Low-Code vs. No-Code

Capabilities

	Low-Code	No-Code
Front-end Development	Basic functionality can be configured via a visual editor, but complex operations (e.g.,form data validation) require additional scripting	Both basic functionality and complex operations can be configured without need for scripting
Back-end Processes & Workflow	Application workflows can be configured visually, via either pre-built modules or through scripting	Complex workflows can be quickly built and easily managed across legacy systems and third-party solutions through a purely visual UI
Integrations	Modern integrations can be achieved via visual configurations, but legacy systems or more complex data transformations require additional coding	Modern and legacy integrations can be configured without ever having to write a single line of code
Data Transformations	Data transformations & logic require complex code and data transformation	Data transformations are executed with a completely visual ETL tool

Impact

	Low-Code	No-Code
Time-to-first-build	**Faster:** A low-code application can be completed in 3-to-6 months relative to 9-to-12 for a typical enterprise application of equal complexity	**Much Faster:** A no-code application of equal complexity can usually be completed in just 2-to-3 months
Ease of making material changes	**Difficult:** Trained engineers must decipher and debug thousands of LOCs which may be years or decades old and written by programmers who have has long since moved on	**Easy:** Only changes to business logic are required to change applications. No legacy code to sort through.
Ease of hiring & training	**Difficult:** Requires consultants or seasoned developers trained in specific language	**Easy:** Anyone versed in business logic and decisioning can configure on a no-code platform
Total cost of tech maintenance	Less: Basic elements of code maintenance & support are still required	**None:** No legacy code to maintain or upgrade (we take care of all that on the back end)