

# Code vs. Low-Code vs. No-Code

Identifying the right development method for your next project is a big decision — this table will help you make it.



For enterprises looking to remain competitive in today's market, offering powerful digital applications is essential. According to **Forbes**, digitally mature companies are making \$1 for every 38 cents made by their more "analog" competitors.

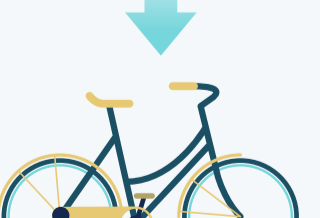



Digitally-mature



Analog

Unfortunately, the path to a complex application is riddled with failure and exorbitant costs. Only **28% of major corporations** achieve effective digital transformations, meaning that the overwhelming majority of large-scale IT projects result in failure. Choosing the right development method for your next project can alleviate friction points and play a large role in the success of your software—but how do you choose the right one? Start by comparing traditional development, low-code, and no-code across these five areas.

	Traditional Code	Low-Code	No-Code
Technology	<p>All processes must be <b>coded</b> using specific, complicated, and soon-to-be outdated programming languages.</p> <p>↓</p> <p>Only <b>skilled developers can make changes</b>, and finding people with experience in certain programming languages gets harder as time goes on.</p> 	<p>Basic functions can be configured using a visual editor or pre-built modules, but <b>complex operations and modifications still require code</b>.</p> <p>↓</p> <p>Like traditional development, <b>trained engineers are still required</b> to decipher and debug lines of code to make and deploy changes.</p> 	<p>Everything from basic functions to complex operations can be configured with visual flows and pre-built components — <b>no coding required</b>.</p> <p>↓</p> <p>It's <b>easy to make changes</b> because all configuration takes place within logical flows on a pre-tested platform—to change the application, just arrange components as needed.</p> 
Integration	<p>Integrations with code can be <b>incredibly complicated</b>, and engineers have to constantly check for compatibility with legacy systems.</p> <p>↓</p> <p>It's <b>extremely difficult to manage integrations</b> with code. If your legacy code can't accommodate integrations, you'll end up with applications that only work in a vacuum.</p>	<p>Modern integrations can be done using visual configuration, but <b>legacy systems or complex data transformations still require code</b>.</p> <p>↓</p> <p>Integrations are a <b>little faster</b> and easier than with code, but you'll still have to code your most complex (and critical!) integrations.</p>	<p>You can <b>seamlessly integrate</b> your application with legacy technologies or modern external solutions without scripting.</p> <p>↓</p> <p>Integrations <b>couldn't be easier</b>. You can hook into any legacy file format and extract data in a visual interface. You can also easily configure APIs using plug-in components.</p>
Cost	<p>Between hiring engineers, purchasing disparate tools, and accounting for legacy maintenance, an application will continue to <b>accumulate significant costs over time</b>.</p> <p>↓</p> <p>Code already comes at a high price point—and 45% of large-scale IT projects run over budget. Before you know it, <b>your enterprise could be saddled with a lot of technical debt</b>.</p>	<p>Compared to traditional development, low-code is a more affordable approach because <b>less code means fewer costs</b> associated with maintaining code.</p> <p>↓</p> <p>Applications can be built faster, but <b>you still have to pay for the basic elements of code maintenance and support down the line</b>. That adds up.</p>	<p><b>Complex projects can be done in a fraction of the time for a fraction of the cost</b>. Not only do you reduce development and legacy maintenance costs, but your speed-to-market also allows you to see returns much faster.</p> <p>↓</p> <p>With no editable codebase to maintain, lower overhead, and an application that actually meets business needs the first time around, <b>your enterprise can save money and focus on generating value</b>.</p>
Security	<p>A <b>not-insignificant portion of your development resources</b> must focus on ensuring your application adheres to critical cybersecurity standards.</p> <p>↓</p> <p><b>Cybersecurity is a never-ending battle</b>. If your engineers have to constantly edit the codebase to keep up with security standards, there won't be time left for anything else, and important updates can easily slip through the cracks.</p>	<p>Security upgrades and patches are uploaded automatically, but <b>many low-code platforms don't have the security protocols necessary to handle enterprise-grade use cases</b>.</p> <p>↓</p> <p>Managing cybersecurity is easier with low-code, but low-code security measures still might not be strong enough to suit your needs—meaning <b>you'll have to use additional code to take your own precautions</b>.</p>	<p>No-code platforms like Unqork have <b>enterprise security built-in from the ground up</b>. Data encryption, role-based permissions, single-tenant deployment in a private cloud, and more will keep your application secure.</p> <p>↓</p> <p>A no-code platform's native security features ensure that you're building an application your customers can trust. <b>Security is constantly updated on the back-end of the platform</b> so you never need to worry about it. Also, remaining compliant with evolving regulations is a breeze.</p>
Speed	<p>On average, building an enterprise solution with code can take <b>9-to-12 months</b>.</p> <p>↓</p> 	<p>A complex low-code application can easily take <b>3-to-6 months</b> to complete.</p> <p>↓</p> 	<p>With no-code, your app could be ready to launch in a matter of weeks—or <b>even days</b>.</p> <p>↓</p> 

When you stack traditional development, low-code, and no-code against each other, it's clear which development method is best-suited for the modern enterprise. No-code with Unqork goes beyond other platforms to allow you to build truly complex, scalable, enterprise-grade applications. Unqork can help your enterprise unlock all of the benefits of rapid app development for a fraction of the cost, without sacrificing security or complexity.

[Schedule a demonstration today to see what no-code can do for your business.](#)