AI n Dot Net

Choosing the Right AI Development Framework

Low Code/No Code vs. Full Code

Understand the strengths and limitations of each approach to make informed decisions for your AI projects

Overview of Each Approach

Low Code/No Code Platforms (e.g., Microsoft Power Platform)

Simplified development for non-programmers and rapid prototyping.

Full Code Development (.NET)

Comprehensive, robust solutions for experienced developers and complex projects.

Key Differences at a Glance

Low Code/No Code: Advantages and Limitations

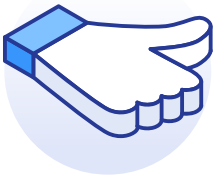
Advantages

Rapid Prototyping

Build applications quickly without extensive programming.

Ease of Use

Drag-and-drop interfaces, minimal coding required.



Good for Simple Workflows

Automate straightforward tasks and integrate easily.

Limitations

Functionality
Often require multiple tools for complete solutions (e.g., separate workflow and API tools like Zapier).

Data Handling
Limited capabilities and integration challenges with large datasets.

Logging and Monitoring
Basic or limited logging, exception handling, and monitoring.

Version Control and DevOps
Minimal support for advanced deployment practices.

Application Types
Limited to specific kinds of apps (e.g., simple web or mobile apps).

Visual
Icons of workflow tools, limited data capabilities, and missing features like DevOps.

Full Code Development (.NET): Advantages and Limitations

Advantages

Comprehensive Capabilities

Develop any type of application: web, web API, desktop, mobile, console.

Enterprise-Ready

Built-in support for DevOps, version control, logging, monitoring, and security.

Robust Data Integration

High-performance, secure, and reliable database integration.

Flexibility

No limitations like low-code/no-code platforms.

Error Reduction

Strong typing and compiled code reduce errors significantly.

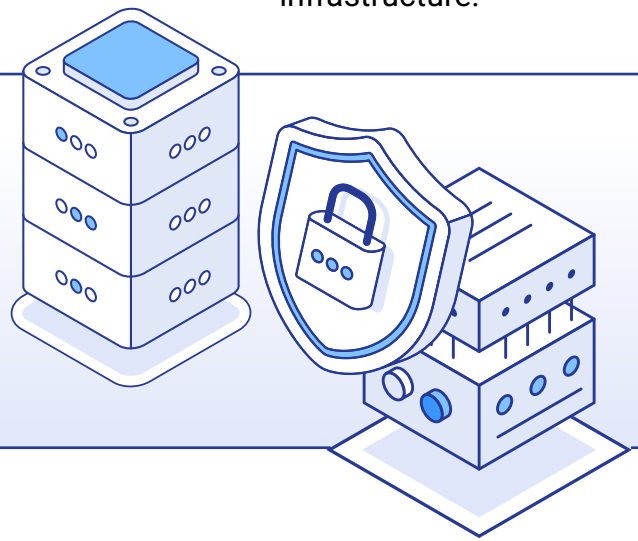
Established Expertise

Most medium to large organizations already have .NET expertise and infrastructure.

Limitations

Time Investment

More development time for complex applications compared to low-code solutions.

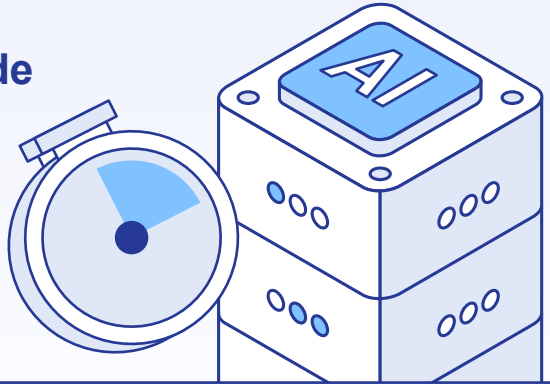


When to Use Each Approach

Low Code/No Code

Best For

Rapid prototypes, simple automation, small-scale apps.



Full Code (.NET)

Best For

Enterprise solutions, complex AI models, data-heavy applications, long-term projects.

Additional Considerations

Microsoft's Low Code/No Code Experience

Often require multiple tools for complete solutions (e.g., separate workflow and API tools like Zapier).

Future Scalability

Full code offers more flexibility for scaling and future-proofing applications.

Visual

Growth charts and a timeline representing Microsoft's experience.

Making the Right Choice for Your AI Projects

Learn more about optimizing your AI development approach at

www.AInDotNet.com