unqork

# Open the Floodgates of Enterprise Innovation

Four enterprise development challenges—and how no-code overcomes them.

# Contents

Introduction 3

Challenge 1: Insufficient Business-Tech Alignment 4

Challenge 2: Legacy Code and Technical Debt 5

Challenge 3: Technology Skill Shortages 6

Challenge 4: High-Risk, Resource-Intensive With No Guarantee of Success 7

In Conclusion 9

## TL;DR

- 70% of large technology projects fail to meet their goals, with up to 20% failing completely. Poor business-tech alignment, the time and cost burden of legacy code, and the acute technology skills shortage contribute to these failures.
- Creating a custom enterprise software solution typically requires around nine months to reach production, and software projects tend to run over time by as much as 33%.
- Enterprises are unable to move forward with their digital innovation because of the development baggage weighing them down. No-code platforms like Unqork break this cycle.
- Unqork improves the alignment between business and technology teams, stops new legacy code from being generated, bridges the tech skills gap, and accelerates application development.`

To remain competitive in today's technology-driven world, organizations from all sectors and of all sizes must embrace continuous digital transformation.

According to a recent study, 80% of digital transformation leaders say they've seen an increase in profitability as a result of their changes, and 71% say digital solutions make it easier to attract and retain top talent.

Global spending on these efforts is expected to reach $2.3 trillion by 2023. Organizations are looking to address rising expectations from customers and employees for fast, on-demand access to information and services. The push toward digital transformation only accelerated with the COVID-19 pandemic as organizations were forced to rapidly—and in some cases, radically—adapt around the global disruption.

However, at the enterprise scale, technology initiatives are not only promised to be long and complex, but they often fall short of goals. Recent analysis has found 70% of large-scale digital programs fail to reach their goals, with up to a fifth of all projects considered total failures.

This increase in partial or total project failure is rooted in the fact that while enterprise ecosystems ecosystems have become increasingly complex, traditional code-based (or even low-code-based) methodologies have not kept pace. Despite new programming languages, tools, and approaches, developer productivity has declined over the last decade.

Enter no-code. These solutions allow organizations to reduce complexities and speed up the digital transformation process by eliminating legacy code and technical debt and empowering companies to overcome skills shortages. A no-code application platform like Unqork helps enterprises address the issues at the core of their bloated tech budgets and high project failure rates.

In this eBook, we will explore the key challenges facing digital transformation and innovation and how no-code can mitigate or eliminate these issues. Specifically, we'll examine how insufficient business-tech alignment, legacy code, and the technology skills shortage lead to project failure and the stagnation of digital transformation—and how no-code can address these issues and more.

# **Challenge 1:** Insufficient Business-Tech Alignment

Poor alignment between business and technology teams has challenged enterprises for decades. It stems from the fundamentally different ways these departments operate. As both business needs and development approaches have grown more complex, this misalignment has only become more entrenched.

Underscoring this lack of alignment and collaboration is code. Code is a literal foreign language that few, if any, stakeholders on the business side are fluent in. Likewise, the needs of your specific business/industry can be just as foreign to engineers and developers that do not speak directly to the market, leading to a fundamental miscommunication between these two business segments.

The result? Business units don't understand code, so they lack clarity on how tech teams solve business problems. Meanwhile, tech teams must attempt to translate business needs into code, despite a similar lack of understanding on their end. This sets development projects up to fail from day one.

## No-Code Aligns Business & Tech Teams

No-code platforms help bridge the gap between business and tech by allowing both sides to communicate in the same language: an intuitive, visual interface that lets business users take a more active role in the development process. The drag-and-drop modules act as a layer of abstraction between the builder and the codebase. Even novice users can understand (and build in) Unqork's no-code platform in a matter of weeks, accelerating alignment between these two departments.

Just as important, no-code also accelerates the development of high-fidelity prototypes that allow stakeholders to offer actionable feedback and ensure business requirements are being met early in the development process rather than months down the line.

# **Challenge 2:** Legacy Code and Technical Debt

Technology budgets are increasing, but that doesn't mean enterprises are innovating more. Indeed, in large companies with more than 500,000 lines of code, maintenance accounts for more than half of the overall development budget.

That's because development technologies are changing so rapidly that any new code instantly becomes legacy code, requiring significant investment to maintain it, update it, or find ways to work around it. Any time an application element or coding language rolls out an update, developers must manually fix any issues that arise. The longer this pattern continues, the more costly the bugs and errors are to improve.

Legacy code always leads to higher maintenance costs, more errors, and an inability to adapt quickly. Engineers end up tied to outdated applications and must spend more time, energy, and resources on keeping the current infrastructure running rather than creating new, innovative projects that drive digital transformation forward.

Enterprise technology resources are further drained by technical debt, which refers to short-term fixes that have to be addressed down the line. One high-profile example of technical debt occurred in 2020 when short-term patches for the Y2K bug created in 2000 began to cause problems two decades later and had to be re-patched.

In many cases, technical debt is the result of overworked technology teams who are forced to settle for workarounds, and kick the problem down the line. It's a common issue—and an expensive one. Researchers have found that an average-sized application of 300,000 lines of code (LoC) contains $1,083,000 worth of technical debt, which translates to an average of $3.61 of technical debt per LoC.

Ultimately, too much legacy code makes the entire business operation complex and inflexible, creating a need for a modern and adaptable solution that reduces errors and maintenance.

## Focus on Business Challenges, Not Code-Based Ones

With a no-code platform like Unqork, your code infrastructure won't break down when backend components are updated since the platform automatically takes care of these changes. Your developers don't have to find quick fixes or code workarounds to get the system working again, meaning your system always works, and your team gets more time to spend on more innovative, transformative projects.

No-code platforms standardize how enterprises work with code and adds an extra layer between the developer and the actual codebase. This eliminates human coding errors because no human will ever type any code. Plus, there's no more need for new developers to interpret the work of their predecessors because all applications are built with a standardized, visual interface. There's no room for coding quirks or syntax errors. Everyone can contribute to development projects by focusing on application logic, not the minutiae of a specific coding language.

# **Challenge 3:** Technology Skill Shortages

Business needs are changing faster than the technology required to meet those needs, and technology is changing more quickly than new developers can be trained. As a result, companies must compete for a limited pool of experienced developers, which can be costly.

The global tech skills shortage is so acute that 87% of organizations struggle to find the right talent due to this rapid technological advancement. Even worse, LinkedIn reports that 55% of executives and employees alike believe that the tech skill gap is continuing to widen as businesses show no signs of slowing down their evolution.

What does this skills gap mean for enterprises? Finding experienced software engineering and developer talent is costly, to be sure, but a lack of appropriately trained developers causes other problems as well. Product releases are delayed, further increasing costs and forcing businesses to miss out on revenue opportunities. Maintenance issues pile up, contributing even more to the problem of technical debt.

Combined, the International Data Corporation estimates that the total financial impact resulting from the technology skills gap will grow to $775 billion by the end of 2022, up from $302 billion in 2019. To mitigate this financial impact, enterprises need a means of finding workers who can help them address the needs of a complex ecosystem. That's where no-code comes in.

## No-Code Bridges the Tech Skills Gap

No-code redefines the skills required to become a technology expert and provides solutions that empower workers to fit this new definition.

Traditional application and software development approaches take years to learn and master. On the other hand, no-code platforms can be learned in a matter of weeks thanks to a visual, configuration-based platform that eliminates the need to write any lines of code at all.

In this new no-code paradigm, less-experienced engineers are equipped to handle basic upgrades and maintenance that previously required a more experienced—and more expensive—programmer. More advanced engineers are then freed from these mundane tasks and can focus their time and energy on innovative projects that drive the enterprise's digital transformation forward.

With a no-code approach, enterprises end up with a more flexible workforce that can get more done with fewer resources. After the initial training period, no-code virtually eliminates the enterprise's technology skills gap while reducing personnel and code maintenance costs. That means more organizational resources can be devoted to innovation and digital transformation.

## **Challenge 4:** High-Risk, Resource-Intensive Projects With No Guarantee of Success

We mentioned earlier that 70% of large technology projects fail to meet their goals, with up to 20% failing completely. Broadly, this high rate of failure is due to changing business needs outpacing even the fastest traditional software development timelines, but there are inherent problems with the process itself too. (And of course, a lack of business-tech alignment, the burden of legacy code, and the acute technology skills shortage as outlined above all contribute to these high-risk, prone-to-failure development projects too.)

Time is perhaps the most significant resource needed to create a custom enterprise software solution, with most projects requiring **around nine months** to reach production, and other research indicating that software projects in particular tend to **run over by as much as 33%**.

Exacerbating this resource problem is the complexity of custom software projects and functionality, typically involving hundreds of thousands of LoCs. As noted above, this code instantly becomes legacy code, saddling the enterprise with more resource requirements—both time and personnel—to maintain and upgrade this software in perpetuity.

Even when custom software projects manage to stay relatively on time and on budget, the disconnect between tech teams and business users still contributes to partial or total project failures. Without consistent alignment between these teams, developers may create applications or software with a poor user experience that doesn't actually meet business needs.

When a project fails, enterprises tend to respond by increasing their investment in development projects, putting an ever-growing percentage of their total budget toward technology. However, they don't address the root problems in enterprise development approaches and simply continue the cycle of failed projects.

To achieve a higher rate of project success, enterprise businesses must break this cycle. They need a faster approach that makes it easier to pivot when business needs change while also establishing checkpoints throughout the development cycle that ensure the project is on track to meet the stated goals.

## No-Code Accelerates Development Cycles

No-code breaks the development cycle by shortening the timeline for completing a custom software project. As development cycles drop from nine months to two or less, the resources required to complete the project drop as well, immediately decreasing the associated risk.

A shorter timeline means a higher chance that developers can meet business needs before they stray too far from the initial request. Still, no-code is also inherently more adaptable to changing requirements than traditional code-based approaches. As mentioned earlier, rapid prototypes ensure projects adequately address business requirements before additional resources are invested.

Reducing the risk of project failure for run-of-the-mill development projects can also be the catalyst for new and innovative enterprise solutions. When the risk for standard projects is lowered, it's easier for enterprises to justify the higher risk for a project that is truly transformative to the organization. Additionally, no-code means less time spent on code maintenance and upgrades, freeing up even more resources for those innovative projects.

# In Conclusion

Traditional software development approaches have long caused massive tech budgets, mounting technical debt, and an increasing rate of project failure across the enterprise landscape. Because of this, enterprises are often stuck in the development cycle for years, unable to move forward with their digital transformation because of the development baggage weighing them down.

No-code platforms like Unqork break this cycle. By making development tools available to more people with less training than traditional code, enterprises spend less on engineering resources and projects while improving their productivity—and their innovation.

A no-code approach standardizes the language of development, so tech teams and business units can collaborate more effectively. It eliminates the problem of legacy code, halting its creation and linking into existing systems for routine maintenance and upgrades. No-code enables less experienced engineers to handle more complex issues, freeing advanced developers to work on fundamental digital transformation initiatives. And finally, no-code reduces the risk associated with large-scale tech projects, making it easier for enterprises to spend resources on projects that will move the business forward.

**Unqork can genuinely revolutionize your enterprise development process. To see how Unqork can enable you to unlock the floodgates of innovation, request a personalized demo here.**