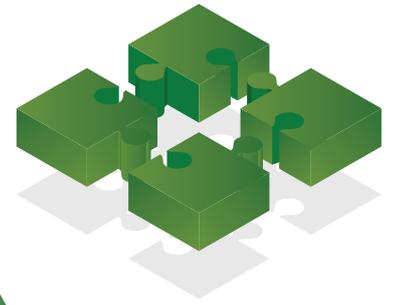


No Code/ Low Code development



Lately there's been a lot of interest in No-code/Low Code development platforms. Every year more and more organizations are adopting this kind of development approach.

The idea of no-code platforms is to use visual development interface with pre-defines templates and pre-built logic models with components interlinking with each other so that the average business user without any prior coding experience can use business cases to build powerful applications themselves. However, in the low-code platform there is still a requirement for some coding with the involvement of development team to customise the application with fine tuning to meet the business requirements.

In this article I will explain the differences between no-code and low-code technologies and discuss advantages and disadvantages of both.

What is No-Code Development

As the name suggests No-code development is a method designed for average user with little or no technical and programming knowledge to create advanced and fully functional applications.

In today's technology driven world the need for mobile and web applications for the digital audience is increasing. Small to medium sized companies have restricted budgets and are less likely to cater for software development companies or in-house developer to develop application at high costs. Creating an application via codeless programming is the most suitable option in this scenario.

What are some of features of No-code development?

Pre-built templates:

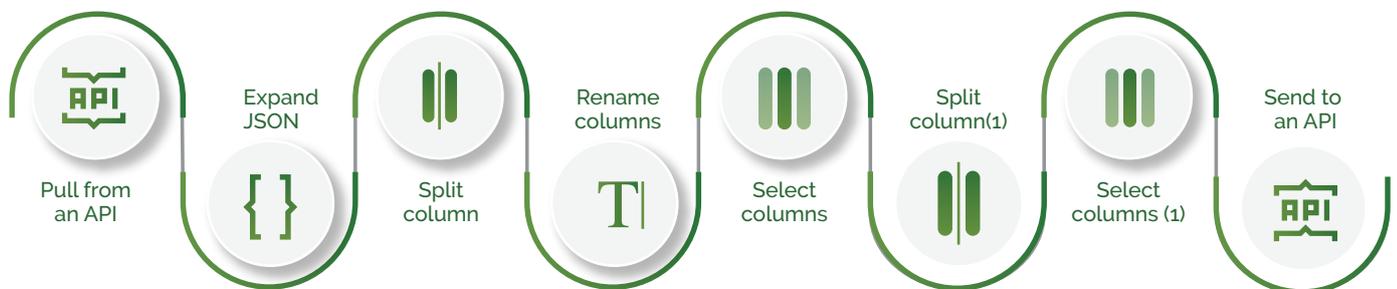
It is now possible to develop required Apps from series of templates available where you get templates around certain workflow challenges within different departments like HR, Marketing and Sales and more

Integration with other platforms: Email and app integrations are available with most no-code platforms. Multiple systems can be interlinked by use of API's and webservice which can extend functionality.

Speed and support:

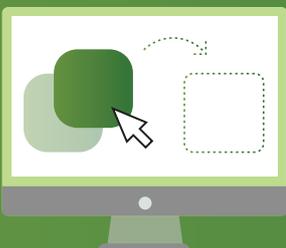
Apps are hosted on the platform which means there is no need of worry if there are any server errors. You can release apps faster than the regular ones.

Built-in actions & workflows:



You can pre-define logic and the workflows inside the applications for ease of use across the team.

Visual Modelling



Complex coding is now replaced with drag-and-drop interface making data management easy through visual modelling.

Reusability

Some components can be reused for other new applications to create a modern layer around outdated legacy systems.

What is Low-Code Development?

Low code development is similar to no-code except there it is aimed for Business users with some coding experience whereas no-code is aimed for people with no experience of writing code. The platform aids developers to use drag-and-drop pre-coded blocks which in turn reduces the need to write code.

Low code offers the flexibility in UI with additional coding requirements as to no-code platforms that relies on UI layer for application design.

Furthermore, Low-code platforms offer security and scalability and Reporting and Monitoring features to allow you to monitor the performance for apps that can adapt to growing business needs

The types of application that can be built with low-code include Mobile applications, Web application and portals whereas no-code can be used to develop mobile application and many more.

Low-Code	No-Code
<ul style="list-style-type: none">● Serves only developers● Little coding required● Mostly used to build complex applications● Complete customization available● Rapid application development tool for future purpose● All platforms provide end-to-end development	<ul style="list-style-type: none">● Serves both developers and non-coders● No coding required● Widely used for reporting, analytics, and tracking apps● Pre-built templates can be customized● Self-service application development tool for business users● Limited capabilities provided by some platforms

What are the Pros and Cons of both No-code and Low code?

Pros

- **Less coding:** Non-technical users can easily use the platforms as they minimal or no coding experience.
- **Ease to understand:** Traditional Codes can be difficult to understand and debug. No-code and Low-code are comparatively easier to grasp and easy to fix any errors.
- **Increase Agility and short release cycles:** with speedy development time and shorter release cycles Apps can be adapted to changing requirements allowing businesses stay ahead of the competition.
- **Increase productivity and Speed:** This is one of the crucial benefits where apps can be built within two to three days compared with traditional hand coded apps. Low-code's inbuilt templates and drag-and-drop functionality together with business process data models help accelerate development time.
- **Lower Cost:** No-code platforms help companies reduce their total cost of building and maintaining application. You don't need highly skilled programmers to build and maintain your applications and due to the module nature of building apps on no-code platforms the total time and effort are significantly lowered as compared to traditional coding.

Cons

- **Limited functionality:** No-code/Low-code are limited on the built-in functionality and therefore the level of customisation is incomparable with applications developed using traditional coding.
- **Security Risk:** NO-code/Low-code app building rely on vendor platforms and since you don't have control over the source code for the apps, it then becomes the responsibility of vendors to address any security flaws.
- **Vendor Lock:** As companies invest in no-code/low-code platforms it becomes more difficult for them to switch to different platforms. However, this may depend on vendor to vendor. Also, vendor will not let you make changes to your applications once you stop using their tool.
- **Price Confusion:** Companies pay unnecessary high price for enterprise level no-code/low-code platforms as compared to mid-level platforms. However, mid-level platform offer less scalability and companies may need to invest in multiple platforms which can further create complication in pricing.
- **Difficult to integrate with other systems:** Low-code platform have limited integration with other systems due to lack of available API's.

Conclusion

More and more companies are investing on the no-code/low-code platforms with combination with traditional coding by developers to bring faster delivery times. Having said that no-code/low-should not be considered as a replacement of traditional code but should be considered as part of the development cycle.

I would encourage companies to adapt no-code/low-code platforms into the business processes, as these tools can be used by both developers and non-developers to help meet business requirements and decrease in development time and increase innovation.



Beyond



Author
Mani Singh
Technical Delivery Engineer