Low-Code/No-Code Development

Ravishka Rathnasuriya
PhD Student

What is Low-Code/No-Code (LCD/NCD) Development?

Low-Code: Creating software with small amount of code by reducing the amount of traditional hand coding.

No-Code: Create software using graphical user interfaces and configuration instead of traditional computer programming. Closely related to Low-Code development platforms.

Applications are developed through model-driven engineering principles and take advantage of cloud infrastructures, automatic code generation, declarative and high level and graphical abstractions.

What platforms are used?

Bubble.io

Airtable

Wix (Editor X)

WordPress

Outsystems

Microsoft Power Apps

Google App Maker

Salesforce App Cloud

Shopify

Zapier

Glide

The part of applications developed by LCD	Platform name
Frontend	*Adalo, Alpha Software, Andromo, Anvil, App maker, Appgyver, Appian, Apprat, Appsheet, AppyPie, Backendless, BettyBlocks, Bildr, Bravo Studio, Bubble.io, CalcuBuilder, Carrd, Caspio, Draftbit, DrapCode, Dribble, DronaHQ, Elementor, Expression Blend, Figma, Fliplet, FlutterFlow, Kony Visualizer, Mendix, Noloco, Outsystems, PixelCraft, Pory.io, Power Apps, pxCode, Reach.at, Retool, Softr.io, Stacker, Storyboard, Tadabase, Thunkable, UI Bakery, Undaku, Unqork, V One, Verastream Host Integrator, Wappler.io, WaveMaker, Webase, Webflow, Weflow, WeWeb, Wix, WordPress, Zyro*
Workflow	*Activiti, Airtable, Amazon Honeycode, Appian, Automate.io, AwareIM, Azure Logic Apps, Bonitasoft, Boomi Flow, Bubble.io, Budibase, Camunda, DronaHQ, Drupal, Google Tables,Parabola, Intalio, Integromat, jBPM, Joget, Knack, MakerPad, Mendix, N8n, Node-RED, Outsystems, Pega, Pory.io, Power Apps, Power Automate, Quickbase, Salesforce, ServiceNow, Slingr, Stackby, Tadabase, TrackVia, Twilio Studio, UiPath Apps, Undaku, UnifiedAI, Unqork, Webflow, Zapier, Zoho*
Integration	*Adalo, Airtable, Alpha Software, Appgyver, Appian, AppMaker, Appsheet, AppyPie, AwareIM, Backendless, BettyBlocks, Bildr, Bravo Studio, Bubble.io, Caspio, Draftbit, DronaHQ, FlutterFlow, Knack, Memberstack, Mendix, NocodeAPI, Parabola, Pory.io, Power Apps, PrestoAPI, Quickbase, Retool, Slingr, Stacker, Syndesis, Tadabase, TrackVia, UI Bakery, Unqork, V One, WaveMaker, WeWeb, WordPress, Zyro*
Backend	*8base, Adalo, Alpha Software, Andromo, App maker, Appgyver, Apprat, Appsheet, AppyPie, Backendless, Bravo Studio, Bubble.io, Byteline, Caspio, Draftbit, DrapCode, DronaHQ, Easy Tables, Firebase, flutterflow, Kelp, Kinvey, Linx, Mendix, Noloco, Outsystems, Power Apps, PrestoAPI, Thunkable, WaveMaker, Webase, Webflow, WordPress, Xano, Zyro*
Framework	*Appsmith, Bildr, funkLang, Glide, Lowdefy, Orientation Aware Control, Picocli, QTKit, Remake, Sonata Admin, Substack, Wappler.io, WordPress*
Database operations	*8base, Alfresco, AwareIM, JayStack, Jhipster, Loopback, Sonata Admin, Stackby, Tadabase , TeamDesk, Webase, Zenbase*
Data visualization	*Google Data Studio, Kelp, Qlik, Tableau*

Programming Languages Used

Java

JavaScript

C#

Python

HTML and CSS

Objective C

PHP

C++

Architectural point of view

Consist of 4 main layers [2]:
1. Application Layer: Graphical environment
2. Service Integration Layer: Connect with different services by using corresponding APIs and Authentication Mechanisms

3. Data Integration Layer: data integration with different data sources
4. Deployment Layer: Deploy on cloud infrastructures or on-premise environments

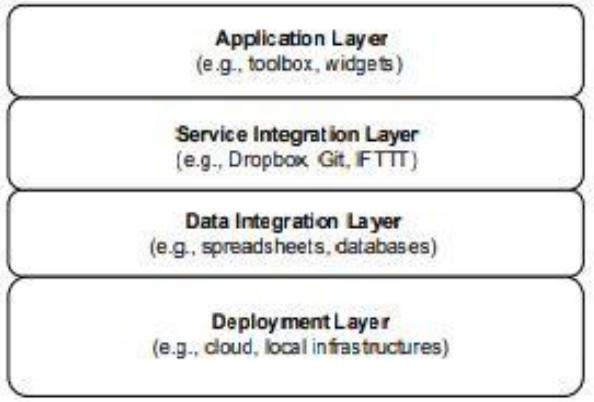

Fig. 1. Layered architecture of low-code development platforms

Main Components

Grouped into three tiers [2]:

1. Application Modeler: Enabling the specifications of applications through provided modeling constructs and abstractions. Define applications.

2. Middle Tier: Takes the application model received from Application modeler and performs model management operations like code generations and optimizations. Consider involved services including database systems, micro-services, API connectors, model repositories of reusable artifacts.

3. External services: services that are integrated with the platform.

Implementation Units

API (backend)

Template (backend and frontend)

Component (frontend)

Services (backend)

Framework (Eg: “Loopback is an awesome framework that offers a full REST API to all CRUD available operations with zero code [1]) (frontend and backend)

Widget (frontend)

SDK

Supporting technologies used

React Native : open-source mobile application framework

Node.js : free, open-source, cross-platform JavaScript runtime environment.

Apache Spark : Unified analytics engine for large-scale data processing

Server-side scripting : In web development that involves employing scripts on a web server and produces a response customized for each user’s request. Allow user to create website with zero code.

SOAP (Simple Object Access Protocol) : Messaging protocol specification for exchanging structured information in implementation of web services. (Eg: Bonita)

JSONata : open-source query and transformation language for JSON data.

Types and Kinds of Applications developed.

Mobile Applications

Web Applications

Integration Applications

Kinds of Supported Applications:

Event Monitoring

Process Automation

Approval Process Control

Escalation Management

Inventory Management

Quality Management

Workflow Management

Domains Used

E-Commerce

Business Process Management

Social Media

Customer Relationship Management

Content Management System

Extract-Transform-Load

Entertainment

Robotic Process Automation

Medical

Benefits

Faster development

Ease of study and use

Lower IT cost

Rich and ready-to-use units

Newbie friendly

Improved system quality

Strong integration and expansion capability

Minimal efforts

Better user experience

Improved IT governance

Limitations and Challenges

High learning curve

High pricing

Lack of customization

Slow loading and publishing

Less powerful than programming

Complex issues still need coding

No access to source code

Limitations for experience developers

Vendor lock-in

Difficulty of maintenance and debugging

Need of basic programming knowledge

Development Process [2]

1. Data Modeling: visual interfaces to configure data schema by creating entities, establishing relationships, defining constraints, and dependencies using drag-drop facilities.

2. User Interface Definition: Configure forms and pages to define application views. Define and manage user roles, security mechanisms. Drag-drop capabilities are important to speedup development and render different views quickly.

3. Specification of business logic rules and workflows:
   Manage workflows among various forms and pages requiring different operations.

4. Integration of external services via third party APIs:
   Provide ways to consume external services through different APIs.

5. Application Deployment

Approaches to Built in Software Applications

- UI to Data: Starts building the application by creating user interfaces and then linking it with needed data sources.
  Eg: Mendix, Zoho Creator, Microsoft Power Apps
- Data to UI: Data-driven approach. Starts from data modeling and then builds the user interface of the application
  Eg: OutSystems, Salesforce App Cloud, Microsoft Power Apps, Appian.

Capabilities that a LCD/NCD platform can Offer and their features[2]

- Graphical User Interface: Represents provided functionalities available in frontend.
  Features: Drag-and-drop tools, forms, point and click approach, pre-built dashboards, built-in workflows

- Interoperability support with external services and data sources:
  possibility of interacting with external services. (Dropbox, Zapier, Office 365)
  Features: Interoperability with external services, connection with data sources

- Security Support: Security aspects like authentication mechanisms, adopted security protocols.
  Features: Application, Platform security

- Collaborative Development Support: Collaboration models.
  Features: on-line, off-line collaborations

- Reusability Support: Enable to reuse already developed artifacts.
  Features: Built-in workflows, Pre-built forms/ reports, pre-built dashboards.

Capabilities and Features Contd.

- Scalability Support: Scale up applications according to different dimensions.
  Features: Scalability on number of users, data traffic, data storage.

- Business Logic Specification Mechanisms:
  Specify the business logic of the application modeled.
  Features: Business rule engine, graphical workflow editor, AI enabled business logic, API support

- Application Build Mechanisms: Ways the application is built.
  Features: Code-generation, models at run-time.

- Deployment Support: Deploying the modeled application.
  Features: Deployment on cloud, local infrastructures.

References

1. https://arxiv.org/pdf/2107.07482.pdf

2.https://www.researchgate.net/publication/344842798_Supporting_the_understanding_and_comparison_of_low-code_development_platforms

3. https://www.youtube.com/watch?v=jaRCENYBuYo

4. https://www.youtube.com/watch?v=JwM9NrePPMc

Acknowledgement

Dr. Wei Yang