Software Platforms Based on the Principles of Graphic Design, Automatic Command Generation and Visual Programming

Dražen Marinković, Zoran Ž. Avramović

Pan-European University APEIRON Banja Luka, Republika Srpska, BiH

(drazen.m.marinkovic, zoran.z.avramovic)@apeiron-edu.eu

A General survay

https://doi.org/10.7251/JIT2102110M UDC: 004.451.9:004.62.032.26

Abstract: This paper presents a new approach to software application development using a graphical interface. The approach is based on a combination of drag and drop elements and logic based on the model's own concept. Low code platforms and principles have been developed and are still being developed precisely to enable the rapid creation and use of applications that meet all the special needs and requirements of various organizations. No code platforms allow professionals and laymen to create applications via graphical user interfaces without any prior knowledge or qualifications in programming. However, code platforms are closely related to low code platforms because they are both created with a similar goal, based on a very similar way of working and almost the same principles of operation. Many vendors point out that the future of software development is based on configuration, not program. We believe that eliminating code is one way to bring development to higher standards in application development. One of the biggest advantages of the LC/NC platform is that they allow us to take advantage of innate problem solving and human skills by removing at least a significant number, if not all barriers to implementing software solutions in today's software world.

Keywords: low code, no code, visually integrated development environment, low-skilled people, professional developers.

Introduction

Low code/no code is an approach to software application development using a graphical interface. The approach is based on a combination of drag and drop elements and logic based on the model's own concept. This approach aims to increase the number of those developers who can participate in the processes of creating and using software to achieve a business goal or facilitate some part of the business process.

Development low code platforms are called visually integrated development environments (IDEs).

The approach to software application development according to the low code/no code concept generally follows the following steps: request determination, API selection, workflow creation, application development, data model, user interface using visual IDE, API connection, SQL queries custom software solution, software user acceptance testing, application publishing and upgrading as needed.

Development using the low code/no code platform helps low-skilled people to create applications or some of its components, without the need for the participation of professional developers. Low code development is also beneficial to professional developers, as it shortens the time needed to develop projects and opens up the possibility for them to distribute parts of their projects to low-skilled staff.

Low code

Low code development platform offers a working environment for creating applications using a graphical user interface, instead of traditional programming. The low code approach to application development reduces the amount of programming, enabling accelerated application development and delivery. The benefit of this approach is that a much larger group of people can participate in the development of the application.

Low code platforms come from fourth-generation programming languages and tools for rapid application development. Similar to their predecessors, low code platforms are based on the principles of model-based design, automatic command generation, and visual programming. Low code platforms came into use in 2011. The term “low code” was first used on 9 June 2014 by Forrester Research.

Increasing the need for automation and constant new applications for all types of business processes places a requirement on developers to create specific applications that are tailored to the specific needs of each organization individually. Low code platforms and principles have been developed and are still being developed precisely to enable the rapid creation and use of applications that meet all the special needs and requirements of various organizations.

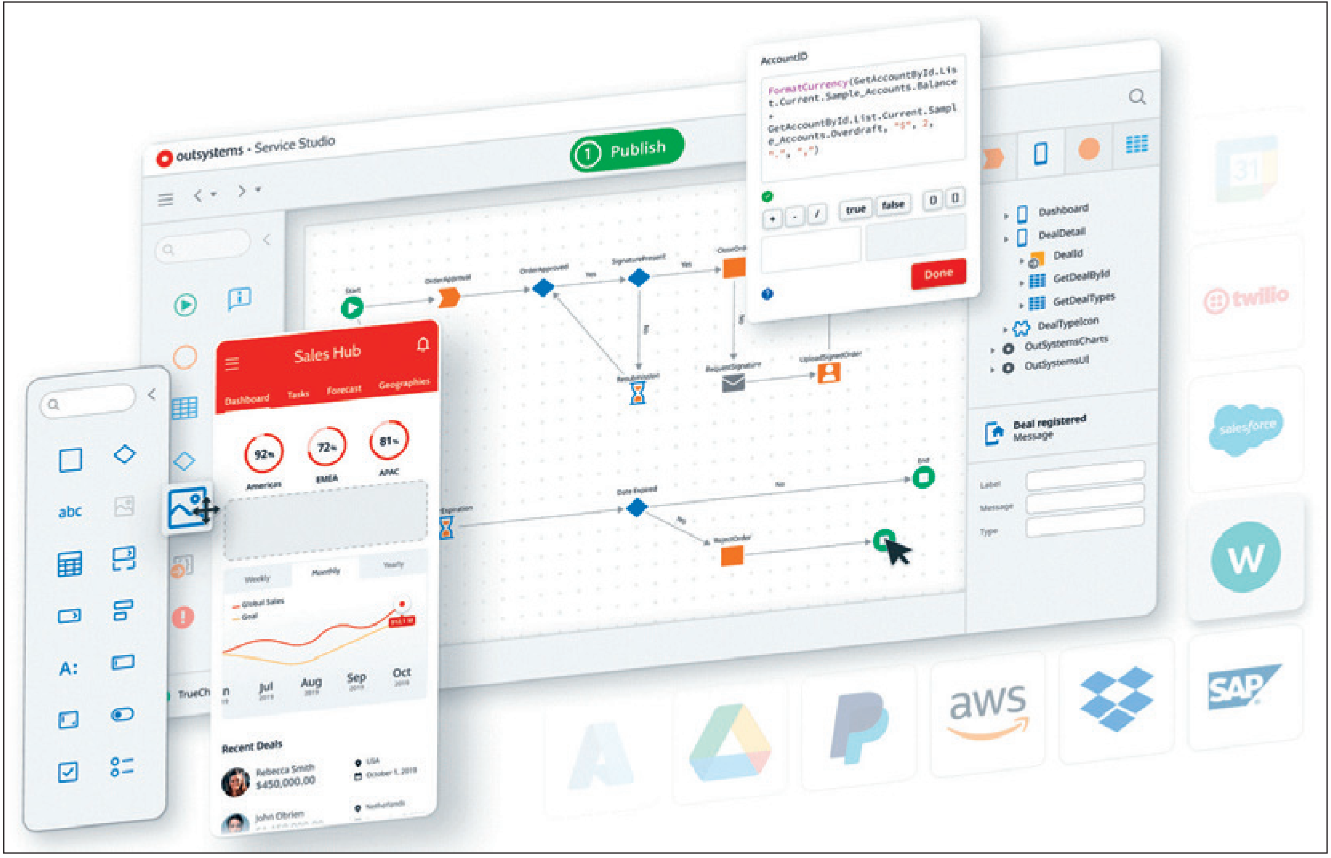
Some of the well-known platforms for low code development are:

- Quixy
- Visual Lansa
- Creatio
- GeneXus
- Appian
- Salesforce Lightning
- OutSystems

No code

No code platforms allow professionals and laymen to create applications via graphical user interfaces without any prior knowledge or qualifications in programming. However, code platforms are closely related to low code platforms because they are both created with a similar goal, based on a very similar way of working and almost the same principles of operation. The popularity of the no code principle of application development is constantly growing, primarily because it offers a huge number of possibilities without requiring any prior knowledge.

No code applications are used to meet the needs of companies that want to digitize processes through applications. Nevertheless, code tools are often created with the goal of adapting to business users, as opposed to traditional IT tools and technologies. This approach aims to accelerate the development cycle by bypassing time constraints, small budgets



AccountID

```
FormatCurrency(GetAccountById.List.Current.Sample_Accounts.Balance
+
GetAccountById.List.Current.Sample_Accounts.Overdraft, "$", 2,
".", ",")
```

Figure 1: *LowCode / https://www.outsystems.com /*

or insufficient resources to develop a complete software. The transition from traditional programming to abbreviated development methods is also making changes in the roles of traditional IT teams.

Some of the well-known platforms for no code development are:

- Zoho Creator
- Caspian
- Tigersheet
- Mendix
- Quickbase
- Retool
- Microsoft Power Apps
- OutSystems
- AirTabel

The difference between low code and no code

Low code application development is a convenient way for developers to develop applications quickly and with a relatively small amount of programming.

However, application code is created by users who do not need to know any programming language to create applications.

While no code platforms are more suitable for business users and non-professionals because the use of the platform does not require any programming knowledge, low code platforms are more suitable for more advanced users and more skilled developers.

Low code/no code in implementation

Practical implementation

There are, primarily, three types of applications that can be built on LC/NC platforms:

- **Portals and Web applications** - a good solution for those who want to create a personal or business website or blog.
- **Back Office applications** - provide ease of administration, automation of some routine or frequent tasks, creating reports
- **Mobile applications** - the most common and widespread type of application today, LC/NC provides companies with the opportunity to get closer to their customers in this way.

Advantages of low code/no code applications

- **Cost reduction**

With LC/NC platforms, development costs are greatly reduced, both due to the reduction of development complexity and the need to hire professional programmers.

- **Speed**

Development with LC/NC platforms allows companies and individuals to develop and modify their applications faster. The skills required to create applications using LC/NC platforms are much lower compared to traditional development.

Furthermore, all configuration processes, including the development of the program itself, its compilation and evaluation, and debugging, take place much faster in LC/NC development than in traditional.

- **Facilitated user experience**

LC/NC platforms automate many more operations critical to the user experience. The simplicity of these applications is reflected in their handling as much as in their creation.

- **Easier path from idea to realization**

By using the LC/NC platform for the development of applications specific to certain types of work, it is

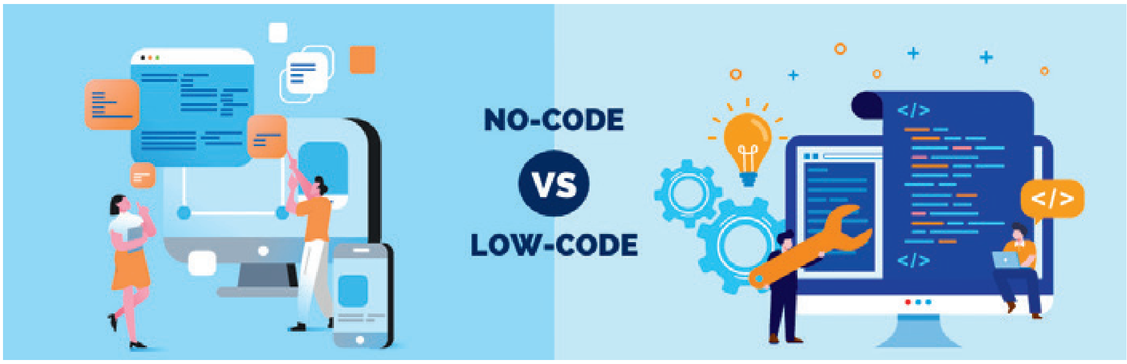


Figure 2: *Low Code No Code / https://quixy.com/blog/no-code-versus-low-code/*

possible to create an application for the person who should perform the work. In this way, the possibility of misunderstanding and disagreement between the one who creates the application and the one who should use it is eliminated. In this way, one of the frequent phases in the design and implementation of software systems is eliminated, and that is working with the client. With LC/NC applications, there is usually no need to collect user requests or user impressions.

- **Compatibility**

One of the main advantages of low code/no code applications is that they can run on almost all operating systems, and most types of devices.

Disadvantages of low code/no code applications

- **Third party dependence**

With a low code/no code platform, the operation of the application depends on the owner of the platform almost as much as on the one who creates it. The creator of the application is not able to work with any background functions or to release updates. For all such jobs they depend on the owner of the platform and their legal engagement in case of any problem.

- **Unable to customize**

Low code/no code platforms generally offer very few options for developing custom software or custom applications.

LC/NC platforms have very few options for integration with existing software solutions, which can be a significant problem for companies or individuals who already have a large amount of data contained in a previously used application.

- **Lack of developers**

The number of developers who have experienced low code/no code platforms is very small, and the number of other potential developers is even smaller. Companies and individuals who want to develop applications through these platforms are usually doomed to their abilities and the very small number of available tutorials and help forums.

- **Impossibility of quality creation of larger software systems**

The low code/no code approach to application development is a good solution for small or medium applications, but complex, large applications with many special functions and procedures still remain strictly in the domain of traditional programming and large, familiar programming languages.

Comparison of LC/NC Approach to Traditional Programming

While low code/no code application development platforms allow developers, users, and businesses to quickly develop applications using a variety of visual interfaces, traditional application development methods use conventional methods that allow developers to create large and powerful applications. Some of the basic differences in these approaches to development are:

- **Prerequisites required**

The traditional approach to application development generally involves the need to know the various tools and functions that developers use in creating complex applications. To be used properly, these applications require highly technical and specialized skills in this area.

Low code/no code platforms usually have a small set of tools that help develop different types of applications without much trouble. Application development with this approach becomes much easier for non-professional users, but the type and complexity of applications that can be developed still remains on the side of traditional programming languages.

- **Application quality**

Due to complex technologies and a very elaborate nature, traditional application development platforms usually produce errors and are themselves often quite complex, making it difficult for users to use applications.

In contrast, low code/no code platforms generally do not produce application design errors. This makes it easier to handle the application on the user side, but only applies to some smaller, more general applications.

- **Price**

Traditional software is quite expensive to build and can range from a few hundred KM to several hundred thousand KM for application design and development, depending on the size and complexity of the system. This is generally a better option for large companies that need extremely complex and specific software.

LC /NC platforms are much cheaper than traditional software for most companies. This is primarily because only access to the platform is paid for, not the development of the application itself.

- **Speed of development**

Traditional development platforms generally have a complicated setup system, which makes them less agile. Due to the complex codes, it takes a lot of time to learn and use them correctly.

By comparison, code platforms are much easier to use and operate, due to their drag and drop functions. Unlike traditional development, the code does not require any program development, but everything is done through inferfaces and already defined elements.

- **Maintenance**

In traditional programming, all kinds of modifications and upgrades go through the developer or team that developed that application. This process usually requires the adaptation of both the technologies used and the program itself. The software of the LC/NC application is usually maintained by the one whose platform it is, which makes the matter much easier for the company or the developer.

On the other hand, traditional software in most cases comes with professional support in charge of all kinds of corrections, troubleshooting and upgrades. In LC/NC applications, part of the job would be the responsibility of the company or individual who developed the application, which, depending on the problem, can be quite a time consuming process.

Conclusion

Without program development, organizations can separate the benefits of software from programming pitfalls. This paradigm is more important than ever because traditional methods have proven incapable of keeping up with the expectations of today's modern affiliate.

Many vendors point out that the future of software development is based on configuration, not program. We believe that eliminating code is one way to bring development to higher standards in application development.

One of the biggest advantages of LC/NC platforms is that they allow us to take advantage of innate problem solving and human skills by removing at least a significant portion, if not all the barriers to implementing software solutions in today's software world. Simply put, LC/NC increases the potential for innovation. This should resonate with organizations that have realized the need to become software-driven if they are to survive, or even thrive, in the years to come. Finally, the benefits of the LC/NC platform are crystallized in their ability to empower the entrepreneurial people in the organization to unleash their visions, ideas and creativity with minimal support through the construction of smart software solutions.

References

[1] "10 best low-code development platformas in 2021," 27.09.2021 [Online]. Available at: https://www.softwaretestinghelp.com/low-code-development-platforms/. [Last accessed 01.10.2021].

[2] T. Shukla, "8 best no-code platforms for building your app," 10.08.2021 [Online]. Available at: https://tigersheet.com/blog/the-8-best-no-code-platforms-for-building-your-app/. [Last accessed 30.09.2021].

[3] B. Kumar, "Low code vs traditional development," 02.06.2020 [Online]. Available at: https://www.zoho.com/creator/decode/low-code-vs-traditional-development. [Last accessed 30.09.2021].

[4] "No Code/Low Code Vs. Traditional Development - Which Team Should You Pick?," [Online]. Available at: https://marutitech.com/no-code-low-code-vs-traditional-development/#Low-CodeNo-Code_Application_Development. [Last accessed 01.10.2021].

[5] "Low-code development platform," 23.08.2021 [Online]. Available at: https://en.wikipedia.org/wiki/Low-code_development_platform. [Last accessed 29. 09.2021].

[6] "No-code development platform," 23.07.2021 [Online]. Available at: https://en.wikipedia.org/wiki/No-code_development_platform. [Last accessed 29.09.2021].

[7] "What is low-code?," 2021 [Online]. Available at: https://www.creatio.com/page/low-code. [Last accessed 29.09.2021].

[8] "No-code/low-code development," 27.07.2021 [Online]. Available at: https://golden.com/wiki/No-code_%2F_low-code_development-NMGMEA6/activity. [Last accessed 30.09.2021].

[9] M. Azoff, "Why low-code/no-code is on the rise," 05.08.2021 [Online]. Available at: https://sdtimes.com/lowcode/why-low-code-no-code-is-on-the-rise/. [Last accessed 01.10.2021].

[10] B. McHugh, "What are the benefits of low-code IT development," [Online]. Available at: https://www.advsyscon.com/blog/benefits-of-low-code-it-development/. [Last accessed 30.09.2021].

Received: September 27, 2021
Accepted: November 18, 2021

About the authors

Assistant professor **Dražen Marinković** was born on 1978 in Banjaluka. He finished elementary school and high school in electrical engineering. The Faculty of Business Informatics at the Pan-European University APEIRON Banjaluka enrolled in the academic year 2009/2010. years and graduated with an average grade of 9.81, defending a master's thesis entitled: "NoSQL databases in theory and practice" with a grade of 10. Academic 2015/2016 enrolls in doctoral academic studies of the third cycle at the Pan-European University APEIRON in Banjaluka, at the Faculty of Information Technology, and in the doctoral program "Information Systems in Communications and Logistics" where he passed all exams with excellent grades - excellent, 10. During his doctoral academic studies, as an author and co-author, he published 15 scientific and professional papers, two of which were published in international peer-reviewed journals.

Professor emeritus **Zoran Avramović** was elected a full member of the Russian Academy of Transport (RAT, St. Petersburg, Russia, since 1995), the Russian Academy of Natural Sciences (RAEN, Moscow, Russia, since 2001), the Serbian Academy of Engineering (IAS, Belgrade, since 2004). (formerly: Yugoslav Academy of Engineering - JINA) and the Academy of Electrical Engineering of the Russian Federation (RAEN, Moscow, Russia, since 2007). He is the scientific secretary of the Electrical Engineering Department of the Engineering Academy of Serbia. So far Avramović has published: 356 scientific and professional papers.

For citation

Dražen Marinković, Zoran Ž. Avramović, Software Platforms Based on the Principles of Graphic Design, Automatic Command Generation and Visual Programming, *JITA – Journal of Information Technology and Applications Banja Luka*, PanEuropien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 11(2021) 2:110-115, (004.451.9:004.62.032.26), (DOI: 10.7251/JIT2102110M), Volume 11, Number 2, Banja Luka, December 2021 (69-140), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004