unqork

How No-Code Transforms Your Organization

Unqork's application platform empowers organizations to accelerate time-to-market, improve quality, and lower costs—without writing a single line of code

Contents

Organizations have always been faced with the difficult “build vs. buy” decision when it comes to software.

On one hand, they can choose to build an application from scratch. This process typically involves either using their own technology team or a consulting firm, collecting requirements from relevant stakeholders, and integrating an assortment of tools to complete the build.

While this method offers almost infinite flexibility, projects tend to run long, spiral out of control (AKA “scope creep”), and break down late in the development process as evolving business needs outstrip what tech teams were originally asked to deliver.

On the other hand, enterprises can buy a commercial off-the-shelf (COTS) solution which has been pre-built by a third party to address a business need. Companies often choose this path because it will, ideally, allow them to avoid the costs and complexity of building a new solution. On this path, organizations must research various packaged solutions, choose one with features that most thoroughly address core business requirements, and then fit that solution into their existing tech stack. Sounds straightforward, until the organization realizes just how much the software fails to match their requirements. Now they must customize, which starts to look pretty similar to a traditional, bespoke tech build.

And as Shakespeare wrote, “there’s the rub”—no matter which path an organization chooses, it is too often a long, unpredictable process to build a solution to meet its business needs. As a result, many projects end up dragging on for years[1], are descoped to minimal functionality, or run way over budget relative to the value they actually create[2].

Enter no-code application platforms such as Unqork, which offer the best of both worlds for organizations by combining the flexibility of custom software development with the standardization of a COTS system. Unqork provides organizations with an accelerator towards a custom application across product lines.

The ability to build complex applications without writing code opens up a world of possibility for the enterprise to create high quality, custom software that meets their business needs.

The most important question isn’t how no-code is different, it’s how no-code can improve an organization’s top and bottom lines.

Today, organizations are faced with increasing operations and technology costs coupled with higher capital requirements.

[1] Studies have shown that 85% of IT projects go over schedule.
[2] A report from McKinsey found that 70% of large-scale digital IT programs fail to reach their stated goals.

Technology project queues are skyrocketing as businesses go increasingly digital to meet customer expectations, reduce operational costs, and manage operational risk.

We'll explore the benefits that no-code unlocks for enterprises, with a focus on six chief outcomes that Unqork's no-code application platform can help you achieve:

- Increasing speed-to-market (and speed-to-value)
- Improving the quality of builds through improved collaboration between business and tech teams and through the prevention of bugs
- Reducing the long-term costs of software maintenance while increasing the ability to make changes down the line
- Attracting and retaining a broader, more diverse range of talent
- Reducing risk through centralized controls and platform-managed compliance engines
- Focusing your organization on true innovation

1 Accelerated time-to-market and time-to-value

The longer a solution takes to deploy, the longer it takes to realize value, and the more time your competition has to catch up—or vault ahead. Enterprise software can take up to a year to go from ideation to production using a traditional code-based approach. With no-code, development cycles are 3x faster[3], which means your organization will:

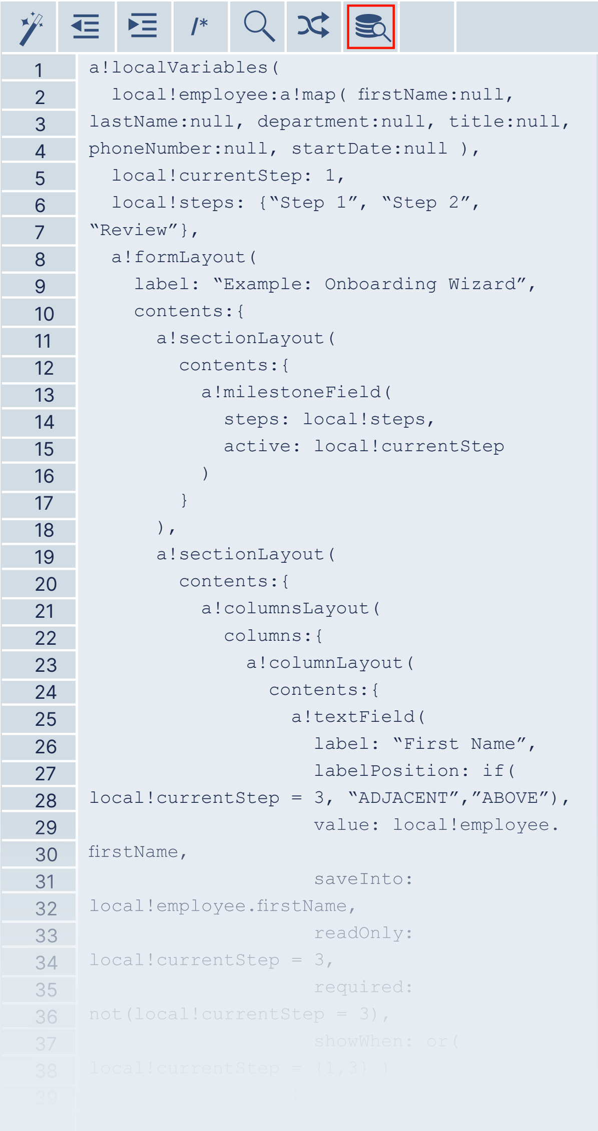
- ✓ Get to market faster than competition relying on traditional development approaches,
- ✓ Reduce development costs via faster development cycles using fewer resources,
- ✓ Receive market feedback earlier to guide iterations,
- ✓ Release more solutions to the market, and most importantly
- ✓ Realize value sooner.

This acceleration is possible thanks to the streamlining of build processes via Unqork's visual UI. "Visual programming" (of which no-code is one flavor) is far more efficient for building software than writing out lines of code. No-code is not a radical new concept (WYSIWYG website designers from the 1990s were technically "no-code"), but Unqork is the industry's first no-code application platform specifically designed for building enterprise-grade solutions in complex, highly-regulated sectors such as Finance, Insurance, Healthcare, and Government.

In Unqork, all crucial parts of an application, including user-facing elements, back-end logic, and third-party integrations are represented as reusable, configurable modules that can be drag-and-dropped into a visual UI.

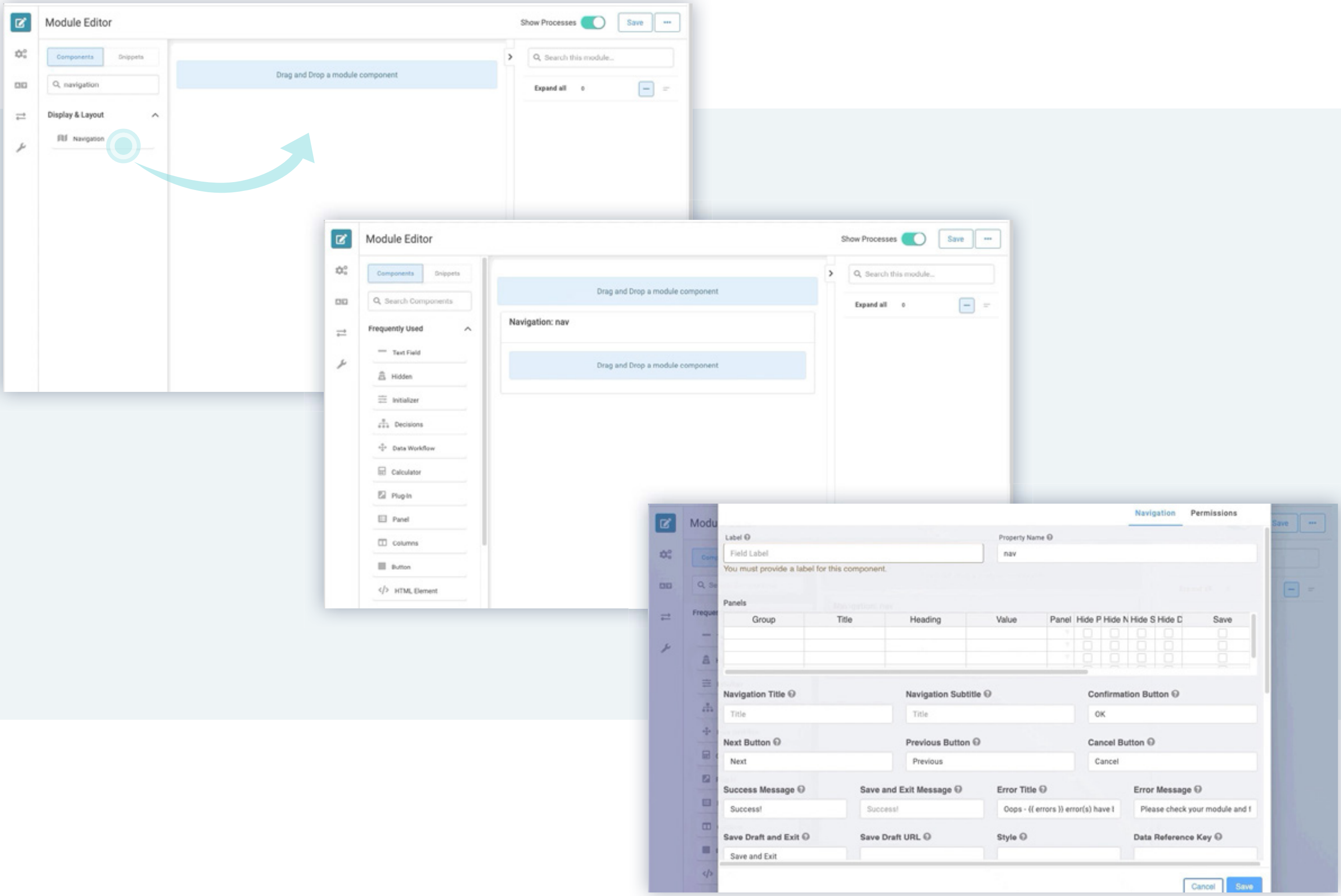
Case in point: In order to build a simple form like the one shown here using a popular low-code platform, a trained engineer must write-out 111 lines of code (or LOCs)—that's according to the platform's own website. This approach adds up in terms of time and resources. And obviously, this process would be even more arduous if taking a purely code-by-hand approach.

To produce the same user-facing form using Unqork, a programmer (or "Creator", as we refer to them) simply drags-and-drops a configurable pre-built module into a workflow and configures its attributes through a simplified graphical user interface (GUI). A trained Creator using Unqork can build the same form in a couple of minutes.



```
a!localVariables(
  local!employee:a!map( firstName:null,
lastName:null, department:null, title:null,
phoneNumber:null, startDate:null ),
  local!currentStep: 1,
  local!steps: {"Step 1", "Step 2",
"Review"},
  a!formLayout(
    label: "Example: Onboarding Wizard",
    contents:{
      a!sectionLayout(
        contents:{
          a!milestoneField(
            steps: local!steps,
            active: local!currentStep
          )
        }
      ),
      a!sectionLayout(
        contents:{
          a!columnsLayout(
            columns:{
              a!columnLayout(
                contents:{
                  a!textField(
                    label: "First Name",
                    labelPosition: if(
local!currentStep = 3, "ADJACENT","ABOVE"),
                    value: local!employee.
firstName,
                    saveInto:
local!employee.firstName,
                    readOnly:
local!currentStep = 3,
                    required:
not(local!currentStep = 3),
                    showWhen: or(
local!currentStep = {1,3} )
                  ),
```

[3] During the spring COVID surge, the city of New York partnered with Unqork to rapidly develop a suite of robust digital applications—some crucial ones in as little as 72 hours.



Label: Field Label

You must provide a label for this component.

Property Name: nav

Navigation Title: Title

Navigation Subtitle: Title

Confirmation Button: OK

Next Button: Next

Previous Button: Previous

Cancel Button: Cancel

Success Message: Success!

Save and Exit Message: Success!

Error Title: Oops - {{ errors }} error(s) have I

Error Message: Please check your module and f

Save Draft and Exit: Save and Exit

Save Draft URL:

Style:

Data Reference Key:

Modern enterprise applications contain dozens, if not hundreds, of individual elements of varying complexity. Unqork has been designed to produce everything from simple forms like the element above to complex processes such as generating quotes for insurance coverage. Taken together, building these various processes visually greatly accelerates enterprise development and empowers organizations to remain at the digital forefront.

[No-code has proven to be] "a minimum of three times faster and three times less expensive."

—James McGlennon, CIO of Liberty Mutual in Forbes

2 Improved quality of builds

Building fast is one thing, but is meaningless if it's not built correctly. Enterprises must ensure that all software they deploy:

- **Performs as intended at all times and in all scenarios.** "Buggy" software can end up being costly in terms of mitigating new problems[4], not to mention strikes against your organization's reputation.
- **Addresses all business requirements.** When using a traditional development approach, business teams are reliant on technical teams to translate their expertise into software. Indeed, research from McKinsey has shown that 70% of large-scale digital IT programs fail to reach their stated goals. Inadequate alignment can lead to additional development cycles and resource expenditure.

Unqork's no-code application platform empowers organizations to build high-quality, bug-free software that businesses can be assured meets their business requirements.

Don't squash bugs, prevent them

A recent analysis has shown that Unqork-built applications contain 600x fewer bugs than those built using a traditional code-based approach. This is chiefly due to the fact that, as detailed above, Creators are only responsible for defining application logic, and leave the technical details to the platform.

When a new element is created for the Unqork platform (e.g., an integration to a third-party API), the feature is tested and reworked until it reaches our standards of quality before becoming part of the platform. Creators can apply these features repeatedly without any fear of creating new bugs.

The platform's continued stability is maintained by a state-of-the-art QA function that combines a global human workforce with advanced automation. This framework ensures Unqork-built applications always perform exactly as they should, even as workflows, workforces, and application components evolve over time. Furthermore, as we identify common logic-, misconfiguration-, or other Creator-made bugs, we can proactively integrate guardrails into the UI to mitigate issues even further.

Bridge the business-technology divide

The complex nature of modern enterprise tech demands that business teams rely on technology professionals to translate their needs into digital solutions. These teams often speak in completely different languages, figuratively speaking. This results in a lot of functionality getting lost in translation because of poorly defined (or poorly understood) requirements, resulting in poor ROI.

[4] Studies have found that programmers spend up to 75% of their time locating and squashing bugs, costing businesses $113 billion in the US alone

No-code bridges this divide by allowing technology teams to rapidly build functional prototypes, so business teams can be assured requirements are being addressed early in the development process, rather than after investing time and resources. Furthermore, no-code's intuitive visual interface allows business users to directly take part in all stages of the Software Development Lifecycle (SDLC). Indeed, as many organizations develop their no-code functionality, they often embed business users in the no-code Center of Excellence (CoE) as both advisors and active participants.

Read The No-Code Journey eBook to learn more about the stages organizations take on their way to becoming a true "DIY Enterprise."

3 Lowered maintenance costs and increased flexibility

It's not uncommon for the average large enterprise to spend 50-75% of its total tech budget on maintaining existing systems. Simply put, complex applications require complex maintenance. Anytime you want to make a modification, integrate a new system, or upgrade an underlying technology, you're typically faced with an unwieldy project.

Unqork alleviates many of these issues because:

- **The same acceleration of initial builds can be applied to updates.** Reworking a workflow via Unqork's visual UI can be done far quicker and with fewer resources than a traditional code-based approach.
- **Changes in Unqork are automatically cascaded throughout complex systems.** Whenever a change in a program is made, it often must be reflected throughout the application. For example, a simple change to an application's front-end (e.g., adding a field to a form) must cascade down to the database, which must then also be modified. These cascading changes are difficult to address with a code/low-code-based approach. With a no-code system, many of these cascading tasks are automatically executed.
- **We handle technological changes on the backend.** In most cases, if a single element of your digital infrastructure is upgraded, it won't disrupt your solution. But with enough changes throughout a complex ecosystem, workflows will inevitably break. With Unqork, you don't have to worry about technical changes because we handle them on the backend.
- **We eliminate the creation of legacy code.** Modern business systems can be years—or even decades—old. Maintaining "legacy code" demands an outsized amount of developers' time, increases overhead costs, and inhibits an organization's ability to efficiently address challenges. Since Unqork divorces application logic and the codebase, enterprises can reinvest resources previously dedicated to maintaining legacy code into other parts of the business.

[5] 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.

4 Digital payments & fulfilment

In order to build and maintain modern enterprise solutions, large organizations must compete for a limited pool of software engineering and developer talent. This pool becomes even smaller when it comes to advanced value-additive digital functionality such as machine learning, cognitive computing, and data science.

One study by Forrester found that while 74% of enterprises have a digital strategy in place, only 15% report having the right skills to deliver it. As a result of this "skills shortage," project logs are backed-up and engineers can't be easily replaced when they leave. This is where Unqork can be a game changer.

While modern programming languages (Java, Python, etc.) can take a year to learn and a decade to master, no-code can usually be learned in a matter of weeks. As a result, organizations can widen their potential talent pool to include a greater variety of backgrounds, from less-experienced engineers to non-technical business analysts, who can take on routine maintenance tasks.

Unqork also benefits experienced tech talent, because they can work on projects appropriate to their skills and experience. No-code eliminates tedious coding tasks and allows developers to focus on building value-additive solutions they signed up for. In a traditional code-based ecosystem, a lot of developer time is inevitably spent on "work about work," such as coordinating work efforts across teams, attending meetings to prioritize coding tasks, or dealing with legacy code across your tech stack. This can enhance the experience of the 30% of workers which studies show don't feel their talents and abilities are being used to the fullest.

5 Reduced risk to your organization

Faced with increasing pressures from global regulations, organizations are challenged to balance compliance and competitive innovation. Unqork makes it easy through centralized controls and up-to-date compliance engines and security protocols baked into the platform.

Unqork is a completely unified SaaS platform, in which components and capabilities are managed centrally, including those for crucial areas like **compliance** (up-to-date regulatory and enterprise rules engines for FATCA, CRS, UK CDOT, Dodd-Frank, EMIR, and MiFID II, etc.), **security** (native encryption both in transit and rest, custom RBAC capabilities, and crowd-sourced penetration tests), and **application management** (SDLC governance, application versioning, and module management).

With Unqork, individual teams don't have to "reinvent the wheel" with every solution in order to maintain compliance—everything is managed by authorized administrators in a single unified location, so changes can be automatically cascaded throughout the ecosystem. This is particularly crucial as companies empower business users and less-experienced programmers.

.

6 Return to innovation

Since coded solutions have high fixed costs to begin with, budget limitations have traditionally dictated that only the projects with an obvious value-add end up moving forward. This makes sense within the constraints of traditional application development, however it means that the boldest ideas may never see the light of day.

Experimentation, however, is where competitive advantages are taken or secured. Consider how large technology companies like Alphabet and Facebook are constantly deploying new digital products and functionality—some prove to be successful, others are simply abandoned. Experimentation allows these titans to maintain their position in a fast-moving marketplace.

Unfortunately, most companies probably aren't in a position where they can allocate massive amounts of resources into developing projects where the result is less than certain—at least, not when using a traditional code-based approach. With no-code, companies are able to mitigate the risk of development, and build-out the capacity to experiment with addressing business problems in novel ways.

[6] In 2019, Facebook invested $13.6 billion on R&D, and Alphabet spent more than $26 billion.

How to Begin

So, what's a good way to get started?

If you're like most organizations, you probably have a very complex technical org structure, lots of projects in flight, and many years of accumulated legacy code.

The good news is that with a no-code platform, you don't have to tackle everything at once. After walking through this process for many different customers over the years, our advice is to start with one use case or process that's been particularly challenging.

Don't pick something too easy! While it's tempting to pick something you can execute fairly quickly for a first use case, you won't be able to assess the true impact of no-code on your organization unless you "kick the tires" thoroughly.

Using this approach, you can start to see exactly how no-code changes your organization in a real, live use case.

Once you see success in one area, you can start to plan for a transition from legacy systems and train Creators in your organization to build on the platform. We find this approach dramatically reduces the risk and unexpected organizational impact of bringing on a no-code platform.

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