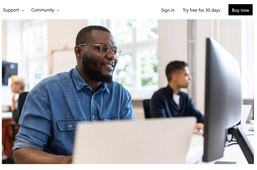
Low-code vs. traditional development

Learn how low-code and traditional development differ from and complement each other.



What is low-code development?

Low-code development is an alternative software development approach that, along with other capabilities, streamlines website and app building by using preset modules, templates, drag-and-drop functionality, and automated processes. Because it requires minimal coding expertise or knowledge, non-developers can easily bring new applications and products to market quickly.

Increasingly, low-code development is a popular option for businesses seeking solutions that can free up time for their IT teams and boost productivity and efficiency. For organizations with limited IT resources, low-code development can empower developers and non-developers to expand their skill sets by developing apps more quickly in-house rather than relying on more costly custom development.

Build applications faster—with fewer resources

Start creating and launching professional-grade apps right away using prebuilt templates, drag-and-drop tools, and rapid deployment. Get started with Microsoft Power Apps today.

Try free ›

Benefits and drawbacks of low-code development

Low-code development is becoming an increasingly integral part of software development, helping to ease workloads and allowing organizations to swiftly design, deploy, and update apps. Rapid deployment and quick updates allow businesses to create improved customer experiences, as they can nimbly respond to changing customer feedback and habits. Building low-code apps with a service like Microsoft Power Apps can help businesses produce professional-grade apps with advanced functionality.

Low-code development platforms can support a range of different business needs and help alleviate IT workloads. Non-IT workers can use low-code development to participate in app development, freeing up time for professional developers to focus on more specialized coding for custom components and applications, and IT professionals can also utilize low-code automations to optimize processes and workloads. While low-code development has some limitations, the numerous benefits make it a great option for a broad range of different businesses.

Examples of low-code apps businesses can develop

- Operational cost management apps
- Scheduling apps
- Business expense management apps
- Timesheet apps
- Help desk apps
- Leave request apps
- Service request apps
- Payroll apps
- Grant and contract activity apps
- Approval process apps
- Inventory tracking apps
- Event planning apps

Explore more low-code app scenarios and real-world examples and learn how Power Apps customers have used the platform in multiple business and nonprofit sectors.

Benefits of low-code development

- **Ease of use:** Drag-and-drop modules and templates take the guesswork out of app development.
- **Reduced IT workload:** By providing IT professionals with an expanded toolkit, low-code development gives them flexibility and space to prioritize more technical labor and custom projects.
- **Low barrier to entry:** With simple tools, interfaces, and drag-and-drop features, low-code development allows a broader swath of workers access to apps and coding projects, helping them improve development knowledge and skills, and promoting a strong learning culture.
- **Bring apps to market quickly:** Preexisting templates and tools help workers produce and launch apps quickly.
- **Cost effectiveness:** Relying on low-code templates and tools means businesses don't have to invest the time and resources required for more complicated, custom development.
- **Simplified updates and maintenance:** Rather than depending on custom development for new updates and routine upkeep, low-code platforms can manage this on behalf of the business.
- **Helps keep organizations secure:** Platforms like Microsoft Power Platform offer built-in security and help bridge independent applications into a single structure.
- **Valuable data and analytics:** Built-in tools and dashboards within low-code platforms can offer visual models and reports with useful, actionable insights.
- **Automate repetitive tasks:** In conjunction with custom development, IT workers can use low-code development to optimize their workloads.

Drawbacks of low-code development

- **Less customizable and scalable than custom development:** While low-code platforms offer a broad range of pre-existing templates, they can be limiting for more complex projects, and don't always offer the ability to scale along with a business's growth and changing needs.
- **Limited integrations with other systems:** Legacy platforms or systems might require a more complex integration solution.
- **Lack of template flexibility:** Working within the specified template means workers may have to create workarounds or shift expectations about an app's functionality.

What is traditional code or custom app development?

Custom app development, which uses traditional coding, relies on professional developers to manually plan, design, code, and implement software or applications. In a traditionally coded website or application, each feature or element has custom code, and developers work to meet specific, often complex requirements. While traditional coding is usually more time consuming and expensive than low-code development, its more expansive customization options can be advantageous for certain projects.

Benefits of custom app development

- **Highly customizable:** Custom developers can plan for, design, and meet specific business needs.
- **Scalable and adaptable:** Custom development is more agile, allowing IT professionals to pivot as necessary, scale up when required, and adapt to the changing needs of the business.
- **Expanded functionality:** Custom code offers a broad range of functionality, and developers can tailor functions to meet specific requirements.
- **Integration with other systems:** Custom developers can adapt the application to integrate with other legacy apps, systems, or products.

Drawbacks of custom app development

- **Continuous maintenance:** Custom apps require a more hands-on approach for security and upkeep.
- **Higher expenses:** A custom initial build can be costly and time consuming, and maintaining the custom app also requires specialized skills that can result in a higher bill.
- **Longer implementation and testing period:** As custom development is often complex, it can be more complicated to implement and test, resulting in a longer waiting period before releasing the app to customers or the public.
- **Updates and changes require more time:** Building a custom app involves a longer development process than using low-code development, and planning and launching updates takes more time as well.

How low-code and traditional app development differ

Low-code development differs from traditional app development in that it extends the ability to create apps or websites to a larger swath of people. While traditional coding requires professional developers with manual coding expertise, low-code development calls for a minimum knowledge base and offers simple, easy-to-use tools, templates, and interfaces.

When deciding which approach is best for your situation or project, consider the following questions:

- How complex is your project? Will it require regular changes and updates?
- Do you need integration capabilities, or will you in the future?
- How soon do you need your application or website to go live?
- How large is your budget?
- What are your security needs?
- How much time does your IT department have to dedicate to the project?
- Do you have non-developers on staff who already have some coding knowledge or are interested in expanding their coding skills?

Differences between low-code and traditional development

Low-code development

- Requires minimal coding knowledge.
- Reduces development costs.
- Uses templates and drag-and-drop tools.
- Provides automated updates and routine maintenance.
- Offers built-in visuals, reports, and analytics.

Traditional development

- Requires skilled IT professionals.
- Requires more time, energy, money, and resources.
- Allows for more agile customization.
- Provides the ability to integrate with a broader range of other systems and apps.
- Scales and adapts more readily.

No-code vs. traditional development

Similarly to low-code development, no-code development can also provide businesses and developers with additional tools to expand app building to a broader range of workers and allow IT teams to focus on more skilled tasks.

While low-code development may require baseline coding knowledge, no-code development requires none. No-code app builders have straightforward, intuitive interfaces that allow any worker to quickly create launch mobile and web apps. While no-code development is often less expensive than both low-code and traditional development, it has fewer customization options and is best-suited for simpler projects.

Using low-code development in conjunction with traditional development

Low-code development is not a replacement for traditional development, but rather a tool businesses can use to augment their coding capabilities and relieve IT departments of repetitive or unnecessary labor. Low-code development allows developers to work faster and dedicate their technical, full-stack skills to more innovative, custom projects.

With low-code development, developers can kickstart projects and add basic functionality to existing pages or create new ones within the context of large, more complicated projects while avoiding writing code line by line. Low-code development can expand the resources and tools available to an IT team while also allowing them to offload simpler tasks to non-development staff.

Getting started with low-code development

Implementing a low-code platform can help businesses support their IT teams and encourage non-professional developers to learn and hone new skills. When considering different low-code platforms, think about your organization's needs, and choose a platform that can support the range of projects you have planned and provide your IT team with tools to augment their skills and alleviate their workload. Platforms and tools like how Microsoft Power Platform and Power Apps can help you empower your entire team to build and launch apps and give your developers tools to extend their app capabilities.

Frequently asked questions

What is the main difference between traditional development and low-code development?

The main difference is that low-code development requires minimal coding knowledge or expertise. Because low-code development makes use of templates and drag-and-drop tools, low-code platforms are accessible to a broader range of workers. Typically, low-code platforms are more cost effective and can help businesses produce and launch apps faster than traditional development.

Will low-code development replace developers?

Low-code development is not a replacement for developers—rather, it provides IT teams with an additional tool, and does not threaten their more skilled work. Low-code development allows developers to automate repetitive tasks, prioritize complicated jobs, and devote their time to more involved and difficult projects. Low-code development can also empower non-IT workers to support the development team by creating professional-grade websites and apps.

What are the advantages and disadvantages of low-code development?

The advantages of low-code development include ease of use, reduced IT workload, a low barrier to entry, cost-effectiveness, simplified updates and maintenance, improved security features, valuable data and analytics, and automation of repetitive tasks. The disadvantages of low-code development include limited customization and scalability, fewer integration options, and inflexible templates.

What are the advantages and disadvantages of traditional development?

The advantages of traditional development include highly customizable options, scalability and adaptability, expanded functionality, and the ability to integrate with other systems more readily. The disadvantages of traditional development include higher upfront costs and potentially higher maintenance costs, a more hands-on approach to security and maintenance, longer implementation and testing periods, and updates and changes that require more time.

How does low-code development differ from no-code development?

While low-code development and no-code development both offer accessible alternatives to traditional coding, low-code development differs from no-code development in that it may require a small amount of coding knowledge and expertise, while no-code development requires none.

How do I choose which development approach is best for my project?

When choosing the development approach best suited to your project, consider the complexity of the project and the budget you have available. Think about what you might need in the future—do you have planned expansions, upgrades, or integrations you'll need? If so, you might choose to use custom development. If your project needs to launch quickly, would get value out of automated updates and security support from a low-code platform, and fits within a template, low-code development might be the best option.

How do I choose a low-code platform?

When choosing a low-code platform, consider the needs of your business and your development team. Do you have a wide range of projects you need to develop, from apps to web pages? How experienced is your team and who will be designing your projects? What kind of reporting and data analysis do you need? Finally, review your budget and choose the option best suited to your needs.

Contact Sales

Chat with Sales
Available Monday to Friday
8 AM to 5 PM Central Time.
Chat now ›

Request we contact you
Have a Power Apps sales expert contact you.
Send a request ›

Call Sales
Available Monday to Friday
8 AM to 5 PM Central Time.
Call 18552700615 ›

© 2023 Microsoft