# LOW CODE VS NO CODE EXPLAINED



The biggest change since cloud-based software.

Low-code and no-code platforms are bringing about the biggest change in the IT world since Google Docs. Cloud-based software forced the rest of the IT world to sit up and listen. Traditional providers were forced to change overnight.

And now the world of DevOps is being forced to listen to the new kids on the block—low-code and no-code platforms, which have the potential to make software development so easy that almost anyone can do it.

Do you need to build applications for immediate use? Maybe it's time to move away from professional developers and pick up a new low-/no-code platform.

## What is "low code"?

Low-code programming removes up to 90% of the coding process. These low-code development platforms cut down on development by using innovative drag-and-drop tools that:

- Streamline the process
- Save on turnaround time

Specialized language knowledge becomes less of a problem because all the groundwork is already written. Application development is faster because specialist coders with narrow skills are not needed.

When you need to develop and release lots of applications, low code helps minimize opportunity for

bottlenecks in the workflow. In-demand skills are no longer stretched in every direction: the low-code platform handles that for you.

Because the demand for mobile app development services is outgrowing what the industry can deliver, low-code platforms make it easier for non-developers to create apps. The definition of a developer broadens. Now, non-specialists using low-code platforms can answer the call for more apps—*without* support from specialists.

## Examples of low-code platforms

Two of the most successful low-code platforms are Salesforce and Zoho. Both platforms allow users to create their own applications based on a wide range of frameworks, such as through Salesforce's low-code development platform AppExchange.

By providing the majority of the code for their clients, both Salesforce and Zoho allow clients to customize what they show their customers and end users. Rapid application development is easy because the low-code requirements allow a content management system or customer relations program to be rolled out almost instantly.

# What is "no-code?"

Defining no-code is somewhat problematic when it comes to the question of low-code vs no-code. The Gartner Magic Quadrant Low-Code Application Platforms 2020 report grouped low-code and no-code together as one. Although it is a distinct style of application development, no-code is still seen as a subsection of low-code.

No-code development platforms are growing in popularity because they allow non-technical individuals to create apps and other tools. There is no need to code anything in the application.

Instead, they use simple, intuitive interfaces, generally with drag-and-drop functionality. This makes application development an agile process there is no need to wait for a developer to create even the last 10% of an application.

The real benefit of no-code platforms is that application developers can quickly respond to business needs. Business process management no longer needs traditional programming experts to build apps or tools.

## Examples of no-code platforms

If you have used Pipedrive for your CRM needs, Airtable for cloud workspaces, or Canva for graphic design, you will understand the benefits of no-code solutions. They have taken difficult areas that generally require a high level of coding skill and made them available to users.

Canva is a particularly interesting case. Its popularity boom has led to the no-code platform being valued at $3.2 billion. The market for no-code solutions is expanding and professional developers may have to worry about their job security.

bmc

| Low Code | No Code |
| --- | --- |
| Use for more complex applications | Use for reporting, analytics and tracking apps |
| Usually for apps that are foundational or run important processes for a business | Apps that evolve with frequent updates and changes in use-case |
| Apps with:<br>• More than 5 years lifecycle<br>• Fewer updates | Can be integrative or stand alone |
| Can be mission critical | Good for self-deploying apps |
| Offers more developer control | Mobile responsive |

## How low/no-code is changing the industry

It's impossible to deny that no-code and low-code tools are having a significant impact on the market. As knowledge about low-code/no-code platforms spreads, so is the expected market share and the total worth of these innovative companies.

By 2024, Gartner expects that 65% of all app development will originate with low-/no-code tools. Business needs will have become too great to wait for people fluent in programming languages to build everything from the ground up. Instead, business users will take their needs into their own hands.

Forrester agrees with Gartner, estimating that low-code and no-code resources will be worth up to $21 billion in 2022. This incredible growth is due for several reasons, as noted above, but the most important benefit is speed.

Enterprises want to have their applications developed quickly—but are they minimizing the problems with this approach?

## Who has to change?

The group that may have to change the most is those in classical DevOps positions. Although the demand for traditional coders will not disappear overnight, there is the potential for fewer positions to be filled.

This may address the gap in the number of developer jobs that go unfilled throughout the year, but does it spell disaster for specialized workers?

Possibly, but it's not likely. Whereas specialized coders working in rare languages or one particular

area of framework development might become less "necessary" to a business, there are still going to be jobs available. But the rise of citizen developers is a concern for many in the industry.

# Are low/no-code platforms flawless?

In short, no. These platforms solve one problem, but they may expose you to additional risk.

## Problems with low-code

Many low-code platforms are difficult to master. Although creating an app on someone else's chassis is easier in the short-term, long-term scalability of high-quality applications may be out of reach.

These platforms may struggle to balance both high performance and creation ease. If it's so easy, can it handle complexity? That's why low-code apps may only be short-term solutions for business needs.

## Problems with no-code

No-code causes more problems than low-code for two reasons:

- The risk of shadow IT
- General technical debt

Because no part of the platform has been made by a developer, a non-specialist may make mistakes that developers wouldn't. From a technical point of view, having your entire code base built by someone else could lead to vulnerabilities or inefficiencies that cause difficulties later on.

This leads to a great deal of technical debt—issues you'll eventually need to deal with—that could cause an enterprise to have slow or ineffective apps. For example, the coding that makes up the basis of the application is full of unnecessary or irrelevant filler.

Of course, no-code options make it so easy to do what you want quickly, members of your organization may experiment with unsanctioned or untested apps—shadow IT—that might not meet your organization's compliance and security needs.

# Is low-code/no-code the future?

Low-code and no-code platforms absolutely have a place in the world of development. Gartner is expecting them to start to dominate the market within only a few years, so ignoring the benefits that these approaches bring could mean missed opportunities for an enterprise.

But low-code and no-code aren't going to erase the need for traditional coders just yet. Relying on others for all your application development needs can be a risk, no matter how useful the platforms are.

DevOps processes still have a place in the workplace, even if low-code solutions may have a majority share of the market in the years to come.

## Related reading

- BMC DevOps Blog
- Application Developer Roles & Responsibilities
- The State of DevOps Today
- Managing Containers and Code for DevOps
- What is Code Refactoring? How Refactoring Resolves Technical Debt
- API/Developer Portals: How To Create Great API Portals