



What is low-code? A guide to enterprise low-code app development

February 2023

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

This guide explores how low-code concepts and practices can help enterprise developers be more efficient, create valuable apps more quickly and contribute to broader business initiatives.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

What is low-code? A guide to enterprise low-code app development

STEPHEN BIGELOW, SENIOR TECHNOLOGY EDITOR

Businesses thrive by tightening work cycles and driving value. Meanwhile, experienced developers and teams are committed to long-term project roadmaps. At this intersection is the rise of low-code software development initiatives, tools and techniques.

Low-code is a visual, highly abstracted, largely automated approach to software development that defines desired tasks at a high level and then relies on tools to generate much of the underlying code base. Professional developers and line of business (LOB) staff understand business problems can use low-code concepts and practices to tackle a wide range of everyday programming chores. This can free developer teams to focus on larger and more complex projects.

HOW DOES LOW-CODE WORK?

Conventional software development has long been a painstaking and detailed effort. Developers write individual lines of code that represent instructions and data. They organize

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

that code into functional routines and modules that provide the software's features and functionality.

This approach demands detailed knowledge of aspects across the application development spectrum: development languages, development environments such as integrated development environments and compilers, testing and deployment tools, and the various policies and practices used to approach coding, testing and deployment.

By comparison, [low-code](#) technology abstracts and encapsulates much of the programming knowledge otherwise needed to create software. Rather than write individual lines of code, users select from a menu of reusable functional components through a drag-and-drop visual interface. They arrange and organize the available functional components to form the intended software flow, similar to creating a flow chart to approach a business problem or task. Users can easily add, move or delete functional components to build the final process. At that point, the low-code tool incorporates the underlying code and support tasks, such as testing and deployment.

The main benefit of [low-code development vs. traditional app development](#) is cost and time savings, which can enable an enterprise to deliver some products and features faster and for less money. There are many reasons a business should continue to use conventional development models, especially for complex or specialized enterprise software. Many

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

projects demand functions and operations, such as optimized performance, that do not fit easily into low-code approaches, and the amount of work to create and modify low-code to meet those requirements can't be justified.

Low-code vs. no-code: Low-code techniques can simplify and accelerate many everyday business programming tasks, but low-code is not designed nor intended to eliminate all coding. Many low-code projects require some amount of programming knowledge, either to modify existing functional components or create new ones, and assemble components in specific ways that a low-code tool may not accommodate.

Low-code and no-code systems are fundamentally identical, but there is a key difference: [no-code](#) aims to provide all of the intended features and functions and eliminate the need to add or modify components, which means users need no knowledge of programming or application design. No-code systems are typically seen as a subset of low-code platforms.

LOW-CODE DEVELOPMENT PRINCIPLES

Low-code incorporates a high degree of automation, but the platforms and development processes are not automatic. Best results from low-code projects also depend on sound principles of business and technology.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

Understand the low-code platform. Low-code platforms don't do the work for you. It's still important that all stakeholders, from developers to business analysts to project owners, participate in the evaluation, selection and use of the low-code platform. The time invested to learn the platform's features and capabilities will pay dividends when stakeholders can identify applications for low-code in future projects and potentially add more value to the initiative.

Avoid customization. The power of low-code is in its predefined components that are dragged and dropped into the workflow. Components are typically generic and fit many different use cases, but don't always address the needs of a programming task at hand. It's possible to modify existing components and add new ones, such as UI and visual design elements, but such customization increases development time and work (and cost), and can blunt the speed and simplicity that low-code technologies promise.

Generalize customizations. When an application needs customization, consider features from the context of high-level or common use goals. Design the customized component for potential reuse in other projects.

Don't overlook the team. The team that selects and uses a low-code platform must understand the requirements and business goals for the project at hand, even if those requirements may be simpler compared to traditional software projects. Product owners

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

and other key stakeholders should be available to answer questions and review builds in rapid low-code environments.

Keep business in the lead. Low-code makes it easy to iterate and experiment with new or different features and functions. Still, a project plan and roadmap can be valuable to keep a low-code development on-track. Test and review builds early and often, and involve business stakeholders for any changes to requirements and features.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

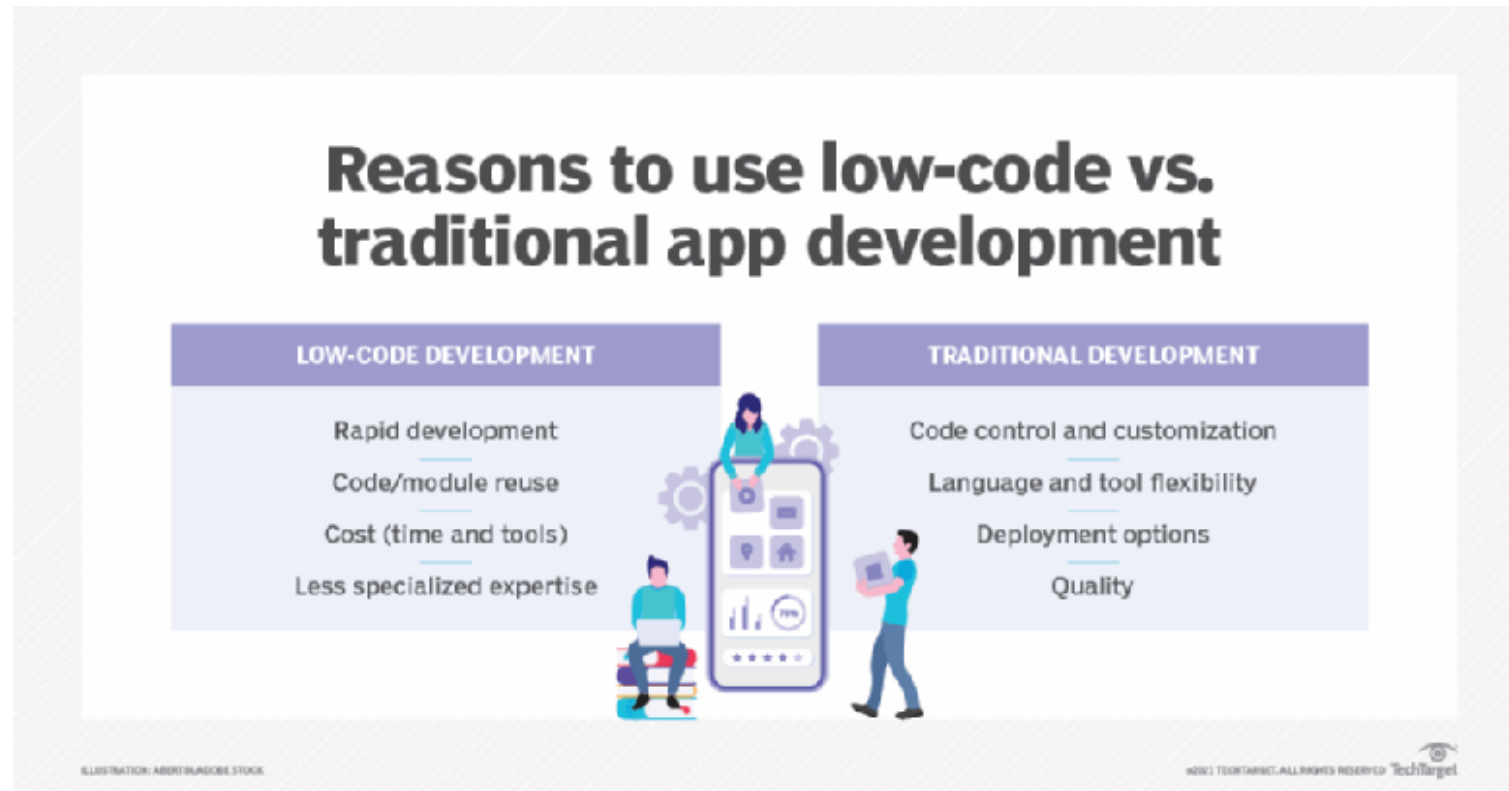
[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)



In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

WHAT ARE THE BENEFITS OF LOW-CODE DEVELOPMENT?

[Low-code development offers many benefits](#) to a business:

Accelerated development speed. It's a significant effort to write individual lines of code and master complex syntax. Low-code allows practitioners to assemble complex ideas and workflows as packaged functions or components, arranged in a convenient visual interface. This can also speed future updates and iterations of the project.

Broader staff availability. Skilled developer talent is at a premium in most enterprises. Low-code allows less-skilled programmers (or even non-programmers) to participate in the software development process -- they possess a clear knowledge of the business goals at hand. Any required customization or hand-coding can often be accomplished with less support from professional developers.

Improved efficiency. Businesses can use low-code to create tools for individual departments. For example, HR may want a tool to help model or forecast salaries and benefits to make better payroll decisions.

Lower-cost innovation. Similarly, businesses can experiment with ideas that might be costly or time-consuming with traditional development -- for example, creating a tool for

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

marketing units to analyze advertising campaign spending versus response and determine the most efficient ways to reach customers.

Accommodation of niche projects. Low-code can accommodate narrow applications and small user bases quickly and cost-effectively. For example, a business might not budget for the development of a tool used only in a finance department, but low-code might provide an attractive option.

Management for performance, governance and compliance. Low-code platforms typically track and manage the software project and its associated elements. This makes it easier for the business to maintain control over development and set guidelines that help ensure business governance and compliance. Low-code platforms also usually provide instrumentation, analytics and reporting that can help gather important details about a project's performance and utilization, which can help teams plan updates and troubleshoot.

LOW-CODE AND ENTERPRISE DIGITAL TRANSFORMATION

Today, every enterprise needs a digital strategy, a blueprint that describes how to apply digital technologies and services to support business goals, create or enable new goals, improve business performance or re-create business processes -- often in ways that were impractical or impossible without digital technologies. A comprehensive digital strategy can

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

fundamentally change policies, priorities, investments and long-term goals across an enterprise.

[Low-code aligns with digital transformation strategies](#) because many of these efforts involve software development projects that low-code can accommodate or even accelerate. Examples include user portals, mobile applications and business tooling.

One of these areas is business process management (BPM), which usually involves manual paper-based business processes, analyses and time-consuming managerial approvals. Businesses embrace BPM methods, such as a BPM platform, to help organize and analyze business data, evaluate budgets, streamline approvals and more. [Low-code and BPM complement one another](#) -- BPM functions can be added as low-code initiatives, and some BPM tools offer low-code abstractions when digitizing business processes.

Of course, low-code is only part of a digital transformation strategy. It also requires strong collaboration between teams of business and technology leaders to understand the business, identify digital products and opportunities that best suit the organization's needs, and evaluate and introduce various technologies.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

WHAT ARE THE CHALLENGES OF LOW-CODE DEVELOPMENT?

While low-code approaches and tools make compelling arguments for enterprise use, [low-code also presents some significant disadvantages](#):

Knowledge beyond the tools. Low-code is not a way to skate around skilled staff and a well-designed infrastructure -- apps built with reliable code require knowledge of enterprise software development and business practices. Enterprises that ask individuals who lack programming knowledge to take on enterprise-level programming tasks can actually spend more to fix bad code than if they'd crafted good code from the start.

Inefficient and unoptimized code. When code is abstracted into generic, reusable components, it can create underlying code that is bloated and excessive for the actual task at hand. It's easy to lose many of the potential optimizations that could reduce software size and improve performance. Quality, handwritten code can address such coding problems with efficiency and even elegance.

Complexity under the hood. Low-code's promise of simplicity is true when the resulting code functions as expected. What if it doesn't work or performs poorly? Is the application secure and consistent with the organization's compliance strategy? Someone within the organization must possess a comprehensive knowledge of what's happening under the hood

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

of the low-code platform to understand underperformance, address errors or enforce software security and other coding standards.

Limited use cases. Low-code platforms effectively solve some business programming problems, but not all. It may stumble with many complex tasks, or require so much customization and integration that conventional software development approaches and tool sets are the better choice.

Potential vendor lock-in. When a team uses a low-code tool to create an app, the tool may produce code that is poorly documented and difficult to maintain outside of the tool. Evaluate a low-code tool's ability to export, access and edit code using other editors and software development tools.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

Low-code development use cases in specific industries

INDUSTRY	REQUIREMENTS	HOW LOW-CODE HELPS
Education 	Address fluctuating enrollments, evolving educational strategies and assessments, and hybrid and remote learning	- Build and rebuild applications that measure learning outcomes as variables change - Integrate various digital platforms, such as enrollment management platforms and learning management systems - Help schools manage reopening amid the pandemic, e.g., template-based websites for COVID-19 information
Healthcare 	- Make data more accessible to patients on more platforms - Adhere to stringent security requirements	- Allow developers to build customer-facing apps that integrate with enterprise systems on the back end so that patients can access information more easily - Simplify compliance requirements by generating code that is designed to conform with frameworks like HIPAA out-of-the-box
Financial services 	- Allow customers to easily monitor financial accounts and data - User-friendly interfaces must work across devices and platforms (PC, mobile and web)	- Developers use low-code to build integrations and UIs that tie into complex or legacy back-end systems - Updating limited-functionality customer apps is easier than updating extensive back-end systems
Retail 	Adapt to an increasingly digital world with apps that serve customers and also manage back-end operations	- Develop and deploy apps for order processing, inventory and online order fulfillment - Avoid dedicating or hiring a large team of developers to build apps using traditional methods - Low-code tools in BPM platforms can help retailers build the software they need at a lower cost

ICONS: ALEXANDRE ASSIRE STOCK

©2023 TECHTARGET. ALL RIGHTS RESERVED.  TechTarget

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

WHAT ARE LOW-CODE USE CASES AND EXAMPLES?

Once IT and the business-side identify and align the business goals with clear IT requirements, low-code projects can fill an array of potential opportunities. [Common low-code applications](#) include the following:

Web portals. Portals are common and popular means for customers to interact with businesses, find services or products, get quotes, check the availability of resources, schedule work or place orders, and make payments. Low-code can help to quickly create an array of portals with common front ends or user interfaces instead of manually coding HTML and back-end components.

Line-of-business systems. Businesses rely on LOB systems and apps to conduct everyday tasks. For example, a mortgage lending company adopts a system to organize mortgage documents and paperwork, integrate appraisals, and conduct credit checks and financial analyses of borrowers. Businesses often buy such a platform from a vendor or build it in-house using traditional coding processes. Low-code brings a third option to help businesses build and add adaptive and scalable applications, and even migrate them to single or multi-cloud deployments.

Digitized business processes. Traditional paper- or spreadsheet-based business processes are time-consuming and error-prone. Businesses can use low-code to create applications

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

that gather required information, pass the information and requests through the company's approval process, deliver results to requesters and integrate with conventional business systems, such as ERP. For example, low-code can be used to streamline a capital request application.

Mobile apps. With low-code, a business can build a wide range of apps for mobile devices that present data and business interactions to customers. For example, an insurance company's mobile app lets customers file claims and upload documentation of an incident, such as collision photos, from their smartphones. Today's low-code platforms can assemble apps for both Android and iOS devices from the same project.

Microservices applications. A microservices architecture creates highly scalable applications from a series of independent components that communicate through APIs across a network. The components can be developed, deployed and maintained independently, which enables faster development and easier updates with less regression testing than conventional monolithic applications. [Low-code is a viable platform for microservices-based components](#), to quickly create and re-imagine core applications for greater performance and stability, and to help to translate traditional legacy code into modern, agile microservices applications.

IoT-based apps. Businesses grapple with ways to derive meaningful business data from the volumes of sensors and real-world devices that make up the internet of things (IoT), and to

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

monetize the resulting data or services. Businesses can use low-code to build apps and functions that integrate IoT endpoints and collect data, send IoT data through back-end computing infrastructures, and provide resulting data requests to internal or external customers. For example, a low-code horticultural application might use moisture and temperature sensors – combined with data about growing cycles and conditions – to automatically control indoor lighting and irrigation for cultivated crops.

LOW-CODE VENDORS AND SOFTWARE

The marketplace for low-code tools and platforms continues to mature and evolve. Many vendors combine low-code and no-code functions in a tool or platform. Businesses can choose from various cloud-native and third-party [low-code products and platforms](#) for uses that range from simple websites and mobile apps to enterprise-class application development. An incomplete list of current offerings is as follows:

- AppGyver
- Appian low-code platform
- AWS Honeycode
- Google AppSheet
- Landen 2.0
- Looker 7 business intelligence
- Mendix low-code platform

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

- Microsoft PowerApps
- OutSystems application platform
- Rintagi
- Salesforce Lightning
- ServiceNow
- Sisense business intelligence
- Skyve platform
- VisionX
- Wix Editor X
- Yellowfin 9 business intelligence
- Zoho Creator

A recent trend is the emergence of major enterprise and cloud vendors -- notably Amazon, Microsoft and Google -- to compete with more traditional low-code vendors such as OutSystems, Mendix, Appian and others. These large vendors have made significant investments to provide customers with deeper integrations to longstanding enterprise systems as well as cloud platform stacks.

Businesses should evaluate a low-code platform's specific capabilities and pay particular attention to how the [low-code platform handles components](#). For example, the low-code tool's packaged models and components should closely map to the high-level building blocks and functional methodologies that the business already uses, which will reduce the need for changes and new custom components. A low-code tool might offer strong API support and

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

integrate with traditional code projects -- but if the tool doesn't fit the use model, the customization work will negate the benefits.

An enterprise that determines it must develop new components should evaluate the ease and efficiency with which the customization engine creates new visual blocks or custom code. It's often better to create a new component than change an existing one since developers can ensure optimal and secure code for specific purposes or projects.

THE FUTURE OF LOW-CODE AND THE LOW-CODE PLATFORM MARKET

The [future for low-code development](#) and platforms will largely build on the trends and behaviors that shaped 2020 due to the [COVID-19 global pandemic](#), which has placed a tremendous strain on development budgets, personnel availability and team dynamics. Many businesses have struggled to develop apps, portals, web forms and automated workflows because employees are now remote and less available.

Low-code helps businesses fill this gap and build apps for specific business purposes with a wider range of IT and business participation. Low-code and no-code represented less than 25% of all application development in 2020 but will account for 75% of it by 2025, according to analyst firm [Gartner](#). Over that same time frame, Gartner projected, annual revenues for low-code and no-code platforms will surge from a little over \$9 billion to nearly \$30 billion.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

Low-code projections, by 2025

70%

Amount of new applications developed by enterprises using low-code and no-code development, vs. less than 25% in 2020.

4+

Number of low-code tools used by three-quarters of large enterprises for IT application development and citizen development initiatives.

50%

Amount of low-code tool customers that are business buyers outside the IT organization.

SOURCE: "FORECAST ANALYSIS: LOW-CODE DEVELOPMENT TECHNOLOGIES,"
GARTNER, JANUARY 2021
©2021 TECHTARGET. ALL RIGHTS RESERVED

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)

Low-code will benefit from automation capabilities realized from emerging technologies, including [hyperautomation and AI](#). Hyperautomation -- the drive to automate every business function that can be automated -- will depend on the ready availability and reliability of low-code capabilities. Similarly, AI can play a key role in determining the business need and functional requirements for future low-code projects, such as what a low-code project must do and even how the project is built.

As low-code platforms evolve with more speed and automation, the need to ensure proper coding standards and manage defects (testing and bug fixes) will drive additional testing capabilities and push test automation deeper into the CI/CD pipeline.

Ultimately, the developer community must embrace the reality of low-code and low-code business strategies. Developers must learn low-code platforms and work more closely with low-code practitioners to extend and customize low-code components. This requires improved communication and collaboration skills, and the ability to interact with business personnel and even "[citizen developers](#)" to build future enterprise applications.

In this guide:

[How does low-code work?](#)

[Low-code development principals](#)

[What are the benefits of low-code development?](#)

[Low-code and enterprise digital transformation](#)

[What are the challenges of low-code development?](#)

[What are low-code use cases and examples?](#)

[Low-code vendors and software](#)

[The future of low-code and the low-code platform market](#)



CONTINUED READING

[A practical take on low-code vs. traditional development](#)

[Assess low-code platform pros and cons](#)

[Review these 9 low-code use cases and industry examples](#)