Adaptavist

New developer realities

Special report: How AI is reshaping the developer experience, the identity of the developer, and the future of software engineering

2026



Table of contents

FOREWORD 3

INTRODUCTION 4

CHAPTER 1 5
An impending DevEx(odus)

CHAPTER 2 13
The new developer divide

CHAPTER 3 16
Entering the age of agents

CONCLUSION 21





Foreword

From **Tom Lasparini**, Chief Operating Officer, The Adaptavist Group

Our report **'The human cost of digital transformation'** found that technology introduced without the right strategy, culture, and support leaves people stressed and quietly disengaged. Our follow-up research, **'The human cost of AI transformation'**, found that AI has added new pressures, including time spent checking AI output and not understanding why it should be used, creating feelings of nostalgia for work before the age of AI.



Our newest research looks at the people who work closest to the technology: developers. This group has always worked under pressure to ship fast without sacrificing quality. After seeing simplistic metrics like 'lines of code' gamed, organisations have evolved to measure developer outcomes more effectively, but in an AI-enabled world, finding the right measurable outcomes to monitor has become more difficult again.

What is new is that, according to our findings, many organisations now measure the volume of developers' AI use rather than the quality of their work, which is driving engineering teams to use it for show.

The research also dives into a paradox we have seen before. Most developers say AI has made them more productive, and that its benefits outweigh its drawbacks. Yet they are being asked to do more with fewer resources, and many worry about how the next generation of engineering talent will build foundational experience.

At Adaptavist, we believe that technology itself is rarely the problem. How an organisation introduces it, measures it, and supports the people using it is a much stronger predictor of the outcome. Developers are clearly telling us they want to be measured on the quality of their work and to have time to solve problems properly. To retain this essential talent and gain real returns from AI investments, business leaders need to listen.

Introduction

Today, software development is experiencing a period of rapid change, and we are still figuring out what the sustained results of it will be.

Developers are shifting from using AI chatbots as coding assistants to leveraging fully-fledged agentic workflows. They have moved beyond experimenting with AI to approaches such as vibe coding and 'tokenmaxxing', which are fast becoming routine ways of working for some. Increasingly autonomous systems are changing what it means to write, review, and manage software.

Much of the conversation around AI to date has centred on what this means for businesses: faster delivery, greater productivity, and the prospect of doing more. Far less attention has been paid to the developers on the ground who are expected to make it work.

The productivity case is strong. **83% of developers say AI has made them more productive,** and **65% say its benefits outweigh its drawbacks,** against just 11% who disagree.

But the research shows that how AI is introduced, managed, and measured matters as much as whether it is used at all.

Developers are being asked to do more with fewer resources, judged on adoption rather than outcomes, and increasingly left to manage AI systems with little support.

This report explores the gap between AI's proven productivity gains and the conditions organisations are creating around its use. It examines the developer experience, how AI is changing developer identities at every career stage, and the shift in developers' day-to-day work as managing agents becomes an essential part of the developer's role.

'New developer realities' shows that the organisations getting the most from AI measure the right things and give developers room to decide how to use it, and where human expertise and problem solving skills are required. Usage levels alone say little about who is doing well.

Methodology

The research surveyed 1,000 developers in the UK, US, Netherlands, Spain, and Germany in March 2026. The research was conducted by Attest.

CHAPTER 1

An impending DevEx(odus)

AI is making developers more productive, but poorly managed adoption risks creating a developer experience defined by pressure, speed, and reduced autonomy.

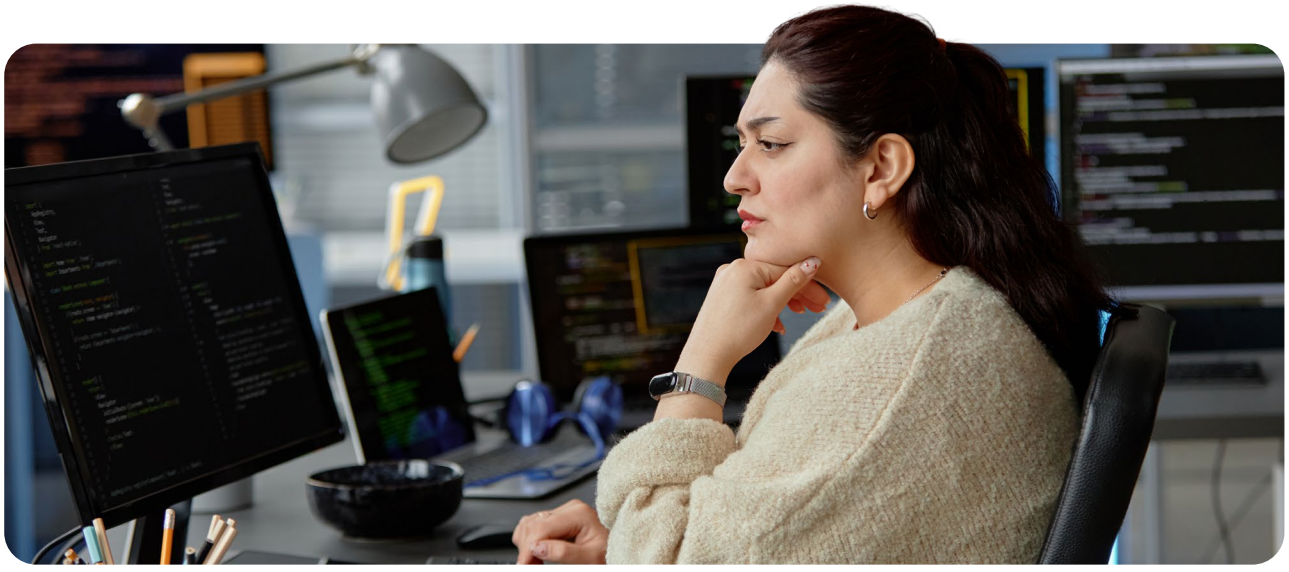
1a. The pressure to do more

Developers are being asked to **work faster and do more with fewer resources,** while AI adoption is becoming an organisational expectation. For many developers, AI has raised expectations without lightening the load.

60%
say they are being asked to **do more** with fewer resources because of AI expectations

55%
say AI creates pressure to **work faster**

61%
say AI is changing software engineering **too quickly**



That pace is taking a toll on how developers feel about their futures, and on how well they do their jobs. **54%** report **increased anxiety about their long-term career prospects** because of AI, rising to **61%** among developers who experience mandated AI usage.

Anxiety about the future is no longer a background worry. For more than a third of developers, it is already getting in the way of work itself, with **36%** reporting that this anxiety **negatively affects their performance at work.**



AI is also generating work of its own, **with 54% admitting they spend more time verifying code because of AI.**



STEPHEN CIHAK

Field CTO, Harness

harness

"AI is making code generation faster than most delivery systems were designed to handle. The time teams save by writing code tends to be spent on reviewing, testing, and debugging a larger volume of code.

The fix is automating verification with the same rigour applied to generation: continuous quality gates, automated test maintenance, and deployment verification that catches a bad release before someone has to stay late handling an incident."



```
for (let pos = 0; pos < 6;
  console.log(`Cycle ${pos
Processing...`);
  if (pos % 2 !== 0) {
```



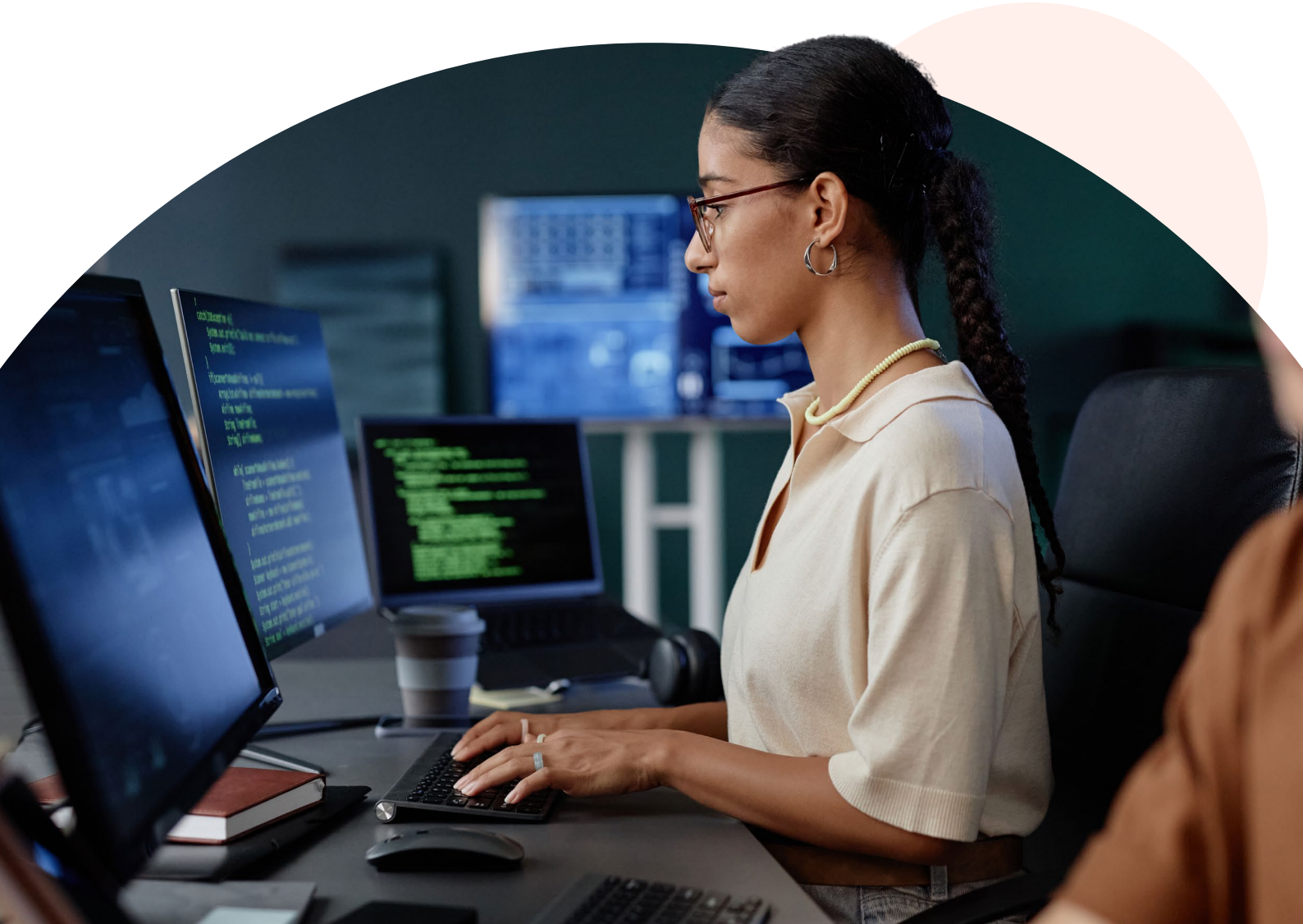
MIKE LONG

CEO, Kosli

"It's surprising that only 55% of developers say AI creates pressure to work faster. With AI entering the software development lifecycle, we're seeing pressure to work faster placed on everyone involved.

Developers are told AI should make them more productive, so there's far greater expectation on them. The **platform teams** that facilitate them are expected to scale pipelines built for a fraction of today's change volume. Then the **governance, risk management, and compliance** teams across all three lines of defence face a growing queue of changes to review, managed by the same number of people using the same manual processes. And we shouldn't forget about the **security** folks who are asked to deal with huge numbers of new vulnerabilities being discovered by tools like Mythos.

Speed in one part of the SDLC creates huge strain everywhere else. AI is creating interesting new challenges that are getting bigger every day."

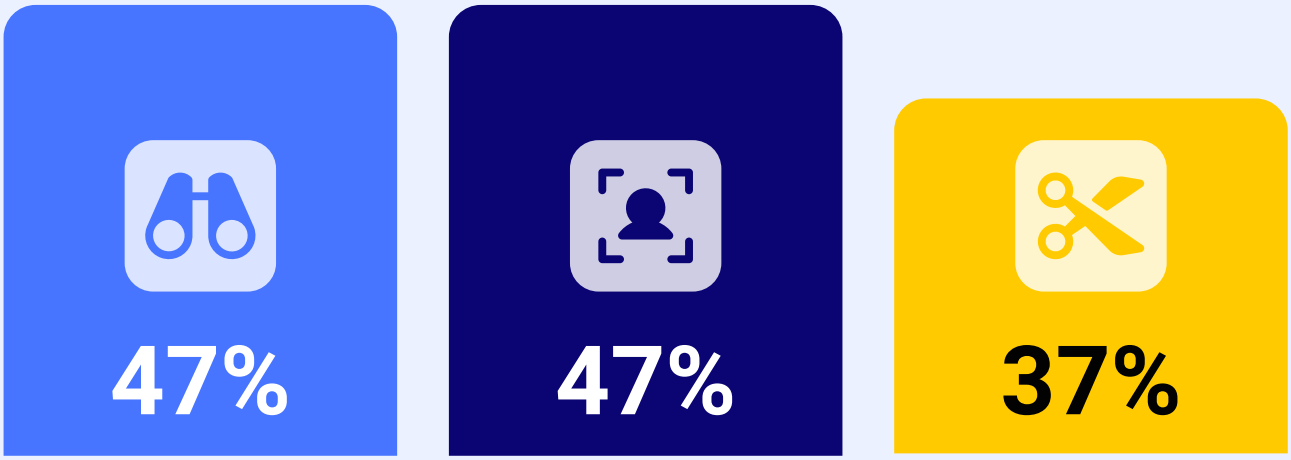


1b. When AI adoption becomes performative

Mandated usage metrics are driving developers to theatre too, with 'performative' use surging. This is when employees use AI tools to signal compliance to management, rather than to produce better software.

Half of developers (50%) admit to using AI primarily to demonstrate adoption to management rather than improve outcomes. 52% admit to using AI more often than necessary simply to meet organisational expectations.

This pressure comes from how developers believe they are judged:



of developers say they are assessed on **how much they use AI** rather than on the quality of their work

say their organisation values **AI usage over AI outcomes**

worry they could **lose their job** if they do not use AI

When usage is what gets noticed, usage is what gets delivered. This pressure is counterproductive however, as **more than half of developers (51%) say management pressure to use AI actually reduces productivity.**

Tokenmaxxing, the practice of maximising AI token consumption to appear busy or productive, is the clearest symptom. 51% say it creates unnecessary pressure, 41% say it damages developer experience, and 46% describe it as an indicator of organisations measuring the wrong things.

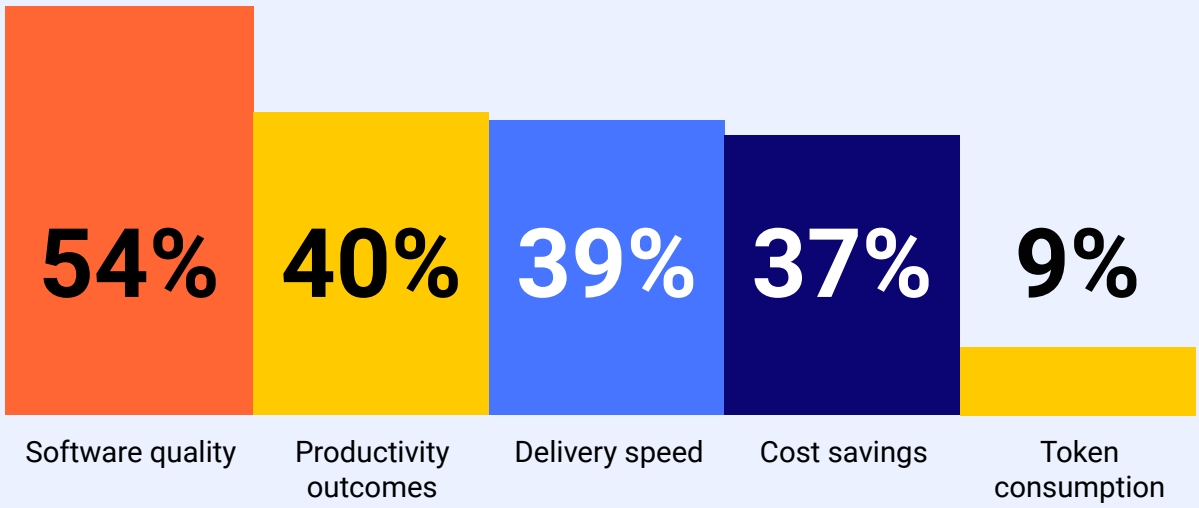
1c. Tracking the wrong metrics

Organisations are mismanaging AI adoption by **rewarding raw usage rather than outcomes,** and developer experience is paying the price.

Usage is fast becoming the yardstick by which AI impact is measured, even though few believe it should be:

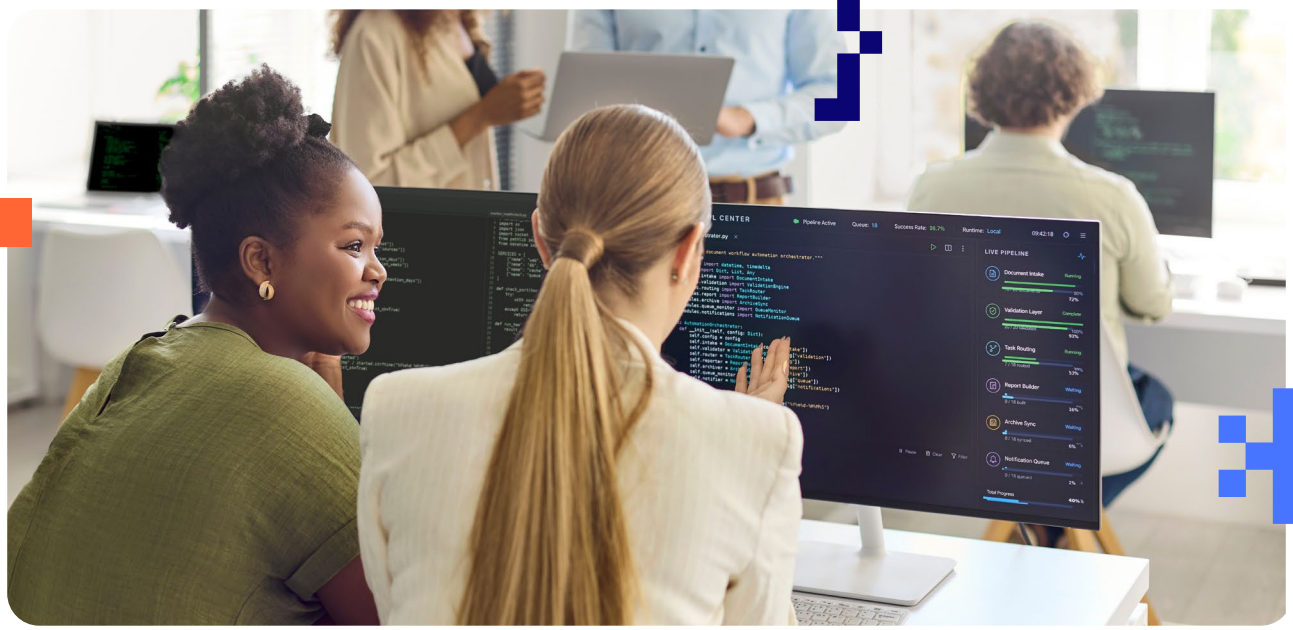
- **58%** say AI usage is becoming a **formal performance metric** or KPI
- Yet only **22%** believe usage levels are a valid measure of success
- Only **24%** say their organisation formally **measures AI usage against outcomes,** while **21%** say usage is encouraged but **not measured at all**

What do developers think is a meaningful indicator of AI success?



These numbers point to a developer community focused on the end-user and the value its product delivers. Leadership, meanwhile, appears to be distracted by AI usage metrics that say little about either. If the obsession with adoption crowds out attention to quality and user experience, customers and the business will ultimately feel the consequences.

But there is also a wider disconnect. Developer sentiment is not always aligned to the official company line on AI, and that gap carries real consequences for morale, retention, and the quality of what gets shipped.





MATT SAUNDERS

DevOps Lead
The Adaptavist Group

"Every organisation is scared of being left behind when it comes to AI adoption, and with rivals frequently showing early wins, many have mobilised by throwing money at the perceived problem. We're in a time of heavy AI experimentation, so there is merit to this strategy, but organisations must get better at surfacing how they measure actual deliverable value to their customers with AI. This goes far beyond usage metrics—considerations like DORA and Value Stream metrics remain valuable and must not be lost."

1d. The DevEx(odus)

We may not yet be at the point of a developer exodus. But AI is already making software development a less attractive career for a significant proportion of developers.

40% feel AI is making software development **less enjoyable**

37% say it is making it **more stressful**

46% believe software development is becoming a **less attractive** career because of AI

43% say it **already has**

These numbers show a profession under strain, though not yet in revolt. When close to half of developers say their career is losing its appeal, organisations that depend on them face a retention problem in the making.

The scale of this dissatisfaction points to a growing developer experience challenge. But developers are clear about what organisations can do to address it. Far from resisting AI, they want the conditions to use it effectively:



72% say **good developer experience** is more important than ever in the AI era



81% rank having **time to solve problems properly,** rather than constantly working to deadlines, as critical to good developer experience



MATT SAUNDERS

DevOps Lead
The Adaptavist Group

The Adaptavist Group

"Good DevEx is still attainable when working with AI tools, it just looks different. Developers still need autonomy, mastery, and purpose. Tooling and processes around these ideals with a semi-autonomous agent in the loop means carefully crafting agentic roles so that human developers can make the most of their thinking brains, whilst handing the toil to the agents. Prioritising human welfare over that of the agents can easily be forgotten, and it shouldn't be."



1e. The technology *is* delivering

It's important to note that developers still report clear productivity and quality gains using AI, despite these pressures.



83% say AI has made them **more productive**, and **81%** say it has made them a **better developer**



73% say AI helps them deliver **better results**



80% say AI is making development **faster**, **74%** say **better,** and **68%** say **cheaper**



Overall, **65%** say AI's **benefits outweigh its drawbacks**, versus **11%** who disagree

Advice from Adaptavist

The pressures developers describe clearly trace back to how AI is being rolled out by leadership. It's clear that usage numbers do not paint a true picture of AI success, so organisations should replace these targets with measures tied to software quality and value delivery, as well as giving developers a genuine say in where, and how, AI is applied. The aim here is to direct AI to the repetitive, manual work that slows teams down, and to protect the time and autonomy developers need to do the rest of their jobs properly.

Organisations need to analyse the effects of adding AI to existing processes, adapting carefully where the AI adoption moves bottlenecks from one place to another in the software development lifecycle, and looking at the counter-effects of what on the surface looks like increased productivity. The effect on developers may not always be obvious.

CHAPTER 2

The new developer divide

AI adoption is becoming a talent development issue as well as a productivity one, with juniors getting less hands-on learning and mid-career developers the most pessimistic about AI's impact on their progression.

AI is changing not only what developers do, but who they are at each stage of their career, including how they learn, how they build expertise, and how they see their future. This risks creating a new divide across the engineering career ladder and, more broadly, a shift in the identity of what it means to be a developer.

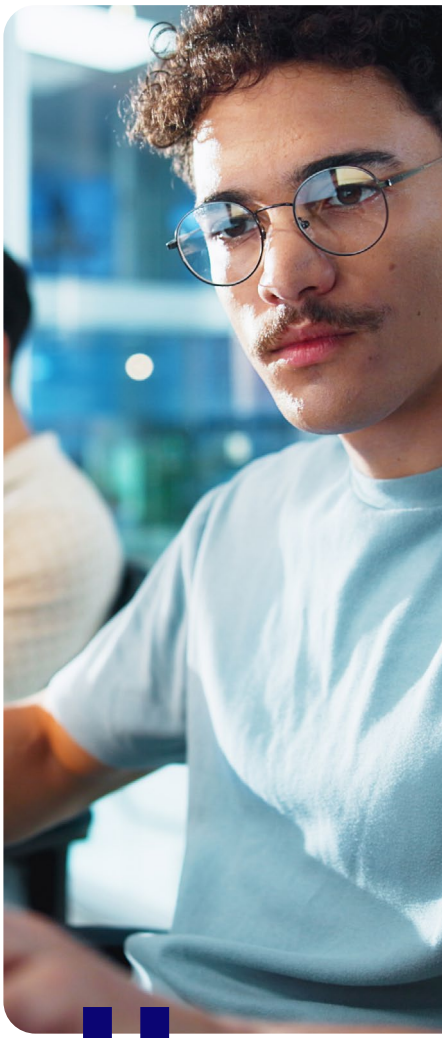
2a. The skills developers are no longer learning

As AI continues to take responsibility for the tasks junior developers have traditionally completed and learned through, the pipeline of future technical expertise is put at risk.

70% worry that junior developers are becoming overly reliant on automation. This concern is remarkably consistent across seniority: **68% of early-career, 69% of mid-career,** and **71% of experienced developers agree.**

What's more, the effect on learning is already visible:

- **63%** say AI has **reduced the amount they learn** through hands-on coding
- **43%** feel they have **fewer learning opportunities** because of AI
- **44%** believe AI is making **core engineering skills less important**



2b. The career progression gap

Early-career developers report the smallest progression benefit from AI.

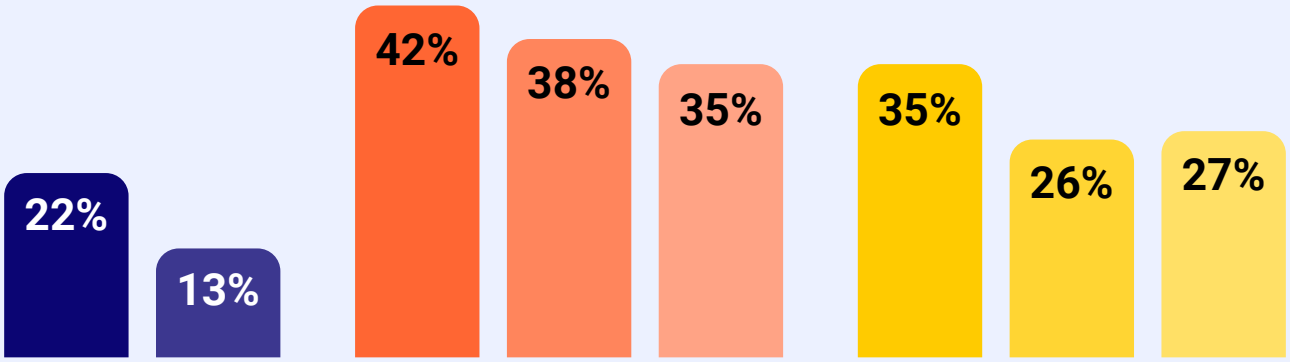
- On average, **69%** of developers say AI has **accelerated their career progression**
- But only **59%** of early-career developers agree, against **71%** of mid-career and **71%** of experienced developers
- Overall, **31%** believe AI is **negatively affecting their progression**

Attitudes to AI itself differ by stage too. 44% of mid-career developers would get rid of AI from the workplace if they could, compared with 31% of early-career developers. The group with the most experience of life before AI is the most inclined to say it would go back.



2c. The mid-career squeeze

Developers in the middle of the career ladder are experiencing the greatest anxiety about their progression. At the same time, they are taking on more responsibility for managing, scrutinising, and working alongside increasingly autonomous AI systems.



22% of mid-career developers deploy four to five AI agents for a task, compared with 13% of early-career developers

42% of mid-career developers have increased their scrutiny of AI-generated work following high-profile security incidents, compared with 38% of early-career and 35% of experienced developers

35% of mid-career developers say AI is negatively affecting their career progression, versus 26% of early-career and 27% of experienced developers

Early-career developers, meanwhile, are gaining less from AI in terms of progression. This raises a longer-term question about how the next generation will build the experience needed to move into senior roles.

Advice from Adaptavist

Organisations need to treat AI adoption as a talent development challenge as well as a productivity opportunity. Junior developers need meaningful opportunities to build foundational expertise, while mid-career and senior team members need support to develop the judgement and leadership skills required to manage increasingly autonomous systems. Only by supporting developers at every stage can businesses ensure that short-term AI productivity gains do not damage long-term talent development and retention.

CHAPTER 3

Entering the age of agents

Most developers now treat AI agents as teammates and run several at once, but managing them adds overhead and governance risk.

3a. The humanisation of AI

The data clearly shows that AI is no longer just a tool developers use. Increasingly, they are treating it as a member of their team.

- **64%** see AI as a teammate
- **57%** rely on it in the same way they would rely on a human colleague
- **43%** have named or customised an AI assistant or agent, or developed a personal attachment to one

For the majority of developers, AI has crossed a line from something they operate to something they work with, and in many cases something they feel attached to.



JOSEPH SHAMOON

Developer Desktop Engineer Adaptavist

CASE STUDY

"I see an AI agent as a teammate. It works alongside me on the same queue, it remembers things I've forgotten, and it hands me a draft I then edit. I spend more of my day managing agents than using them, which is a colleague relationship. The only difference is accountability. The agent never signs off on anything, I do.

My agents don't have names. I refer to them by what they do: the triage one, the drafting one. Naming them would imply they have a personality to manage. They don't—they have a context to manage.

Nearly every time I'm frustrated with an agent it's because I spoke to it badly or gave it the wrong context. It's less like being annoyed at a colleague and more like being annoyed at myself for writing a bad brief. Once you accept that the quality of the output is a mirror of the quality of the input, the frustration mostly goes away."

This shift is changing developers' relationship with their roles and responsibilities:

- **Over half (51%) feel they are increasingly managing AI rather than simply using it**
- **At the same time, 47% feel AI is replacing parts of their role**

Evidently, developers are gaining a collaborator and losing parts of their traditional role at the same time.



EDD BRISLEY

Software Engineer
The Adaptavist Group

CASE STUDY

"On a scale of 'spanner' to 'best man', AI agents are like wing-nuts or screwdrivers to me—a useful tool when used correctly.

Writing agents is just like writing any other code. The focus is on boundaries, not conventions. I still get frustrated when my screwdriver runs out of battery and makes mistakes, though.

I'm in an early cohort of devs who have never engineered without AI. I know that misusing AI can take learning opportunities from me, so my agents do work which I already understand well enough not to want to do myself. For example, the Changelog Agent monitors all apps, and proposes fixes when it spots upstream breaking changes."



ANAND UNADKAT

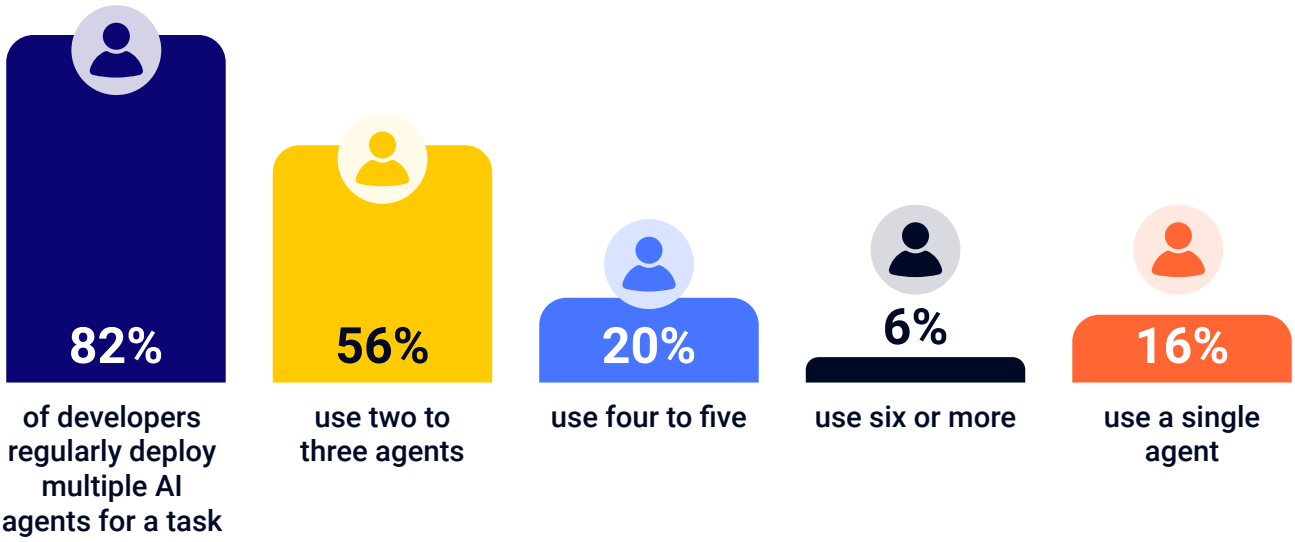
Senior Solutions Architect
Atlassian

"The shift from viewing AI as a passive tool to an active teammate marks a fundamental evolution in how knowledge work gets done. When nearly two-thirds of respondents see AI as a teammate and over 40% are already personalising and shaping their agents, it signals that effective AI adoption isn't just about raw compute or automation; it's about context, trust, and collaboration.

The most successful organisations won't just deploy tools; they will onboard and integrate AI agents alongside their human teams into everyday workflows."

3b. The multi-agent workforce

Multi-agent workflows are already standard practice, but they bring mounting tool sprawl and governance risks.



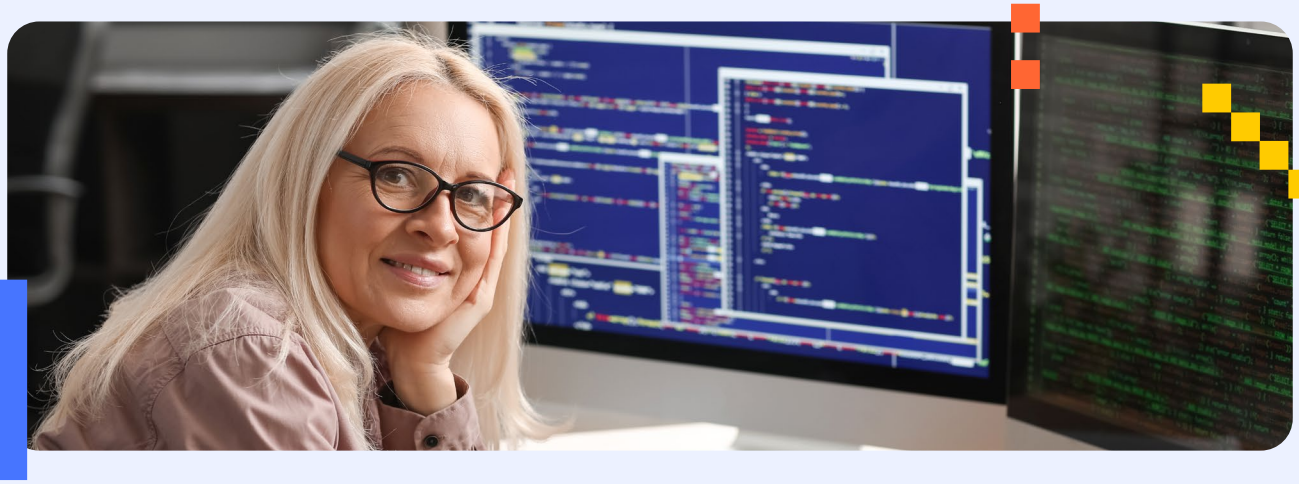
The single-agent developer is now the exception as managing agents becomes a core part of developers' day-to-day activities:

- **68%** say it is becoming an explicit **part of their daily role**
- **70%** believe it will become a **core software engineering skill**

And the benefits are real:

- **80%** say agents save significant **time**
- **79%** say they improve **productivity**

But the overhead is real too. 58% say managing AI tools creates additional operational overhead, and 57% say agents contribute to tool sprawl.



3c. Trusting agents

Following high-profile AI agent security incidents, developers are recalibrating how much autonomy they are willing to give agents.

Trust remains high: **66% trust agents to complete tasks autonomously,** and **60% trust them to make important technical decisions,** but confidence has been shaken:

39%
now review AI-generated work more closely

29%
require tighter human oversight

21%
are less willing to give agents autonomy

14%
explicitly trust agents less than before

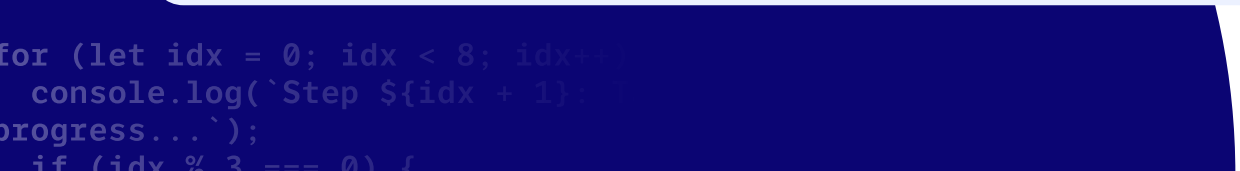


BEHN WILLIAMS
Field CTO
Chainguard

"Reviewing AI output more closely is the right instinct, but most reviewers are reading the wrong layer. The logic in the function is the easy part. The supply chain decisions the agent made on the way to that function are where the exposure sits, and they rarely get a careful read in a code diff.

There are four things I'd look for in every new dependency: does it exist, is it the package you think it is, who maintains it, and did the agent pin a version it remembers from training data that is two years stale? In one study, about one in five packages that AI models recommended didn't exist, and the same fake names came up run after run. That consistency is what makes them worth registering. Every FROM line and install step: the agent picks the most common base image it has seen, and popular images routinely carry hundreds of unpatched vulnerabilities. And everything the agent read while working.

A reviewer cannot catch all of that at agent speed, so don't make the reviewer the control."



Advice from Adaptavist

Understanding the dynamics of adding agents to workflows is key, and finding the best places to add them and how to best organise them and review their output is something that every engineering leader needs to figure out.

As agents become more capable and more numerous, developers need the skills and guardrails to orchestrate them effectively. The challenge is ensuring developers can judge when to delegate, when to intervene, and how to maintain oversight as autonomous workflows scale.

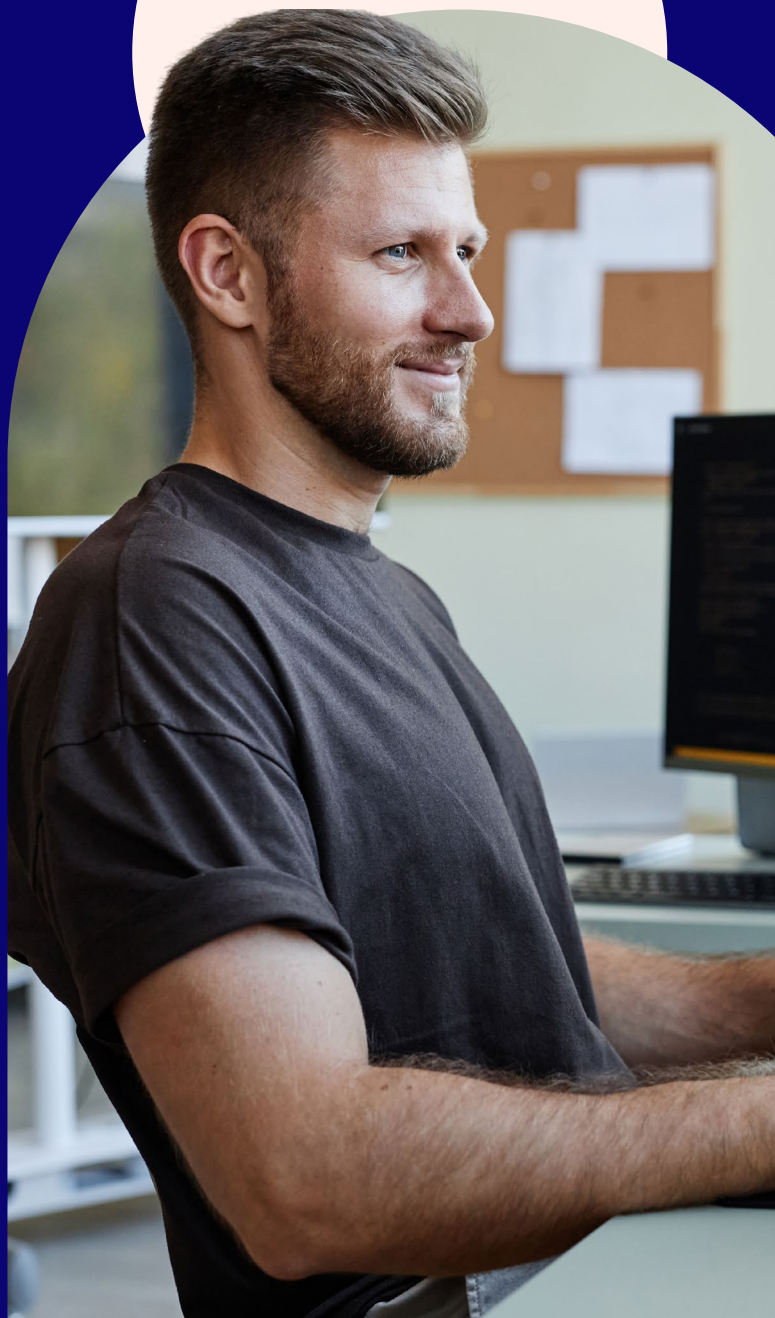


SERGIU-VASILE PETEAN

Field CTO EMEA
Atlassian

“Work itself is being redefined by AI, as well as individual roles. To keep pace with this change, organisations should deliberately design themselves to be less rigid and more open to experimentation, flexible topologies, and supporting personal growth.

Bringing AI next to HR, like Atlassian has, is one way to define every new, and old, role through its relationship with AI, supporting dev teams and beyond in this journey.”



```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++)
        printf("%d\t", i * j);
```

Conclusion

The attitudes and challenges of today's developer workforce show that AI is not the source of their problems. The way many organisations are introducing it is.

Across every chapter, the same paradox emerges. Developers overwhelmingly say AI has made them more productive, and that its benefits outweigh its drawbacks. Yet the conditions around its use are creating pressure and a growing sense of lost autonomy. Usage-based KPIs and performative behaviours such as tokenmaxxing are rewarding the look of adoption rather than the substance of value delivery and good software. Developers know the difference. They want quality measured, and they want time to solve problems properly.

The research also shows that AI is not only changing what developers do, but who they are. Early-career developers are learning less through hands-on coding, mid-career developers are carrying the heaviest load of scrutiny and anxiety, and experienced developers are being asked to oversee systems that grow more autonomous by the month. Meanwhile, developers increasingly treat AI as a teammate and spend more of their day managing agents. That is a real shift in the nature of their role, and it is happening faster than most organisations have adapted their team structures, skills, and support to address. Together, these pressures are making software development a less attractive career for a significant proportion of developers. The data does not point to an exodus yet, but it does provide a warning that should be taken seriously.

The encouraging news is that developers are not asking organisations to slow down. What they ask for is practical: clear expectations, room to build and apply their expertise, and guardrails they can trust as agents take on more. Where those conditions exist, AI becomes an enabler rather than a source of friction. The difference is not technological capability, but organisational intent.

Only 24% of organisations formally measure AI usage against business outcomes. Starting there, and asking developers what quality means in their own teams, is the clearest next step.



Adaptavist

About Adaptavist and The Adaptavist Group



At Adaptavist, we help organisations remove friction, unlock value, and achieve long-term, exponential growth.

Founded in 2005, we're a specialist business and technology consultancy and a trusted partner to Atlassian, monday.com, AWS, GitLab and many more. Our expert consultancy and practical delivery connects technology, people, and ways of working with powerful tailored solutions that simplify complexity and achieve sustained business outcomes.

Experts in cloud, DevOps, service management, work management, AI, and agile, we are the pioneer brand of The Adaptavist Group, a family of companies whose team spans over 1,000 employees, with a 22,000+ customer base representing more than half the Fortune 500. We combine the best talent, technology, and processes to make it easier for our customers to excel–today and tomorrow.

Contact us: **hello@adaptavist.com**