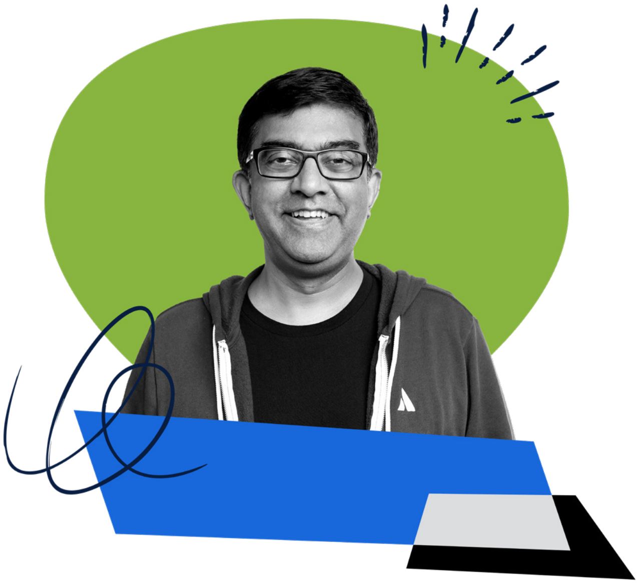
ATLASSIAN — DX

State of developer experience report 2024

Get a fresh look at what's getting in the way of developer satisfaction and productivity in the age of AI.

Table of contents



RAJEEV RAJAN
Chief Technology Officer at Atlassian

A note from Atlassian's CTO

Software development teams are dealing with more complexity than ever. They've moved from monoliths to microservices, seen an explosion in APIs, started to explore generative AI tools, and often work in a distributed environment. Developers are being asked to manage everything from security vulnerabilities to cloud configuration, which leaves little time for writing code.

None of this is surprising, given the recent spate of layoffs in the tech sector and tighter purse strings across the board. Teams are being asked to do more with less. Maximizing productivity is the order of the day. The trouble is that knowledge worker productivity is notoriously difficult to quantify—and as the saying goes, you can't improve what you can't measure.

Meanwhile, there's a significant body of research showing that happy workers are productive workers. So instead of worrying about what makes developers more efficient, what if we think about what makes them happy? Not the surface-level happiness you can buy with high-dollar swag and on-tap kombucha, but the deep sense of fulfillment that comes from creating something great.

Atlassian engineering is betting big on the idea that carrots are more effective than sticks. If we make developers' jobs more satisfying, we believe productivity will improve organically.

That's why we embarked on this research. We surveyed 2,100+ developers and managers across a range of industries to get a fresh look at what keeps work flowing smoothly vs. what introduces friction, and how they feel about their work environments in the age of microservices and AI. While this aggregated picture won't perfectly reflect your team's situation, it should provide some useful clues.

That said, reporting problems without offering solutions isn't the Atlassian way. So we also partnered with DX, the engineering intelligence platform designed by leading researchers.
By incorporating their existing work with this new data, we were able to include a section at the end with specific recommendations for engineering leaders.

- If you want to learn more about the challenges to developer experience, see Part 1 on page 4
- If you want to improve the developer experience at your organization, see Part 2 on page 13
- If you're looking for more resources on developer experience, see Part 3 on page 20

Speaking candidly, developer experience is an ongoing challenge for Atlassian, too. This research uncovered new information that my team and I are figuring out how to act on. We don't have all the answers, but we do have a mission to unleash the potential of every team–starting with our own. As we continue our developer experience journey, we'll keep sharing what we learn along the way.

I hope the information in this report will spark ideas for experimentation and investigation as you seek to unleash your team's potential.

RAJEEV RAJAN
Chief Technology Officer at Atlassian

Where we sourced our data

In February 2024, we worked with Wakefield Research and DX, an engineering intelligence platform, to learn more about:

1. The state of the developer experience at global organizations
2. Progress and limitations to developer experience improvements
3. The future of developer experience, including the impact of AI

To understand the perspectives of both engineering leaders and developers, we ran two global surveys:

- Wakefield Research surveyed 1,250 engineering leaders in the US, Germany, France, and Australia.
- DX surveyed 900 developers around the world, including the US, Germany, France, and Australia.*

*DX also surveyed developers in the UK, Sweden, Lithuania, Estonia, Spain, Ireland, Ukraine, Denmark, Switzerland, the Czech Republic, Canada, Brazil, and India.

Developer experience in 2024: the highlights

Developers and leaders know developer experience is critical to attracting and retaining talent.

86% of leaders believe attracting and retaining the best developer talent will be almost impossible without improving developer experience.

Developer experience is important or very important for 63% of developers when deciding whether to stay in their current job.

But that doesn’t mean leaders and developers are always on the same page.
There are mismatches between leader attitudes and what developers actually experience.

Less than half of developers think their organization prioritizes developer experience.

41% of leaders use tools that measure productivity to assess team satisfaction.

2 out of 3 developers are still losing 8+ hours a week to inefficiencies in their roles.

Leaders believe AI is the most effective way to improve productivity and satisfaction.

But 2 out of 3 developers aren’t seeing significant productivity gains from using AI tools yet.

PART 1

The state of developer experience in 2024

Developer experience – n. [dih-vel-uh-per ik-speer-ee-uhns]
Developer experience focuses on the lived experience of developers and the points of friction they encounter in their everyday work.

Developer productivity isn't well understood—or enabled

69%

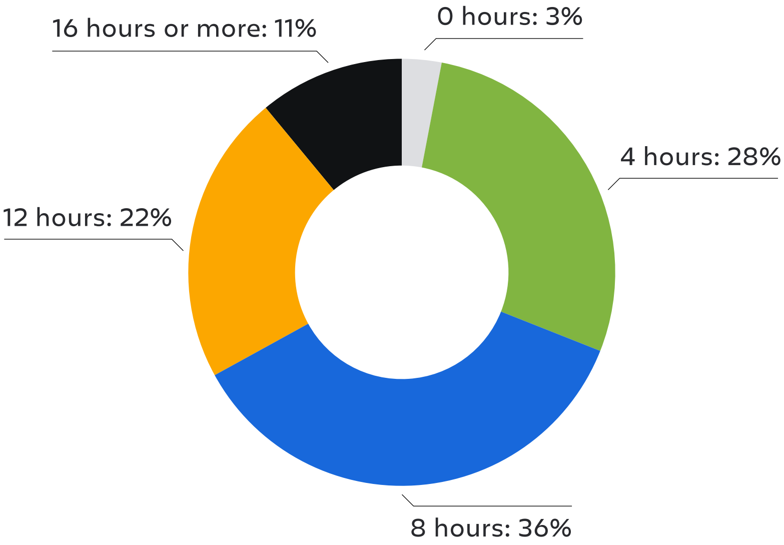
of developers lose 8 hours or more of their working week to inefficiencies.

44%

Less than half of developers believe leaders are aware of these issues.

The last few years may have seen a lot of interest in developer productivity, but that doesn't mean organizations are enabling their developers to be as productive as possible. In our survey, 69% of developers say they're losing 8 hours or more a week to inefficiencies in their role.

How many working hours a week do developers lose to inefficiencies?



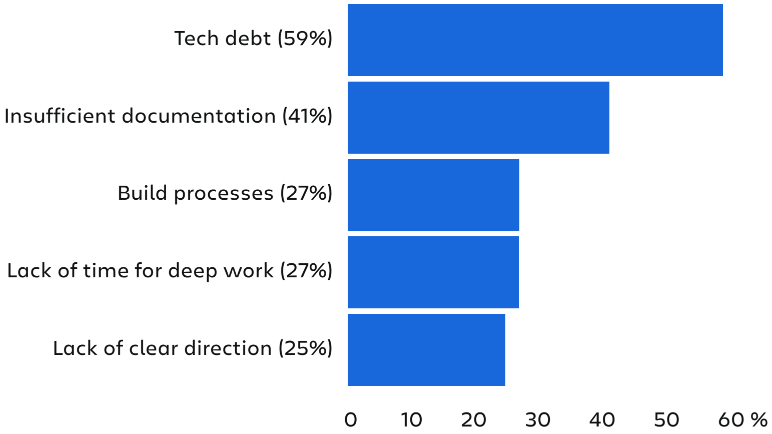
Hiring top talent only to have them blocked by organizational impediments is pointless–and expensive. For an organization with 500 developers, losing 8 hours per week costs roughly $6.9 million over the course of a year[1].

With such significant losses on the line, it's essential to understand the main friction points in the daily lives of software developers. Our study found that the top contributors are technical debt, along with a lack of documentation, processes, focus time, and direction.

*According to Stack Overflow's 2023 Developer Survey, the average salary for developers worldwide in 2023 was $69,767. If a developer loses 8 hours weekly on average to inefficiencies, those inefficiencies come at an annual cost of $13,954.40 per engineer per year. The larger the organization, the higher the cost of inefficiency.

Top 5 areas of developer time loss according to developers*

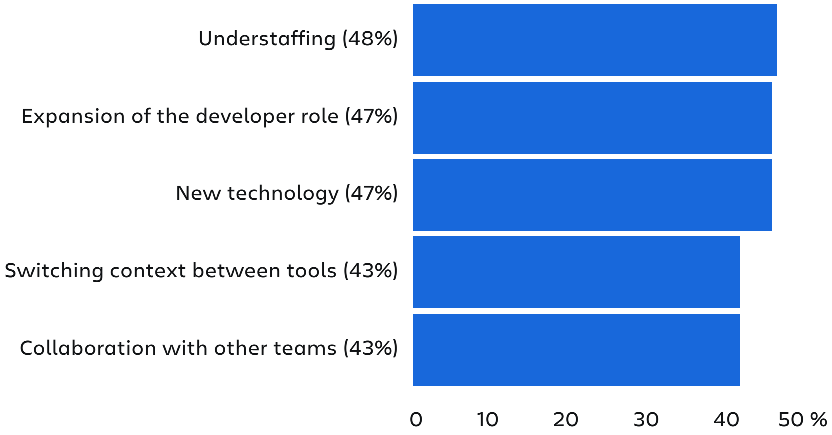
*multiple choice question



While just under half the developers we surveyed (44%) believe that leaders at their organization are aware of these issues, we found that almost all leaders (99%) do acknowledge that the developer role has become more complex. According to these leaders, these complexities are primarily seen as being driven by understaffing, technology, and tooling.

Top 5 areas of developer role complexity according to leaders*

*multiple choice question

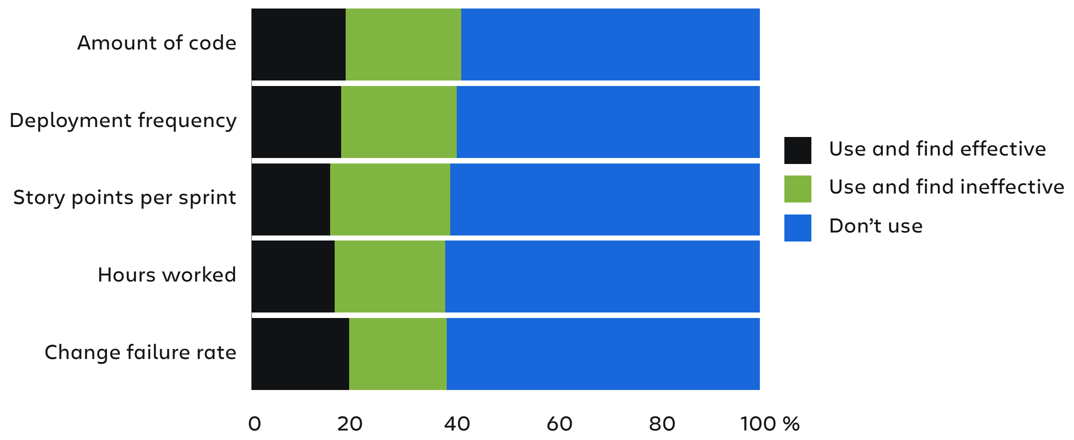


The problem with productivity metrics

Most leaders admit the metrics they track are ineffective for measuring developer productivity. Our survey of engineering leaders shows the most commonly used metrics are the amount of code written, the number of story points completed, and hours worked. We also found that leaders aren't exactly enthusiastic about them. For example, more than half of the engineering leaders using these metrics find them ineffective as a measure of developer productivity.

Top 5 ways organizations are measuring productivity and how effective these measurements are*

*multiple choice question



38%

of organizations measure developer productivity by hours worked.

69%

of developers are losing 20% or more of their time to inefficiencies at work.

Measuring hours worked is particularly problematic since it only shows how long a developer spends on their laptop. As we've seen, that time doesn't all go into coding. In fact, 69% of developers lose 20% or more of their time to inefficiencies at work. This might explain why 55% of leaders who use "hours worked" find it ineffective.

Experience and productivity are seen as being closely intertwined

51%

of organizations are focused on measuring developer productivity, while 49% are focused on developer satisfaction.

41%

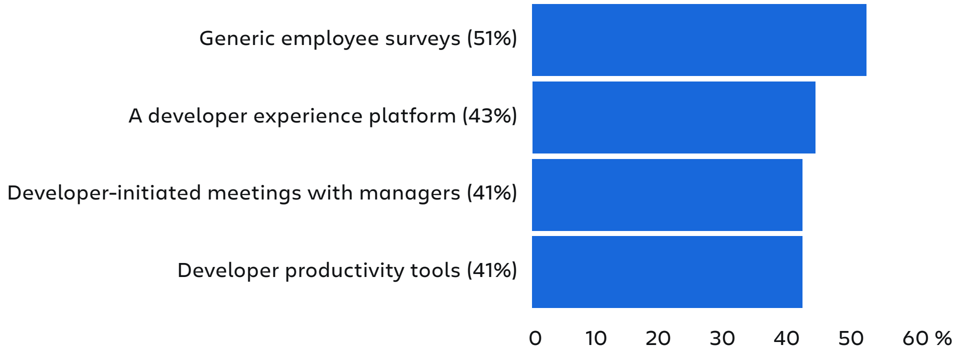
of organizations use tools that measure developer productivity to assess the satisfaction of their development teams.

While some organizations seem to recognize developer experience as a distinct concept, many others appear to conflate developer productivity and experience. For example, 41% of respondents said their team uses the same tool to assess developer productivity and experience.

Our survey didn't provide visibility into exactly which tools are running double-duty on productivity and satisfaction assessment, or exactly which metrics are being pulled. But it does present a red flag about making sure you're tracking the right metrics using the right tools.

Top methods leaders use to assess developer satisfaction*

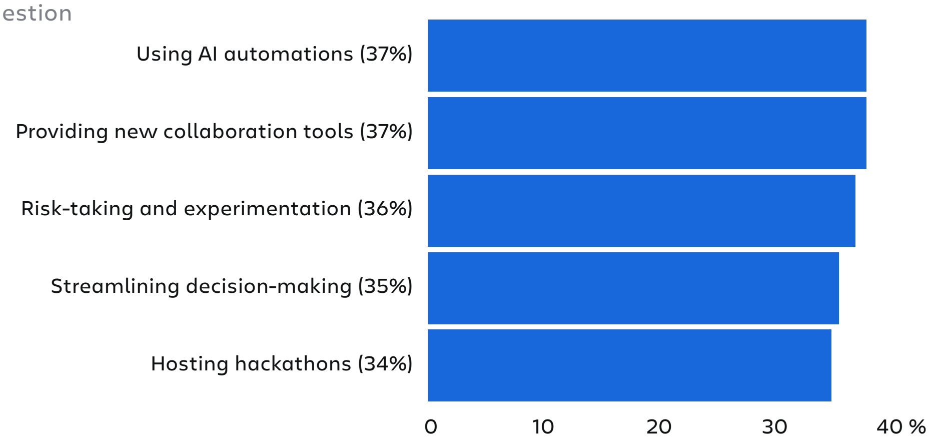
*multiple choice question



The survey also revealed engineering leaders are looking at a mix of automation, tooling, and culture and process shifts to increase autonomy and collaboration.

Top areas leaders believe will improve both developer productivity and satisfaction*

*multiple choice question



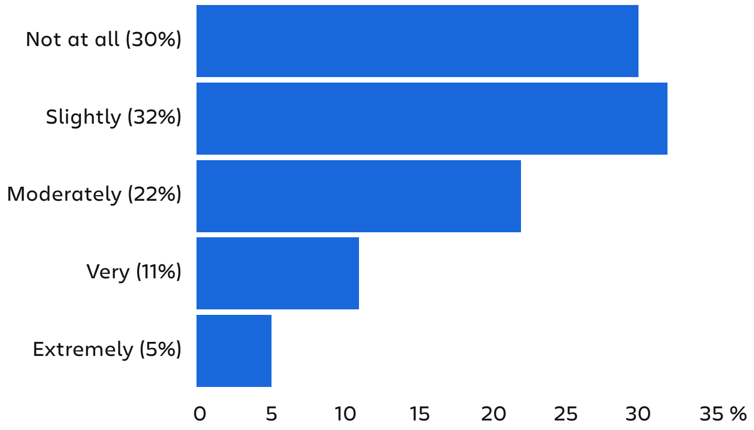
IDC predicts that organizations worldwide will spend $40 billion on GenAI tools this year. This investment will quadruple to $151 billion within three years. According to our survey, leaders believe using AI to automate processes and testing is the most effective way of improving developer productivity and satisfaction.

While much of the current AI focus is on helping developers save time in their code editors, AI shows potential to improve daily developer workflows by addressing tech debt, translating requirements, documentation gaps, and reducing interruptions.

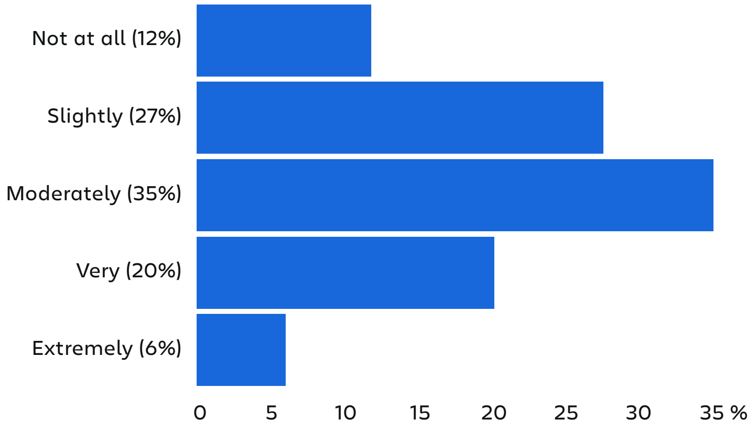
However, most developers are underwhelmed when asked to what degree AI-based dev tools improve personal productivity: 30% said not at all, while 32% said only slightly.

> **38%**
>
> of developers are seeing more than a moderate productivity improvement from AI-based dev tools, despite leadership's belief that using AI is the most effective way to improve developer productivity.

How much AI tools are improving developer productivity today



How much AI tools will improve developer productivity within the next two years



Developers are more optimistic that AI tools will improve their productivity in the future, with 35% moderately convinced and 26% very or extremely convinced. But this is still less than software developer leaders, who all believe AI will improve the developer role.

AI can help improve developer experience, but it can't solve all the pain points of development teams to improve productivity and satisfaction. It's critical for leaders to ask developers about their friction points and then focus on implementing the right tools and cultural changes to make a difference.

Learn more

How to measure GenAI adoption and impact

This guide from DX explains how to integrate GenAI into your software development process in a data-driven way and measure its impact on your business.

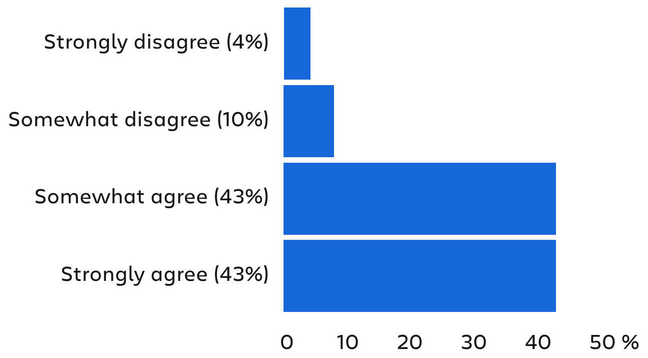
Interest in developer experience is growing, but efforts are lagging

> **23%**
>
> of developers are satisfied with the amount of time spent on improvements. Both developers and leaders know developer experience is important for attracting and retaining talent.

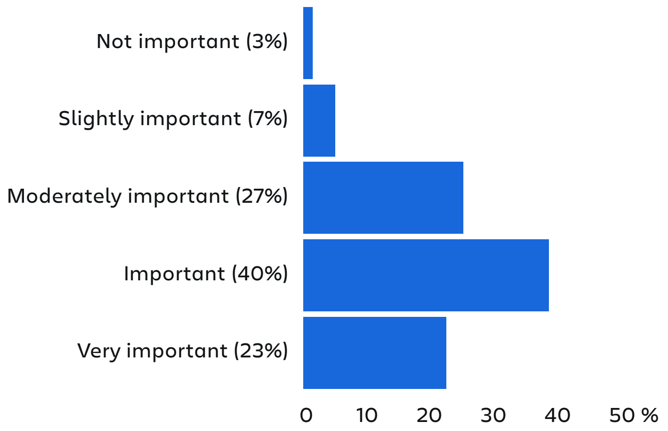
Providing a great developer experience can be an effective recruitment and retention tactic. 63% of developers said developer experience is important or very important when deciding whether to stay in their current job.

While 86% of leaders believe that attracting and retaining the best talent will be nearly impossible without improving developer experience, less than a quarter of developers (23%) are happy with their team's investment in this area. The existence of this gap isn't surprising; it can be difficult to budget for new tools and get time to tackle tech debt.

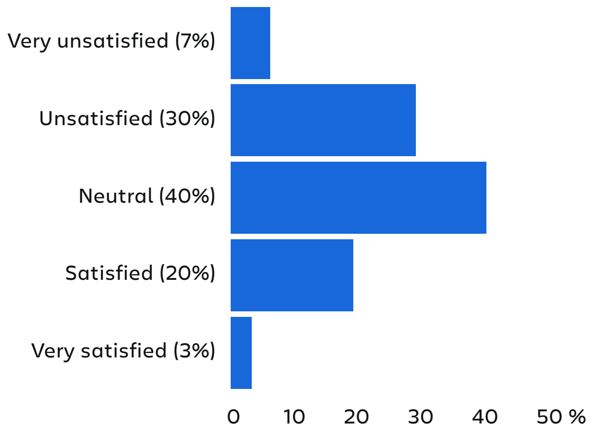
Leaders' take: Attracting and retaining the best developer talent will be near impossible without improving the developer experience



How important developer experience is to developers



Developer satisfaction with time spent on developer experience improvements



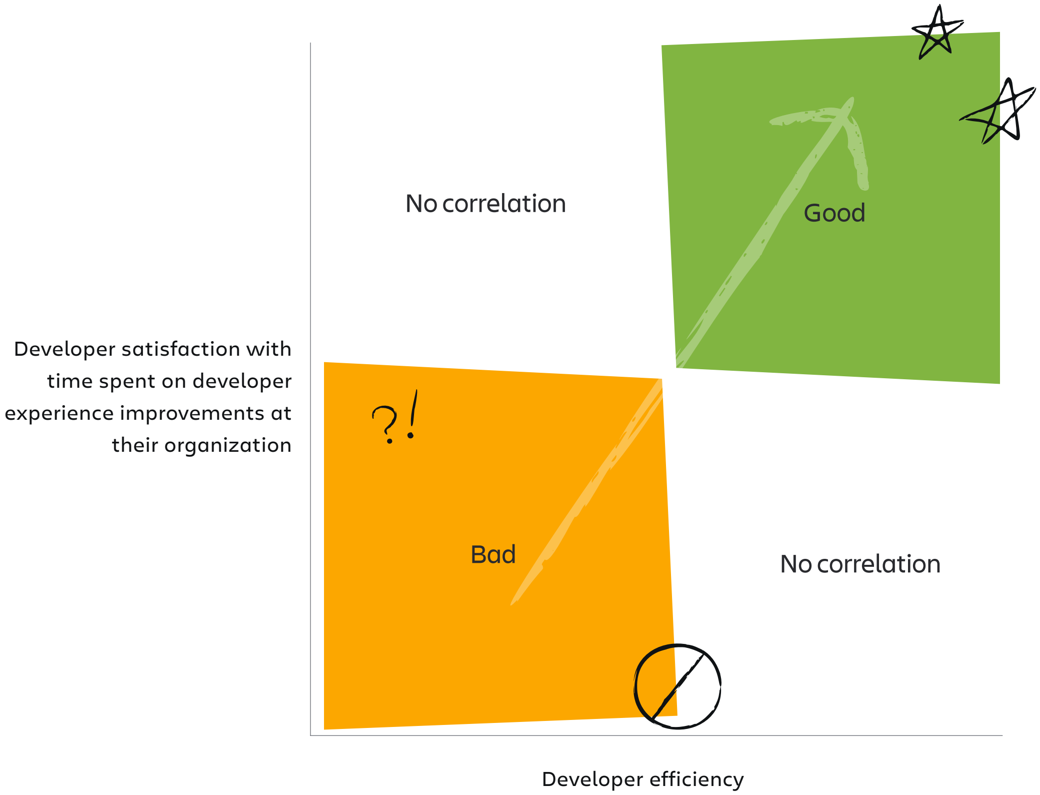
As our survey shows, almost two out of three developers consider leaving their roles when they aren't satisfied with the developer experience at their organization. But there's hope. 76% of organizations plan to invest more in developer experience within the next year, and according to our survey, that investment is poised to pay off.

We cross-referenced two survey questions we asked developers:

1. How satisfied are you with the amount of time your organization spends on developer experience improvements?
2. In a typical week, what percentage of your time is lost due to obstacles or inefficiencies in your work?

We found that less time lost to obstacles and inefficiencies correlates with higher satisfaction with developer experience investment. In other words, investing is a double win. Not only does productivity improve, but employee sentiment improves as well.

Correlation between developer satisfaction with time spent on developer experience and developer efficiency



76%

of organizations plan to invest more in developer experience. And when organizations spend more time on developer experience, developers have more time to work on what matters.

PART 2

How can we improve developer experience in 2024 and beyond?

According to research by DX, a positive developer experience has three core elements:

- **Feedback loops** that allow for continuous improvement through learning and adjustments
- **Manageable cognitive load**, thanks to well-organized code and easily accessible documentation
- The ability to get into a "**flow state**" where they're moving fluidly through tasks and completely engrossed

Let's step through each element and look at ways to improve them.

The building blocks of a great developer experience

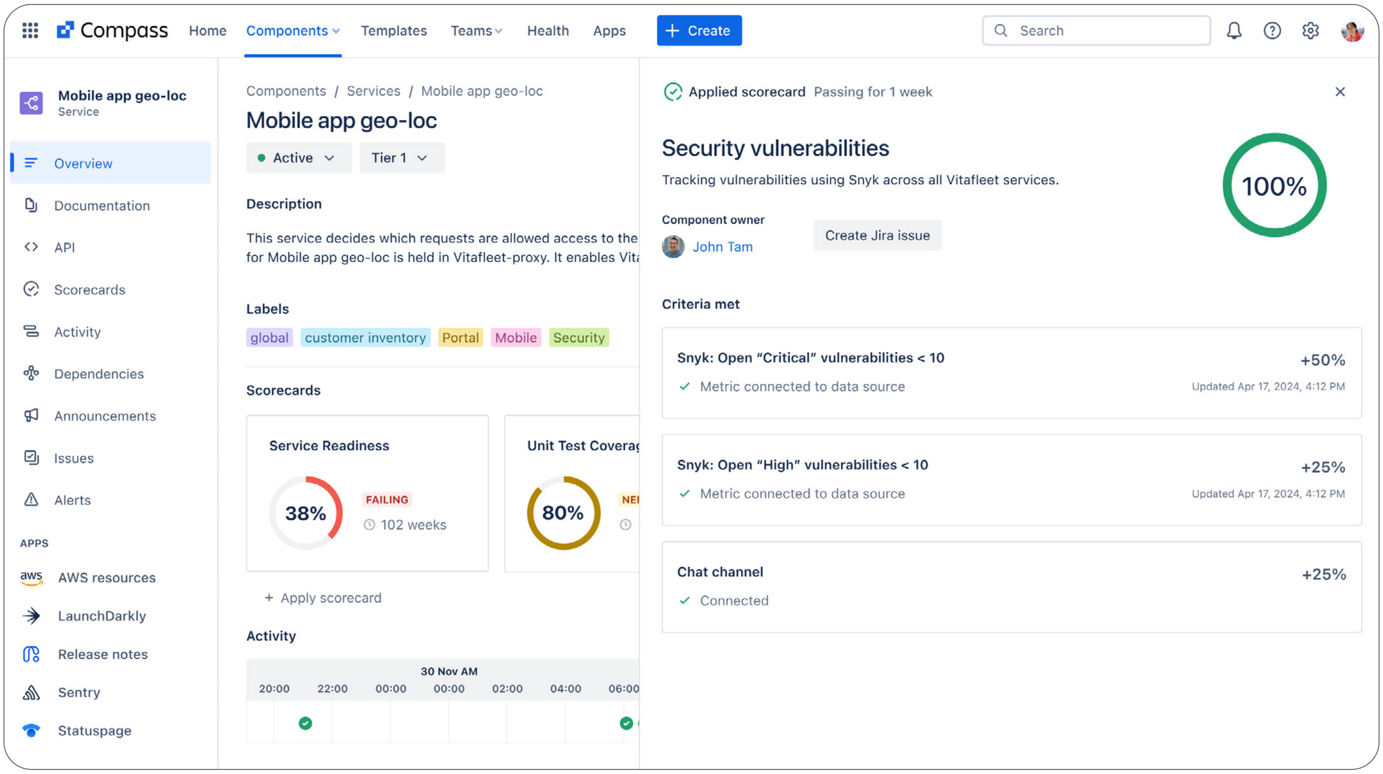
Our survey showed that engineering leaders believe things like AI tools and more streamlined processes will boost developer productivity and developer satisfaction at the same time. It is possible to address both areas in one fell swoop. But you have to understand the needs of each area separately in order to design effective solutions. With that in mind, we're going to focus solely on improving developer experience in this section–again, with the idea that productivity gains will follow.

Feedback loops

Regular **team retrospectives** give developers and leaders a chance to reflect on what's going well vs. what's not. Discussions can range from cultural topics like workloads and deadlines to technical issues like API design. You can find basic retrospective instructions and templates, as well as variations for specific situations, in the Atlassian Team Playbook's Retrospective play.

Dashboards in your project-tracking tool of choice (may we suggest Jira?) provide a wealth of disparate data points at a glance. Add modules showing in-progress tasks, bug counts, time to resolution, SLA compliance, release readiness, sprint burndown, and build status for a 360° overview of your team's work.

For a deeper view of the microservices and other components your team works with, try the **component catalog and system scorecards** in Compass, Atlassian's Developer Experience Platform. Developers can quickly access the information they need to work with various components, such as who owns them and how they interact with other systems. Meanwhile, leaders have an easy way to monitor and maintain systems' health to help identify bottlenecks for their teams and keep development work flowing smoothly.



Components / Services / Mobile app geo-loc

Mobile app geo-loc

Active | Tier 1

Description

This service decides which requests are allowed access to the for Mobile app geo-loc is held in Vitafleet-proxy. It enables Vita

Labels

global | customer inventory | Portal | Mobile | Security

Scorecards

Service Readiness: 38% FAILING 102 weeks

Unit Test Coverag: 80% NEI

+ Apply scorecard

Activity

30 Nov AM

20:00 | 22:00 | 00:00 | 00:00 | 02:00 | 04:00 | 06:00

Applied scorecard Passing for 1 week

Security vulnerabilities

Tracking vulnerabilities using Snyk across all Vitafleet services.

100%

Component owner: John Tam

Create Jira issue

Criteria met

Snyk: Open "Critical" vulnerabilities < 10 ✓ Metric connected to data source	+50% Updated Apr 17, 2024, 4:12 PM
Snyk: Open "High" vulnerabilities < 10 ✓ Metric connected to data source	+25% Updated Apr 17, 2024, 4:12 PM
Chat channel ✓ Connected	+25%

Service details and scorecard in Compass

Think of it as a "developer experience hat trick" in that it scores a win in all three areas:

Maintain the flow state

Developers don't need to dig through the archives or disturb team members for answers. It's all available in a single location and standardized format.

Reduce cognitive load

Developers access information in one centralized space, eliminating the need to remember the details of how everything fits together.

Provide feedback loops

Teams can review progress together through weekly data-informed retrospectives to improve performance continuously.

These improvements make getting work done easier and more enjoyable, creating a better engineering experience at your organization.

Cognitive load

This is where AI-powered tools have the potential to make significant gains in developer experience.

> **Code generation**: Code generation tools like GitHub Copilot, Tabnine, Codeium, and Cody can produce output in a variety of languages, including Python, Javascript, Go, and PHP. Some of them can even convert natural language instructions into code.

Incident response: When an incident strikes, AI can summarize the relevant tickets and documents so developers can get up to speed quickly.

Searching for information: According to the latest Stack Overflow Developer Survey, over a quarter of developers spend an hour or more each day searching for answers or solutions to problems. AI tools like Atlassian Rovo can gather and summarize the latest information from internal documentation and project management tools to reduce time spent away from software development and wasted on tedious searching.

Documentation: Inefficient documentation is a major hindrance for 41% of developers in our survey. But that doesn't mean that developers have the time or desire to write the documentation they need. AI can create code and process documentation quickly and painlessly, allowing developers to focus on more impactful tasks.

Flow state

Staying in the flow is impossible without large blocks of uninterrupted focus time. That means no meetings, no pings, no shoulder-taps. Some teams designate one or two days each week as "no-meeting days." For a more ad-hoc approach, encourage team members to put a handful of recurring 90- to 120-minute holds on calendars at whatever time works best for them. Either way, empower developers to silence their phones and turn off chat notifications. For more ways to regain control of your calendar, check out Atlassian's 2023 report on meetings.

Now trending: platform engineering teams

Platform teams provide internal services to software delivery teams, enabling them to work autonomously. These services can include infrastructure and developer tools like those for CI/CD, monitoring, and logging. They make it easy for software teams to comply with organizational standards and practices.

According to Gartner, by 2026, 80% of large software engineering organizations will establish platform engineering teams. Given their impact on developer experience (and therefore, productivity), enterprises will soon find themselves at a disadvantage if they don't establish a platform team.

Learn more:
DX discusses how to advocate for serious investment in platform work for long-term business impact.

Measuring developer experience

You *can* measure developer experience. But, even with all the metrics available, there isn't one metric, or even a set of metrics, that rules them all.

The DORA metrics of Deployment Frequency, Lead Time for Change, Change Failure Rate, and Mean Time to Restore have become the default way to measure DevOps performance. As valuable as these metrics are, they don't provide insight into development teams' daily challenges and experiences. Even the SPACE framework, which expands on DORA to consider the less-quantifiable aspects of the developer role, is difficult to convert into applicable productivity metrics for developer teams.

This is because developer experience and productivity are highly contextual between teams and organizations. Even if all companies purchased the same tools, the developer experience would still vary significantly between teams and companies.

Since no single metric or framework can be applied across all contexts to quantify developer experience, you need to look at things from a few different angles to get the 360° picture:

Perceptions

How developers feel about their work and their working environment

Workflows

How efficient and reliable systems and processes are

Key Performance Indicators (KPIs)

The measures your team obsesses over, based on your specific situation

Let's dive into each one.

Developer sentiment

Generic employee surveys are the most common way of measuring developer satisfaction, according to our study. But they don't include questions that address developer pain points. Carefully crafted developer experience surveys, on the other hand, give you the specific details you need to understand your teams' challenges. The responses will show you which areas to prioritize.

Learn more

Developer Experience Survey Play

Understand precisely what's holding developers back and which improvements would yield the highest value for the team with this Atlassian play.

A word of warning, however: all talk and no action can exacerbate dissatisfaction. In our developer experience efforts at Atlassian, we've found it helpful to bring developers in as collaborators.

- Publish anonymous survey results and attach potential next steps for teams to vote on. This builds a feedback loop and gives developers agency in the process, which can improve job satisfaction.
- Be transparent and accountable. Explain the steps you'll take and set a date to check in on progress.

Workflows and processes

There's a simple framework for measuring and improving developer processes:

1. Ask developers what causes the most friction in their workflows
2. Start tracking and measuring that area of friction
3. Implement changes and monitor their impact

Here's how this might look in practice:

> Build processes were the third most common obstacle slowing developers down in our survey. When developers report build processes as a blocker, track build and deploy times. Also, see the number of manual steps required to deploy changes, kick off tests, or initiate a pull request. Can the process be automated? If so, see what impact those automations have on build processes, based on the measurements *and* team feedback.
>
> Do teams report a lack of focus time like over a quarter of developers in our survey? A calendar analysis will reveal your team members' meeting load, while a report pulled from your service management tool can show the frequency of incidents that interrupt the team. Try implementing and monitoring the impact of meeting-free days and an updated incident response plan.

Given that no one metric will reduce friction for every team, the first step is to speak with developers to find the right metrics to track and improve.

Key performance indicators (KPIs)

Most people think of KPIs as "hard numbers" or "objective measures," and that works for quantitative concepts like revenue or failure rates. However, developer experience is subjective, which means its KPIs will be somewhat qualitative. DX recommends choosing one of the following and incorporating it into your developer surveys:

- Overall perceived ease of delivering software
- Perceived productivity
- Employee engagement or satisfaction

Perceived ease of delivering software is great for teams that have a thriving culture with effective collaboration practices (e.g., decision-making, workflows) but are mired in technical debt and/or brittle systems.

Similarly, **perceived productivity** zeros in on the team's work but is less specific to making software, making it suitable for platform, site reliability, and operations teams.

Employee engagement or satisfaction covers the entire work experience, and might be a good fit for teams struggling with retention or other culture-related issues.

We cannot overstate the importance of choosing one of these metrics (or something similar) as your developer experience KPI. Productivity metrics like pull request turnaround times or user story cycle times provide useful signals about efficiency, but they don't speak to whether the job is a joy or filled with obstacles.

Notably, some of the metrics that showed up in our survey don't speak to developer productivity or experience. For example, hours worked only shows how long someone was logged into their laptop. Deployment frequency is equally meaningless; there's no indication of how significant vs. trivial the change was, or whether that code was riddled with bugs.

We recommend choosing a meaningful KPI as the first step. In fact, we recommend it as the first step in your developer experience journey. Once you've identified your North Star, deciding how to go after it becomes much easier.

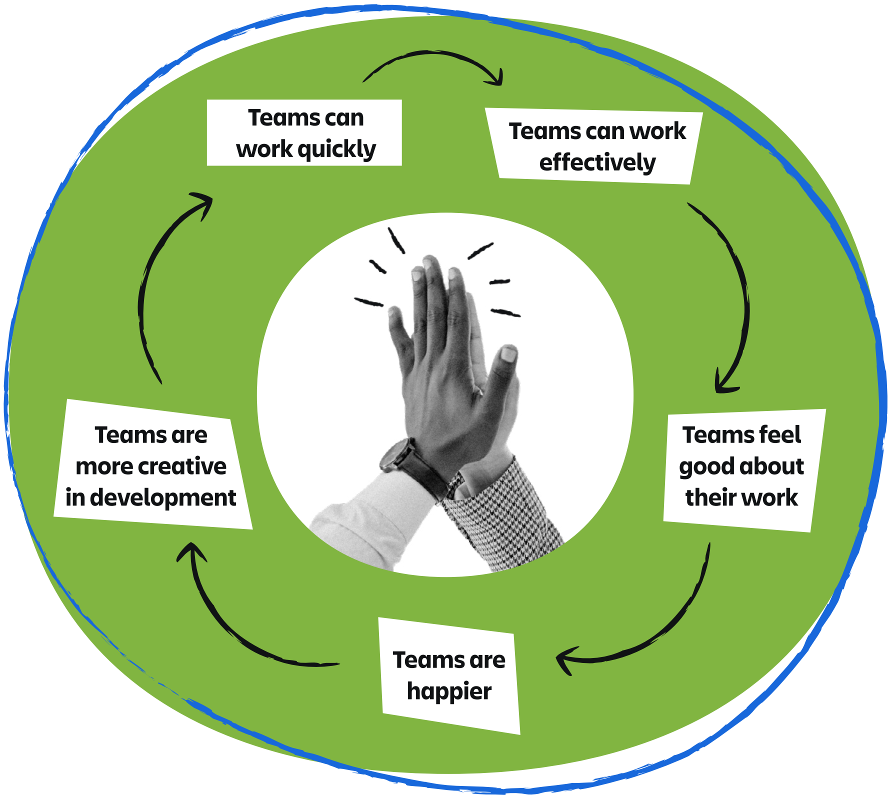
Learn more

Developer productivity metrics at top tech companies

This DX report shares the top-level metrics that 17 tech companies, including Atlassian, use.

DORA, SPACE, and DevEx: Which framework should you use?

DX's CTO describes how companies should think about and use the DORA, SPACE, and DevEx frameworks for measuring developer productivity.


Teams can work quickly
Teams can work effectively
Teams feel good about their work
Teams are happier
Teams are more creative in development

Atlassian's "developer joy" initiative

Context-switching, missing documentation, and inefficient processes are a drag. But what happens when your teams can enjoy more of the frictionless flow state and feedback loops of a positive developer experience? You get a virtuous cycle of accomplishment and fulfillment–something we call "developer joy."

At Atlassian, developer joy is a company-level OKR. If joy is the objective, then think of teams, tools, and culture as the three key results, each with baseline metrics so we can track progress. These are a few of the tactics we've found to be successful.

Awesome tools	Empowered teams	Amazing engineering culture
Initiative ideas: ▪ Introduce a new pull request automation ▪ Migrate to new test libraries ▪ Invest in a developer experience platform **Measurement ideas:** ▪ Deployment cycle times ▪ Pull request times	**Initiative ideas:** ▪ Involve development teams earlier in the planning process to show the connection between tech and goals ▪ Give development teams more control over their roadmaps ▪ Spend 10% of time spent on tackling tech debt **Measurement ideas:** ▪ Whether teams spend the allotted time on tech debt	**Initiative ideas:** ▪ Build guidelines for development standards ▪ Host quarterly focused hackathons ▪ Create a demo culture with weekly showcases **Measurement ideas:** ▪ Team satisfaction scores

We first shared our journey and early results in this blog. Overall developer satisfaction has risen roughly 25 percentage points from 49% to 74% over the last two years of this initiative. Drilling down into specific areas, we're also tracking satisfaction with documentation and our cloud platform.

Our progress has been steady but not linear. Events like a restructuring or a string of high-severity incidents have an unavoidable impact on developer sentiment, and Atlassian engineering has experienced both. Nevertheless, we'll keep pressing forward and sharing what we learn through blogs, events, and reports like this.

To sum it up...

1. Most developers lose at least 20% of their workweek to inefficiencies, which costs organizations.
2. Many organizations still use developer productivity metrics they feel are ineffective.
3. When looking at AI tools, go beyond code generation. Work with your developers to understand use cases that support them, from task prioritization to trend identification and vulnerability assessment.
4. Direct conversations with your development team are the best way to understand their needs and challenges.
5. Improving developer experience has a positive impact on productivity and satisfaction.
6. Developer joy can take your developer experience to the next level for more gains.

```
    (!@#$%^&*()*+=~`) Not allowed")){
var firstname=document.getElementById('fname');if(
var fi
    Pa
var fi
var fi
    Use
```

PART 3

Additional resources

About Atlassian: Atlassian unleashes the potential of every team. Our collaboration tools help distributed teams organize, discuss, and complete shared work. This includes Compass, our developer experience platform that allows you to track all services, improve software health, and create a better developer experience at your organization.

Learn more about Atlassian and get more of our insights on developer experience:

See how we help improve developer experience from our Compass customers:

Developer Experience reel

Learn about our new features and developments that help support customers like OVO Energy identify developer blockers, manage APIs, and speed up software delivery.

Watch the video

Boden case study

Hear from James Crowe, Head of Software Development at Boden, about how they use Compass to help their engineers work smarter and more efficiently.

Watch the video

ExpressVPN case study

ExpressVPN's global software teams rely on Compass to understand, standardize, and improve their software components while maintaining a culture of autonomy.

Watch the video

About DX: DX is an engineering intelligence platform designed by leading researchers. They give engineering leaders and platform teams the data they need to lead with confidence and drive higher impact per developer. We are privileged to serve hundreds of the world's most iconic companies including Dropbox, Etsy, Twilio, Vercel, and Pfizer.

Follow along:

Learn more from DX:

Developer Productivity resources

Get industry reports that answer questions like 'What % of headcount are companies allocating to Platform teams?' and 'How do we measure the impact of GenAI tools?'.

Explore industry reports

Plaid case study

Plaid chose DX over building their own developer productivity metrics and surveys in-house. They implemented DX and got the insights they needed in a week.

Read the story

SiriusXM case study

After running in-house surveys and quantitative metrics, SiriusXM's Platform Engineering team chose DX to bring all metrics into one place, inform roadmaps, and quantify the impact of their work.

Read the story

ATLASSIAN