GitHub Copilot

Realizing the potential of GitHub Copilot

WRITTEN BY GITHUB WITH ♥

Table of contents

Maximizing the benefits of GitHub Copilot

GitHub Copilot is designed to help developers and organizations drive faster time to value, operational efficiency, and promote talent retention.

Through a variety of usage patterns, GitHub Copilot provides multiple types of time-saving assistance:

Pair programmer	Expert system	Code translator	Faster typing
Acts as a collaborative partner, offering critiques, alternatives, and answers queries.	Delivers advanced code suggestions and more comprehensive guidance.	Aids in deciphering intricate code segments and assists with unfamiliar programming languages.	For seasoned developers, it swiftly transforms their conceptualizations into structured and styled code.

A single suggestion from GitHub Copilot can combine multiple types of assistance to further increase time savings by replacing multiple slower steps.

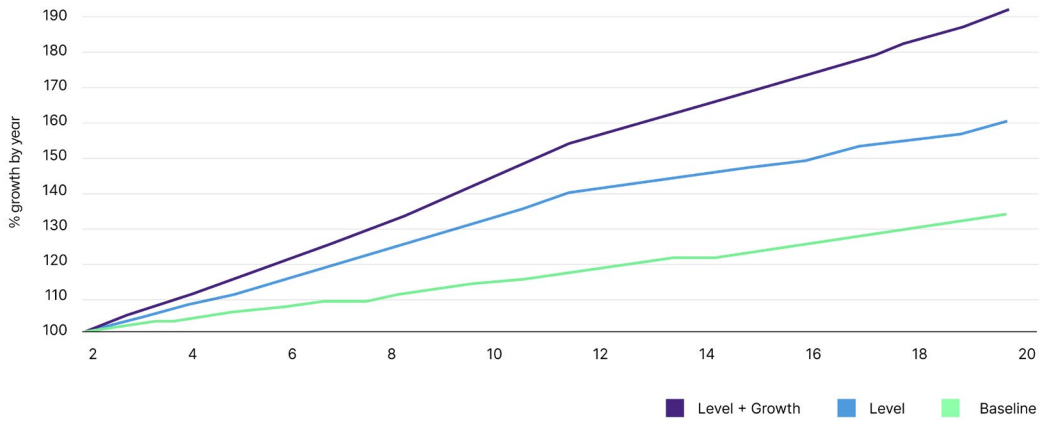
Time savings is the most tangible and directly measurable benefit of GitHub Copilot, but customers routinely observe that learning, autonomy, and even innersource collaboration are improved as well.

Justifying the subscription cost of GitHub Copilot is straightforward. By saving one hour per week, the value from time savings conservatively outpaces the costs by two-to-one or more (given that the GitHub Copilot Enterprise subscription price is $39 per developer per month). This already high rate of return rapidly improves with levels of experience (i.e. higher compensated developers) and with more time spent coding.

According to a report from the Brookings Institution, Machines of mind: How generative AI will power the coming productivity boom, generative AI might increase the productivity rate by 18% over the next 10 years on top of existing expectations for productivity growth.

The report anticipates both direct benefits of higher worker productivity as well as an increasing rate of innovation that will further drive productivity growth in the economy:

> *"Large language models such as ChatGPT are emerging as powerful tools that not only make workers more productive but also increase the rate of innovation, laying the foundation for a significant acceleration in economic growth."*



Simulated Productivity Growth Trajectories reflecting AI-boosted productivity (Level) and AI-boosted productivity plus accelerated innovation (Level +Growth) both compared to the pre-existing (Baseline) forecast. (Source: Brookings Institution)

However, like any tool, GitHub Copilot's ultimate potential can only be unlocked with proper adoption, training, and continued empowerment. This is captured in a quote from a McKinsey research report titled Unleashing developer productivity with generative AI:

> "With the right upskilling and enterprise enablers, these speed gains can be translated into an increase in productivity that outperforms past advances in engineering productivity, driven by both new tooling and processes."
>
> - McKinsey June 2023

Organizations that embrace and integrate such advancements will not only see immediate benefits but will also position themselves at the forefront of the next wave of technological evolution.

This paper reviews the high-level rationale and detailed research that can help support evaluation, justification, and the collective decision to move forward.

Understanding and measuring improvement

Measuring and improving software processes is full of pitfalls.

One thing is clear, however: quickly reducing gaps in information and accumulating less uncertainty are proven ways to make the entire process of creating software more efficient and predictable.

It is impossible to directly see or measure gaps in information or the level of uncertainty in outcomes that can arise in software processes. Therefore, the industry has developed practices that, if integrated, can help keep the information gaps and uncertainty levels as low as possible.

The following table describes four wise practices or warnings that remind development teams to keep uncertainty low:

Classic insights on reducing uncertainty

Brooks' law	Little's law	Fail fast principle	Test-driven development
"Adding people to late software project makes it later." Adding people adds uncertainty and knowledge/info gaps. Summarized from *The Mythical Man-Month* by Fred Brooks	Limit work in progress to minimize the time unfinished work spends in queues. Increasing 'work in progress' adds uncertainty. Based on kanban	Rapid iteration allows faster understanding of what does and does not work. Delays in feedback increase uncertainty and prolong knowledge/info gaps. Summarized from *The Lean Startup* by Eric Ries	TDD leads to higher test coverage and the ability to catch issues early because every piece of code is developed with quick feedback from a growing suite of tests. Written but untested code is a unit of uncertainty or gap in information. Summarized from *Extreme Programming Explained* by Kent Beck

As described in the above insights—and as what GitHub Copilot attempts to accomplish—it's best to **close the uncertainty gaps as quickly as possible.**

Reducing gaps in information and uncertainty is undeniably valuable for improving the software creation process. Developers regularly report that using GitHub Copilot saves large chunks of time while coding. The measured improvements found in studies and by customers range from 10-55%.

Research shows GitHub Copilot allows developers to complete tasks more quickly:

Research: quantifying GitHub Copilot's impact on developer productivity and happiness (GitHub, 2022)	Research involved: 95 devs total Same coding task 45 used GitHub Copilot 50 did not	Findings: 55% faster time to complete tasks	10% more likely to complete tasks
Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle (GitHub, Harvard Business School, Keystone.AI, 2023)	Research involved: Large sample of GitHub Copilot users (n =934,533)	Findings: Within the first year on the market, users accept nearly 30% of code suggestions from GitHub Copilot and report increased productivity from these acceptances.	The study also found that less experienced developers benefited more.
Unleashing developer productivity with generative AI (McKinsey, 2023)	Research involved: Study showed that software developers can complete a variety of coding tasks up to twice as fast with generative AI.	Findings: 45-50% faster for code documentation 35-45% faster for code generation	20-30% faster for refactoring <10% for complex coding tasks
Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence (MIT, 2023)	Research involved: Study focused on the use of ChatGPT to complete mid-level professional writing tasks.	Findings: Results showed that ChatGPT substantially raises average productivity (~15%) and quality, but shifts how time is allocated to phases of writing:	Effects on task structure: *Brainstorming* reduced 25% → 20% *Writing rough draft* reduced 50% → 25% *Editing* increased 25%→ 40% Effects by skill level: Increased the output quality of less skilled writers.

These studies clarify that not all tasks or developers are equal. However, in each combination tested, less time is spent completing the required task. In the cases of junior skill levels, quality also improved.

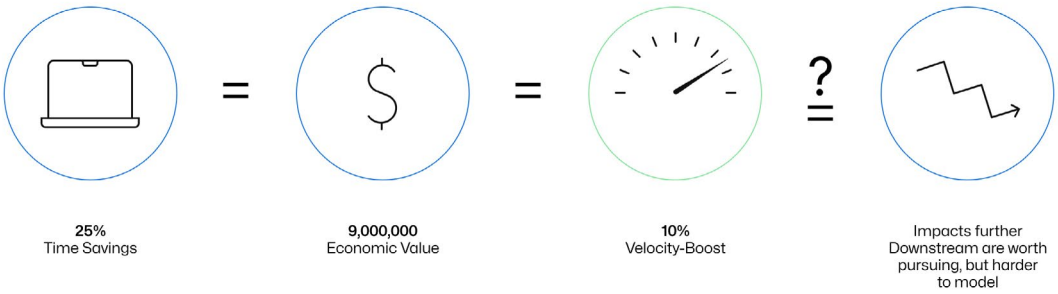
Doing the math... what is realistic?

Time savings is the primary benefit and measurable impact of GitHub Copilot in the near term, and especially for evaluations.

As developers within organizations get more experience and skill with GitHub Copilot, the time saved can be focused on improving downstream objectives like reducing technical debt, improving automation, and team health.

The GitHub Copilot value calculator is a tool that uses the above research as a starting point and provides an estimate of the economic value derived from using GitHub Copilot. For instance, for a group of 1,000 developers with an average annual salary of $100,000, and a 25% time savings while coding translates to an economic value of about **$9 million per year.**

Modeling shows that Value is easier to measure as Time savings or Economic Benefit rather than Velocity:



The graphic shows that 25% time savings is equivalent to $9 million and also equivalent to 10% increase in velocity (assuming all time savings is converted to velocity). Impacts further downstream, like improvements in test automation and defect levels are best pursued once time savings is achieved and measured.

It's important to mention that developers have the choice to spend the time savings on velocity or on other valuable activities that do not affect velocity. This optionality tends to make velocity (e.g. the rate of pull requests, releases, or deployments) an unreliable evaluation metric for GitHub Copilot.

Example experiences

Different organizations have reported varying levels of time savings from GitHub Copilot.

The average "coding task time savings" customers measure varies from 10% to 25%, as reported by Duolingo, to over 50%, as reported by Mercado Libre. Meanwhile, others have reported a 50% efficiency boost specifically for tasks like terraform scripting, as observed by Coyote Logistics.

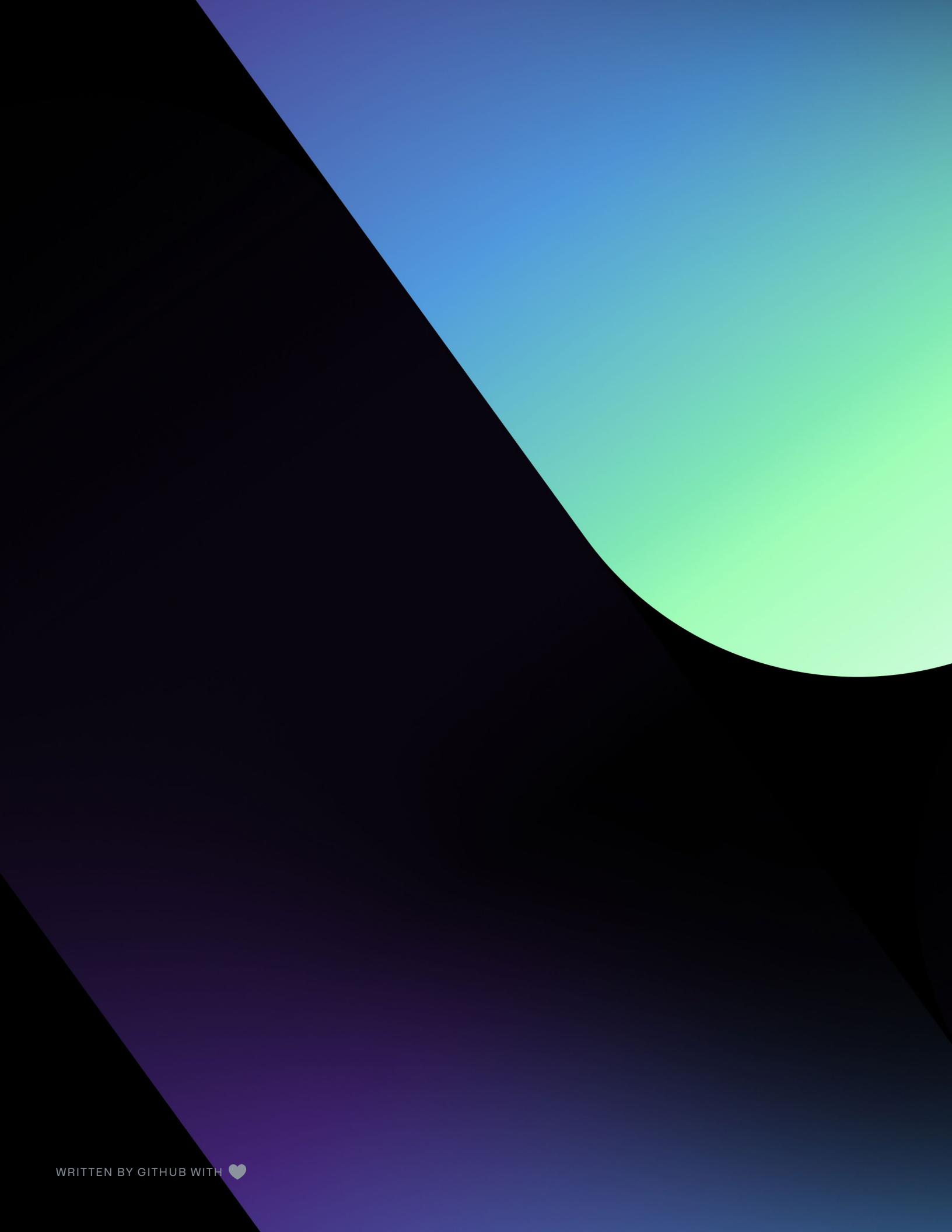
> *"Barrios says he spends half as much time to arrive at the correct code as he did before GitHub Copilot. The first time he used GitHub Copilot, it wrote an entire script for him based on a single comment."*
>
> - Mercado Libre

Most organizations are evaluating or planning to evaluate generative AI tools soon. (GitHub Copilot has over one million paid subscribers in over 37,000 organizations, making it the most widely-adopted AI developer tool.)

Many GitHub Copilot evaluations are comparable to the following summary from commercetools, the global leader in composable commerce, which evaluated the GitHub Copilot for three months and shared their experience on their website.

> *"Once enabled, GitHub Copilot was used continuously, and the usage even increased. Suggestions were often accepted and of good quality. Users were able to embed it easily into their existing work environments and got huge productivity gains out of it — all of which means for us that we will continue to roll it out wider across our organization in the coming weeks."*
>
> - Tobias Deekens, Principal Engineer of Frontend Architecture at commercetools

In today's competitive business climate, with a high demand for skilled developers, GitHub Copilot is becoming the productivity edge that developers recognize and request by name. It promises to be an essential tool for developers, increasing their discretionary time and safeguarding their well-being, which can make room for strategic areas of focus like continuous learning and innovation that every business seeks.


WRITTEN BY GITHUB WITH