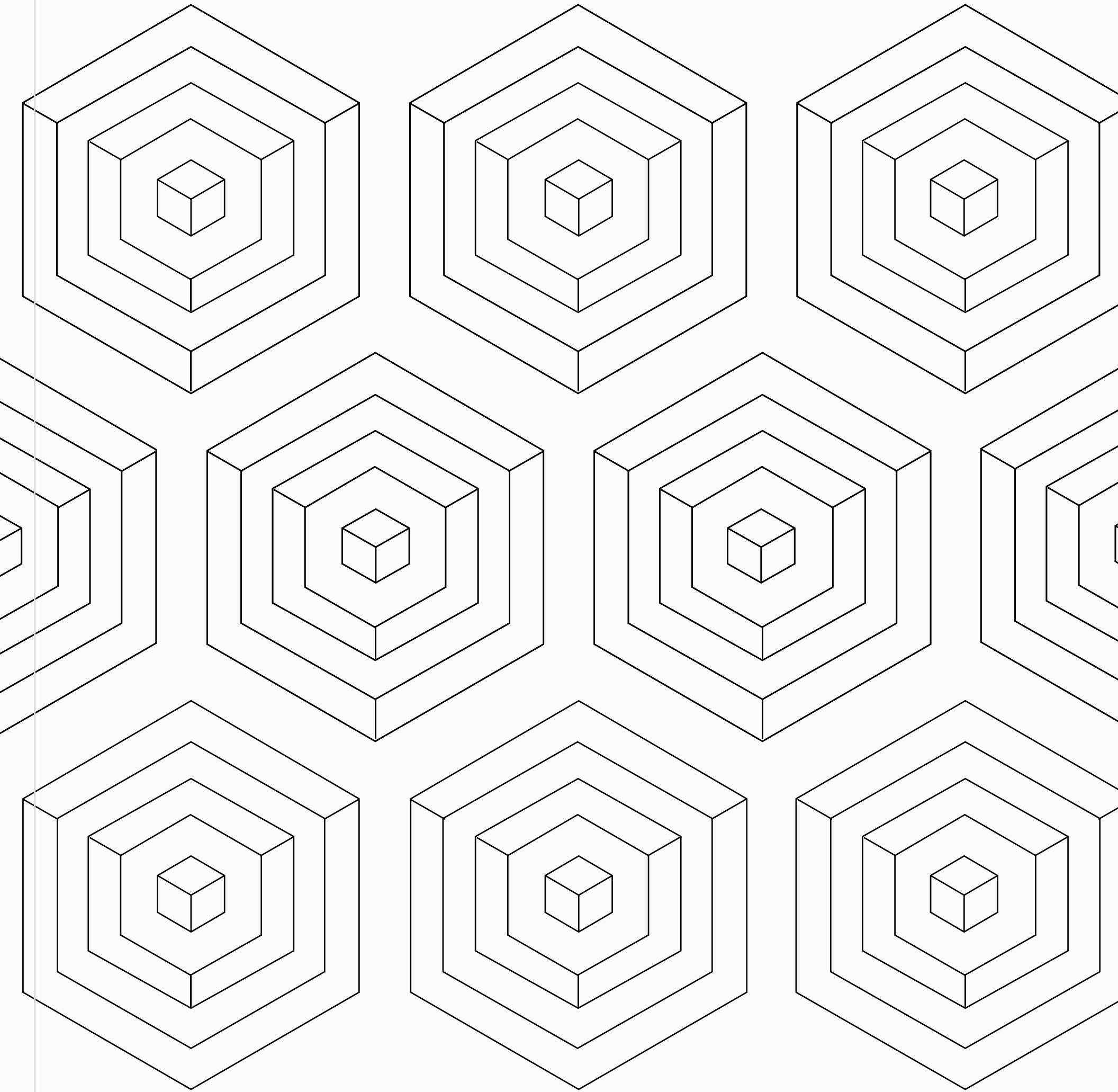


# How to measure GenAI adoption and impact



GenAI is all the hype right now. From boardrooms to newsrooms, the narrative centers around the revolutionary impact of GenAI tools such as GitHub Copilot on developer productivity and efficiency. Recent studies underscore this sentiment, revealing compelling insights:

- Developers using Copilot complete tasks 55% faster
- Copilot reduces review time by an average of 19.3 hours
- AI-assisted PRs have a 1.57 times higher likelihood of being merged

With these kinds of reports being published by the likes of Microsoft and Google, it's no surprise that C-suites at nearly every organization we talk to are opening up their company's coffers to purchase and roll out GenAI tools as fast as possible to their software developers. There's never been such a rapid rush around something like this, at least in recent history.

Yet, despite the widespread enthusiasm, there are a lot of challenges as well. Chief among them is determining the tangible impact of GenAI on developer productivity. Leaders need this information to validate and inform their investments. However, developer productivity has always been a complex problem, and measuring the impact of GenAI is no different.

Other organizations are seeing suboptimal developer adoption that is difficult to explain, and are looking to better understand why this is happening and how to address it. This problem is especially painful for leaders responsible for rolling out these tools. These people are doing so without feedback loops or insights into what the best use cases are and where developers are seeing the biggest gains.

Across these challenges, we see that there's a common thread: it's difficult to get useful feedback, signals, and measurements on how GenAI is impacting developer productivity. At DX, we've been working with a number of organizations to solve these challenges, and are seeing promising results. In this article, we will share our learnings on the different approaches organizations are using, and provide guidance into how to combine the methods available into a holistic approach that gives organizations adopting GenAI tools the insights they need.

• • • •

## How organizations are collecting feedback and data today

In the rush to understand and measure the impact of GenAI tools, organizations are scrambling to collect and analyze developer feedback and data. But this process is often a big struggle.

Developer adoption of GenAI tools is a key concern, yet we hear from many leaders that their only sources of insights are basic access reports and a weekly screenshot of utilization metrics sent to them by their GitHub account representative (or, [this](#) just-released alpha REST API).

To measure the impact of GenAI tools, telemetry-based productivity metrics like pull request count are of heavy interest right now. But for many organizations, these metrics aren't telling a compelling story (and in many cases, not showing any changes at all), worrying and confounding leaders.

Some organizations have launched efforts to collect data through surveys, but struggle with survey design and collecting enough responses to produce reliable baselines. Experience sampling – the least familiar of the methods – holds a lot of promise, but putting it into practice can be challenging, as we'll discuss later.

At DX, we've witnessed many of the benefits and challenges of these different approaches, and find that many organizations' challenges stem from the misapplication or misunderstanding of how to properly utilize each method. Telemetry metrics, experience sampling, and surveys can all provide leaders with rich and useful data. Deploying each method successfully is the challenge.

Method	Good for	Not good for	Challenges
<b>Telemetry metrics</b>	Measuring GenAI impact on developer output	Quantifying ROI (i.e., time savings) Understanding how GenAI tools are being used	Limited insight Doesn't tell a compelling story
<b>Experience sampling</b>	Quantifying specific ROI (i.e., time savings) Identifying what developers are using GenAI tools for (best use cases and opportunities)	Collecting large amounts of data at once	Difficult to set up Must be run over a period of time
<b>Surveys</b>	Measuring GenAI adoption, developer satisfaction, and self-reported productivity	Quantifying ROI (i.e., time savings)	Can only be run periodically Participation rates

## Three approaches to measurement and insights

Understanding and measuring developer productivity has always been a difficult problem. But add GenAI into the mix and it's gotten even harder. Thankfully, this problem is solvable, but only when organizations adopt a mixed-methods approach.

In this section, we'll outline the three different methods for measuring and collecting data on GenAI utilization and impact. We'll tell you where each approach shines, how it can be utilized, and the common pitfalls we see organizations running into. Then in the next section, we'll show you how to combine the different approaches together into a holistic insights strategy.

### Telemetry metrics

The first place that many organizations look to for data on the productivity impact of GenAI is telemetry metrics from tools like GitHub. Common metrics used to track productivity include pull requests per developer, code review time, and cycle time. Some organizations counterbalance these measures with metrics like number of incidents, to ensure that GenAI-fueled speed increases don't come at the cost of quality.

Telemetry metrics are a useful way to get a high-level gauge of how developer output and activity levels are being affected by GenAI. Many organizations observe small but noticeable increases in their metrics, ranging from 5-10%. Other organizations, however, see little or no change in the numbers at all, leaving leaders concerned about the large investments they're making in GenAI.

The challenge we've seen many leaders run into with telemetry metrics is that these metrics often don't tell a clear or compelling story on their own. There's skepticism around whether GenAI tools are the direct driver of fluctuations in developer activity levels. And these metrics alone don't provide a concrete picture of how GenAI tools are being utilized to realize the benefits.

Telemetry metrics can't answer some important questions like: How much of developers' time is actually being saved thanks to GenAI tools? How are developers using these tools? What are the most beneficial use cases for GenAI tools that can be taught to the rest of the developers?

## Experience sampling

Experience sampling is not quite as familiar as the other methods we're discussing. In technical terms, experience sampling refers to a set of data collection methods for gathering systematic self-reports of behaviors or experiences as they occur in the individual's natural environment.

Applied to developer productivity and GenAI: experience sampling involves taking a continuous random sample of developers as they complete tasks, and surveying or interviewing them in real-time to understand how they're using GenAI tools and what benefits they've realized.

Question	Response options
Did you use Copilot in developing this PR?	Yes / No
How much less time did the coding take compared to your previous experience coding without Copilot?	0% up to 10% between 10% and 20% between 20% and 30% between 30% and 40% over 40%
How did you use Copilot to save you time on this PR?	Open text

Experience sampling is a powerful data collection method that can provide your organization with two key insights that are difficult to obtain otherwise.

The first insight is around concrete time savings or ROI, which is a critical question that's otherwise difficult to answer. Telemetry metrics and traditional surveys can only provide high-level numbers, whereas experience sampling can tell you exactly how many minutes or hours were saved on specific development tasks thanks to GenAI, from which organizations can extrapolate total estimated ROI in terms of time and dollars.

The second key insight gained through experience sampling is how exactly developers are using GenAI tools to positive effect. This is key for driving adoption: early adopters in your organization are likely to be self-driven and discover use cases, but for everyone else, simply dropping a tool like GitHub Copilot on them isn't going to lead to adoption or positive results. To achieve successful adoption, organizations must provide guidance around practical and beneficial use cases, as well as proactively identify gaps and opportunities for further tooling improvements.

Experience sampling comes with great reward, but is also the most challenging of the discussed methods to implement. If not using a tool that's built for this such as PlatformX, we've seen organizations build sophisticated tooling to deploy event-sampled data collection campaigns, and the required duration for these studies can be a point of friction for organizations that want complete answers immediately.

## Surveys

Surveys are a powerful tool for capturing measurements and feedback about GenAI. In particular, surveys are highly useful for measuring developer adoption, satisfaction, and self-reported productivity.

Most organizations, for example, don't have individual-level telemetry data on how regularly developers' are utilizing GenAI tools for specific types of tasks. Periodic surveys where developers are directly asked questions about their level of utilization for different types of tasks can provide fast and reliable data points.

Similarly, while telemetry metrics provide one lens into developer productivity, self-reported measures of satisfaction and productivity can tell a different side of the story on the positive benefits of GenAI in your organization (for example: we've seen GenAI have direct measurable benefits on developer fulfillment and ease of completing development tasks).

The difficulties we see organizations face when it comes to surveying aren't a surprise. Designing proper surveys is always a challenge, especially under the time pressures many leaders face to collect data. The periodic nature of surveys means that timing is important, and high-enough participation rates are required in order for reliable insights to be drawn from responses.

## Putting it all together

We've outlined three methods of data collection and insights, and discussed how each can provide unique insights. Telemetry metrics are primarily useful for quantifying the impact of GenAI on developer output. Experience sampling is most useful for quantifying the ROI of these tools and their specific use cases. Surveys are best for measuring adoption and satisfaction with these tools, and developers' self-reported productivity as a result of using them.

Organizations can get the most success by applying all three methods together to get the fullest insights into how GenAI is being used by developers, and how it's impacting productivity.

Method	Description	What to capture
<b>Surveys</b> <i>Every 6-12 weeks</i>	Fastest way to establish comprehensive baselines	Perceived productivity Developer adoption and satisfaction Job areas where GenAI is providing greatest benefit
<b>Experience sampling</b> <i>Targeted, 4-week studies</i>	Close up insight into impact and usage	ROI (i.e., time saved) Specific examples of how GenAI was used
<b>Telemetry metrics</b> <i>Continuously as needed</i>	High-level view of how GenAI may be increasing throughput	Number of PRs per engineer Incident rate (to counterbalance velocity) GenAI code suggestion acceptance rate

getdx.com

Organizations should deploy surveys as soon as possible to establish baselines early, before GenAI tools have been fully rolled out. Running these surveys regularly, about every six to twelve weeks, helps track changes in developer adoption and satisfaction.

At the same time, organizations should keep an eye on their telemetry metrics to spot any changes or trends in developer productivity levels as GenAI tools are adopted. It's important to dedicate effort to properly cleaning and normalizing data to ensure that you're getting reliable signals.

Lastly, we strongly recommend that organizations run experience sampling studies in focused, four-week intervals. These studies can yield powerful data on the dollar-value ROI of GenAI tools, along with close-up insights into how developers are using GenAI to realize their productivity gains. These learnings can be shared back with other developers and internal platform teams, helping make clear the best use cases for GenAI as well as gaps and opportunities.

## Final thoughts

GenAI represents a significant opportunity to boost developer productivity and job satisfaction. Effective collection of developer metrics and feedback is key to optimally rolling out and realizing the full impact of these tools.

As discussed, data can be used to better understand and drive adoption, as well as validate the financial ROI of productivity gains being captured. Insights on specific GenAI use cases can help with educating developers across your organization on how to best apply these tools.

The earlier organizations can establish baselines and put data mechanisms in place, the better: this provides a longitudinal view of how GenAI impacts your business over time.

[Learn how DX](#) can help you implement the measurement approaches described in this guide.