

unqork



Blending Business Vision With Technical Precision

Three ways no-code improves business and tech alignment.

Contents

Introduction	3
No-Code Bridges the Business-Tech Divide	4
1. Rapid Prototypes	4
2. Empowered Business Users	5
3. Faster, More Responsive Software Iterations	7
Conclusion	9

TL;DR

- To remain competitive, companies are embracing digital transformation, building a wide range of applications for their customers and clients.
- The role of building software has traditionally fallen to technical people, with the business perspective limited to broad goal-setting and high-level feedback.
- A lack of business-tech alignment slows the development process down and undermines the quality of the product.
- A no-code approach to software development overcomes this barrier, allowing development teams to build faster prototypes, foster higher quality engagement from business users, and support greater agility in adapting software to evolving customer demands.

Enterprises are under increasing pressure to “go digital.” By 2022, 65% of the world’s GDP will originate from digitally mature organizations, market intelligence firm [IDC](#) predicts.

Adding to the complexity of this shift, companies are embracing digital transformation on multiple fronts. At a more operational level, digital transformation helps companies ramp up the efficiency with which they do business. Cloud-based technology, AI, and automation have vastly accelerated the pace of commerce. At the same time, enterprises are harnessing digital transformation to meet and exceed ever-rising customer expectations while staying ahead of the cybersecurity curve.

A lot rides on going digital. And this trajectory looks set to dictate the pace of change in the coming years. But that change isn’t always easy. One of the biggest challenges of digital transformation is that it isn’t purely—or even chiefly—a technological process. It’s also a business process. Only by closely and carefully aligning the technological possibilities with business objectives can your company find that elusive common digital ground and reap the benefits.

Bridging the business-tech divide generally requires solutions to a few interrelated challenges:

- Code has traditionally called the shots: In a conventional software development process, business users are often forced to rely on coders to translate their expertise and vision into functional software. There’s a communication gap, and something vital gets lost in translation.
- Budget and time blowouts: The expertise and experience gap between developers and business users creates friction. Coders often come to see the business perspective as an impractical hindrance and overly “big-picture.” Business users, meanwhile, frequently march in the opposite direction—seeing the coders as lost in the weeds. The result? A misalignment of priorities and a development process that both takes longer and costs more than projected.
- Disappointing ROI: As technology and business teams struggle to meet in the middle, the solution of compromise becomes increasingly inevitable. Big ideas become diminished in impact. The product frequently falls short of business requirements and delivers only a mediocre return on investment.

In this ebook, we’ll show you a better way to rise to the challenges of digital transformation. We’ll explain how using a no-code approach to software development can support clearer goal-setting and communication between business and technology team members, improved cost-effectiveness, and above all, enhanced agility to give your customers and clients what they want, when they want it.

No-Code Bridges the Business-Tech Divide

No-code can be a game-changer in building a more efficient relationship between business outcomes and the development process. The no-code approach provides an intuitive visual development environment that empowers business teams to directly participate in the application building process.

What does a no-code solution like Unqork's mean, practically, for application development?

Instead of coding a software solution line-by-line, a layer of abstraction and simplification sits between the development team and the codebase.

Using a drag-and-drop interface, development teams can achieve in days what might require months of conventional coding and bug testing.

No-code platforms aren't new. What-you-see-is-what-you-get (WYSIWYG) web builders from the 90s and 00s helped consumers build highly functional websites with limited or no understanding of writing HTML. Unqork, however, is the first no-code application platform to bring that same concept to enterprise-grade application development.

Enterprise no-code application platforms such as Unqork help bridge the technology-business divide in a number of key ways.

1. Rapid Prototypes

A [2019 Gartner panel](#) exploring critical factors in digital transformation found that: "Two critical aspects of ... design thinking are to fail early and learn from the failure. If you don't ... you're not being aggressive enough, you're playing it too safe."

Prototypes play a crucial role in quickly establishing a shared understanding of what works, and what doesn't. Developed early enough, a high-fidelity prototype gives your team valuable intelligence on how to position your product competitively, and with minimal wasted resources.

The problem with conventional software development is that it usually just isn't feasible to build a prototype. Instead, development teams will typically rely on a "wireframe"—a basic mockup image of a software interface accompanied by a series of flowcharts offering an overview of its logic. While this is better than nothing, as a tool for testing and building on an idea, a wireframe is cumbersome and imprecise. With nothing but words and diagrams, it's difficult to explain how software works—particularly to predominantly business-oriented counterparts. The conventional approach rarely leads smoothly to consensus and fails to adequately bridge the business-tech divide.

Using no-code, development teams can abandon this early step, shifting instead to rapidly building the real thing. In a fraction of the time—and without writing a single line of code—

organizations can develop sophisticated, scalable prototypes. In fact, using Unqork, a proficient tech user can often build a fully functioning prototype in the time it'd take to build a wireframe.

This sets a development team up for:

- Earlier consensus: Using functional high-fidelity prototypes, business and technology teams gain a shared visual language for collaborative development. It's a process both far more intuitive and significantly less prone to communication breakdown.
- More efficient use of time and resources: With layers of wireframing abstraction removed, business users are better positioned to confirm that the planned software successfully aligns with a company's business goals. This equips tech teams to sidestep dead-ends, and to avoid squandering time and resources.
- Deeper insights into the end-user's experience: A high-fidelity prototype gets a development team closer to pragmatically understanding the user experience. Tech teams can accurately identify potential pain points and avoid them.

2. Empowered Business Users

From early ideation right through to testing and launch, business users hold critical information to the success of a piece of software. It's predominantly the business users—not the programmers—who bring an intimate understanding of the corporate ecosystem, of what your customers want, and where your company as a whole is going. The business users on a development team know (in fact, often set) corporate objectives. They know what drives their customers and have granular data and intimate insights in regards to the customer experience. And all that information is vitally important.

We know that empowered and actively participating business users improve the development process. But how do you make that participation happen? The problem in getting business users more directly involved is that they are not programmers. A modern advanced coding language (like Python or Javascript) takes a year to learn and several more to master. It's simply unfeasible to expect a business user to tackle this body of knowledge before entering the conversation.

The result? A conventional enterprise software development process separates coders from business users with a near-impenetrable wall of jargon and mismatched expertise. The task of making software is hampered by both teams "throwing over the wall" with each new iteration of development. Ideas get lost. Opportunities are squandered.

With a visual programming platform such as that offered by Unqork, business users no longer throw their requirements over the wall and hope for the best. Instead of being relegated to a distant, advisory role, non-programmers can be using Unqork, often in as little as [three weeks](#). A tech user, of course, will still be able to pick it up faster and it will help them do more. But a non-technical person should at least be able to pick up the basics and handle smaller, more routine tasks. No-code empowers business users to:

Get closely involved in designing the user experience

By removing an impeding barrier to collaboration, the no-code development process allows business users to become much more closely involved in critical development decisions. Even a business user with little-to-no coding experience can actively participate in the development process.

The Unqork no-code platform allows teams to consider user experience, design, back-end development, and marketing goals in one unified and intuitive environment.

Instead of being consigned merely to a broad conceptualization of the application, your business team gets to see how all these elements interact early in the process. They can be directly involved in pivotal decisions about how the software should look and feel to the end-user.

Lift your gaze beyond the busy work

Finally, a managed no-code platform like Unqork does much of the “heavy-lifting” in terms of security compliance, legal requirements, and component upgrades.

For example, Unqork emphasizes efficiency, giving users many opportunities to reuse components they’ve already developed. Let’s say a tech user designs a customer portal for one segment of a software package. Using a no-code approach, it’d be a relatively trivial task for a business user to take that design and repurpose it for a similar task elsewhere. This reusability gives business users far more leverage for active involvement in the software development process. This recently became even easier with [Unqork Marketplace](#), where users can choose from a library of reusable application components.

No-code allows developers to shift focus away from the minutiae, from the myriad technical challenges they need to overcome to make software minimally viable.

Instead, business and technology teams can shift their gaze to how the software can best meet their company’s strategic business objectives.



3. Faster, More Responsive Software Iterations

Flash forward to a few months after your application was released into the world. It's all but an inevitability that a steady flow of customer feedback and marketplace change will, sooner or later, trigger a need to build on what you have. Your application has to grow and change in pace with demand, and if you're unable to adapt quickly, you'll quickly get left behind. But rapid software development is no small task. And it isn't just a job for the programmers. Business and tech team members face the challenge of continuing to work together to efficiently align the right business trajectory with software development best practices.

For several decades, speed has been a key criterion for software development. Most coding languages devised across the years were built to enhance a coder's productivity to get their job done quicker. But despite the growing sophistication of programming languages, the software development trajectory began to flatten.

By 2010, developers were making software 20% slower than they were in 2000, [research](#) by business analytics firm, QRM, reveals. A conventional coding approach just can't keep up with growing business demands.

A conventional programming approach tends to add layers of complexity to the code of each subsequent release. A host of factors come into play here:

- Evolving business requirements may push programmers into coding workarounds that cause unexpected and undesirable results.
- New third-party services may need to be brought into play.
- A new tech team member not knowing the history of decisions that shaped the original code.

Software updates frequently cause time and budget blowouts. As additional resources are poured into resolving legacy code constraints and bug testing, business and technology teams have no choice but to release updates at a glacial pace, with little to no prospect of responding to meet rapidly evolving demands.

Unqork's visual programming platform simplifies the process of releasing software iterations. No-code allows business and tech teams to more efficiently join forces in an intuitive drag-and-drop interface. Business users are therefore positioned to tackle routine upgrades themselves, leaving tech teams to handle more involved coding tasks.

Rapid and more responsive software iterations allow your business to:

Obtain early feedback from real users

No-code allows development teams to swiftly get their software to minimum viable product (MVP)—a product that is sufficiently developed to offer early-adopter users a real taste of the software's features and user experience.

By reaching this point earlier in the development cycle, companies gain valuable insight into how software can be improved and refined. And because that data is obtained early, business and tech teams are better positioned to make smart design decisions at a formative stage of development, saving both money and time.

Respond quickly to marketplace demand

A typical conventionally-developed software application requires [300,000 lines of code](#). Every line of code must be checked—not just to ensure its accuracy, but to be sure it doesn't break some other software component elsewhere in a company's shared software ecosystem. The result is mounting "technical debt," a growing and compounding expense that comes part and parcel with keeping a codebase organized and adequately resourced.

No-code removes the entropy of technical debt. Flowing on from this, the no-code approach significantly streamlines how technical teams translate business objectives to updated software design and enhanced user experience. At the height of the COVID-19 pandemic, [for example](#), New York officials were able to develop an information portal, from scratch, in days.

With greater agility comes increased responsiveness to what consumers want. [Interviewed for Forbes](#) in 2020, Liberty Mutual CIO James McGlennon said: "[No-code has proven to be] a minimum of three times faster and three times less expensive."

Foster innovation

Organizations that spend less time on maintenance can spend more time on innovation. Consider the initial time outlay at the onset of planning even a minor software update. Conventional developers spend an average of 17 hours per week simply studying code while implementing a codebase change, according to [2018 research](#) commissioned by the online payment platform, Stripe.

No-code separates application design from the codebase. Further enhancing the ease of design, the Unqork platform [guardrails](#) how Unqork users build applications, actively preventing them from making mistakes. This means you can update software with a focus on what your software does, instead of how to get there. Freed from worrying about programming language updates, third-party service changes, or a myriad of unexpected coding glitches, your team gains extra time to focus on what counts: Innovation.



In Conclusion

No-code isn't just about securing a faster and less expensive path to a minimum viable product. No-code application development also helps bridge the divide between technical and business team members. No-code makes it easier to blend business insight with coding acumen, and this can position your company to produce better software.

Enterprise no-code platforms like Unqork can empower your development team to build rapid prototypes, get business users more closely involved in development, and react more swiftly to a changing marketplace. To learn more about how Unqork can help improve the alignment between the technology and business teams at your organization, request a demo [here](#).

unqork

Enterprise application development, reimagined

Unqork is a no-code application platform that helps large enterprises build complex custom software faster, with higher quality, and lower costs than conventional approaches.

[Request a Demo](#)

[Learn More](#)