unqork

How No-Code Ramps Up (Not Replaces) Coding Talent

How no-code saves developers time, boosts productivity, and paves the way to higher quality, more innovative work

Contents

Introduction	3
3 ways no-code makes developers more productive	4
Benefits of transitioning from coding to configuring	8
Final Thoughts	9

TL;DR

- Many coders and software developers spend an inordinate amount of time on high-volume, low-value tasks.
- Bogged down as they are maintaining and rewriting code, effecting bug fixes, or modernizing legacy systems, some developers worry no-code solutions will replace their hard-won skills and experience.
- Quite the opposite is true, in fact: Top-performing companies take advantage of no-code to empower less experienced business users while freeing seasoned coders to focus on the most challenging tasks.
- By removing the need to work in a codebase, Unqork's no-code enterprise application platform makes Creators of all types more productive by reducing development times, speeding up or eliminating low-value tasks, and allowing coders to spend more time engineering inventive solutions to complex business challenges.

Despite distinct time limitations, coders and software developers are often painfully aware of the number of hours they squander on high-volume, low-value tasks. Research suggests most coders spend **just four hours** or less each day actually writing code. What else are they spending their time on? A 2019 survey fielded by **The New Stack** showed that:

- **Meetings, management, and operations** take up 14% of a software developer's time
- **Testing and security issues** occupy another 18%
- **Code maintenance** chews up a full 22% of the typical developer's day

Managing and **modernizing legacy systems**, for example, can mean researching hundreds of thousands of lines of code, while at the same time struggling with antiquated tools or development processes[1].

Worse, given the subjectivity of building with code—and the high **turnover rate among technology workers**—coders often find themselves devoting large amounts of time to **researching the work of their predecessors**, further complicating upgrades and updates.

Identifying and **fixing bugs** in written code, meanwhile, not only wastes valuable development time, it can significantly increase stress and frustration. One recent study found that roughly **a quarter of developers** would rather pay their bills or endure a trip to the dentist than work on bug-related errors.

Finally, because they spend an inordinate amount of time **maintaining and rewriting code**, programmers have significantly fewer hours to devote to building out vital new features and functionality.

One of the biggest time-wasters where today's tech teams are concerned is the lack of an efficient, streamlined development process[2]. Fortunately, Unqork's no-code platform is changing all that by allowing developers of all skill levels—including the most seasoned programmers—to optimize application development.

While some traditional developers may be concerned that no-code technology is poised to replace them and the skills and experience they've so painstakingly acquired, **quite the opposite is true**.

[1]Some **legacy code** bases—like those involving single-tiered, monolithic applications, for example—while perfectly functional and feature-rich 10-20 years ago, were frequently written using techniques that don't suit modern software development.

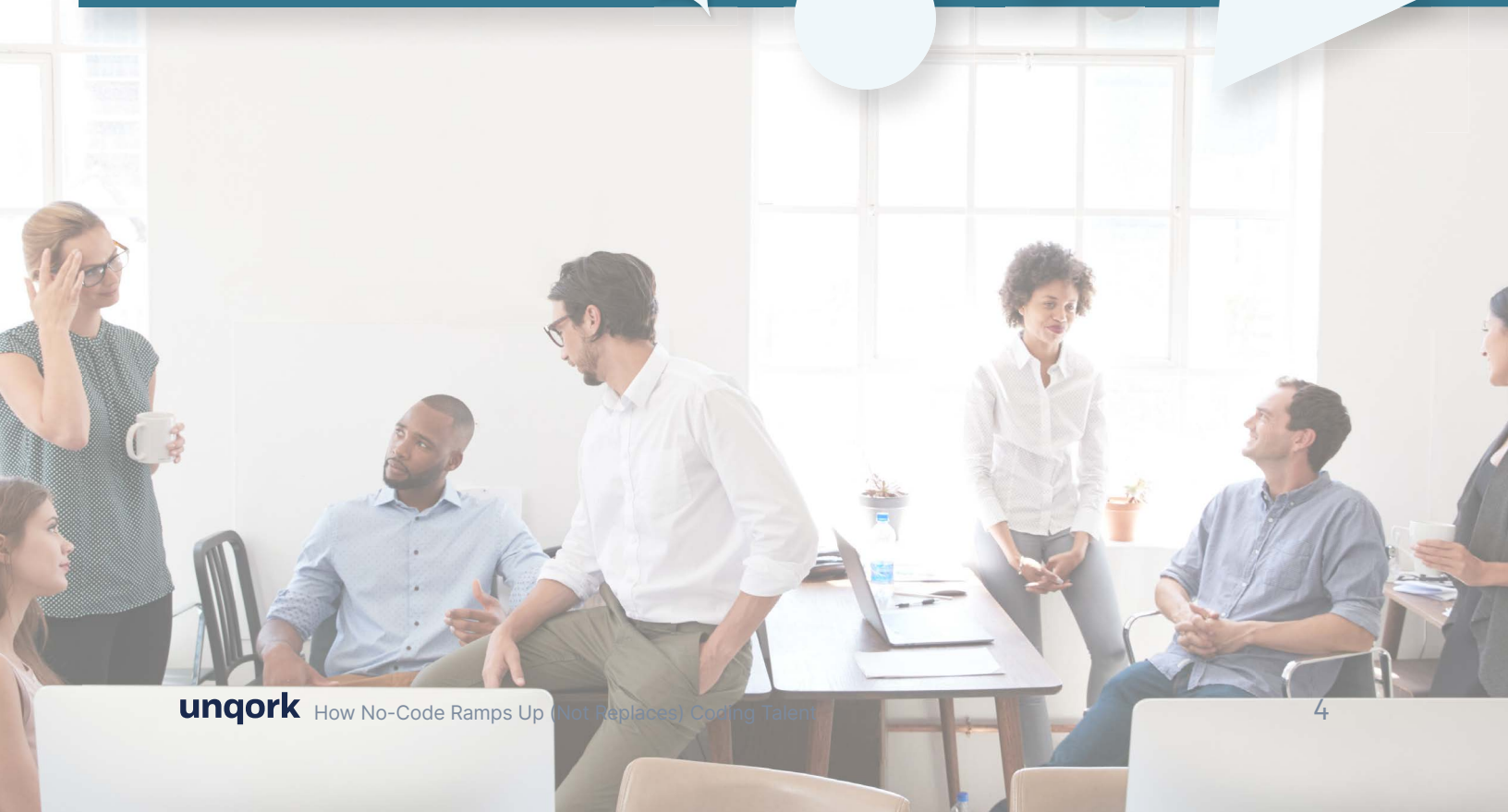
[2]While essential, bug fixes and code glitches **waste valuable work hours**. It's in every development team's best interest, therefore, to find and plug common productivity drains.

According to McKinsey, companies that score in the top quartile for “developer velocity” (an ability to unleash the full potential of their development talent) invest in tools like no-code platforms to:

- Enable the average business user to develop applications without any software experience
- Free up seasoned developers to focus on the most challenging tasks
- Score 33% higher on innovation than bottom-quartile companies as a result

By effectively removing the engineering—not the engineers—from the development process, no-code ramps up a coder’s ability to do more meaningful work by freeing them from routine development tasks. Even the most experienced engineer can benefit from using no-code to maximize their time, effort, and workload efficiency.

In this eBook, we’ll discover why taking advantage of a no-code platform like Unqork allows coders of all types to save time and increase their output so they can focus on more innovative pursuits.



3 ways no-code makes developers more productive

Although new programming languages, coding strategies, and development tools like low-code escalated developer productivity over the PC era from around 1980 to 2000, things started to go in the opposite direction shortly thereafter. By 2010, the data shows it was taking 20% longer on average to complete an enterprise application project than it had in the previous decade.

In addition to tech talent scarcity, slower development times have largely been the result of:

- An upswing in digital landscape and application complexity
- Compounding legacy code challenges
- Stop-gap coding solutions that contribute to technical debt

The good news is that while code-based tools and methodologies have done little to keep up with the demands of today's digital ecosystems, no-code solutions are both a proven antidote to slower development times and one of the best ways to mitigate dwindling coder productivity.

By auto-generating and managing code for you—and replacing it with graphical representations in an intuitive, visual UI—Unqork's no-code enterprise application platform makes it easy for tech and business users alike (or Creators, as we refer to them at Unqork) to leverage the latest technologies and build sophisticated applications without compromising development timetables.

Because no-code's visual programming tools and drag-and-drop components dramatically shorten the application building process, project development cycles that used to be measured in months, or even years, can now be measured in weeks.

Here are three ways that moving at the speed of no-code with Unqork can shave precious hours off development schedules and free coders to spend more time on the projects that matter.

You'll spend less time working on legacy code

Because keeping software up-to-date is a must for staying competitive in today's rapidly changing market, many companies commit substantial resources—including their developers' time—to modernizing legacy systems.

Given that such systems may have been written years or even decades before, in a language that few now understand[3], a significant portion of your day may be spent struggling to keep legacy code functioning properly or orchestrating change into outdated systems.

This can prove especially time-consuming if it means dealing with:

- Software that relies on older code designs, techniques, languages, or libraries that are no longer in common use
- Poorly detailed or undocumented software built by past developers who have since moved on from the company (because equally talented engineers can overcome the same business challenges in different ways, almost 50% of the time a developer spends on code maintenance goes to trying to understand the code they're maintaining)

[3]One surprising upside of the buildup in legacy systems is that coders with the ability to write in relatively ancient languages, like COBOL for example, have suddenly become essential again.

Fortunately, the problem of legacy code modernization (something nearly every software engineer is tasked with) can be greatly improved with the help of no-code.

Whereas researching and maintaining years of legacy code might normally take an enormous bite out of your daily schedule, no-code makes it easy to "repackage" pieces of old software as APIs so you can revitalize previous work and breathe new life into individual services. By removing the need to write in code, Unqork also eliminates the problem of code subjectivity.

The upshot is that you'll have more time to deal with other pressing tech challenges (like the fallout from Shadow IT[4], for example) while advancing more high-value tasks.

You'll spend less time chasing down and squashing bugs

Coders know all too well how monotonous and time-consuming debugging can be. With 13 errors per 1,000 LOCs[5] accompanying the average coded application (a number that can reach into the millions for enterprise-grade applications) it's no surprise that:

- Nearly one-third (32%) of developers say they spend up to 10 hours a week fixing bugs
- 16% report squandering as many as 15 hours weekly
- 6% say that, rather than writing code, they devote up to 20 hours a week to bug fixes.

Since the vast majority of bugs and security vulnerabilities stem from human error in written code, by taking code out of the equation, you also remove the potential for such time-wasting problems to occur.

In 2020, all applications built with Unqork averaged just 0.2 bugs per LOC. Not only does that represent an improvement of more than 600x over code/low-code approaches, it includes all kinds of bugs—from mathematical errors, to misconfigurations, to log errors.

Squashing bugs is resource-intensive.

No-code Unqork frees software specialists to:

- ☑ **Create enterprise software quickly with fewer bugs.** Overall, enterprise applications built using Unqork's no-code platform aren't just notably less buggy, they're also more reliable and significantly faster to build than code-based applications
- ☑ **Offload technological upkeep**. Because Unqork prevents bugs from the start, you won't have to waste valuable time fixing them later. We also take care of emerging bugs on the back-end of your application if any of the underlying application elements shift (such as a programming language or third-party integration updating from v2.0 to 2.6). In most cases, small changes won't break things. However, with traditional coding, if enough changes happen, then breakage can, too. That's not the case with Unqork's platform.

[4]According to McAfee, not only is Shadow IT cloud usage at least 10x the size of known cloud usage, 80% of workers admit to using SaaS applications at work—in many cases without IT approval.
[5]Using Commercial product code, the benchmark is ~13 bugs per KLOC.

Building dependable, high-quality enterprise software with no-code can save you the time, stress, and frustration so often associated with chasing down and squashing bugs—both today, and into the future.

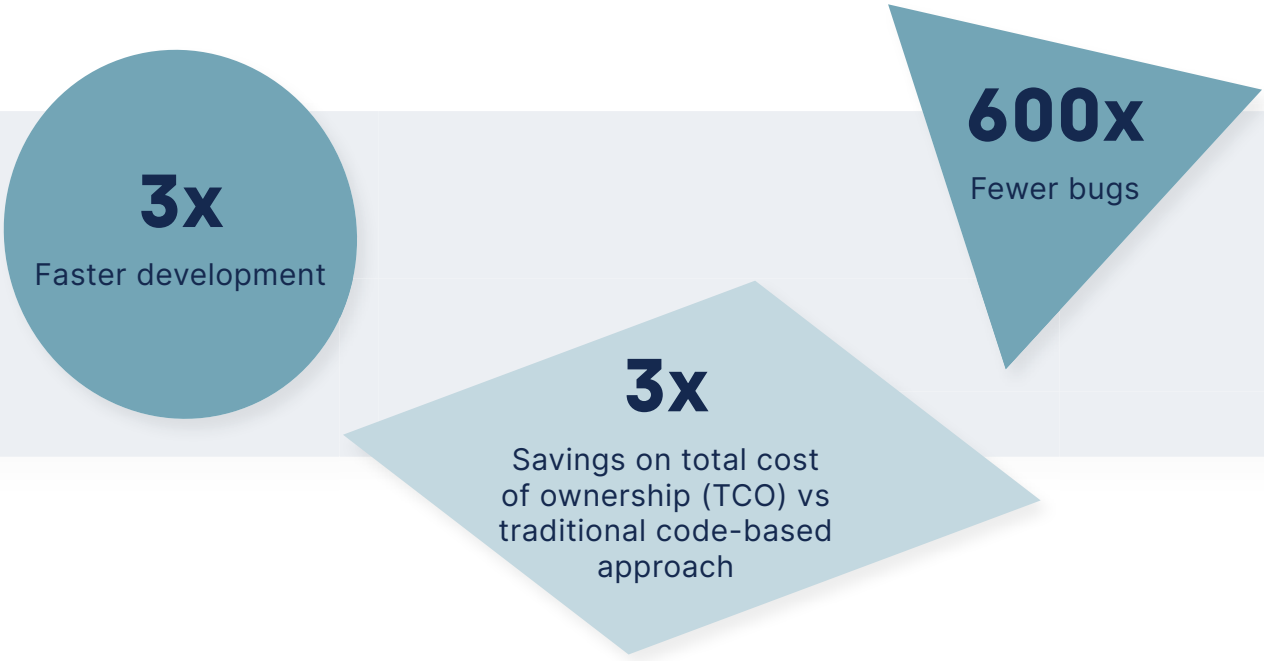
You'll spend less time on high-volume—but low-value—development tasks

Modern companies are under **increasing pressure to "go digital"** so they can feed operational efficiency and meet users' evolving expectations for digital services.

As a result, limited pools of experienced programmers spend a lot of their time working on high-volume tasks and activities that, while necessary, offer little in the way of a value-add where competitive innovation is concerned.

As the world's first enterprise **no-code application** platform, Unqork was designed to take on much of the heavy lifting that goes into program development and upkeep.

Specifically created to streamline development of sophisticated applications in **complex, highly regulated industries** (like finance, insurance, healthcare, and government, for example), when compared with code-based solutions, Unqork ensures coders can deliver:



Because no-code technologies eliminate the need to write in or manage a codebase altogether, programmers and engineers can feel less confined in their abilities to address business growth challenges.

Unqork also allows less-experienced Creators and business users to contribute to the software development process so they, too, can spend less time on routine tasks.

Benefits of transitioning from coding to configuring

Chances are you learned to write code to create dynamic new solutions—not perform repetitive, monotonous tasks. The time you'll save ramping up productivity with a no-code platform like Unqork can open the door to more impactful and meaningful work.

Accelerate feedback and iteration cycles

The ability to create rapid, high-fidelity prototypes and **MVPs** (minimum viable products) is essential to ensure the application you're developing aligns with business requirements and meets user needs.

Discovering after the fact that a workflow isn't properly designed—or that you've missed an opportunity to optimize a certain function—will most certainly slow you down. But building out prototypes and trial versions with code makes for an incredibly complicated and time-consuming process.

Because you can build and deploy so quickly using Unqork's drag-and-drop components, you can get the rapid feedback you need to satisfy internal stakeholders—or to market test, adjust, and re-test your software—considerably faster.

Engineer inventive solutions to complex business problems

No-code is a great avenue for getting back to the work that inspires you. Instead of coding to carry out recurring, colorless tasks, adopting a no-code platform lets you dedicate more time to developing creative solutions to intricate problems.

By focusing less, for example, on programming languages—or the likelihood that one code line change here will break an entire element elsewhere—you'll increase your capacity to innovate and tune into business logic. Not only can this help differentiate your business, it can make a significant impact on your day-to-day work experience.

Because Unqork is built to be accessible to anyone, it also makes it easy to delegate portions of the development lifecycle to other Creators. By redistributing your workload, you'll be less likely to experience **burnout** and better able to liberate the time needed to work on more meaningful projects.

Final Thoughts

As co-founder and CEO of Twilio, and a former product manager at Amazon, Jeff Lawson has stated that, "In the digital economy, whoever builds the best software wins."

Unfortunately, code-dependent enterprises often underutilize their software developers' talents[6] by relying on them to automate simple processes or keep legacy applications alive.

According to Lawson, companies that want to compete need to:

- Empower their developers
- Assign problems, rather than tasks
- Run more experiments, tolerate failures, and prioritize speed

By giving their existing knowledge and expertise a serious boost, Unqork's no-code enterprise platform allows developers to accelerate time-to-market by building and deploying applications 3x faster than code-based approaches—all without sacrificing quality.

Not only can this ultimately impact a company's bottom line, delivering outcomes sooner—and for one-third the cost—isn't likely to go unnoticed.

Between bolstering cross-functional teams, and reaping recognition from leadership, both developers and the companies they work for can benefit from the boon that no-code represents in terms of ramping up—rather than replacing—coding talent.

Want to explore the value and power Unqork's no-code platform can unleash compared to older delivery techniques? Get in touch and let's set up a demo.

[6]With Twilio research indicating companies will need to accelerate digital transition plans by an average of six years due to COVID-19, CEO Jeff Lawson has authored a book explaining how underutilized software developers should be increasingly integrated with strategy-based decision making.