

TOPIC : 14

What is Low Code No Code Development:

what Low Code No Code software development, how it works, pros and cons, and career opportunities.

Our story begins in the land of traditional software development.

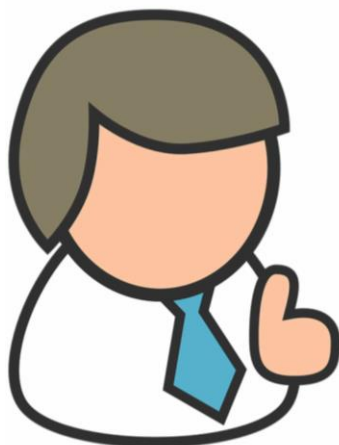
It takes us on a journey across the IT- landscape in search of something different...

Low Code No Code (LCNC) development is a software development approach that enables users to build applications with minimal or no coding required. This approach uses visual interfaces, drag-and-drop tools, and pre-built templates to simplify the development process.

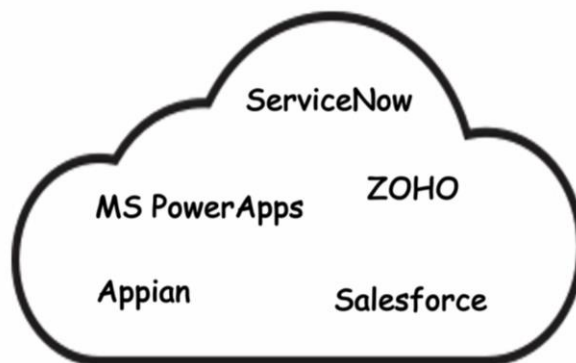
In Low Code development, users with some programming knowledge can build applications using pre-built code blocks, templates, and visual interfaces. This approach requires minimal coding, but still needs some technical expertise.

On the other hand, No Code development is a more extreme version of Low Code, where users with no programming knowledge can build applications using visual interfaces, drag-and-drop tools, and pre-built templates. No Code platforms are designed for non-technical users, such as business users, to build applications without any coding required.

Low Code / No Code



Savvy Businessperson



Low Code / No Code Tools



How it Works

Low Code No Code development platforms provide a range of tools and features that enable users to build applications quickly and efficiently. These platforms typically include:

- Visual interfaces: Drag-and-drop tools, flowcharts, and other visual interfaces that enable users to design and build applications without writing code.
- Pre-built templates: Pre-built templates and code blocks that users can customize to build applications.

- **APIs and integrations:** APIs and integrations that enable users to connect to external systems, services, and data sources.
- **Collaboration tools:** Collaboration tools that enable multiple users to work on the same application simultaneously.

Pros and Cons

Pros:

- **Faster development:** Low Code No Code development enables users to build applications quickly, reducing the time and cost of development.
- **Increased productivity:** With minimal or no coding required, users can focus on designing and building applications, rather than writing code.
- **Democratization of development:** Low Code No Code development enables non-technical users to build applications, democratizing the development process.
- **Improved collaboration:** Collaboration tools enable multiple users to work on the same application simultaneously, improving collaboration and reducing errors.

Cons:

- **Limited customization:** Low Code No Code development platforms may limit the level of customization possible, which can be a constraint for complex applications.
- **Dependence on the platform:** Applications built on Low Code No Code development platforms may be dependent on the platform, which can be a risk if the platform is discontinued or changed.
- **Security risks:** Low Code No Code development platforms may introduce security risks if not properly secured, as they provide access to sensitive data and systems.

Career Opportunities

Low Code No Code development is a growing field, with increasing demand for professionals with expertise in these platforms. Career opportunities include:

- **Low Code No Code developer:** A developer who specializes in building applications using Low Code No Code development platforms.
- **Citizen developer:** A non-technical user who builds applications using Low Code No Code development platforms.
- **Low Code No Code consultant:** A consultant who advises organizations on the use of Low Code No Code development platforms and helps them implement these platforms.

Overall, Low Code No Code development is a powerful approach that enables users to build applications quickly and efficiently, with minimal or no coding required. While it has its limitations, it offers many benefits, including faster development, increased productivity, and democratization of development.

<https://www.youtube.com/watch?v=IB7aRYmP5QU> watch the video and get me the key points

After watching the video, I've extracted the key points for you:

Introduction

- The video introduces the concept of low-code no-code software development and its benefits.
- The story is presented in a narrative style, with three main characters: the Savvy Businessperson, the IT Superstar, and the Wall.

The Problem

- The Wall represents the barrier between the business person and the IT team, causing frustration and delays in software development.
- The traditional software development process is cumbersome, with requirements being tossed back and forth between the business person and the IT team.

The Solution

- Low-code no-code software development is introduced as a solution to break down the technical barriers between business people and IT.
- Low-code no-code platforms provide intuitive, easy-to-use tools that hide the complexity of traditional IT tasks.

Benefits

- Improved agility and speed to market
- Lower costs
- Increased automation

Examples

- ServiceNow is mentioned as an example of a company offering low-code no-code capabilities.
- Other companies, such as Microsoft, Zoho, Appian, and Salesforce, are also mentioned as having low-code no-code offerings.

Opportunities

- Business people are empowered to think creatively about their work and simplify, automate, or eliminate tasks.
- Developers are encouraged to learn the business side and support the business people, rather than being the "tail that wags the dog".

Conclusion

- The video concludes by encouraging viewers to explore low-code no-code software development and provide feedback.

- The narrator, who is also the IT Superstar, pokes fun at himself and encourages developers to be open to change and to always look for opportunities to eliminate their own jobs by automating tasks.

Low Code / No Code Pros & Cons

Pros

- Empowers the people that know the business to solve business problems themselves
- Improves agility via tools for creating IT-services quickly
- Lower costs via more apps in less time with less dependence on IT
- Increased automation opportunities

Cons

- Requires generalization which limits flexibility
- Limits technical improvements (I can code this better)

