Open Journal of Applied Sciences, 2023, 13, 1864-1869
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Prospects of Using Low-Code in the Creation of Automated Systems

Conrad Onesime Oboulhas Tsahat, NGOULOU-A-NDZELI, Patrick Aurélien Ampiri Kwai

Ecole Nationale Supérieure Polytechnique, Université Marien Ngouabi, Brazzaville, Republic of Congo
Email: oboulhas@yahoo.fr, becker20000@yahoo.fr, pat_ampiri@yahoo.fr

How to cite this paper: Tsahat, C.O.O., NGOULOU-A-NDZELI and Kwai, P.A.A. (2023) Prospects of Using Low-Code in the Creation of Automated Systems. *Open Journal of Applied Sciences*, **13**, 1864-1869.
https://doi.org/10.4236/ojapps.2023.1310147

Received: September 25, 2023
Accepted: October 28, 2023
Published: October 31, 2023

Abstract

Ever-changing market conditions and a rapidly changing IT landscape call for fast and cheap ways to meet software demands. In order to tackle these problems, low-code development platforms (LCDPs) have emerged. These platforms are designed with the idea to limit recurring traditional hand-coding and programming. This article provides a theoretical overview of low-code solutions. The advantages and disadvantages of using LCDP in the creation of automated systems are considered. In conclusion, a conclusion is formulated about the prospects of using low-code technology.

Keywords

Low-Code, Coding, Automated Systems, Low-Code Development Platform

1. Introduction

Different development approaches assisted the acceleration of digital transformation: traditional, low-code and no-code development approaches. It is essential to realize the differences among them since they vary from each other in several features. Some of those properties are the size of the implementation team, the required capabilities, the prototype and the product development time, cost, modification, and investment risk [1]. In the traditional programming approach, the programmers must write a huge amount of code to develop an application. This causes a lot of costs in terms of time, money, and effort. The developer needs to have good experience and knowledge in the development activities and programming languages to be able to develop an application [2] [3].

Low-code is considered the fourth generation of programming languages. It is a visual integrated development environment that is based on automatic code generation and model-driven design [4] and defined as "a software that provides a development environment used to create application software through graphical user interfaces and configuration instead of traditional hand-coded computer programming". Finally, no-code development which is considered as a part of the LCDP market; but it focuses on building applications without writing any line of code, where only some text entry is needed for expressions, formulas and the other requirements for the application development that are held throughout the visual modelling and configuration [5]. The functionalities of the developed application using the no-code approach is fairly limited comparing to the other approaches. It is also considered by Gartner as a part of the low-code approach as both were developed to overcome the limitation of IT staff and the high demand on IT departments [3].

Low-code development has recently exploded as one of the biggest trends in software development [6] [7]. It can skyrocket engineer productivity, cut time to market, and decrease project costs. Due to the novelty of many NoCode and LowCode environments and the advantages of modern teaching methods over traditional technical literature, there are relatively few books on these technologies. These promises and predictions sound great, yet there is a severe lack of research looking into the performance of LCDPs in practice. There are several white papers which highlight the main benefits. Especially papers from the market research companies of Gartner and Forrester. Besides these sources, several research papers from LCDPs vendors themselves will be taken into consideration. We would especially like to note the following:

- The ELMA company, published the book in the public domain. This guide uses the currently relevant Low Code platform ELMA365, which will increase the efficiency of the company without crutches and code. This book briefly and clearly explain the features of automating already built-in processes, teach how to use the ELMA365 platform to solve practical problems, and also talk about how to properly maintain documentation [8];
- The book "*Building a No-Code/Low-Code Web Application On Airtable*, *Zoho Creator*, *Bubble and Caspio*" [9] is dedicated to the peculiarities of working with Airtable, Zoho Creator, Bubble and Caspio LowCode systems. This is not the author's first book on this topic: he previously published a tutorial on working with Google Sheets—"Anatomy of a Google Sheets Project". These frameworks allow you to create a quick, simple, and straightforward application to achieve your immediate goal. This is where fast, low-code or no-code platforms come in. Each development platform reviewed here is online, so they provide a platform for multiple clients to access the same application from anywhere there is an Internet connection. The book provides a detailed comparison and a detailed overview of each of the systems.

Currently, researchers use many "grey" sources to evaluate and analyze this new technology, limiting the accuracy and objectivity of the current scientific literature. The focus of this paper will be on the comparison of LCDP and traditional approches in specific or environments and development methods.

2. Comparison of Specific or Selected Environments

In order to understand the pros and cons of low-code environments compared to traditional development methods, we can compare writing websites in the traditional way (in HTML markup language) and low-code in the Figma engine. The result is shown in Table 1.

3. Comparison of Development Methods

Despite the global turbulence, the IT sector is currently experiencing a new round of rapid development. Particular attention of specialists is paid to low-code technology—experts predict that low-code solutions will take stable leadership positions in the IT market by 2024 [10] [11]. In this regard, an assessment of the prospects for using this development method in the creation of automated systems is presented not only an interesting, but also a relevant task.

Low-code is a technology for developing applications and websites in a visual interface with minimal manual programming (in some cases without it). In low-code platforms (Low-Code Development Platform, LCDP), the user creates software with the necessary functionality using visual modeling tools—moving blocks with ready-made code. Thus, the solution in question places significantly lower demands on users in terms of coding skills, which ideally fits into the paradigm of many owners of the largest IT companies, who believe that development and programming should cease to be highly specialized areas [12].

When designing and creating automated systems, specialists have considerable flexibility in their functionality. Professional developers use the low-code method to quickly develop simple software in order to focus their efforts on developing more complex applications that have greater meaning and value for the company. Non-IT employees with some programming skills also use low-code

Table 1. Comparison of HTML and Figmalow-code.

Average	Advantage(+)	Less(−)
HTML	High quality optimization	A greater barrier to entry
	The ability to easily connect user-needed styles and scripts (CSS, JavaScript, etc.)	The need to know the basics of programming
	Easy to work with complex elements	Slower development speed
Figma Low Code	Ease of learning	Low scalability
	Large set of templates and standard solutions	The Problem of Inaccurate Exporting
	Development can be done by a person who does not know the basics of programming	Improper operation of extensions in non-standard conditions

platforms to develop simple applications or extend the functionality of existing software [13].

Low-code technology describes the use of a variety of visual, flexible, and rule-based development tools and pre-configured elements to catalyze the software development and prototyping process, eliminating time-consuming manual coding. So low-code platform tools contribute to solving the existing problem of lack of programmers and developers, attracting specialists with non-core education, but with some experience in programming [14].

Low-code and No-code technologies are actively used for testing hypotheses—often small business owners do not have extra resources to conduct tests, but they need to learn about customer preferences. Low Code solutions are flexible—where templates cannot be used, we can add the necessary code.

Taking into account the described advantages, LCDP platforms seem to be an impeccable, ideal solution, however, this method is also characterized by a number of disadvantages.

First, LCDPs require a change in the company's culture, which in turn requires the understanding and approval of the manager [13].

Secondly, learning low-code requires time and considerable effort. Despite their apparent accessibility, LCDPs are not trivial; for example, nested loops are complex programming constructs and learning them is not an easy process.

Third, experience, resources and community support are currently very limited. Many platforms, due to their recent emergence, are currently immature. With traditional programming, the situation is different—there is enormous experience, developed communities, millions of programmers and developers, and many courses and literature are available for languages such as Java or C#. The newer the low-code platform, the more noticeable this problem is. In addition, many experienced programmers are conservatives who find it difficult to retrain on new platforms.

Fourth, enterprise LCDP platforms are generally expensive solutions. Low-code medium and small markets are less expensive, but also less scalable. Involving several platforms for a comprehensive solution to the problem of creating automated systems further increases the final cost.

So, in practice, low-code development is convenient and rational to use for simple, routine development tasks, but it performs poorly when working with complex and experimental tasks that traditional programming languages can handle. LCDP platforms make it possible to simplify the development of application solutions, providing the ability to design, modify and develop automated systems with a minimum of manual coding with a maximum visual component of development, so their use is an appropriate solution.

At the same time, LCDP has a number of significant problems that indicate that this technology is unlikely to become an equivalent alternative to traditional programming and development. But at the same time, LCDP has many advantages in creating ready-made template solutions for small and medium businesses: sites on Tilda, applications on Glide and Adalo.

4. Conclusion

Low-code covers a wide range of solutions in terms of creating automated systems, and the use of technology solves several urgent problems of the IT sector at once: it reduces the amount of labor-intensive manual coding, increases the productivity and efficiency of software development while reducing time costs, and reduces the severity of the problem of staff shortage.

It is obvious that IT companies that are less sensitive to trends and slow to use the latest technologies risk being ahead of more progressive competitors. Analysts at Gartner, Inc. call for a transformation of thinking towards process automation, since the introduction of low-code, combined with a composable product, will give companies the opportunity to make huge strides in the ability to adapt to changing market conditions and introduce innovation [14].

Thus, despite the existing limitations, low-code development in creating automated systems, seems to be a promising solution.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Moskal, M. (2021) No-Code Application Development on the Example of Logotec App Studio Platform. *Automatyka Pomiary*, **11**, 54-57. https://doi.org/10.35784/iapgos.2429

[2] Sanchis, R., García-Perales, Ó., Fraile, F. and Poler, R. (2020) Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences*, **10**, 12. https://doi.org/10.3390/app10010012

[3] Metrôlho, J., Araújo, R., Ribeiro, F. and Castela, N. (2019) An Approach Using a Low-Code Platform for Retraining Professionals to ICT. 11*th International Conference on Education and New Learning Technologies*, Palma, 1-3 July 2019, 7200-7207. https://doi.org/10.21125/edulearn.2019.1719

[4] Sahinaslan, E., Sahinaslan, O. and Sabancıoglu, M. (2021) Low-Code Application Platform in Meeting Increasing Software Demands Quickly: SetXRM. *AIP Conference Proceedings*, **2334**, Article ID: 070007. https://doi.org/10.1063/5.0042213

[5] Vincent, P., Iijima, K., Driver, M., Wong, J. and Natis, Y. (2019) Magic Quadrant for Enterprise Low-Code Application Platforms. https://fr.scribd.com/document/489488225/gartner-magic-quadrant-for-enterprise-low-code-application-platforms-august-2019

[6] Al Alamin, M.A., Malakar, S., Uddin, G., Afroz, S., Haider, T.B. and Iqbal, A. (2021) An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. 2021 *IEEE*/*ACM* 18*th International Conference on Mining Software Repositories* (*MSR*), Madrid, 17-19 May 2021, 46-57. https://doi.org/10.1109/MSR52588.2021.00018

[7] Beranic, T., Rek, P. and Hericko, M. (2020) Adoption and Usability of Low-Code/

No-Code Development Tools. *Proceedings of the Central European Conference on Information and Intelligent Systems*, Varaždin, 97-103.

[8] Loschenova, A. (2023) A Brief Guide to Creating Low-Code Solutions on the ELMA365 Platform. https://community.elma365.com/ru/forums/LC-book-platform/

[9] Love, P.E. (2020) Building a No-Code/Low-Code Web Application on Airtable, Zoho Creator, Bubble and Caspio. https://www.amazon.com/Building-No-code-Low-code-Application-Airtable-ebook/dp/B08MV8QDPB

[10] Vincent, P., *et al.* (2019) Gartner Magic Quadrant for Enterprise Low-Code Application Platforms. https://www.gartner.com/en/documents/3991199

[11] Margaria, T. and Steffen, B. (2020) Extreme Model-Driven Development (XMDD) Technologies as a Hands-On Approach to Software Development without Coding. In: Tatnall, A., Ed., *Encyclopedia of Education and Information Technologies*, Springer, Cham, 732-750. https://doi.org/10.1007/978-3-030-10576-1_208

[12] Magomadov, V.S. (2021) Low-Code and No-Code Platforms as a Way to Make Programming More Accessible to the General Public. *International Research Journal*, **6**, 100-103.

[13] Yakovlev, G.S. and Ivanov, F.F. (2020) Using Low-Code Platforms when Switching to a Process Approach in Creating Automated Systems. *Physical and Mathematical Sciences*, **30**, 120-126. https://doi.org/10.26117/2079-6641-2020-30-1-120-126

[14] Forbes (2022) Predictions: Low Code and the Future of Work. https://www.forbes.com/sites/forbestechcouncil/2022/02/28/2022-predictions-low-code-and-the-future-of-work/?sh=4bbf1964ecfb