Low-Code/No-Code Platforms

Citizen Development

Bhavika Prasannakumar

Lekha Ajit Kumar

California State University, Los Angeles

LOW-CODE PLATFORMS - CITIZEN DEVELOPMENT

Abstract

The project report on Low-code/No-code platforms - Citizen Development introduces users to creation of enterprise level applications faster and with higher efficiency through LCAP(Low code application development) in comparison to the traditional applications. Low-code is a software development approach that requires little or no coding in order to build quality applications. It uses visual interfaces with simple logic and drag-and-drop features instead of using extensive coding languages. These intuitive tools allow users even with no formal knowledge of coding or software development to create applications for business. Hence, developing applications for citizen developers becomes easier with low-code platforms. As a result, the number of applications in an organisation grows, which, if not governed, may lead to Shadow IT apps (which is the use of information technology systems, software, applications and services without explicit IT department approval). This aims at solving this industry problem.

***Keywords*:** Low-code, No-code, Shadow IT, Citizen Development, Security, LCAP.

Introduction

The majority of software programs and applications are written using high level complex programming languages.Non-technical people do not have the same skill set as coders. This is where the importance of Low-code/No-code platforms comes into play.

Low-code/no-code makes it easier for IT companies to improve their productivity by accelerating development processes even with non-technical staff allowing them to build applications more quickly. Low-code frameworks' versatility makes it simple to transition from legacy methods, and digital transformation will proceed without complexity, allowing technologies to be implemented quicker than ever before.

So, in today's cloud/SaaS environment, anyone can be a developer. Many businesses find it necessary for citizen-developed apps to be governed for better efficiency and monitoring.

With the complexity of systems, software, networks, and hardware, IT organisations quickly become overburdened only keeping up with day-to-day operations and ensuring that the systems in place are operational. From the viewpoint of the business department, this often results in a lack of constructive IT advice and help. More often than not, this is perceived as either incompetence of the IT department in comprehending the requirement or indifference of the IT department to the needs of the User". This leads to the businesses trying to develop applications on their own which can turn out to be very chaotic and also difficult to secure and maintain. If this is paired with lengthy IT approval cycles for proposed solutions, as well as counter-proposals with much longer implementation times, IT is quickly cut out of the loop when new solutions are being considered[16].

Central IT should be able to identify a low-code platform that meets an organisation’s criteria and needs after conducting a comprehensive review of low-code platforms. A low-code framework can then be used to create solutions that meet market requirements more quickly alongside business users, bridging the gap and establishing a shared understanding[17].

Any technology incorporated into the company to deliver applications that have not been approved by a central IT department is referred to as shadow IT. Individual business efficiency tools (such as BI-tools, Access, or Excel spreadsheets), departmental spreadsheets, or even organisation-wide solutions that have slipped past IT's notice are examples. These aren't always rudimentary spreadsheets or databases, but they may also be enterprise-grade SaaS solutions obtained without formal IT inspection and control[14].

From modelling to coding, Shadow IT is the dark side of IT, where users are left with software that has not been accepted or handled by IT. Will low-code platforms be able to solve this issue? How do we encourage business users to create applications that meet their organisational needs? This is referred to by Gartner as Citizen Growth. Citizen Development is receiving a lot of attention, with investments leaning towards low-code platforms or citizen development[19].

Discussion

1. Literature Review

There have been a number of articles that have focused on low-code/no-code platforms. Research paper "**Programming Without Code: The Rise of No-Code Software Development**" [11] notes that these platforms allow people who don't know how to write code to create the same applications that any developer would. No-code creation has been regarded as a type of visual programming where the user can manipulate code elements via drag-and-drop user interfaces.
The research paper "**Case Study: The Use of Low-Code Creation Technology to Help Research Data Collection**" [21] discusses data collection, methods for collecting data, and how low-code development can be used to keep data collection secure and efficient in the healthcare industry.
According to the website blog "**Forrester**," [22], development teams will work faster and produce more applications by empowering "citizen" developers thanks to low-code platforms. According to the site, low-code developers are very likely to reach your security neighbourhood.
The website blog "**The Low-code Regular**" [23] provides daily news and perspectives from the world's leading low-code tech providers. This demonstrates that low-code platforms are common today, and people want to know about the latest developments.
The IEEE research paper "**Sagitec Software Studio (S3) - A Low Code Application Development Framework**" [24] discusses the Sagitec platform, which allows businesses to create broad enterprise level applications with features such as file processing, reporting, business process modelling (BPM), rule creator, and so on.
The research paper "**Supporting the understanding and comparison of low-code development platforms**" [25] by the IEEE compares eight different LCDP platforms to help users determine which one is better for their specific needs.
The research paper "**The End of 'Corporate' Governance: Hello 'Platform' Governance**" [26] examines the distinctive features of this new business model and its implications for regulation, notably corporate governance.

2. Why Low-code/No-code?

RAD is a type of Agile software development that prioritises rapid prototype releases and iterations. It gives enhanced flexibility and adaptability as developers can make adjustments quickly during the development process. Since RAD was not easily adoptable by citizen developers as it focuses more on complex solutions whereas LCNC platforms were easier due to their features.
Over the years, programming languages have progressed from 1GL to 5GL(present). In the 21st century, the 4GL systems emerged as low code environments to counter the problem of RAD. With this progression, the enterprises have transitioned towards low-code enablement due to the following reasons:

- User-friendly business
- Faster innovation & smarter work
- Enhance customer experience
- Improve operational efficiency
- Stay on top of ever changing risks & regulations
- Automate at scale
- Increase productivity

3. Low-code vs No-Code

Users frequently mistake low-code with no-code platforms since there is no apparent differentiation between the two. Even the major analysis companies appear to have difficulty distinguishing between them. Gartner's 2019 Magic Quadrant for Enterprise Low-Code Application Platforms included "no-code" capabilities as one of the report's inclusion criteria. However, Gartner claimed that no-code platforms were not included in its newest Magic Quadrant for Enterprise Low-Code Application Platforms, 2020 (LCAP). However, platforms or technologies that their manufacturers claim are no-code, such as Honeycode and AppSheets, were evaluated as LCAPs.
Low-code is a technique that allows developers of various skill levels to construct apps fast and with minimal hand-coding by dragging and dropping visual pieces of existing code into a workflow. Producing software with low-code is similar to building software in any other method, with the primary distinction being the sorts of

shortcuts available. Instead of hand-coding a user management system, learning the latest programming framework, or writing ten tests before writing a single line of code for your app, you go right to work on building something unique and important. Because they are not burdened by repeated coding or repeating work, skilled developers can work smarter and quicker with low-code. Instead, they concentrate on the 10% of an application that distinguishes it, architecting it entirely with their development expertise and abilities and leaving the grunt work to the low-code tool or platform.
Drag-and-drop visual development is also available in no-code solutions. Unlike low-code, they are primarily targeted at business people or other IT professionals who may not be familiar with programming languages but want to create an application for a specific use case—often for their own department. To put it another way, no-code enables organisations to provide teams with the tools they need to build apps without requiring formal development training. Everything the no-code vendor believes the user will need to create an app is already included in the tool. No-code solutions are similar to popular blogging platforms and e-commerce website design companies, in that they provide pre-built pages that you can use to launch your blog or business in minutes.

4. Objectives of the study

The supply-and-demand mismatch is widening as the world is embracing digitalization at a much faster pace than before. The rising demand for software with a limited skilled human resources, can to a large extent be met by improving the efficiency of the developers by adopting No-code/low-code programming paradigms. To combat this, more citizen developers should be encouraged to build applications; some of the goals that can be achieved include:

- To allow IT to ensure that low-code or no-code apps are safe and can be integrated with other systems.
- To allow protection through configuration rather than relying on citizen developers to meet security needs.
- To enable IT to better manage these apps and add more power while also aligning with LOB requirements.

5. Requirements

Functional requirements are generating **security code** side by side or on top of the application business logic code, allowing security configurations of all data sources to be used by citizen apps. One of the most important requirements for widespread adoption of these platforms by IT corporates is the assurance that the generated code and data is secure. The primary aim of the project is to demonstrate a governance platform where one can see the **citizen app landscape, operations of the user(audits), app metrics, integrations with other apps** and **setting access control for use by citizen apps**. The non-functional requirements include **performance, security, reusability, interoperability, maintainability** and **data integrity.**

6. Architecture of a low-code/no-code platform

Fig 1 shows a generalised architecture of a Low-code/No-code platform. The bottom layer consists of the data services in which data can be stored in a number of ways. The data store may be hosted on the cloud or locally on a data server which is on-premise.
When a user submits a request, the API call retrieves and uses data from the data store, regardless of its location. This is done by the Data services layer as shown in Fig 1.
The workflow engine and the app generator are both present in the logical layer which is the middle layer. In addition to these, all the functional modules are defined here which serves various purposes.
The topmost layer has the UI templates and integration points. The citizen developers can use these templates or create their own UIs based on their needs. This enables them to add new features or even create completely new apps.
Through an app governance platform, the administrator can supervise, monitor, and secure the apps developed by citizen developers.
To these, a security layer is added on top of and alongside the other layers, resulting in secure code which enables the safekeeping of information.

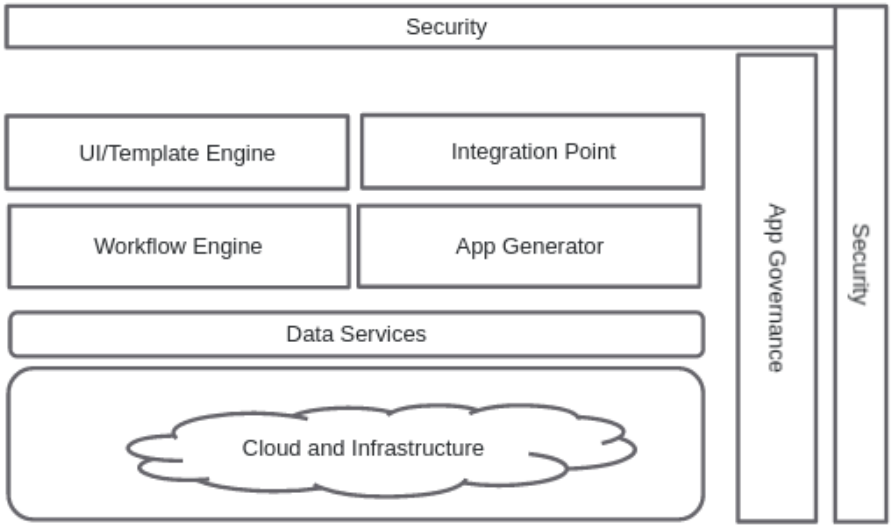

Fig 1. Architecture of Low-code/No-code Platforms

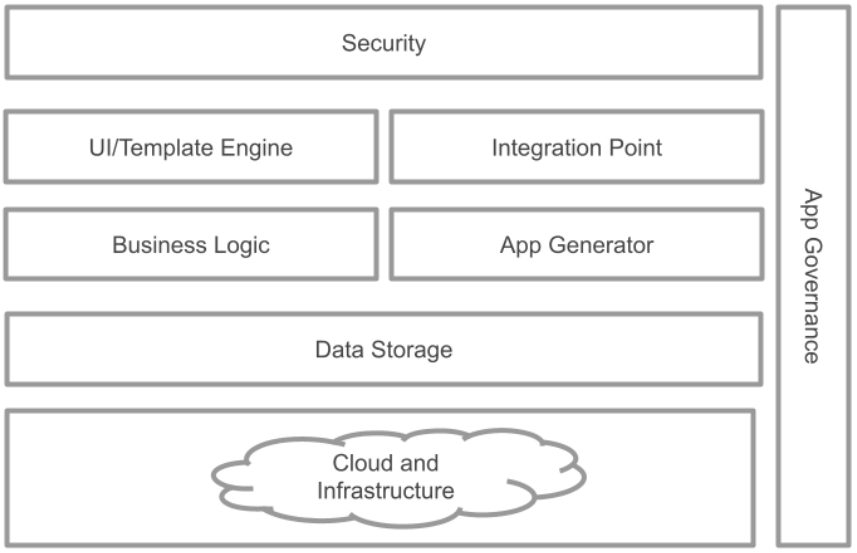

Fig 2. Modified Architecture of Low-code/No-code Platforms

IMPLEMENTATION

A thorough investigation of low-code platforms was conducted in order to gain a good understanding of their capabilities, functionalities, usage, application domains, market share, and future potential. Low-code architecture was first introduced in 2014. According to a Forrester study from 2019, the low-code market grew at a 40% annual pace, with spending expected to reach $21.2 billion by 2022. In 2019, Gartner added a Magic Quadrant for Enterprise Low-Code Application Platforms (LCAP).

7.1 Open source low-code/no-code platforms

Following the realisation of the value of low-code platforms, a few open source platforms namely, OpenXava, Joget and Skyve were investigated. Platform independence, open source code base operation, cloud adoption, domains supported, database used, security feature added, licensing, test framework used, web/mobile app support, responsive design, and database/file dependency were all factors considered.
Joget is an open source framework that makes it simple to create, run, and manage applications. Issues like it being a heavy platform with a big project size tend to slow down the platform's performance. In addition, joget necessitates further pre-configurations like setting it up in the local environment. The construction method is lengthy and it is difficult to add plugins due to a lack of adequate documentation.
OpenXava is another open source platform that uses Java-related technology. Owing to the lack of support for incorporating role-based access mechanisms, this was also eliminated. It is not possible to operate the platform locally. In addition, extensive configurations are needed to construct the applications, including the explicit addition of the Tomcat server.
Skyve is an open source framework that can be used to create complex and reliable applications. It's a versatile platform with a Java domain-driven architecture that focuses on role-based access control (RBAC) which is an existing feature. Skyve is platform agnostic, allowing for cloud adoption. It has a responsive architecture and supports several databases (H2, MySQL). TLS/SSL encryption, CORS, and user input sanitation have all been implemented.

7.2 Concerns including the use of low-code/no-code platforms

7.2.1 Security

Skyve offers Role Based Access Control (RBAC) mechanisms in the security layer. Along with this, OAuth2 and JSON Web Token (JWT) are integrated for authorization purposes. OAuth2 is an authorization framework that enables applications to obtain limited access to user accounts over an HTTP service. All the API requests have to pass this security layer and the token (JWT format) is shared among the applications during an API call.

7.2.2 Governance

Every IT organisation needs a centralised portal for viewing and monitoring all of their applications. For an organisation to be work-efficient, platform governance becomes important to be able to provide the best user experience to ultimately please both the user and the business. The low-code/no-code apps built by citizen developers usually lack security, for this, it becomes essential for them to be monitored which might otherwise lead to Shadow IT. The administrators can use the governance app to register these low-code/no-code apps and monitor them to have more control on the organisational data.

Some vital features that need to be included in a governance platform. Only low-code/no-code applications will be governed by this governance app. Figure 3 shows the schema diagram for the Governance app.
The basic functionalities of this app include:

- **Registering applications** - Particulars of the app such as app name, database dialect, a deployable war file of the application are captured using a form. The information is saved in a SQLite database after submission. Also, the other requirements to run the application are generated and stored locally on the administrator’s device.
- **View applications -** This is a dashboard view that shows all the apps registered on the SQLite database. Along with this, there is an added feature to start or stop the application manually with the help of a toggle button. The apps can be used from here when they are turned on. Upon selecting a particular app, the details of the apps are displayed which include the app name, database dialect, the date of registration, performance metrics and the audits in a tabular format. These can be obtained if and only if the application is running.

7.3 Performance Metrics

The Java Melody plugin is integrated with Skyve. Using this plugin, the performance metrics of the application are obtained. The rest API endpoints provided by Java Melody are called using an AJAX request from the flask application. On clicking the 'refresh' button, the AJAX call is made to the low-code platform and the metrics are refreshed onto the page. The metrics include, server, process id, start date, host name, operating system, number of available processors, java version used, process CPU time, average system load, system CPU load, thread count, used buffered memory, used non-heap memory, total used memory, maximum memory and garbage collection time. Along with these, graphical representations of the number of HTTP sessions, active threads, active JDBC connections, HTTP hits per minute and percentage of HTTP errors. These graphical representations can be viewed for time ranges of one day, one week, one month and one year.

7.4 Audits

The RDBMS interceptor Java class in Skyve is modified to capture the audits for all the operations of the applications (CRUD operations). Using the bizlets of Skyve, only the valid audits are saved into Skyve database. Subsequently, using the configured API calls, the audits are retrieved and displayed onto the governance app in the form of a table. New audits are appended onto the table in real time and also stored in the database for future use. Information such as the module name, document name, username, operation carried out on the data and the timestamp when the operation was carried out, are obtained for every audit.

7.5 Workflow

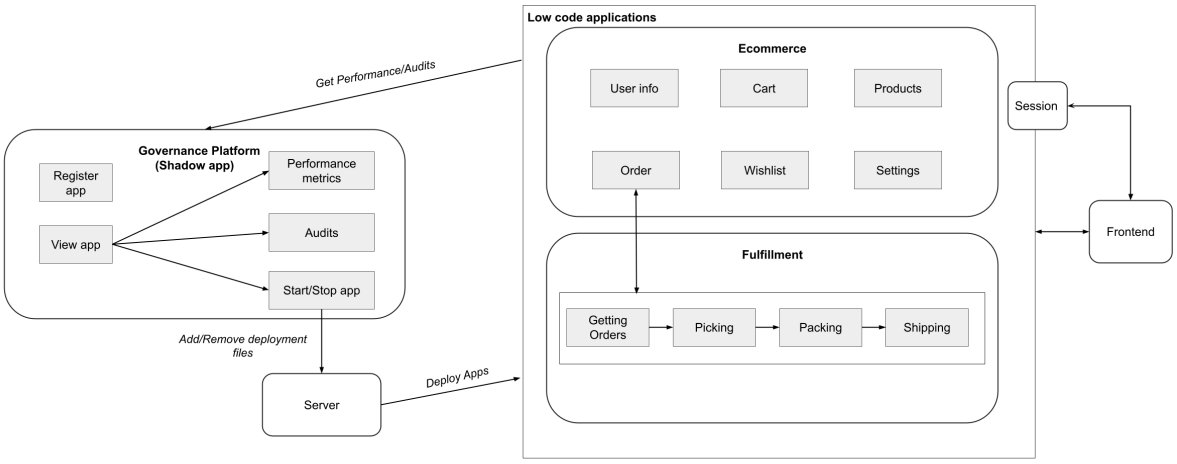

Fig 3. Workflow of the system

Figure 5 shows the workflow diagram followed in the project. Two Skyve apps namely E-commerce and Fulfillment (deemed as test cases) are developed using the modified low-code platform, Skyve, to demonstrate how the citizen developers can use the platform to build and deploy applications. These apps have a security layer alongside or on top facilitated by the OAuth2 and JWT security features. Along with these apps, a frontend ecommerce app is built using Angular. This app gets its data from the backend low-code ecommerce app by using the integrated REST services.

The governance platform is integrated as a Shadow IT app. All the low-code apps (here, e-commerce and fulfillment) are registered here for the administrators to have control and to monitor apps in the organization. The apps can be used only after registration. This platform is made interactive with the features as described in section 7.3. The governance platform uses the Wildfly server to deploy the registered apps. Each time an app is turned on/off by the administrator, the server restarts.

8. RESULTS

The main aim of this paper is to ease the work of citizen developers with monitoring of applications of any kind(including Shadow IT apps). The previously mentioned objectives are met in the following manner:

- **Integration with other apps** - the citizen developers can integrate the apps made using the modified low-code/no-code platform (here, Skyve). A demonstration of this feature is done by integrating the fulfillment app with the e-commerce app (both built using Skyve). Similarly, additional features can also be integrated with the low-code/no-code apps like the wishlist feature, which is integrated with the e-commerce app.
- **Generate security code alongside or on top of the business logic code for the application** - addition of the role-based access mechanisms as one of the access layers, to be able to detect valid and invalid access. Alongside, a security feature 'OAuth2' has been utilized for the authorization process of the customers.
- To **manage all the apps** in the organization, this work evolved a solution that is, a governance app. This includes the following features:
    1. The citizen app landscape (here, e-commerce and fulfillment)
    2. Operations of the user (audit)
    3. App metrics

CONCLUSION

Low-code/no-code platforms would be able to improve the speed at which applications are created. Since these applications are self-contained, citizen developers would be allowed to incorporate protection and monitoring functionality. More non-developers would be able to develop software which can be incorporated into the IT mainstream as the IT can govern the operations of the application and ensure that the systems are not compromised. This also eases the pressure on a company's technical staff.

Shadow IT is the use of IT-related software or hardware by a person or a department without the knowledge of the organisation's IT or security groups. Software, hardware, and other resources are all included. With the governance app, the project aims to combat Shadow IT by giving IT organisations more power over the visibility of their applications.

Users will not have to worry about security problems with the integration of new security features such as OAuth2, which can help them keep pace with the fast-developing market.

REFERENCES

[1] Gartner report-
"https://www.gartner.com/doc/reprints?id=1-24BBDEDZ&ct=201005&st=sb"
[2] Importance of Citizen Development and Citizen IT
"https://blogs.gartner.com/jason-wong/importance-citizen-development-citizen/"
[3]Evolution of Citizen Development
"https://techbeacon.com/enterprise-it/how-it-can-stop-worrying-learn-love-citizen-developer-movement"
[4]Shadow IT- "*https://en.wikipedia.org/wiki/Shadow_IT"*
[5]The Future of Citizen Development- "*https://www.gartner.com/en/documents/3970067/*"
[6]Considerations to adopt low-code development-
"https://www.computerweekly.com/feature/Gartner-What-to-consider-before-adopting-low-code-development"
[7]Low-code Development Platforms - *"https://www.trustradius.com/low-code-development"*
[8]Low-code platform Skyve- *"https://skyve.org/"*
[9]Low-code platform Joget- *"https://www.joget.org/"*
[10]ShadowIT-*"https://blog.kintone.com/business-with-heart/tackling-the-challenges-of-shadthe-future-of-apps-must-include-citizen-developmentow-it-with-no-code/low-code-solutions"*
[11] Rise of low code development -
"https://spectrum.ieee.org/tech-talk/computing/software/programming-without-code-no-code-software-development"
[12] Shadow IT defined - *"https://www.forcepoint.com/cyber-edu/shadow-it"*
[13] Rise of Low code - "*https://www.techadv.com/blog/rise-low-code-2020-and-beyond*"
[14] Low code and Shadow IT - "*https://www.genus.no/blog/low-code-and-shadow-it*"
[15] Introduction to Low-code development - *"https://www.creatio.com/page/low-code"*
[16] Shadow IT and Citizen Development -
"*https://tips.viscosityna.com/low-code-shadow-it-citizen-developers-forrester/*"
[17] Shadow it Disadvantages - "*https://www.myhubintranet.com/shadow-it/*"
[18] Integration of low code platform - "*https://www.xapix.io/post/low-code-integration*"
[19] Introduction to Citizen development -
"*http://cdn2.hubspot.net/hubfs/172645/eBook%20Introduction-to-Citizen-Development-Bringing-Shadow-IT-into-the-Light.pdf?t=1477583182996*"
[20] Low Code Platforms -
"*https://in.pcmag.com/migrated-30861-onlinecloud-backup-services/115106/the-best-low-code-development-platforms*"
[21] Case Study: The Utilisation of low-code development technology to support research data collection
-*"http://www.iacis.org/iis/2018/2_iis_2018_132-139.pdf"*
[22] Forrester - "*https://go.forrester.com/blogs/category/low-code-platforms/"*
[23] The Low Code Daily - "*https://low-code-daily.com/#/*"
[24]Sagitec Software Studio (S3) - A Low Code Application Development Platform
"https://ieeexplore.ieee.org/document/9102703"
[25]Supporting the understanding and comparison of low-code development platforms -
"https://ieeexplore.ieee.org/document/9226356?signout=success"
[26]The End of "Corporate" Governance (Hello "Platform" Governance) -
"https://sci-hub.se/https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3232663"
[27] Challenges and opportunities in Low code Testing - *"https://dl.acm.org/doi/10.1145/3417990.3420204"*
[28]Decentralized Governance for Digital Platforms - Architecture Proposal for the Mobility Market to enhance Data Privacy and Market Diversity - *"https://ieeexplore.ieee.org/document/9369659"*
[29] Influence on Shadow IT on innovation in organisations -
"https://www.researchgate.net/publication/309714077_Influence_of_Shadow_IT_on_Innovation_in_Organizations"
[30] Shadow IT and Business-Managed IT: Where Is the Theory? -
"https://ieeexplore.ieee.org/document/8808025"