Forrester states that the term “low-code development” was first used in a 2011 report on new productivity platforms for building custom apps. Bubble was the first no-code platform launched in 2012 that enabled visual programming by replacing traditional coding. But where did the low-code trend start? That's a question worth exploring to understand why low-code development platforms continue to dominate the market.

2.1 Limitations of No Code/Low Code

Certainly, there are also limitations of no-code/low-code platforms, Some of the main limitations of NO CODE /LOW CODE include:

1. **Flexibility and Customization :** No-code/Low-code platforms often have predefined templates and limited customization options, making it challenging to build highly specialized or unique applications.

2.Performance Concerns : No-code/Low-code platforms may have performance limitations when dealing with resource-intensive tasks or handling large volumes of data.

3.Long-Term Viability : No-code/low-code platforms may not have the same long-term viability as traditional development, as they are subject to the evolution of the platform and vendor support.

4.Vendor Lock-in : No-code/low-code platforms may lock organizations into proprietary solutions, making it challenging to switch to other platforms or services.

5.Scalability : While no-code/low-code platforms support scalability to a certain extent, they may face limitations when it comes to highly complex or resource-intensive application.

2.2 Common Features of No Code/Low Code

While no-code and low-code platforms have distinct characteristics, they share several common features due to their common goal of simplifying and accelerating the application development process. Here are some features that are commonly found in both low-code and no-code development platforms:

1.Visual Development Interface: Both no-code and low-code platforms provide a visual interface for users to design, build, and modify applications.

2. Pre-Built Components and Templates: Both types of platforms offer a library of pre-built components, modules, and templates that users can leverage in their applications.

3. Workflow Automation: Both no-code and low-code platforms enable users to create and automate workflows without extensive coding.

4.Cross-Platform Compatibility: No-code and low-code platforms often support the development of applications that can run on multiple platforms (web, mobile, desktop) without requiring separate codebases.

5.User-Friendly Interface: Both types of platforms prioritize ease of use, making app development accessible to users with varying levels of technical expertise.

6.Collaboration and Version Control: Collaboration features and version control are often integrated into both types of platforms to facilitate teamwork and track changes

7.Security and Compliance: Both no-code and low-code platforms address security concern and compliance requirements to ensure that developed applications meet industry standards

8.Integration Capabilities: Both types of platforms offer integration capabilities, allowing users to connect their applications with external services, APIs, or databases.

3. WORKING OF NO CODE/LOW CODE PLATFORMS

Low-code development platforms and No Code development platforms are based on the principles of model-driven design, automatic code generation, and visual programming.

These platforms are intentionally designed to target users that are familiar with processes and workflows within their business department, regardless of their coding experience.

And once again, not only does this empower non-technical users, it matches them with experienced developers.

Below are some of the fundamental steps in the Low Code/No Code development journey:

- **Define your needs and the desired outcome:** Whether you are working on a customer facing application or an internal business process, it's essential to identify the business needs and the desired outcomes of your project at the outset.
- **Draw a business process or workflow:** Using Low Code/No Code business process management and development tools, users specify and document the desired processes and workflow. This is often achieved by identifying modules by purpose within the application, developing them as independent entities. For example, some modules may gather data, others may trigger an action or event. At this stage, the citizen developer can first build, then integrate the modules to achieve the desired outcome.
- **Test and deploy your project as an Low Code/No Code application :** With a couple of clicks, the Low Code/No Code platform resolves all the backend complexity of the process for users. When it is ready, IT specialists and or beta testers can evaluate the application, once their recommendations are implemented, the app can be deployed for general use.

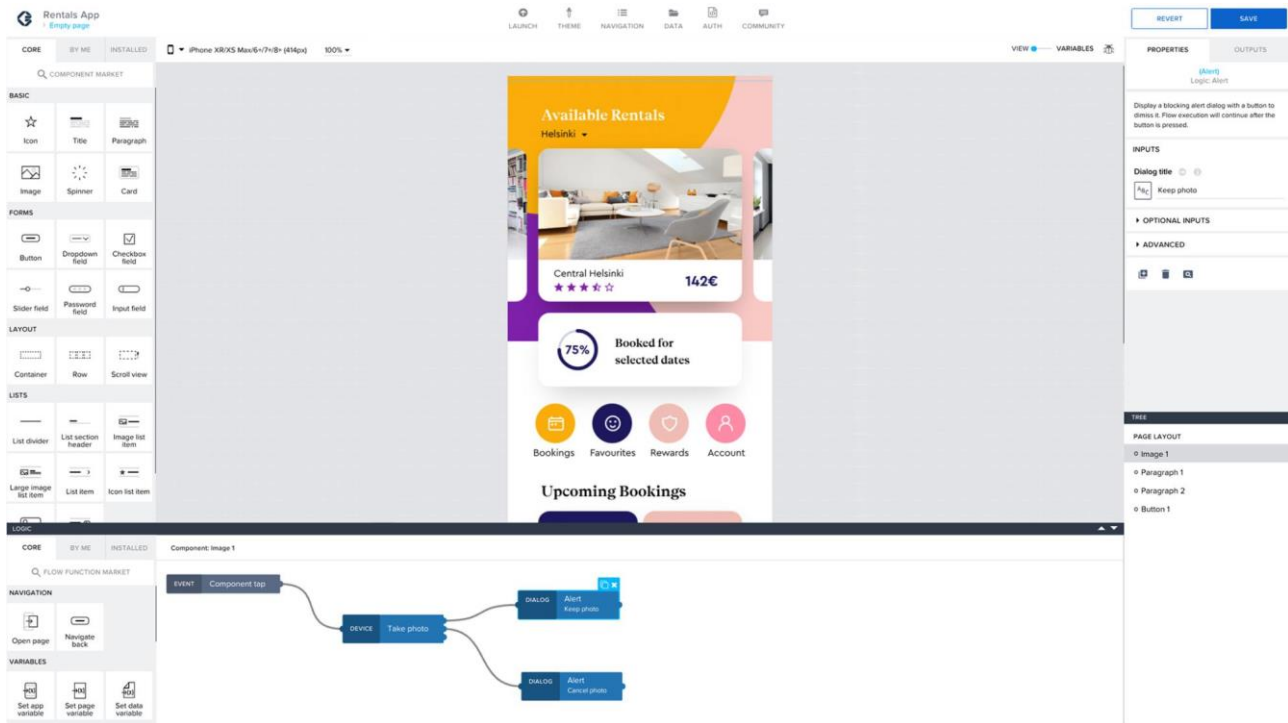


Fig: App View of a Low Code/No Code development platform

4.SUMMARY

No Code/Low Code development has emerged as a transformative paradigm in the software development landscape, offering a revolutionary approach to building applications. At its core, these platforms empower individuals with limited or no coding experience to actively participate in the creation of software, marking a departure from traditional development methodologies that heavily rely on programming expertise. The essence of no-code lies in its ability to enable users to construct applications through intuitive visual interfaces, dragging and dropping elements, and utilizing pre-built components. One of the distinctive features of No Code/Low Code platforms is the emphasis on visual development. By providing an array of pre-built components, widgets, and templates, these tools facilitate the creation of user interfaces, workflows, and data models with unprecedented ease. The visual development environment not only reduces the learning curve for non-technical users but also accelerates the application development lifecycle by fostering rapid prototyping and iteration. This agility allows users to experiment, refine, and adapt their applications swiftly, aligning with the dynamic demands of modern business environments. Accessibility is a cornerstone of the No Code/Low Code movement, ushering in a new era where business analysts, domain experts, and individuals outside the traditional realm of software development can actively contribute to the creation of applications. This democratization of development aims to bridge the gap between business needs and IT capabilities, fostering collaboration and empowering diverse teams to bring their ideas to fruition.

No Code/Low Code platforms go beyond just simplifying development; they integrate capabilities for seamless connectivity with external services. Pre-built connectors and APIs facilitate the integration of third-party tools, databases, and other systems, allowing users to build comprehensive applications without delving into the intricacies of backend coding. This interoperability enhances the versatility of applications and promotes a modular approach to development. While these platforms cater to users with limited coding skills, they do not compromise on the potential for customization and scalability. Skilled developers can inject custom code when necessary, ensuring that complex projects and unique business requirements can still be addressed. This flexibility positions No Code/Low Code as a viable solution across a spectrum of projects, from simple prototypes to more sophisticated applications. Collaboration is another hallmark of No Code/Low Code development. Many platforms include features that enable teams to work cohesively on projects, fostering a collaborative environment. Additionally, version control mechanisms ensure that changes are tracked and managed effectively, allowing for a structured and organized development process. However, it's important to acknowledge the challenges associated with No Code/Low Code development. While these platforms excel in accelerating development for certain use cases, they may face limitations in handling highly intricate or specialized projects that demand extensive custom coding. Striking the right balance between simplicity and flexibility remains an ongoing consideration in the adoption of these tools.

5.REFERENCES

- [1]. Low code use cases: What can you build with low code platforms. (2021, September 6). Gramener Blog. <https://blog.gramener.com/low-code-platform-use-cases/>
- [2]. Balo , S. (2020, February 6). 5 best use cases of no code. Medium . <https://medium.com/@senolbalo/5-best-use-cases-of-no-code-5141e362b329>
- [3]. Johannessen , C., & Davenport, T. (2021, June 22). When low-code/no-code development works — and when it doesn't. Harvard Business Review. <https://hbr.org/2021/06/when-low-code-no-code-development-works-and-when-it-doesnt>
- [4]. Alexander, F. (2021, January 8). Low-code and no-code: What's the difference and when to use what? Out Systems . <https://www.outsystems.com/blog/posts/low-code-vs-no-code/>
- [5]. Tozzi , C. (2021, March 1). A practical take on low-code vs traditional development. Search Software Quality. <https://searchsoftwarequality.techtarget.com/tip/A-practical-take-on-low-code-vs-traditional-development>

6.PUBLICATION

Revolutionizing Software Development: The Rise of No Code/Low Code Development Solutions in Digital Era

Reshma k Ramesh¹, Ms.Divya P²

¹MCA Scholar, Department of MCA, Nehru College of Engineering and Research Centre, Pambady, India

²Assistant Professor, Department of MCA, Nehru College of Engineering and Research Centre, Pambady, India

Abstract

Low/No-code improvement is a program development method that gives clients with a stage for outwardly creating applications with small or no coding. Companies and organizations require program applications and information systems for different commerce purposes like administration in the technology period. Low/No-code advancement gives non-IT professionals a helpful device for quickly building simple business applications they require without or with small coding. In this paper, we investigated the benefits & confinements of Low/No Code advancement and advanced Low/No Code development platforms in the industry. In expansion, we analyzed how it can be improved and prospected the impacts of Low/No Code development on society and related businesses in the future. In conclusion, we discover that Low/No code improvement is a promising trend that can essentially affect future program development and computerized change.

Keywords: Digital transformation; Low code development; No code development; Software development.

1. INTRODUCTION

Traditional software development approaches typically find it difficult to keep up with the fast changing technical landscape in the digital era, where agility and creativity are critical. But a new force in software development is upon us: low- and no-code development solutions. Without requiring a great deal of coding experience, these cutting-edge platforms enable anyone with different degrees of technical proficiency to design, develop, and implement apps.[1] No code and low code solutions provide an efficient way to develop software by using pre-built components and intuitive visual interfaces. This lowers the barriers to entry for both corporations and aspiring developers and significantly reduces time-to-market.

The use of low-code and no-code development solutions is growing as businesses come to understand the need of agility and quick iteration in the current competitive landscape. We will examine these platforms' revolutionary potential to change software development processes in this seminar.[2] We will discover how low-code and no-code solutions are changing the software development landscape, spurring creativity, and helping businesses succeed in the digital era through a thorough examination of

major trends, obstacles, and opportunities.

The emergence of low-code and no-code development methods has caused a radical change in the way software is designed, developed, and used. These platforms offer a democratized approach to software development, putting the ability to create complex applications in the hands of non-technical individuals in an era defined by digital disruption and unrelenting innovation. No code and minimal code solutions free people and organizations from the burdens of traditional coding, enabling them to experiment with new ideas, iterate quickly, and launch products more quickly than ever before. The emergence of low-code and no-code solutions is not just a fad; rather, it represents a fundamental rethinking of how technology is used, as is becoming more and clearer as we go further into the world of software development.

2. LITERATURE SURVEY

A thorough analysis of the literature on low-code/no-code software development reveals a wealth of knowledge and perspectives that illuminate the development, advantages, difficulties, and promise of this cutting-edge methodology. Researchers have looked into how these platforms came to be in response to the increasing need for software development solutions that are more accessible and speedier.[3] By conducting a methodical examination of multiple studies, it is apparent that low-code/no-code platforms have become popular in a variety of industries because of their capacity to enable non-technical users, expedite development cycles, and promote cooperation between developers and subject matter experts. Although much of the literature points out the benefits, it also discusses issues such as these platforms' possible security threats, scalability issues, and customization restrictions. Through the synthesis of these findings, this literature review advances our understanding of low-code/no-code software development and helps practitioners and academics make decisions and identify areas for future research in this rapidly evolving field.

3. OBJECTIVE

A revolutionary method of creating software, low-code/no-code development has several alluring benefits. It makes application development accessible to even those with low technical knowledge by abstracting and simplifying complicated coding procedures. The democratization of software creation promotes cross-departmental collaboration and innovation by shortening the workload for specialized development teams and speeding up the development cycle. Additionally, low-code/no-code platforms' visual interfaces and pre-built components speed up prototype and iteration, improving agility in responding to changing business needs. Increased productivity, lower development costs, and the opportunity for a larger group of participants to participate in the software development lifecycle are the end results, which eventually lead to quicker innovation and more responsiveness to the market. [4] Even people without a lot of coding experience may effectively create and implement apps thanks to its user-friendly interface. Rapid prototyping is made possible by Low Code/No Code, which reduces development costs and time. This method simplifies the creation process by encouraging cooperation between technical and non-technical teams. The platform's pre-built modules and templates make it easier to assemble complicated capabilities and remove the need to start from scratch. [5] This quickens the pace of innovation, enabling companies to maintain their competitive edge. Moreover, Low Code/No Code platforms facilitate upgrades and improvements by guaranteeing scalability and ease of maintenance. By lowering entry barriers and democratizing app creation, Low Code/No Code platforms

usher in a new era of creativity and efficiency and make a substantial contribution to inclusive and agile software development.

4. METHODOLOGY

A methodical approach is used in research methodology for low-code/no-code software development to examine and understand the benefits, drawbacks, and implications of these cutting-edge development paradigms. Presenting unique ideas and giving due credit to all sources are crucial in order to prevent plagiarism. [6] In order to collect information on user experiences, development speed, scalability, and security, the technique may include a review of the literature, case studies, surveys, and empirical analysis. Researchers can make significant contributions to the area while maintaining the moral norms of academic honesty by using a well-structured technique.

Here are some common research methods used in the low-code and no-code software development domain:

1. USER SURVEYS AND INTERVIEWS:

User interviews and surveys are essential when it comes to low-code and no-code software development. In this quickly changing market, these strategies provide insightful information about the needs, preferences, and pain areas of consumers. Developers can find usability issues, unmet needs, and improve the user experience by interacting directly with users. Teams acquire a more profound comprehension of the ways in which these platforms influence productivity, cooperation, and invention by means of systematic surveys and in-depth interviews.

2. CASE STUDIES:

Case studies in the field of low-code/no-code software development offer insightful information about the advantages and usefulness of these cutting-edge techniques. These studies usually focus on real-world situations where developers and organizations have successfully accelerated and streamlined the development process by using low-code/no-code platforms. Stakeholders are better able to comprehend how these platforms allow complex apps to be created with little to no traditional coding effort thanks to these examples. These case studies demonstrate the adaptability of low-code/no-code solutions across a range of industries, demonstrating how they enable experts, both technical and non-technical, to work together and effectively realize ideas. Through an analysis of the particular difficulties encountered, tactics utilized, and results attained, these case studies provide a framework for those thinking about using low-code/no-code solutions, promoting a more knowledgeable decision making process.

3. COMPARATIVE ANALYSIS:

In the field of low-code/no-code software development, comparative analysis entails a methodical assessment of various platforms and tools intended to facilitate the construction of applications. Through a thorough analysis of the features, capabilities, scalability, and ease of use of different solutions, developers and organizations may make well-informed decisions that are in line with their unique requirements and goals. This procedure makes it easier for stakeholders to fully comprehend the advantages and disadvantages of each platform, allowing them to choose the one that will optimize productivity, speed up development cycles, and enable people with different levels of technical expertise to actively participate in the software development process.

4. PROTOTYPING AND TESTING:

In the field of low-code/no-code software development, prototyping and testing are essential phases that enable effective and user-centered solutions. Early iterations of an application's functionality and

interface are created through prototyping, which enables stakeholders to see and improve the idea at an early stage. This iterative procedure promotes quick feedback gathering and aids in detecting possible problems with the user experience or design. Testing subsequently verifies the generated software's resilience and dependability. Developers can verify that an application works in a variety of circumstances by using testing approaches like functional, usability, and performance testing. Testing and prototyping work together to improve low-code/no-code software quality, increasing user happiness and reducing the need for post-production adjustments.

5. VENDOR EVALUATION:

Vendor evaluation for low-code/no-code software development is a careful examination of possible suppliers to guarantee that their offerings meet quality standards and business requirements. A thorough examination of the vendor's platform capabilities, scalability, security protocols, integration possibilities, user interface, and support services are usually part of this process. [7] Organizations can make well-informed decisions regarding the acceptability of a vendor's product, its compatibility with their project objectives, and its capacity to expedite development processes by carefully examining these elements. In order to take full advantage of low-code/no-code solutions while preserving the integrity and effectiveness of software development projects, a thorough vendor evaluation is an essential first step.

5. RESULT EVALUATION

After examining the top Low/No Code Development platforms available today, we discovered that advancing LNC technology requires a focus on AI and ML. It is also a conversion key from Low-Code to No-Code. As Chris Wanstrath predicted, rather of creating code and squandering a lot of time on implementation details, people can concentrate on high-level software prototyping and design in the future.[8] A time like this will come when strong coding AI & ML technologies have beautifully merged with Low/No Code technology. After contrasting Low/No Code with VB, Thomas Stiehm came to the conclusion that Low/No Code's future should be the same as VB's. The conclusion is flimsy and unlikely to come to pass—at least not in the manner Thomas thought. Children are among the many possible users of Low/No Code systems.

The field of Low/No-Code Development is very new, and its acceptance rate is increasing quickly, suggesting that the future looks promising. To address the shortcomings and problems with the current Low/No Code technology, researchers from all around the world should conduct additional studies and evaluations. To make improvements to their platforms, organizations can work with Low/No Code platform providers. Better user experiences for organizations are just as important as it is for the suppliers. [9] Vendors of Low/No Code platforms should constantly investigate and resolve restrictions by examining the work of rivals and attempting to integrate the newest technology with the platforms. Low-code and no-code platforms have established themselves as the most widely used software development option to solve the issue of a shortage of trained workers to meet all of the demands that businesses and organizations have in the modern world. But because of the high demand curve for workers in these locations, it is not yet seen as a fully feasible alternative to traditional growth. The sector is still in its early stages of adoption and development, despite the fact that many observers see a good evolution in the technological capabilities of Low- and No-Code platforms.[10] The most noticeable drawbacks of Low/No Code development platforms are the best indication of their early stages. Issues with applications' scalability, dependability, and obsolescence in terms of support, depth, and customization suggest that more work needs to be done in these areas before it can become a

significant contender in the years to come. [11] By 2024, these technologies are expected to be established with a penetration rate of up to 75% according to consulting firms like Gartner and Forrester. Nevertheless, it is still unclear if development organizations will be able to attain the necessary technological advancement and stay at the forefront of the industry.

6. CONCLUSION

The demand for progressive transformation within organizations creates an opportunity for Low Code/No Code innovation to demonstrate its value. The requirements for trade applications will become increasingly specific and complicated as digital transformation progresses. [12] While Low Code/No Code has advantages that enable companies to respond quickly to market movements, Low Code/No Code stages demand greater flexibility and customization. Although Low Code/No Code provides safe building squares that may be used to create secure apps, security concerns are raised by data breaches and the requirement for access to source code. [13] While maintaining Low Code/No Code applications should be simple given the minimal or non-existent code, vendor lock-in arises when suppliers stop supporting the stages. [14] In summary, Low-Code/No-Code Development will be critical to the digital transformation of the software development industry and lead to a turnaround, according to our analysis in each component. Systematic study on teaching approaches that can address this demand is found to be extremely critical, as there is little to no academic research on the topic and little of it focuses on the training needs of professionals who wish to be taught in this field. [15] It is to be expected, nevertheless, that any approach to the subject must consider the necessity of a multidisciplinary treatment of teaching, considering that in this new scenario, professionals in software development will be interested in training not only in the field of computer science but also in topics related to computer science.

REFERENCES

1. Sanchis, R., García-Perales, S., Fraile, F., & Poler, R. (2020). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences*, 10(1), 12. <https://doi.org/10.3390/app10010012>
2. Richardson, C. & Rymer, J. R. (2014, June). New Development Platforms Emerge For Customer-Facing Applications. Forrester. <https://www.forrester.com/report/New-Development-Platforms-EmergeFor-CustomerFacing-Applications/RES113411>.
3. Rizwan, O. A Snapshot of Programming Language History, Increment. April 2018. [Online]. Available online: (accessed on 12 November 2022)..
4. Outsystems. It Began with a Vision. 2001. [Online]. Available online: (accessed on 9 November 2022)..
5. HP, Computer History: A Timeline of Computer Programming Languages. 15 October 2018. [Online]. Available (accessed on 12 November 2022)..
6. Swinhoe. GitHub CEO: The Future of Coding is No Coding at All. 13 October 2017. [Online]. Available online: (accessed on 3 November 2022).
7. Vincent, P.; Iijima, K.; Driver, M.; Jason, W.; Natis, Y. Magic Quadrant for Enterprise Low-Code Application Platforms. Gartner. Rep. 2016..
8. Microsoft, "Microsoft News Center Latin America," Microsoft, 30 November 2021. [Online]. Available: <https://news.microsoft.com/es-xl/aplicaciones-low-code-de-microsoft-para-potenciar-la-transformaciondigital/>. [Accessed 1 June 2022].

9. B. Fernandez, "Silicon Technology Powering Business," 27 April 2022. [Online]. Available: <https://www.silicon.es/la-plataforma-de-appian-mucho-mas-que-low-code-2456612>. [Accessed May 20, 2022].
10. Appian, «Process Modeling Tutorial,» Appian, 2022. [En línea]. Available: https://docs.appian.com/suite/help/22.1/Process_Modeling_Tutorial.html. [Último acceso: 1 June 2022].
11. R. Martins, F. Caldeira, F. Sá, M. Abbasi y P. Martins, «An overview on how to develop a low-code application using OutSystems,» de 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, 2020.
12. Warren, N. (2018, October 9). Low-Code Myths, Fears, and Realities: Security. Outsystems. Retrieved December 11, 2021, from <https://www.outsystems.com/blog/posts/low-code-myths-security/>
13. Peterson, B. (2017, October 11). The CEO of GitHub, which caters to coders, thinks automation will bring an end to traditional software programming. Business Insider Nederland. Retrieved December 11, 2021, from <https://www.businessinsider.nl/github-ceo-wanstrath-saysautomation-will-replace-software-coding-2017-10?international=true&r=US>
14. Sarabyn, K. (2021). What's Wrong with Low and No Code Platforms? Pandium. Retrieved December 11, 2021, from <https://www.pandium.com/blogs/whats-wrong-with-low-and-no-codeplatforms>
15. M.Heusser, "Tech Target," 23 March 2021. [Online]. Available: <https://www.techtarget.com/searchsoftwarequality/tip/A-closer-look-at-6-popular-low-code-developmentplatforms>. [Accessed June 5 2022].
16. Woo, M. (2020). The Rise of No/Low Code Software Development— No Experience Needed? Engineering, 6(9), 960–961. <https://doi.org/10.1016/j.eng.2020.07.007>
17. Kulkarni, M. (2021). Deciphering Low-Code/No-Code Hype – Study of Trends, Overview of Platforms, and Rapid Application Development Suitability. International Journal of Scientific and Research Publications (IJSRP), 11(7), 536–540. <https://doi.org/10.29322/ij srp.11.07.2021.p11570>
18. ALSAADI, H. A., RADAIN, D. T., ALZAHIRANI, M. M., ALSHAMMARI, W. F., ALAHMADI, D., & FAKIEH, B. (2021). Factors that affect the utilization of low-code development platforms: survey study. Revista Română de Informatică Şi Automatică, 31(3), 123–140. <https://doi.org/10.33436/v31i3y202110>
19. Stangarone, J. (2019, May 21). Pros and cons of low-code development platforms - mrc's Cup of Joe Blog. Mrc's Cup of Joe Blog. Retrieved December 11, 2021, from <https://www.mrcproductivity.com/blog/2019/03/pros-and-cons-of-low-codedevelopment-platforms/>

