MAKERS

FOR THE TRANSITION

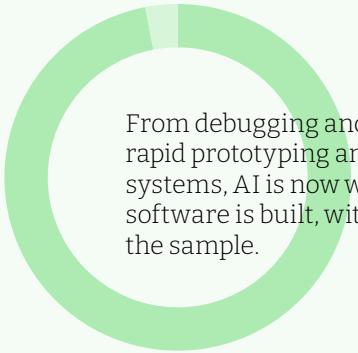
The AI leadership challenge – deep dive on engineering

INSIGHTS FROM THE 2026 MAKERS AI ENGINEERING SURVEY

EXECUTIVE SUMMARY

High adoption, low orchestration

In 2026, the question for engineering leaders is no longer whether their teams use AI, but if they are using it for the right things. Having guided tech careers for over 13 years, we at Makers wanted to understand how AI is shaping day-to-day engineering work. With a community of 5,000+ alumni - representing organizations from **agile startups to global enterprises** - the results of our survey make it clear; AI use is ubiquitous.



From debugging and code explanation to rapid prototyping and learning unfamiliar systems, AI is now woven into how modern software is built, with **97%** adoption across the sample.

But while adoption is widespread, progress is uneven.

Our research reveals a landscape where engineers are moving faster as individuals, but organisations are struggling to turn that momentum into collective advantage. For around half of respondents, AI adoption is happening **with limited strategic direction from senior leadership**, emerging instead through independent experimentation or small informal groups. In other words, much of today's AI use is happening in the shadows of formal strategy, without clear guidance on where and how it should be applied. The result is a paradox: high levels of activity, but shallow organisational learning.

> **AI is still mostly used as a faster autocomplete instead of a true engineering partner that understands system context, tradeoffs, and long term ownership.**
>
> Survey Respondent - Software Engineer, Mid-Market

This matters because AI is not simply an efficiency tool. As productivity increases, so does ambition. The data shows that instead of using the saved time to take on more complex problems, developers are often pulling back due to hesitancy around acceptable AI usage. Or worse; being forced to dedicate more time to tackling downstream problems caused by unstructured AI use.

Engineers are also making deliberate choices about where not to use AI. High-stakes areas such as system architecture, security, and critical decision-making remain firmly human-led for a significant proportion of respondents. This restraint reflects professional judgment and accountability - and highlights where trust, governance, and leadership clarity are still missing.

Taken together, the data points to a clear conclusion: while some engineers remain unconvinced by the benefits of embracing AI, the main blocker to AI-enabled productivity is not technical. It is organisational. The challenge facing engineering leaders is understanding how AI is actively being used and then building a roadmap to support the right people to use it in the right ways.

ADOPTION IS WIDESPREAD, PROGRESS IS UNEVEN.

THE FRONTLINE REALITY

How developers are choosing tools in practice

Before examining behaviour in detail, we wanted to understand something simpler: what tools are engineers actually using?

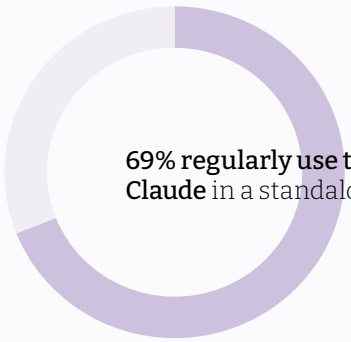
Tooling is often the first practical signal of AI maturity. It shows whether adoption is coordinated or improvised, standardised or self-directed.

Across the dataset, OpenAI models remain the most widely used overall. In many enterprises this may also reflect existing platform ecosystems - for example, OpenAI access bundled through Azure - rather than purely bottom-up developer preference. But when you cut by organisation size, a more nuanced picture emerges. In enterprises (5,000+ employees), usage consolidates: **62% of enterprise respondents use OpenAI**.

In scale-ups (51–1,000 employees), that dominance softens, and the spread widens. Scale-ups use a wider mix of models. **59% of scale-up respondents use Anthropic Claude**, more than double the enterprise rate. Gemini appears in both segments at similar levels, but without clear dominance in either. In some organisations, these choices are also shaped by the broader technology stack already in place, with models often adopted through existing cloud or tooling ecosystems. Larger organisations standardise. Smaller ones experiment.

If model choice shows how organisations think about AI, tooling shows how engineers actually use AI day to day.

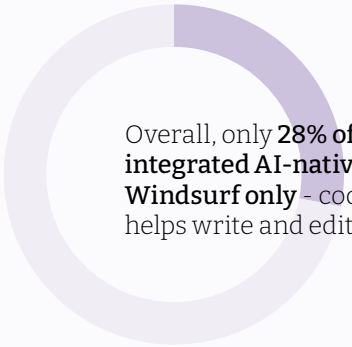
Most engineers still rely on browser-based tools.



69% regularly use tools like ChatGPT or Claude in a standalone web interface.

More than half also use IDE-integrated assistants such as Copilot or JetBrains AI. The dominant pattern is layered rather than singular - engineers move between tools, combining conversational reasoning in the browser with in-editor suggestions.

Where the divide becomes clearer is in AI-native development environments.

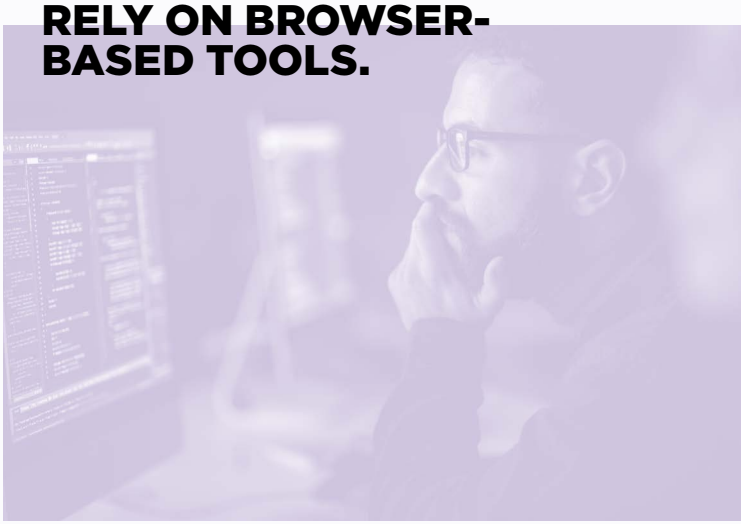


Overall, only **28% of respondents use integrated AI-native IDEs like Cursor or Windsurf only** - coding tools where AI helps write and edit software.

But when you look at company size, the difference is striking. In scale-ups, 39% have moved to AI-native environments. In enterprises, that figure drops to 10% – a reflection, perhaps, of the relative ease with which smaller companies can change their development environments.

This difference reflects how easily organisations can change their tooling. Larger companies tend to add AI into existing editors through approved plugins, keeping their established development environments intact. Smaller companies are more willing to switch tools entirely and build their workflows around AI from the outset.

MOST ENGINEERS STILL RELY ON BROWSER-BASED TOOLS.



WHERE ENGINEERS DRAW THE LINE

How professional judgment shapes AI use

Once you move beyond tooling, a deeper question emerges: where do engineers trust AI - and where do they stop?

The value of AI shifts significantly as engineers move from tactical delivery into technical leadership. For most Software Engineers, AI is primarily used to reduce friction in delivery. **81% use it to decipher or fix existing code**. At this level, the tool helps manage cognitive load. It shortens the time spent navigating unfamiliar logic and speeds up day-to-day execution. It acts as a filter for complexity, allowing developers to spend less time lost in unfamiliar logic and more time shipping features.

As responsibility increases, the way AI is used begins to broaden. Among Seniors and Tech Leads, 71% use AI to research best practices, and 57% use it to learn unfamiliar frameworks. The emphasis expands beyond fixing code to include exploring best practices and learning unfamiliar frameworks.

One respondent captured this shift:

> **"I would like to see more people using AI to teach them things or to advise on best practice, rather than simply doing work for them."**
>
> Survey Respondent – Tech Lead / Principal Engineer, Enterprise

But across roles, a consistent boundary appears around high-risk work. Engineers are explicit about where they will not use AI, particularly in areas involving security, privacy, and sensitive system context. This distinct set of "No-Go" zones reveals a shared professional boundary that is most visible where the stakes for the business are highest.

68% of engineers actively avoid using AI for security- or privacy-sensitive work.

49% refuse to use AI for tasks requiring deep domain context, acknowledging it cannot replace an understanding of a system's unique history.

27% draw the line at architectural or system design decisions.

This suggests that Engineering is not yet ready to outsource "judgment." While AI is trusted to generate syntax, it is not trusted to oversee the integrity of the system. This collective avoidance suggests that engineers view AI as an assistant for the practical parts of the job - writing and fixing code - but see architectural and security decisions as a matter of professional accountability that cannot be delegated to a model.

What is the root of this hesitation? The primary barrier to deeper adoption is not a lack of interest, but a calculated assessment of risk.

28% of respondents cite a fundamental "lack of trust in outputs" as their main reason for holding back.

25% are specifically wary of "subtle errors or hallucinations" that could bypass a standard review process.

24% point to "accountability and professional responsibility," refusing to put their name to work they did not fully author or cannot fully verify.

Nearly 80% of practitioner hesitation is rooted in the fear of silent failures. This is why confidence remains stuck; engineers feel they have the permission to use these tools, but they lack the guidance and support required to trust them in high-stakes situations. The survey suggests most engineers are not resisting AI itself - they are weighing the risks around where and how it should be used. They use it where speed helps and mistakes can be spotted quickly. But when security, architecture, or long-term system health are involved, they slow down and rely on human judgment.

"I don't like that many models seem to be goal oriented heavily toward just making things work at the expense of correctness."

Survey Respondent – Software Engineer, Mid-Market Org

THE JEVONS PARADOX

When efficiency expands the scope of what gets built

In the 19th century, economist William Stanley Jevons observed that when coal engines became more efficient, the expectation was that Britain would use less coal. Instead, it used more. Because each ton of coal could power more work, coal became more useful - and demand increased.

A similar pattern is emerging in engineering.

AI has dramatically reduced the cost of writing code. Generating functions, refactoring modules, scaffolding features - all of it is faster.
The friction of execution has dropped.

But efficiency does not automatically create increased output.

62% of engineers report "noticeable time savings" in their weekly work

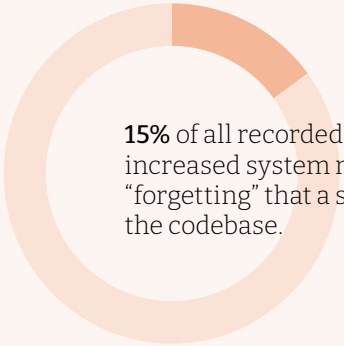
Yet only **10%** of engineers are using that extra capacity for high-level planning, estimation, or design work

This suggests a disconnect between the "time saved" and the "value added." Much of the reclaimed time appears to flow back into delivery work rather than higher-level planning or design.

At the same time, frustration signals are rising. Several respondents describe AI creating redundant logic or "re-inventing the wheel." One engineer put it plainly:

> **"AI often likes to 're-invent the wheel' and write new components/utils instead of reusing existing ones (because it literally forgets that they exist)"**
>
> Survey Respondent, Software Engineer, Mid-Market Org

Further, the ease of generation is creating a specific kind of technical debt where AI models, lacking a full view of the existing system, create redundant solutions.



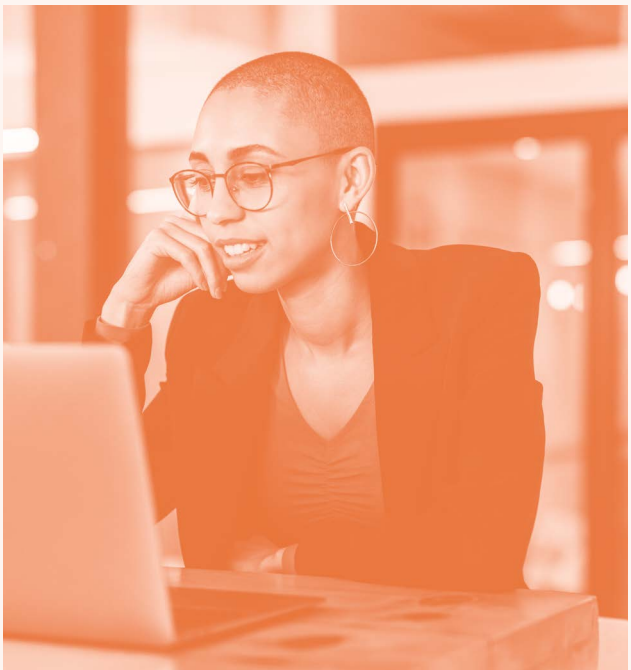
15% of all recorded frustrations relate to increased system messiness or the AI "forgetting" that a solution already exists in the codebase.

When generation becomes easy, duplication becomes easier too.

As code becomes easier to generate, the constraint shifts from writing code to maintaining system coherence.

Without intentional reallocation of effort, time savings risk reinforcing the ticket treadmill rather than freeing engineers to shape the system itself. If routine implementation becomes easier, engineering expertise becomes more valuable higher up the stack - in architecture, integration, and solving complex system problems.

AI can accelerate delivery. What remains unclear is how organisations choose to use that speed.



WHERE AI IS REALLY TAKING HOLD

Individual use, limited oversight

AI adoption inside engineering teams has largely evolved through practice rather than policy. In many organisations, usage patterns are emerging faster than formal strategy. That decentralisation is visible in the data.

THE EVIDENCE: INDIVIDUAL INITIATIVE VS. FORMAL INTEGRATION

48% of engineers report that AI usage is driven either by individuals experimenting independently or by a small number of motivated engineers informally influencing their peers.

In contrast, only **6%** of respondents work in an environment with a dedicated enablement or platform group to guide usage.

46% of respondents state that the impact of AI is not yet formally measured or discussed within their organisation.

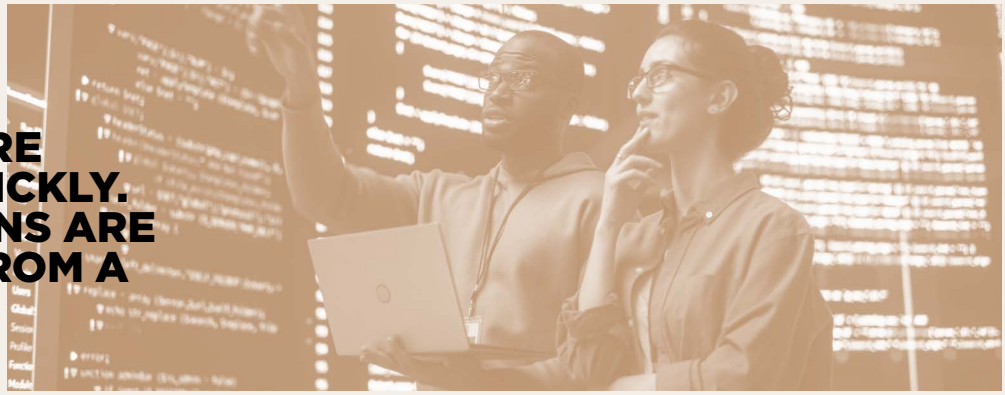
The picture is not one of resistance. It is one of decentralisation.

This raises a practical question: if AI has reduced the cost of building routine features, internal automations, and lightweight tooling, how should organisations rethink where that work sits? As the barrier to building smaller tools falls, organisations may begin to reassess which work truly requires engineering input. It also raises a leadership question: how do organisations move from engineers figuring this out in practice to leadership directing it with intent?

In many teams, those tasks still flow into the engineering backlog by default. Yet as AI lowers the technical barrier, some of this work begins to resemble operational implementation rather than deep systems design. Without visibility into how effort is shifting, it becomes harder to tell which tasks truly require engineering judgment - and which could reasonably sit elsewhere.

Over time, some of the smaller internal tools or automations that currently sit with engineering may become accessible to a wider set of teams. But without clear ownership or direction, most of this work continues to land where it always has.

ENGINEERS ARE ADAPTING QUICKLY. ORGANISATIONS ARE OBSERVING FROM A DISTANCE.



BEYOND EXPERIMENTATION

Why engineers need structure to go further

Engineers have successfully woven AI into their daily habits. But the "No-Go Zones" are established, and optimal AI workflows are still emerging. Where are engineers asking for help? The picture painted by our survey is one where many engineers feel they have reached the limit of what can be achieved through individual initiative and the primary bottleneck is lack of organisational support.

CONFIDENCE SNAPSHOT:

When asked to rate their confidence in steering AI to achieve complex, specific results, the most common response (41%) is a **3 out of 5**.

Only **13%** of the workforce feels highly confident (a score of 5) in their ability to direct these models through professional-grade, production-ready logic.

At a tactical level, engineers are comfortable. They can generate code, refactor functions, explore new libraries. But as complexity increases - legacy systems, cross-team dependencies, production risk - confidence drops.

When asked what would most help them deepen their use of AI, engineers were clear. The top response was stronger prompting and steering skills, selected by 53% of respondents.

Close behind, **49% asked for real-world examples** of how AI can be applied inside engineering workflows.

The signal is clear: engineers are asking less for new tools and more for clearer examples of how AI should fit into established engineering practices.

Time and structure also surfaced. 39% said dedicated time to experiment safely would help them go further, while 30% wanted a better understanding of AI's limitations and risks. **Fewer asked for new tools or access.**

The signal is practical. Engineers are not waiting for the next model release. They are looking for guidance, examples, and the space to integrate AI safely into the systems they already maintain.

One engineer described frustration with:

> **"Its use is assumed, and its fallibility underestimated. Meaning that high level management set deadlines assuming use of AI, but expect the reliability of it being done manually and properly. "**
>
> Survey Respondent – Software Engineer, Enterprise Org

When engineers are left to figure out integration "safely" on their own, they naturally gravitate toward low-risk, tactical use cases. This creates a disconnect: management expects transformative shifts in velocity, but because there is no architectural "safety net" or guided policy, engineers remain tethered to the tasks they know they can verify - debugging and boilerplate.

ENGINEERS NEED CLEARER EXAMPLES OF HOW AI SHOULD FIT INTO ESTABLISHED ENGINEERING PRACTICES.

CONCLUSION

What becomes possible when AI is directed

Even in an environment where adoption is largely organic and uneven, many engineers are already reporting meaningful time savings. 32% of engineers report saving **between four and ten or more hours each week** through the use of AI. Within that group, **15% are already reclaiming the equivalent of a full working day** every week or more.

In mid-market firms (201–1,000 employees), the shift is even more pronounced. 64% report these kinds of time savings. These reported gains sit alongside the tensions described earlier. AI can accelerate individual tasks even when its broader impact on systems and workflows remains uneven.

And yet many organisations are not measuring these changes. **46% of respondents say the impact of AI is not formally tracked or discussed within their company.**

Which raises an obvious question: if engineers are already reclaiming meaningful time every week, what is happening to that capacity?

In many teams, the answer appears to be simple. It is absorbed back into delivery.

More tickets. More features. More code.

But many engineers see a different possibility emerging. When asked about the biggest missed opportunity:

> **"Less time spent on boilerplate code, more on design and high-level concepts."**
>
> Survey Respondent – Software Engineer, Enterprise Org

> **"Companies focus on accelerated delivery rather than the cognitive space it gives engineers to think about architecture and deeper problem areas."**
>
> Survey Respondent – Software Engineer, Mid-Market Org

The shift is subtle but important. As AI lowers the cost of generating code, the value of engineering work begins to move upward - away from routine execution and toward system design, integration, and oversight.

That transition does not happen automatically.

Several respondents pointed to the same missed opportunity: organisations are encouraging AI use but not yet defining how that reclaimed time should be reinvested. Another possibility is:

> **"Developing strong tools that automate smaller focused areas of work as a consistent guided way of working, rather than making big changes happen without guidance or policy."**
>
> Survey Respondent – Software Engineer, Start-up

Others pointed to an even broader opportunity: expanding who participates in building internal tools and automation.

> **"We are failing to use AI as a 'Red Team' partner for experts to stress-test complex decisions, choosing instead to use it as a shortcut for non-experts."**
>
> Survey Respondent – Software Engineer, Enterprise Org

In other words, AI is already increasing technical capacity. What remains undecided is how that capacity should be used.

Some work that currently sits in engineering backlogs could reasonably move elsewhere as AI lowers the barrier to building smaller tools and automations. At the same time, the engineering function itself can move further into the work that matters most: designing resilient systems, managing technical risk, and guiding the architecture of increasingly complex platforms.

The opportunity is not simply to build faster.

It is to rethink where human judgment adds the most value.

Because the time dividend already exists.

The organisations that recognise it - and design around it - will be the ones that turn individual productivity gains into lasting advantage.

Is your team experimenting – or is it evolving?

LET'S START A CONVERSATION.
MOVE YOUR TEAM OUT OF THE SHADOWS.

BOOK A STRATEGY SESSION

contact@makers.tech
0203 817 8870

in @makers-academy
www.makers.tech

MAKERS
FOR THE TRANSITION

Certified B Corporation