# Recent Trends in Software Development: Low-Code Solutions

**Chapter** · January 2022

DOI: 10.1007/978-3-030-89912-7_41

2 authors:

Ryan Benac
Augustana College
**1** PUBLICATION **0** CITATIONS
SEE PROFILE

Tauheed Khan Mohd
Augustana College
**20** PUBLICATIONS **15** CITATIONS
SEE PROFILE

# Recent Trends in Software Development: Low-Code Solutions

Ryan Benac[1] and Tauheed Khan Mohd[2]

Augustana College, Rock Island, IL 61201

**Abstract.** This paper mainly serves as a summary of the current software development trend of low-code platforms. It discusses many uses of the applications, company created low-code programs, and risks and benefits in using low-coding solutions. Low-coding applications discussed include Xatkit, ktrain, RESTsec, and applications in machine learning, the Internet of Things, and mobile application development. This paper also examines several leading companies in low-code applications including OutSystems, Microsoft, and Salesforce. Each of these companies are shown to be increasing in the efficiency of their low-code solutions and are continuously growing and evolving. Although each company continues to evolve, there are differences in their uses mainly rooted in how the applications interact with other products offered by each company. The findings show that low-coding is still gaining momentum and will continue to increase and develop in the future. This may in fact be the future of most coding experience. Future plans for this research include a continual analysis of applications and overall impacts of this trend as low-code applications continue to grow and develop in usability and scaleability.

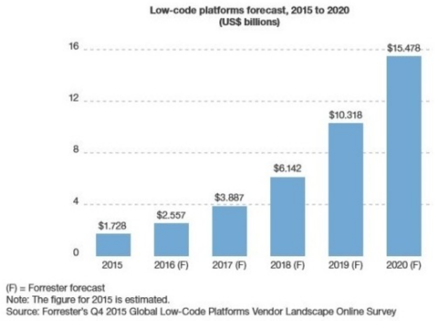
**Keywords:** low-code, software development, applications, software companies, coding, computer science

## 1 Introduction

Software development methods and practices have been expanding and improving over the last several years and will continue to do so. One such recent trend in software development is the push for low-code or no-code applications. Low-code applications are easy to use visual environments that can be used by developers to build applications without needing extensive coding knowledge or experience [1]. No-code applications are a branch of low-code methods whose main difference is in the name: no-code should be required for developers to create the programs and applications [2]. This recent trend in software development allows for almost anyone to easily create applications quickly and efficiently with minimal coding knowledge.

The trend of low-coding applications is actually an extension of model driven engineering. This type of engineering uses models, also known as visuals, as an attempt to reduce the need for complex and extensive coding [2]. Although both use models, low-code is slightly different. This extension is more restrictive because it targets a concrete type of software application [2]. This means that each portion of low-code development needs to focus on a particular concrete software application, and each low-code application needs to be tailored to those needs. One example of a low-code application can be recognized by website designers. Many times, websites need to be created by coding in HTML and CSS, but using Adobe DreamWeaver, a user doesn't need to type any code at all; they simply rely on a visual interface to create their website. This is also the case for many machine learning programs, but these will be discussed in detail later in this paper.

As this trend continues, the developers continue to find useful solutions to problems using these applications. In fact, a 2018 survey found that twenty-three percent of global developers used low-code platforms and another twenty-two percent planned to implement them within a year [3]. These companies will find these applications quite efficient and useful. In fact, the following visualization shows the total projected revenue from just five years of data [4].



Low-code platforms forecast, 2015 to 2020
(US$ billions)

(F) = Forrester forecast
Note: The figure for 2015 is estimated.
Source: Forrester's Q4 2015 Global Low-Code Platforms Vendor Landscape Online Survey

Fig. 1: There is an increase in forecasted revenue for low-code platforms that reaches over 15 billion dollars by 2020 [4].

Figure 1 shows the revenue nearly doubling each year to reach a peak in 2020 of 15 billion dollars. This upward trend will no doubt continue as software developers continue to create new applications and develop better practices.

This paper will focus on summarizing the recent trend of low-code applications through developers. It will explain some categories of programs that these applications will fall into. Then, it will discuss and analyze some of the leading companies that are developing and producing these applications for users. Finally, it will discuss low-coding as a whole, and it will explain several of the applications' benefits and risks as well as what those mean for the future of this software development trend. The following section will analyze several applications of low-code solutions and the benefits of each solution.

## 2 Findings

This section will mainly focus on the utilization of low-coding in both created programs used for specific applications and companies that are leading the creation of these programs. Figure 2 shows the presence of several top leaders in the low-code arena [3].

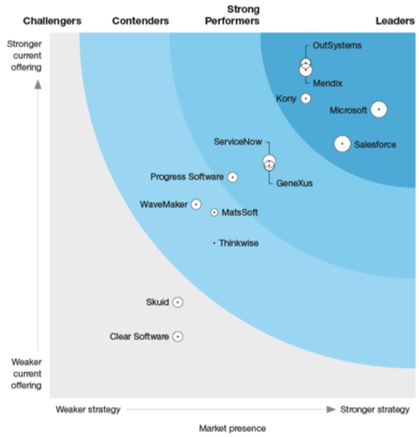


Fig. 2: Leaders in market presence for low-code solutions and software include Outsystems, Microsoft, and SalesForce [3].

Many of these companies will be discussed in detail in the second portion of this paper.

Low-code applications can be used for a wide variety of needs. When a company decides that they need a computerized solution to an issue, their need will likely fall into one of the following five categories: general-purpose, process application, database application, request-handling, or mobile application platforms [4]. Selecting one of these categories will depend on the requested product features and the vendor ambitions for the project. In any case, low-code solutions provide features for common use cases that allow a vendor to add depth including web and mobile user experience, page navigation, basic data and management reporting, and even incorporation of machine learning and case and content management [3]. The following are examples of low-code utilization for specific needs and purposes.

## Platforms and Applications

### 2.1 Xatkit: Chatbot

One example where low-code applications can be utilized to create functional and helpful programs is for bots. Xatkit provides a set of DSL, domain specific languages, to define mainly chat and voice bots [5]. These applications are increasingly adopted in e-commerce and customer service environments and can be used as a direct communication channel between customers and end-users [5]. These frameworks allow for easier access to help for customers as well as increased efficiency and time management of employees. On the technical side, it also allows for easier method definition and deployment of applications. However, because of the need for increased understanding of back-end applications, there is increased development and maintenance costs for this application [5]. Xatkit is available online as an open source download.

### 2.2 ktrain: Augmented Machine Learning

As machine learning programs continue to evolve and advance, there is a greater need for visual, low-code based solutions that allow non programmers to create machine learning models. One such program to do this is called ktrain. This application contains a Python library for machine learning accessibility [6]. This program provides features that support text, data, graphic, and tabular data in a sophisticated and unified interface that allows the user to create models for a wide range of data in as little as three to four lines of code [6]. This allows for any machine learning developer, regardless of coding knowledge or experience, to build and train models and assign computers to do both supervised and unsupervised tasks [6]. Low-coding applications will help advance the machine learning field much quicker because it reduces gate keeping and allows for a scientist in any field to train and build their own models that suite their specific needs.

### 2.3 RESTsec: Security in Enterprise Solutions

There is an increasing need for low-code solutions in enterprises, namely in their security. Businesses are currently more likely than ever to switch to an online business model because it will allow them to reach more users, increase their customer base, and eventually increase revenue and profits [7]. RESTsec is a low-code platform that supports the rapid security requirements that companies will need to secure their data as they use an online business model [7]. A low-code solution would be best for this scenario because it does not require extensive coding experience to create these solutions. RESTsec allows for developers to deliver secure and by design enterprise solutions in a rapid and automated manner [7]. This is just one example of a low-code solution that can be used in a business enterprise.

### 2.4 Mobile Application Development

Low-coding can also be used in the development of Mobile applications. It is predicted that by 2021, the demand for mobile application development will increase about five times faster than developers can deliver applications [8]. There is also a chronic shortage of IT personnel with the appropriate skills to quickly release extensive IT applications, let alone perform updates or maintenance on them [9]. Because of the demand for new applications at such an immediate and high rate, low-coding has been found to be a solution that allows for applications to be created quickly and efficiently. Low-code and no-code solutions remove the need for extensive programming experience which allows anyone to create a mobile application. One such tool used to achieve this is the Smart Maker Authoring Tool. This tool allows a developer to use Word or Excel to create applications instead of typing code into a shell [8]. Another tool that inexperienced developers can take advantage of is the Web Ration Mobile Platform. Features of this tool include model checking and full code generation that produces mobile applications only requiring a small amount of polishing [9]. These tools offer a low-code solution with a massive impact on the creation speed of applications.

### 2.5 Internet of Things (IoT) Solutions

The Internet of Things is another recent trend in software development. The Internet of Things, or the IoT, is a collection of automated procedures and data that interact with each other and with their environment to reach common goals [10]. Figure 3 is a visual representation for how the IoT works and interacts with its surroundings [10].

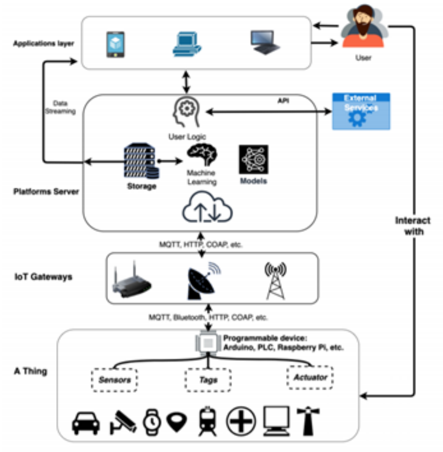


Fig. 3: How each component of the Internet of Things is related [10].

Figure 3 shows how one "thing", such as a smart device, interacts with other things to reach a common goal through IoT gateways, platform servers, external services, data streaming, application layers, and even the user of the application.

Low-code solutions to the IoT allow for users to program their devices for their own specific needs. However, there are a few drawbacks for using these methods. There is a wide lack of standards to support the model-based IoT systems, so each one much be tailored for a specific purpose [10]. There have also been several issues among many large industrial companies producing these platforms because the simple tools of low-code design cannot be implemented as a single abstract entity but instead as individual ones [10]. This requires more time and money for a company to achieve the task. Further, most approaches for the IoT's low-code solutions rely heavily on run time environments like Eclipse and OSGi [10]. These are not however available on most devices. Although low-code solutions are helpful in many instances, it may seem that widespread use of the IoT is still a ways in the future.

## Leading Companies in Low-Code Applications

Low-code solutions are offered by many vendors across the globe. Figure 4 shows just a few of the leaders in producing these trending applications [11]:

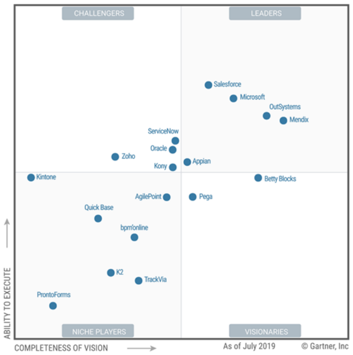


Fig. 4: The magic quadrant relating ability to execute to the completeness of vision for a given company [11].

It shows the companies that have both a high ability to execute and a completeness of vision include Salesforce, Microsoft, and OutSystems. There are quite a few other companies that are producing applications, but the diagram suggests that they are falling behind, especially in comparison to the leaders. These top three are major companies that are pushing the trend of moving toward low-code applications.

### 2.6 OutSystems

There are several leaders in the low-code application field. One of these leaders is OutSystems. The OutSystems Platform is a development environment used to quickly build and validate web and mobile applications [12]. They allow for users to model different perspectives including interfaces, data models, custom business logic, and the construction of process models [12]. They are likely one of the top performers in this field because their products are all user based meaning that the low-code solutions are created around the needs of the user. These solutions generally come through visual interfaces and business logic with easy integration of data stores and services [13]. However, simple abstractions of the created code make it difficult to integrate all of the diverse data and process models [13]. The OutSystems team is also working on another method to add to their low-code solutions. They are creating a polyglot solution that can add a data access layer that will combine the SQL database language with optional code snippets [13]. This is the next big step for the company.

### 2.7 Microsoft

Microsoft is another leading company in the creation of low-code platforms. Although they are in the early stages of their low-code programs, Microsoft takes advantage of their own Excel program to create an interface in PowerApps that uses an Excel-like expression language and a drag-and-drop approach for designing canvas apps [11]. This company has many strengths associated with their solutions, however, there are a few details that users should be weary of. One of these is their sales execution. Microsoft has switched to their Office 365 platform that requires a paid subscription to use their services [11]. Further, Microsoft does not yet support unnamed external users or less predictable usage patterns [11]. This is something to keep in mind when choosing a low-code provider. It appears however that they are still doing well with customers, as Microsoft reported above-average satisfaction rates and an excellent strategy for product creation [11]. Their revenue and business plan is likely the explanation for how they are able to continue funding and expanding the development of their low-code applications.

### 2.8 Salesforce

Salesforce is another major leader in low-code application development. Their major platform, the Lightning Platform, focuses mainly on applications that relate to customers, and it is supported by over five thousand third-party applications [11]. This allows Salesforce to integrate with a larger number of sources to better both customer application experience and scaleability [11]. There are however a few cautions that should be taken when using Salesforce applications. Although their business has experienced continuous growth for over twenty years, the company is integrating and diversifying their portfolio which complicates their level of innovation and evolution [11]. Also, again due to the recent integration into the field, Salesforce applications require developers to use both Lightning no-code and the Salesforce Apex language features in order to deliver any complex functionality of applications [11]. It is understandable how Salesforce is a leader in the fields of low-code and no-code applications, yet is still short of completely efficient products; they are still growing and evolving as technology and software development methods change.

## 3 Result

This paper mentions applications of low-code platforms in many fields including bots, machine learning, mobile applications, security, and the Internet of Things. Low-code is is a growing methodology that allows for rapid application, execution, and management of applications using declarative, high-level programming abstractions, such as model-driven and metadata-based programming languages [11]. This sounds as though there is great benefit in using low-code to create applications. However, there are some risks still involved. Companies are able to create an application at a high rate of speed, but they risk too many smaller vendors selling similar products which causes the loss of business from the customers who do not understand how each low-code platform will fit into their business model [4]. Currently, customers are having difficulty choosing because there are hundreds of platforms, not to mention specific purposed applications, for users to choose from [1]. As each vendor creates similar low-code environments, it makes it nearly impossible for a company to choose the best solution without proper guidance. Perhaps a solution to this issue is to create a detailed classification source to describe each existing low-code platform so that users can find what is best for their specific requirements. Further, other risks involve issues in transparency and the ability to scale the product. Increasing the size of the software development model leads to issues in their efficient interaction [14]. Perhaps one solution to this issue is to mix strategies in order to build a multi-stage model-management system [14]. This will allow companies to create applications that run more efficiently and are better able to be scaled for specific use.

Many of the previously mentioned risks have potential solutions. These solutions are still being considered and explored in order to better the evolution of low-code applications. In a study of twenty-five professional software engineers, it was found that an increase in transparency of new versions resulted in not only an improved developer experience, but also an increase in readability, usability, and scaleability [12]. Simply by creating a program that is more transparent, this allowed for immediate and noticeable improvement in the structure of the low-code applications.

## 4 Conclusion

Low-code applications are increasingly prevalent in coding frameworks and platforms throughout the world. This paper has provided several examples of how low-code environments can be applied to more specific fields including bots, machine learning, mobile applications, security, and the Internet of Things. There are also many companies that are driving the evolution of low-code applications including OutSystems, Microsoft, and Salesforce. These companies and their low-code applications may involve many risks, but this is simply because the low-code trend in software development is relatively new and still being expanded. Software engineers are continually creating new ways to solve the issues that low-code solutions may present which, in turn, continuously result in better transparency, scaleability, and readability of the code. Low-coding is being promoted as the key infrastructure for the digital transformation of our society [15]. It will allow any user, regardless of coding experience and knowledge, to create programs and platforms that will better our society at a higher rate than ever before. Although currently a young trend in software development, low-coding will likely be the next big evolution of computer science.

Future plans for this research will include continually monitoring and analyzing new releases of low-code applications. It is likely that as more of these applications are developed, there will be more companies that develop and release new software. Future work will also include monitoring both the usage of low-code applications in a business setting and how much the projected revenue continues to increase. A limitation of this research is the budding growth that is this software development trend which has resulted in little information about low-code applications and small time frames of statistical results about the impact of the trend as a whole. It is difficult to predict the exact future of this trend other than the inevitable increase of its usability and scale.

## 5 Related Work

The purpose of this paper is mainly to summarize and partially analyze the impacts of low-code applications. All cited material should be used for further reading and explanation. If you are looking for a more in depth analysis of anything discussed in this paper, I would suggest reading "Positioning of the low-code movement within the field of model-driven engineering" by John Cabot [15] for a basic introduction to the combination of both low-coding and model-driven engineering. If you are looking for more specific platforms that enable users to utilize low-code programs, I would also suggest reading "Magic Quadrant for Enterprise Low-Code Application Platforms" by Paul Vincent and contributors [11].

## References

1. A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, "Supporting the understanding and comparison of low-code development platforms," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 171–178, IEEE, 2020.
2. J. Cabot, "Low-code vs model-driven: are they the same?,"
3. J. R. Rymer, R. Koplowitz, S. A. Leaders, K. Mendix, S. are Leaders, G. ServiceNow, S. Performers, W. MatsSoft, and T. are Contenders, "The forrester wave™: Low-code development platforms for ad&d professionals, q1 2019," 2019.
4. C. Richardson and J. R. Rymer, "Vendor landscape: The fractured, fertile terrain of low-code application platforms," *FORRESTER, Janeiro*, 2016.

5. G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: a multimodal low-code chatbot development framework," *IEEE Access*, vol. 8, pp. 15332–15346, 2020.
6. A. S. Maiya, "ktrain: A low-code library for augmented machine learning," *arXiv preprint arXiv:2004.10703*, 2020.
7. C. Zolotas, K. C. Chatzidimitriou, and A. L. Symeonidis, "Restsec: a low-code platform for generating secure by design enterprise services," *Enterprise Information Systems*, vol. 12, no. 8-9, pp. 1007–1033, 2018.
8. Y.-H. Chang and C.-B. Ko, "A study on the design of low-code and no code platform for mobile application development," *International journal of advanced smart convergence*, vol. 6, no. 4, pp. 50–55, 2017.
9. R. Acerbis, A. Bongio, M. Brambilla, and S. Butti, "Model-driven development of cross-platform mobile applications with web ratio and ifml," in *2015 2nd ACM International Conference on Mobile Software Engineering and Systems*, pp. 170–171, IEEE, 2015.
10. F. Ihirwe, D. Di Ruscio, S. Mazzini, P. Pierini, and A. Pierantonio, "Low-code engineering for internet of things: A state of research," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–8, 2020.
11. P. Vincent, K. Lijima, M. Driver, J. Wong, and Y. Natis, "Magic quadrant for enterprise low-code application platforms," *Retrieved December*, vol. 18, p. 2019, 2019.
12. H. Henriques, H. Lourenço, V. Amaral, and M. Goulão, "Improving the developer experience with a low-code process modelling language," in *Proceedings of the 21th acm/ieee international conference on model driven engineering languages and systems*, pp. 200–210, 2018.
13. A. N. Alonso, J. Abreu, D. Nunes, A. Vieira, L. Santos, T. Soares, and J. Pereira, "Towards a polyglot data access layer for a low-code application development platform," *arXiv preprint arXiv:2004.13495*, 2020.
14. J. Philippe, H. Coullon, M. Tisi, and G. Sunyé, "Towards transparent combination of model management execution strategies for low-code development platforms," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–10, 2020.
15. J. Cabot, "Positioning of the low-code movement within the field of model-driven engineering," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–3, 2020.