

unqork



# Code No More: The Power of Visual Programming

Unqork is leading a new generation of enterprise-grade visual programming platforms that empower users to build complex business applications **without writing a single line of code.**

# Contents

Introduction	3
What Is Visual Programming?	5
How Visual Programming Benefits Organizations	6
1. Reduced Bugs	7
2. Accelerated Development	8
3. Increased IT Resource Flexibility	9
4. Streamlined Development	10
5. Eliminated Legacy Code	11
The Future Is Visual	12

**TL;DR**

- Enterprise development has become less efficient over the past decade due to complexity and limited IT resources
- Visual development is a way of building software without writing code
- Visual development isn't new, but has become more critical in the enterprise space thanks to no-code application platforms like Unqork
- Visual development makes development faster, accessible, more affordable, and of a higher quality

Over the past several decades, the north star in enterprise software development has been to build faster, faster, faster. Anything developers and enterprises could do to hasten time-to-market was good. These improvement efforts included the evolution of preferred programming languages from COBOL to Java to Python as well as new software development methodologies like Agile, Scrum, and Kanban.

And for many decades, these efforts succeeded in making developers more productive. But then, something strange happened. Around 2010, developer productivity began going in the other direction.

Research shows that developers were creating software 20% slower than they did in the aughts despite the advances in languages and methodologies. There are several reasons enterprise development suddenly became more difficult, but the chief drivers are:



**Complexity:** Modern enterprise ecosystems must incorporate multiple teams worldwide, a vast array of internal legacy solutions, and a growing number of external third-party services. Resources must be invested to tie all these elements together effectively.



**Legacy Systems & Technical Debt:** Modern organizations dedicate an increasing amount of IT resources to managing legacy systems (some of which may be decades old and built by programmers who have long since moved on<sup>1</sup>) and paying down their technical debt (i.e., workarounds that must eventually be addressed to keep systems functioning). All this maintenance demands an outsized amount of developers' time and makes it more difficult to build new functionality.



**IT Skills Gap:** Organizations are forced to compete for experienced developers within a limited pool of talent—and competition is particularly fierce when it comes to advanced fields like machine learning and data analytics.

<sup>1</sup>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.

What ties these three trends together? Code. This is why enterprises are increasingly sidestepping editable codebases altogether and embracing visual programming approaches such as no-code application platforms like Unqork.

In this eBook, we will explore the power of visual programming and how it can help your organization innovate at speed, stay ahead of the competition, and lower overall costs while increasing your productivity and efficiency.



# What Is Visual Programming?

[Visual programming](#) is a development method that empowers developers to define complex processes using modules, components, and illustrations rather than text, which is a far more intuitive way for humans to convey ideas.

Before moving forward with visual development, let's break down the nature of traditional code-based development. Code is, at its core, a series of instructions presented into a language that a machine can understand. When building with a traditional code-based approach, developers must "translate" each step in the instructions into machine-readable code—and if one piece of syntax is out of place, the whole program can break. Modern visual programming tools such as Unqork provide a layer of visual abstraction between the developer and those instructions. This simplification makes building software faster and more accessible, while eliminating the possibility of human coding errors.

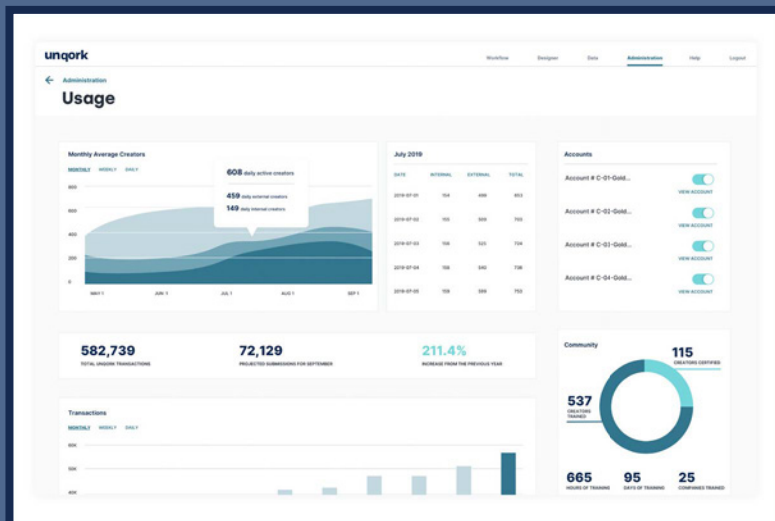
Visual programming isn't a radical new technological innovation. It's actually been around for nearly as long as text-based coding. There's a long history of using visual programming tools to build applications, such as the flowchart-based systems like [Pygmalion](#) or [GRail \(Graphical Input Language\)](#). Consumer-level applications and WYSIWYG web builders in the 1990s were also technically forms of visual programming.

Recently, visual programming has become a topic of renewed interest in the enterprise space thanks to advanced "no-code" platforms like Unqork which have been specifically designed for building robust applications in dynamic and highly regulated environments like finance, insurance, healthcare, and government—all without writing a single line of code.



# How Visual Programming Benefits Organizations

Visual programming empowers developers to design complex business processes using graphical representations of user-facing elements, back-end logic, and third-party integrations.



With visual programming, IT teams can complete initial builds faster, collaborate with business teams more closely, and simplify their maintenance work.

Let's explore some of the specific benefits that visual programming can bring to your enterprise.

**600x**

The factor by which bugs are reduced using Unqork compared to a traditional code-based approach

## 1. Reduced Bugs

Visual programming eliminates—or greatly mitigates—programming bugs because it completely removes the need to write code, and therefore removes the possibility of human coding errors. (It, of course, would still be possible for a human to create a workflow or logic error, but Unqork has installed guardrails to automatically identify and alert users to potential problems.)

Platforms like Unqork use modules, components, and illustrations instead of text, which frees developers from ever needing to type out code themselves. The [QA team at Unqork](#) ensures that all components behave as intended and work in harmony with all other elements. So organizations never need to invest resources in perfecting syntax or ensuring that elements are properly integrated with one another. Indeed, a recent analysis has demonstrated that building with Unqork [reduces bugs by more than 600x](#).

## 2. Accelerated Development

The average software application comes with [300,000 lines of code](#). Building, maintaining, and organizing all that code demands resources. If your development teams are still relying on a traditional code-based approach, they're spending a lot more time and effort than they would be with a visual-based system.

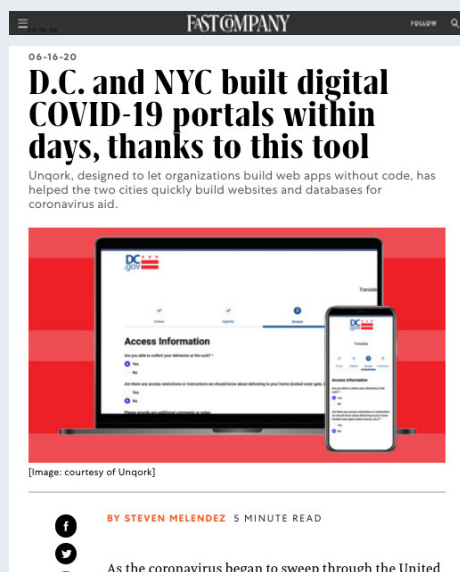
Visual programming reduces the need for these resources because it moves developers one step away from the code itself. Developers can use the drag-and-drop nature of the platforms to achieve the same functionality as hand-coding in less time and with less effort.

**[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](#)

Visual development makes development on average [three times faster](#) than a traditional code-based approach. Recent history has demonstrated that no-code application platforms can help deploy complex solutions [in a matter of days](#), that might have otherwise taken several months to go from ideation to production.

### IN THE MEDIA



The image shows a screenshot of a Fast Company article. The article title is "D.C. and NYC built digital COVID-19 portals within days, thanks to this tool". The sub-headline reads: "Unqork, designed to let organizations build web apps without code, has helped the two cities quickly build websites and databases for coronavirus aid." Below the text is a photograph of a laptop and a smartphone displaying the "Access Information" portal. The article is attributed to Steven Meleendez and is a 5-minute read. The beginning of the article text is visible: "As the coronavirus began to sweep through the United".

“Instead of relying on code, the platform for New York-based Unqork provides a drag-and-drop, flowchart-style interface to specify how forms should collect data and how back-end logic, like determining who is eligible for what kind of programs, should function. That means that people who are familiar with the ins and outs of government and corporate operations can often quickly build working digital tools themselves even if they don't have coding expertise.”

— [Fast Company, June 16, 2020](#)”



### 3. Increased IT Resource Flexibility

Visual programming is a far more intuitive approach to development than a traditional code-based approach and can be picked-up **far quicker** than a new programming language. This is particularly apt with younger “digital native” workers who have grown up using technology as a tool for analysis, research, and self-expression. Visual programming combines all of these things, and is often easy for these younger generations to pick up.

The intuitive nature of visual programming empowers non-technical business users to directly take part in the development process rather than relying on IT. Since the platform takes on the “heavy lifting” of development, less-experienced developers can produce the same quality of builds as their more-experienced colleagues.

Visual programming allows organizations to be more flexible cultivating their development workforce. Less-experienced developers and business users can be assigned to routine maintenance or upgrade activities, while more-senior engineers spend their time addressing unique or complex functionality.

This flexibility is becoming increasingly crucial as the impact of the “**IT Skills Gap**” is predicted to expand and industries around the world are forced to become even more competitive for a limited number of IT workers

**\$775**  
billion

Predicted global financial impact of the IT Skills Gap, which is more than double what it was in 2019

**15%**

Number of enterprises that report having the skills in place to deliver their digital strategy



## 4. Streamlined Development

Advanced no-code platforms like Unqork streamline the development process within sprawling ecosystems so developers can focus on overcoming business challenges instead of technical ones.

Complex workflows and capabilities are built with drag-and-drop logic blocks, while behind the scenes, the platform takes care of security compliance, component upgrades, and much more.

No-code platforms provide business and IT users alike peace of mind in that their applications and software will always comply with business, industry, and legal requirements, no matter the situation.

### unqork components



#### **Compliance:**

FATCA, CRS, UK CDOT, Dodd-Frank, EMIR, MiFID II, etc.



#### **Security:**

Native encryption both in transit and rest, custom RBAC capabilities, crowd-sourced penetration tests, etc.



#### **Application Management:**

SDLC governance, application versioning, and module management

## 5. Eliminated Legacy Code

Legacy code is a bit of a catch-all term that developers use to describe:

- Code from outdated programming languages that is still in use
- Code in outdated software applications created with older versions of modern coding languages
- Code that was written by a different developer than the one working on it today

Most developers will say that all code is legacy code because it becomes outdated the moment it's compiled and released.

With each new enhancement or fix, a codebase becomes more complicated for the next developer that must work on it. Research has shown that developers spend [an average of 17.3 hours a week studying code](#) and figuring out how to shoehorn their changes into it so that it doesn't break the application. The continuous stream of resources required to maintain legacy code is known as "technical debt."

Technical debt is a general term representing the effort required to fix problems remaining in code that's released into production.

As mentioned above, the average software application contains 300,000 lines of code (LOCs), and [according to one study](#), every single LOC is associated with \$3.16 in technological debt that will eventually have to be paid. That means the average application has over \$1 million of technical debt right now.

Your company's technical debt grows every year, so if you want to eliminate it entirely, it's time to switch to visual programming. A no-code platform reduces technical debt right away by speeding up the overall application development timelines and eliminating code maintenance entirely. Developers don't need to study the code anymore to see how their predecessors did it. The no-code platform handles all of it "under the hood," so they can concentrate on the process the code solves.

**300,000**

Number of LOCs  
in an average  
software application

**\$3.16**

Average amount of  
technical debt associated  
with each LOC

**17.3**

Average number of  
hours each week  
developer spend just  
studying code

## The Future Is Visual

Visual programming isn't a new idea, but its latest iteration—no-code—takes the concept to the next level. Companies looking to create an innovative environment that fosters creativity and collaboration need to adapt sooner rather than later if they want to remain competitive in today's rapidly transforming world.

Enterprise no-code platforms like Unqork are more attractive to companies who want to save money, shorten product development timelines, and empower all employees to contribute directly to product development.

At Unqork, we believe that visual programming, and no-code in particular, is the future. No-code makes it easier for any user to harness the power of technology to create new and complex enterprise applications without ever having to write a single line of code.

Want to learn more? [Schedule a personalized interview](#) with one of our in-house no-code experts.

**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](#)