2023

Future Of Work Report

Data on the Impact of Remote Work on Software Development

Software

Table of Contents

Introduction

Three years into the pandemic, the future of work is still uncertain. Companies continue to set shifting hybrid work mandates, often trying to pull more employees back into the office.

To understand how remote work has impacted the global software development workforce, we analyzed and collected data from more than 400K software developers over the past three years — one of the most comprehensive datasets on developer productivity compiled to date.

Our key findings: the developer experience has improved during the pandemic, while productivity remains relatively unchanged. Developers are spending less time coding nights and weekends, reclaiming time from interruptions, and making better use of commute times. They are feeling less burnt out. While they are spending slightly less time coding, they are getting more efficient with their time, automating away repetitive tasks with AI and automation tools. These are a few of the indicators we've seen that the shift to remote work has been, overall, a net positive for the global development community.

Why does this matter? While our findings indicate that most developers are happier and just as productive working remotely as at an office, many leaders still distrust remote work — presenting a productivity paradox moving forward.

Leaders that mandate a return to the office risk negatively impacting the developer experience and losing top talent, key drivers of long-term productivity.

With the rise of collaboration and automation tools, like Slack and Zoom, developers can be productive working from anywhere. Development observability tools further enable remote work by solving the productivity paradox. By tracking productivity metrics like releases and release quality, companies can use data to fill in the gap between the developer experience and leadership's visibility, restoring trust within engineering teams.

Our world runs on software. We believe in the power of software to effect positive change in the world. Enabling innovation requires leaders to help development teams perform at their best.

We founded Software.com to equip software developers and engineering leaders with the data, insights, and tools to continuously improve their craft. We hope this report will help more engineering leaders use data to empower teams to work where they are most productive.

— Brett Stevens and Geoff Stevens
Founding Team at Software.com

Key Findings

We set out to understand, three years on, how the development community has adjusted to remote work, the impact on productivity, and what the future holds for engineering teams in a hybrid world. Analyzing the impact across our community of more than 400K developers from around the world, we uncovered **three key insights into how software development has changed over the last three years**:

1

Developer Experience Is Improving

With more flexible schedules, more work is getting done between 9am and 5pm (+5%). Developers are coding 9% more during morning commutes and less work is being pushed to late nights (-11%) and weekends (-9%). As a result of a better work-life balance, developers are feeling less burnt out.

2

Productivity Is Unchanged

While developers are spending marginally less time coding per weekday — 59.9 minutes in 2023 compared to 64.2 minutes in 2020 (-7%) — this change has been offset by a small improvement in efficiency. Keystrokes per minute coded, a proxy for focused work, increased by 4%. At the same time, an increasing number of repetitive tasks are being automated away with code completion and AI tools.

3

Effects of Automation and AI Are Increasing

With the rapid adoption of AI and automation tools, like GitHub Copilot, developers are writing and editing code at a faster rate than ever before. From 2020 to 2023, the average number of characters inserted per keystroke increased by 41% and lines of code edited per minute increased by 39% — indicators of the increased usage of generative AI and code completion tools. The long-term impact of these tools on productivity and code quality remains unknown.

Methodology

The Software.com platform provides visibility into global coding trends from our community of 400K+ developers. Data is analyzed from our Code Time plugin[1] for popular code editors, such as Visual Studio Code and IntelliJ, from September 2020 to April 2023. All data was aggregated and anonymized, based on our company's principles regarding developer privacy, productivity, and happiness.[2]

In our 2022 Global Code Time Report,[3] we found that most developers spend less than an hour per day coding. In this report, we've refined our cohort to developers who code at least two days in any given week, giving us a better understanding of the impact on full-time developers.

To complement our research, we surveyed a representative sample of developers from our community in April 2021, October 2021, and February 2023, using a similar set of questions. These users self-identified as a cross-section of full-time developers working across 18+ industries around the world.

Throughout this report, we have also referenced other published research with the goal of helping teams parse through an increasingly complex body of research on remote, hybrid, and office work.

1

KEY FINDING #1

Developer Experience Is Improving

With fewer disruptions and more flexible schedules, developers have been more productive during work hours (between 9am and 5pm) while working remotely. Overall, the percentage of time spent coding during work hours increased from 56.7% in 2020 to 59.5% in 2023 — a 5% increase.

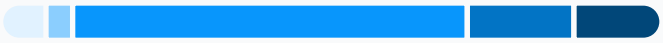
Unlike the early days of the pandemic, when concerns about blurring lines between work and home dominated the shift to remote work, developers are now experiencing a significantly better work-life balance. Our research indicates that developers are spending a smaller fraction of their time — as a percentage of their overall time spent coding — coding late nights and weekends.

The percentage of time coded late nights each week declined by 11% since 2020, while the percentage of time coded on weekends each week decreased by 9%. Developers coding less as a hobby, or less on open source projects, could also partially explain this trend.

Our analysis also reveals that the length of a developer's day (measured as the duration between first and last coding activity) decreased by about 6%, from 6.6 hours in 2020 to 6.2 hours in 2023. Shorter days, however, do not necessarily mean developers are less productive; rather, this trend could indicate that teams are working less time overall but releasing more, higher quality changes. Metrics from development observability platforms, like release frequency and quality, can provide a true lens into productivity.

Work-Life Balance Is Improving

Percentage of minutes coded per day, by time of day



Time of day	2023	vs. 2020
Early morning	7.1%	▼5%
AM commute	3.9%	▲9%
9am to 5pm	59.5%	▲5%
PM commute	16.1%	▼6%
Late night	13.4%	▼11%

Figure 1

> *The lack of commute is by far the most notable impact of remote work. Adding back 1-2 hours in someone's day is non-trivial, but there's also the small efficiency improvements of not having to change venues for meetings, having access to a home kitchen for lunch, and being able to better balance home tasks throughout the day.*

— Laura Tacho
Engineering Leadership Coach

Code Time Is Increasing During Morning Commutes, Work Hours

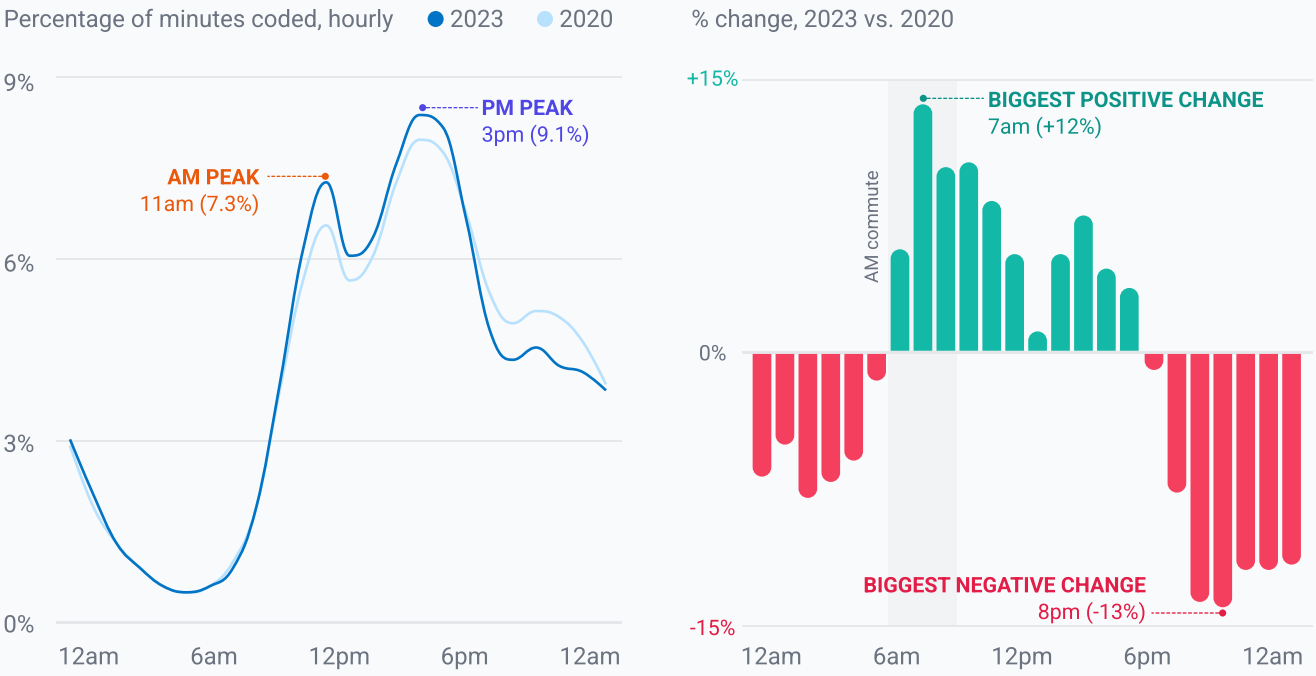


Figure 2

Benefits of Flexible Schedules

According to a survey from Terminal, a platform for remote development teams, the top cited remote work benefit for developers was flexible work hours (47% of survey respondents).[4] Another upside of a flexible work schedule is less time spent commuting; according to the National Bureau of Economic Research,[5] workers on average saved 72 minutes each day while working from home due to a lack of a commute in the first two years of the pandemic. Developers are similarly taking advantage of time saved from morning commutes, coding 9% more between 6am and 9am.

Developers are also reclaiming their Mondays for code time: Mondays saw the largest increase in code time (+1.6%) since 2020, while Fridays saw the largest drop (-2.1%). Wednesday — the long-time favorite for company-wide no meeting days — remains the most popular workday for coding.

With more control over their schedules and better work-life balance, many developers will be reluctant to forfeit these benefits in a shift back to an office-first workplace. According to Netlify, a platform for deploying websites and apps, more than half of developers said they would quit their jobs rather than go back to an office.[6] Companies that require developers to return to the office run the risk of higher attrition and turnover, as top talent leaves for more remote-friendly workplaces.

Developers Are Feeling Less Burnt Out

Since the height of the pandemic in 2021, the percentage of developers reporting their current level of burnout as medium to high decreased by 8% in 2023. Overall, the average level of burnout reported was 4.8 (out of 10) in 2021 compared to 4.5 in 2023 (-6%).

Our research shows that developers are making the most of remote work, with a healthier work-life balance, fewer distractions, and more efficient collaboration tools. The percentage of respondents who felt that poor work-life balance contributed to their burnout saw the biggest decrease (-12%), a sign that teams are embracing the flexibility and autonomy of remote work. Likewise, the number of respondents that said disruptions were a contributor to burnout decreased by 7%, highlighting another one of the primary differences between remote and office work.

As companies weigh the decision to bring employees back to the office, they should aim to fully understand how these factors affect developer happiness and well-being.

> *My boss wants to spend time with us as often as when we worked at the office – that is why we waste almost three hours each day talking, which disrupts our flow. Half of this time is wasted talking on non-work-related things. I would prefer to just code during this time and deliver at my preferred pace.*
>
> — Survey Respondent

Poor Work-Life Balance Is Contributing Less to Burnout in 2023

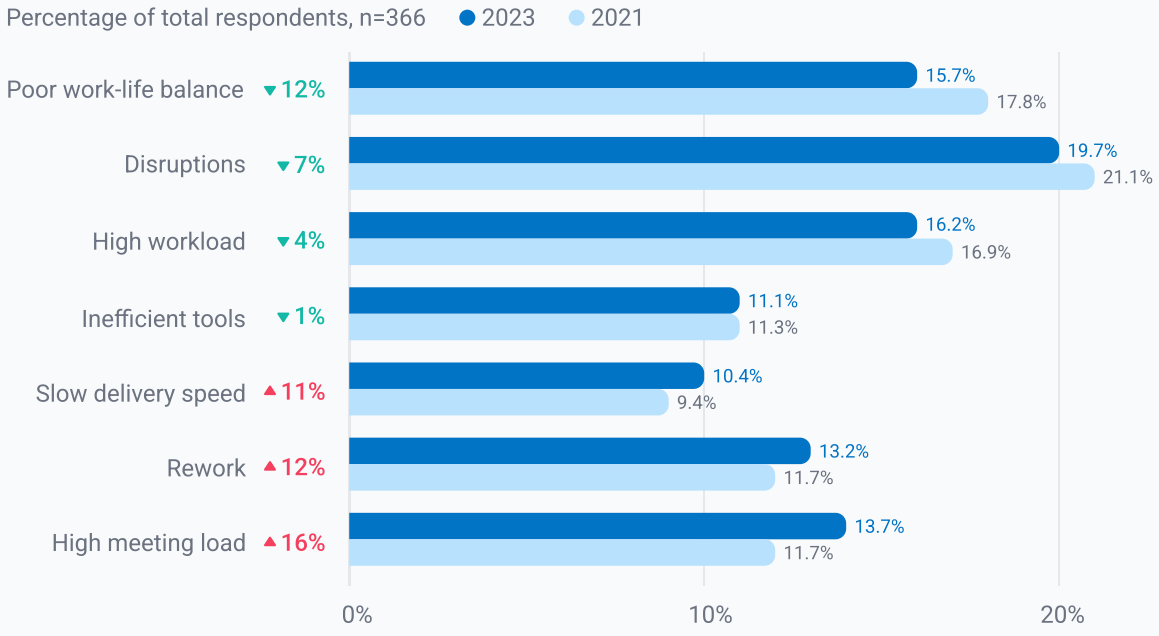


Figure 3

KEY FINDING #2

Productivity Is Unchanged

While developers are coding marginally less per weekday — 59.9 minutes in 2023 compared to 64.2 minutes in 2020 (-7%) — this change has been offset by a small improvement in efficiency. Keystrokes per minute coded, a proxy for focused work, increased by 4%.

Data in the next section of this report shows that developers are automating away more and more repetitive tasks, allowing them to get more done in less time, which may even indicate that productivity is slightly rising.

Metrics like time coded and keystrokes are imperfect and incomplete proxies for productivity; they do not capture effectiveness or code quality, and should never be used by managers to measure individuals. However, on a global scale, these metrics provide evidence that there has not been a large, quantifiable impact supporting either side of the productivity paradox — that remote work has both improved and impaired productivity, depending on whether you ask employees or their bosses.[7]

> *My productivity is the highest when I get to spend some time away from the desk, where I can't focus. On a normal day, it'd be 3-4 hours of deep focus, then away for a couple of hours, and then deep work for a couple more.*
>
> — Survey Respondent

Time Spent Coding vs. Efficiency

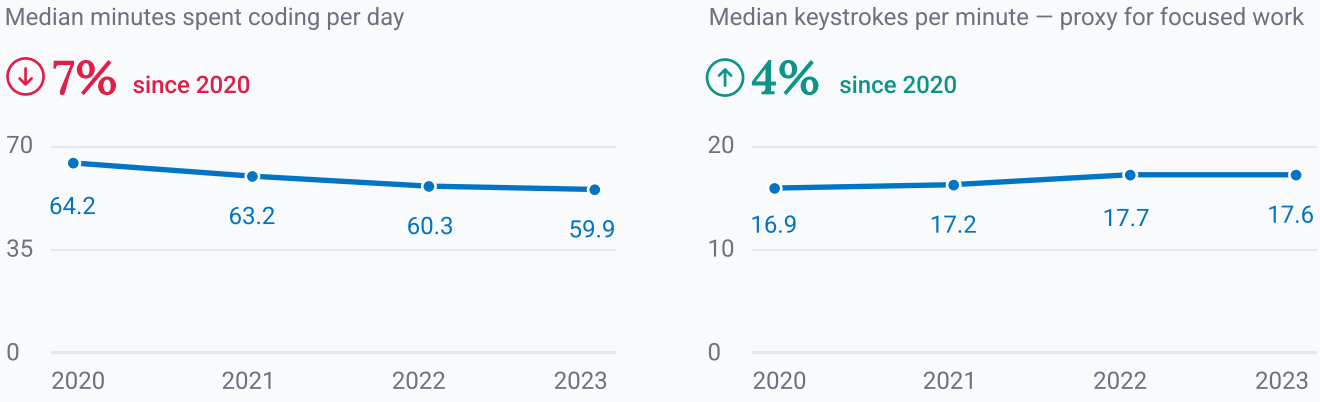


Figure 4

KEY FINDING #3

Effects of Automation and AI Are Increasing

With the rapid adoption of AI and automation tools, like GitHub Copilot, developers are writing and editing code at a faster rate than ever before. Using modern development tools, developers can write multiple characters at once to their code editors with each keystroke, such as when pasting a code block or accepting a code suggestion. From 2020 to 2023, the average number of characters inserted per keystroke increased by 41% and lines of code edited per minute increased by 39% — indicators of the increased usage of generative AI and code completion tools.

So what's driving this megatrend? A growing suite of developer tools in code editors and IDEs, such as linters, generators, and formatters, are helping teams write and rewrite their code more efficiently. Visual Studio Code, the most popular code editor, offers over 45K extensions through its marketplace. With the increasing breadth of coding knowledge on Stack Overflow, in open source repositories on GitHub, and elsewhere, developers are also likely re-using more code from other sources.

Another likely cause is that they're making use of new AI-powered tools — like CodiumAI, Mintlify, and Mutable.ai, among others[8] — that are helping them not just write new code, but also create documentation, add tests, and refactor existing code. GitHub Copilot, for instance, has seen a meteoric rise in popularity — over 5 million downloads since June 2021.

Automation, AI Are Accelerating Development

Characters inserted per keystroke — indicator of usage of AI and code completion tools

↑ 41% since 2020

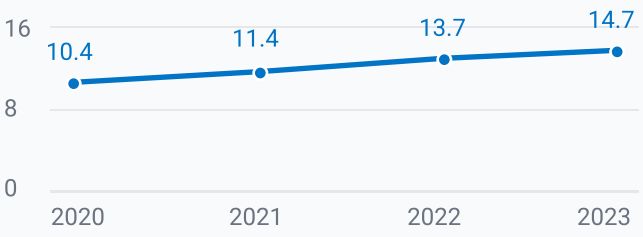


Figure 5

Meteoric Rise of GitHub Copilot

GitHub Copilot installs on the VS Code marketplace, in millions

5M+ installs since June 2021

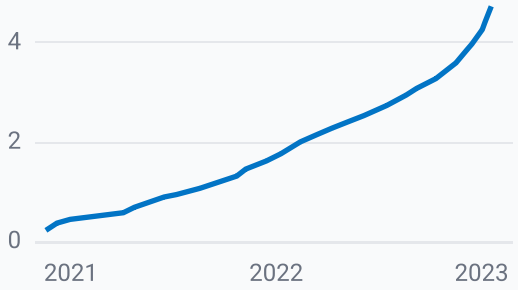


Figure 6

Developers Are Writing Code Faster Using GitHub Copilot

Rate of editing code, GitHub Copilot users vs. baseline

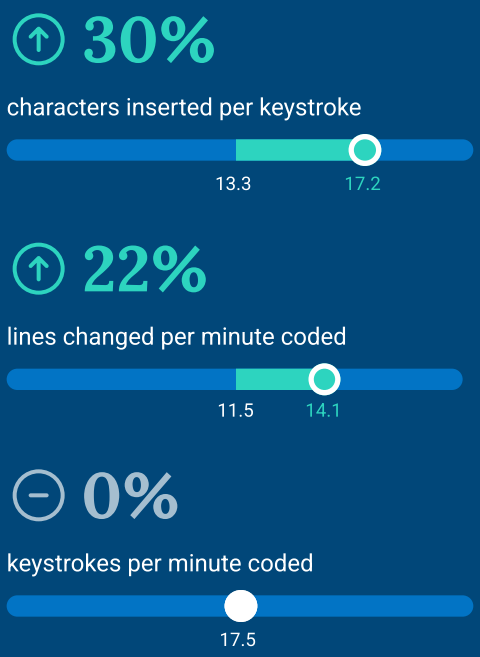


Figure 7

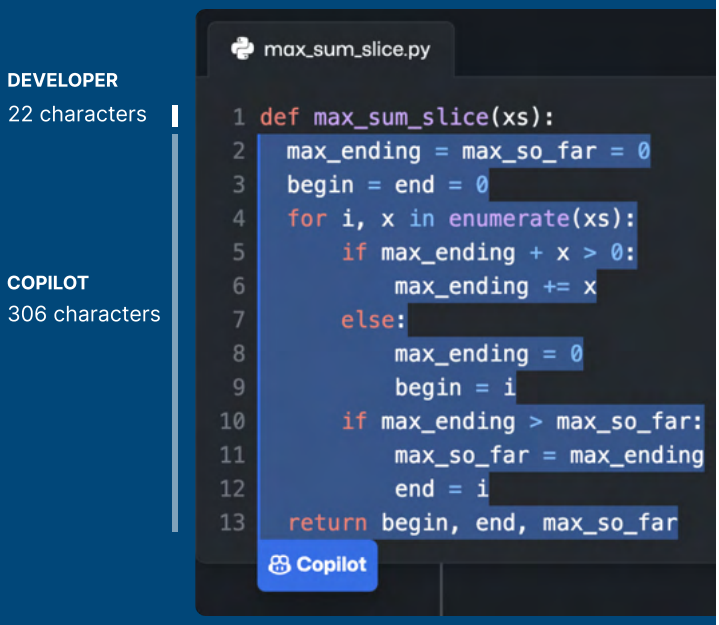


Figure 8: example usage of GitHub Copilot

In our analysis, we found that developers using GitHub Copilot are inserting 1.3x more characters per keystroke and 1.22x more lines of code in the same amount of time as developers not using the AI coding assistant. While changing more lines of code does not necessarily equate to improving productivity, the increasing rate at which developers using GitHub Copilot can write code — whether it's unit tests, functions, or other boilerplate code — provides evidence that they are saving time and effort on repetitive tasks.

Concerns still remain about the long-term impact of AI tools on code quality. As of December 2022, 35% of code suggestions by GitHub Copilot were accepted (up from 27% in June 2022).[9] Generative AI also raises unresolved issues regarding licensing and plagiarism. GitHub Copilot, for instance, faces legal scrutiny over its use of open source code in its training data.[10] Despite the ongoing discourse about the role of AI in software development, our research suggests a growing and measurable trend that will be increasingly irreversible as millions more developers integrate AI-powered tools into their daily work.

> *I use Copilot for everything at this point. It saves me time from having to search through documentation. Basically, I go from writing code to editing code.*
>
> — Survey Respondent

Looking to the future

We asked our community how many days per workweek they typically spend working remotely. In contrast to an accelerating return to on-site work for the general workforce,[11] the number of developers working fully remote increased by 8% since 2021. In 2023, 60% developers are working fully remote, and 85% are working either fully remote or in a hybrid setup.

The results validate other sources that indicate a dramatic increase in remote work for developers. An *Economist*[12] analysis of HackerNews job postings revealed that 77% mentioned remote work in 2022. By contrast, in 2011, only 13% of postings mentioned remote work.

85% of Developers Work In Hybrid of Fully Remote Environment

Work location by percentage of total respondents, n=366

Work location	2023	vs. 2021
Remote	60.1%	▲8%
Hybrid	25.0%	▼11%
Office	14.9%	▼8%

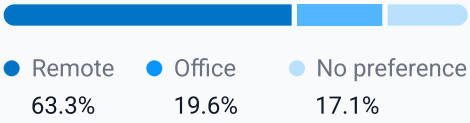
Figure 9

Productivity Paradox

Not only do most software engineers work remotely, but most also believe they are more productive working remotely, a view that has hardened over time. According to our research, 64% of developers said they were more productive working remotely, compared with 55% in 2021. “Remote work is the future,” said one of our survey respondents. “I will be very happy to work remotely rather than go back to a typical desk job.”

Developers Feel Most Productive Working Remotely

Work location preference by percentage of respondents, 2023, n=366



Remote	Office	No preference
63.3%	19.6%	17.1%

Figure 10

While developers are embracing remote work, their managers are more skeptical about the impact on productivity. A study conducted by Microsoft[13] found that only 12% of leaders said they had full confidence that their teams were productive. “My company’s leadership is unhappy with any hybrid work and wants everyone back in the office,” said another survey respondent.

Understanding the Tradeoffs

For companies deciding whether to continue remote work or bring developers back on-site, it is important to understand the tradeoffs — and which matter most for their organization.

Offices can be essential hubs for collaboration, especially early in a company's life cycle when creativity and rapid iteration are most important. Synchronous work, like whiteboarding new ideas or training junior developers, is often most effective when teams are working together in-person. Our research also suggests that developers may spend more time coding when working from an office, although that does not necessarily mean they are more productive.

But are offices worth the cost? Elon Musk, an outspoken critic of remote work, recently shut down Twitter's Seattle and Singapore offices, recognizing the profitability benefits.[14] A study by Density, a workplace insights platform, found that offices at Fortune 500 companies are sitting empty for nearly 36% of working hours — a significant source of waste as companies look to weather a possible economic downturn.[15] Leaders should always weigh the productivity vs. cost benefit of maintaining office space.

Companies that mandate a return to the office also risk negatively impacting the developer experience and losing top talent, which is becoming increasingly important. In a world where most developers prefer remote work, creating a flexible workplace can help companies retain their best developers. It can also help them attract new hires from a wider talent pool with more diverse backgrounds, without the expenses or constraints of offices.

Observability for Development Performance

This report summarizes trends across our global development community. With insights from developers around the world, we can begin to understand how remote and hybrid work have impacted software developers and their workplace as a whole.

Global trends, however, are not representative of every company. Even within engineering organizations, there can be radical differences between teams, projects, and locations. Most companies have only scratched the surface of understanding the long-term impact of remote work.

Weighing the Tradeoffs of Remote Work

A framework for weighing decisions for the future of work

Figure 11

Summary of Key Findings

1	Developer Experience Is Improving	▲5%	Coding during work hours (9am-5pm)
		▼6%	Burnout level
2	Productivity Is Unchanged	▼7%	Minutes spent coding per day
		▲4%	Keystrokes per minute (focus)
3	Effects of Automation and AI Are Increasing	▲41%	Characters inserted per keystroke (auto-complete, paste)
		▲39%	Lines changed per minute

Figure 12

Before driving employees back to the office, companies should first set up a development observability tool to help them understand the impacts of remote work — as well as which tools, processes, and workflows make teams the most productive. Observability tools can answer productivity questions for leadership while maintaining the developer experience benefits of remote work, and more importantly, build trust within engineering cultures.

Here are three ways that adopting a development observability tool can support leaders deciding about the future of work at their companies:

1. **Measure the impact of remote work over the last three years.** Developers as a whole have figured out how to stay productive in the hybrid era. With an observability tool, teams can address concerns from leaders that remote work has negatively impacted productivity.
2. **Experiment with different work models.** Instead of applying a one-size-fits-all model, leaders should enable teams to see where they are most productive — while simultaneously weighing the qualitative impacts of work environments on the developer experience, including work-life balance, satisfaction, and retention.
3. **Understand the impact of new tools on productivity.** We've entered a new era for developer productivity with many unknowns. The accelerating adoption of AI-enabled developer tools means teams should start tracking *now* to measure the impact on developer productivity in 3, 6, or even 12 months. With better data, leaders can gain confidence faster to make broader changes across the organization.

Every team stands to gain by improving visibility into their own data. Do *you* have the data to prove that your team or company has become more productive during the pandemic?

Notes

1. Code Time, VS Code Marketplace
2. We believe in developer privacy, productivity, and happiness above all, Software.com Manifesto
3. Global Code Time Report, Software.com
4. The State of Remote Engineering, 2023 Edition, Terminal
5. Time Savings When Working from Home, National Bureau of Economic Research
6. Findings from the Jamstack Community Survey 2022, Netlify
7. Work Shift: The New Productivity Paradox, Bloomberg
8. Developer Tools 2.0, Sequoia Capital
9. GitHub Copilot now has a better AI model and new capabilities, GitHub
10. OpenAI, GitHub AI Tools Draw Legal Scrutiny Over Fair Use, Bloomberg Law
11. Telework, Hiring, And Vacancies - 2022 Data From The Business Response Survey, Bureau of Labor Statistics
12. For programmers, remote working is becoming the norm, The Economist
13. Hybrid Work Is Just Work. Are We Doing It Wrong?, Microsoft
14. Elon Musk Is Now A Fan Of Remote Work, Forbes
15. Fortune 500 companies are wasting millions on unused office space, Fast Company

Software

We make software delivery more visible by collecting data across the stack, providing insights that help developers and teams measure and improve.

www.software.com @software_hq