unqork

How No-Code Transforms “Build vs. Buy”

Stop settling for off-the-shelf software. With no-code, enterprises can rapidly build custom software using a fraction of the resources.

Contents

TL;DR

- Building custom is better than purchasing commercial-off-the-shelf (COTS) solutions—72% of tech decision-makers claimed COTS would fail to keep up with the pace of change—but the expense and complexity forces enterprises to settle for COTS.
- With no-code, however, you can reap all the benefits of custom, while matching (and even surpassing) the peace of mind that is associated with a simple COTS implementation.
- We explore three reasons why Unqork should be the no-code solution of choice for enterprises: (1) Build custom solutions cheaper and faster, (2) configure the platform to your exact needs, and (3) achieve peace of mind with a future-proof application

unqork

Introduction

Businesses from across all industries are under increasing pressure to "go digital" in order to optimize internal workflows, amplify productivity, and address rising user expectations for digital experiences. According to Gartner, enterprises are on track to spend approximately **$557 billion** this year developing enterprise software, or a little bit more than the total annual **GDP of Thailand.**

When implementing enterprise software, organizations must choose between two separate strategies: building a custom solution or buying a commercial-off-the-shelf (COTS) solution.

Custom builds allow organizations to address all their business requirements and differentiate their offerings in the marketplace. However, building custom software is typically complex[1], resource-intensive, and time-consuming[2].

On the other hand, adopting a COTS solution is simpler and more affordable, but they provide limited customizability. The COTS route often means missing business requirements or having to settle for clunky workarounds. Enterprises have often been forced into COTS, even though custom is in their best interest.

Enter a no-code application platform like Unqork, which streamlines the development of custom enterprise software. No-code offers a combination of low cost, ease of use, and extreme flexibility that empowers enterprises to build robust custom software without necessitating the resources of a traditional code-based approach.

In this eBook, we will explore the ways in which no-code platforms ease the traditional friction points of custom enterprise development and upends traditional Build vs. Buy calculations.

[1] McKinsey & Company, **Finding the speed to innovate**. "Manual deployment of software can take 30-50 individual steps."
[2] McKinsey & Company, **Delivering large-scale IT projects on time, on budget, and on value**. "On average, large IT projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted."

Build Custom Solutions, Faster and With Fewer Resources

Between hiring specially trained engineers, integrating with legacy systems[3], and chipping away at a mountain of technical debt, building custom software gets very expensive, very quickly. While price ranges widely vary based on a plethora of variables (e.g., user requirements, feature set, complexity, size, etc.), an enterprise should expect to pay **at least $100k** for the initial build of a single custom application—though the prices can be far higher depending on the use case.

Building custom enterprise-wide applications also takes a lot of time, which adds to the expense and means a longer time before it can make an impact in the market—and the time frames are only expanding.

Throughout the digital age, software development has become more efficient... until 2010. Studies have found that around 2010, **build times began increasing**. Why the change? Today's digital architectures are disparate, expansive, and built on decades of deeply ingrained legacy systems (some built using **now-obsolete programming languages**). Traditional code-based approaches—or even "low-code" based approaches—are no longer able to keep up with the increasing demands of the modern enterprise. Beyond the effects on initial builds, this complexity introduces challenges throughout the entirety of the software development life cycle (SDLC).

Developing custom software using a no-code application platform requires fewer development-hours and specialized resources, thus lowering the cost of initial development and long-term SDLC management substantially. While enterprise software can take up to a year to go from ideation to production using a traditional code-based approach, no-code development cycles can be measured in weeks, months, **or even days**. In a world where **85% of custom software products go over schedule**, this is a huge difference in productivity.

In a **Forbes profile** on Unqork's CEO Gary Hoberman, Liberty Mutual's CIO James McGlennon said Unqork has proven "a minimum of 3x faster and 3x less expensive". How is this possible?

Unqork provides visual drag-and-drop components, templates, and pre-made workflows that sharply reduce the need for backend work compared to coding from scratch. This substantially reduces the time needed to build, test, QA, and modify custom products. The platform also makes it easy to seamlessly integrate new software with existing legacy systems and external third-party services.

[3] CFO Dive, **Colonial hack a wake-up call to CFOs with legacy systems**. "On average, the cost of managing an outdated IT system can cost a business around $3.61 per line of code or over $1 million for an application with 300,000 lines of code, said Kevin Shuler, owner and CEO of the Quandary Consulting Group, a Denver-based IT firm."

Here is an overview of the key efficiency (and therefore cost-saving) drivers no-code offers:

- **Accelerated Development:** No-code means faster development cycles, requiring the use of fewer developer hours. Enterprise applications like NYC's and DC's digital COVID-19 portals have even been created in days.
- **Increased Functionality:** Unqork comes "out-of-the-box" with all the toolsets and elements necessary to build complex business solutions (e.g., front-end UX, workflow, rules engine, analytics, integrations, and maintenance). Since they're all components of one unified platform, everything works together in instant harmony, allowing engineers to address business challenges instead of technical ones.
- **Improved Collaboration:** Feedback can be received and implemented more quickly for guiding iterations. While no-code necessitates fewer technology resources than code, it also opens the development process to more users, which allows organizations to save money without sacrificing quality.

3x

Acceleration in time-to-market compared to traditional development

600x

Fewer bugs compared to applications built with code

70%

Reduction in Total Cost of Ownership (TCO)

Meet to Your Exact Business Requirements

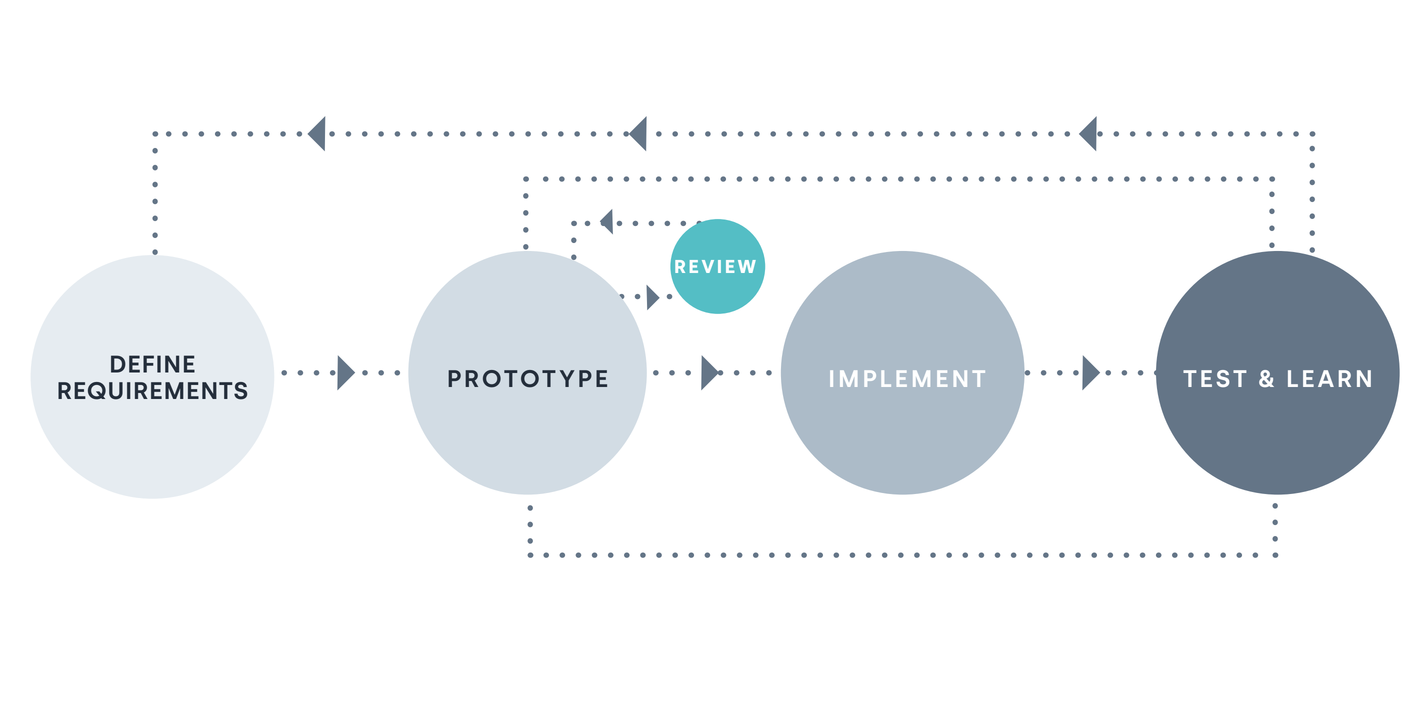
A survey of technology decision-makers found that most respondents preferred customized software over prepackaged software, and 72% claimed COTS solutions were unlikely to keep up with the pace of operational change.

COTS software is designed to address general business processes (e.g., case management or onboarding), not specific business goals or complexities (e.g., case management and onboarding as it is handled at your organization with your customers). The ability to meet all the needs and requirements for both the business and the customer/end-user is the main reason organizations choose to build a custom solution rather than buy off-the-shelf.

A custom-built solution is always better because it can deliver unique services and workflows that are not available in the COTS systems used by competitors, leading to powerful differentiators in a competitive marketplace. The only reason organizations still choose COTS is the aforementioned development challenges of a traditional code-based approach.

In addition to streamlining development, the inherent speed and accessibility of no-code make it far easier to translate unique business requirements into a digital solution precisely. In most large enterprises, there's a significant gap between tech and the business on understanding the objectives and functionality of a build. This misalignment often results in functionality being lost in translation because of poorly defined (or poorly understood) requirements, leading to disappointing ROI. No-code bridges the gap through the acceleration of functional prototypes and increased accessibility into the development process thanks to:

Rapid prototyping: With no-code, organizations can rapidly build—and rapidly iterate—high-fidelity prototypes. That means relevant stakeholders can provide feedback early in the process rather than months down the line.



Empowered business users: While learning to code inside programming languages like Java and Python can take years to learn, and a decade to master, Unqork can typically be learned in weeks. This opens up the playing field for who can add insights and workflows during the initial build process. Platform users can directly participate in the development and testing processes with minimal training. This means business teams, subject matter experts (SMEs), and executives participate in software creation alongside experienced coders and developers to ultimately deliver better, more innovative products. In fact, McKinsey found that companies that empower "citizen developers" score 33% higher on innovation.

Achieve Peace of Mind

One of the benefits of the COTS route is the relative ease it provides compared to building a bespoke solution using a code-based approach. With no-code, however, you can reap all the benefits of custom, while matching—if not surpassing—the peace of mind that is associated with COTS. Below are the four major ways building with no-code can reduce complexity, add value, and streamline operations in the short- and long-term.

1. REDUCED LEGACY CODE

Modern business systems can be years—or even decades—old. Maintaining “legacy code” demands an excessive amount of developers’ time, increases overhead costs, and inhibits an organization’s ability to efficiently address challenges[4]. Since Unqork divorces application logic from the codebase underneath, enterprises can reinvest resources previously dedicated to maintaining legacy code into other parts of the business, which can deliver value for years to come.

2. FEWER BUGS

A recent survey of 950 software developers revealed that nearly a third (32%) spend up to 10 hours a week fixing bugs. The majority (52%) said they would rather use this time to build new features and functionality, while 42% said freeing up this time would allow them to simply “do their job”.

With Unqork, developers can focus all their attention on building creative solutions to complex problems and building applications that enhance the organization’s bottom line. Since there is a layer of abstraction between the user and the codebase, it sharply reduces the risk of human error. A recent analysis found that using Unqork reduces bugs by more than 600x compared to custom applications built via code/low-code.

3. CONTINUOUS UPGRADES AND ITERATIONS

The same acceleration and streamlining that happens during an initial build are also applicable when making changes to applications down the line. If an update is ever necessary due to changing marketplace realities or new regulations, customer expectations, or technologies, tech teams can apply those updates without worrying about causing a software failure or other system malfunction.

[4] Consider last year’s unexpected rush on COBOL programmers. Many government and financial systems still run on COBOL, even though most universities no longer include this half-century-old language in their CS curriculum.

Unqork's configurable modules and components are easily manipulated within a completely visual drag-and-drop UI and include front-end elements, back-end processes, and third-party integrations all in one place. Thus, even as technologies change, the applications you build within Unqork will always work, empowering tech teams to integrate with the hottest new technologies or make updates to existing workflows without fear. Teams can also take advantage of Unqork's always-expanding roster of technology partners that allow cutting-edge functionality to be seamlessly added into any application.

4. FOCUS ON BUSINESS CHALLENGES, NOT TECHNICAL ONES

As technology elements evolve (e.g., one element or programming language goes from v5.2 to v6.1), Unqork handles all the changes on the back-end. All upgrades are vetted intensely by Unqork's Quality Assurance (QA) team to ensure processes go off without a hitch.

Traditional QA methodologies weren't sufficient enough for large enterprises that require rapid changes and have a high demand for quality. That's why Unqork's QA team developed six standard programmatic gateways for preventing defects, alongside an automation framework powered by CodeceptJS and Selenium to guardrail how Unqork users build applications and prevent them from making particular mistakes. Any time an Unqork engineer wants to make a change to the platform, for example, it has to be run through over 1,600 tests as part of the regression testing gateway alone.

Unqork does all the manual checkpoints, so you don't have to, thus future-proofing your organization behind the scenes.

The Time is Now to Build With No-Code

Using Unqork means enterprises can enjoy the benefits of creating custom solutions from scratch while avoiding the budget and time concerns that typically force COTS adoption. Organizations benefit from simplified application development and reduced barriers for updating, upgrading, and iterating applications once they have gone live.

Unqork is a completely visual, no-code application building platform that helps large enterprises bring complex, custom software to market faster and with higher quality and lower costs than traditional approaches. All without writing or editing a single line of code.

Ready to bring your build dreams to life? To see what we can do for your organization, set up your own 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