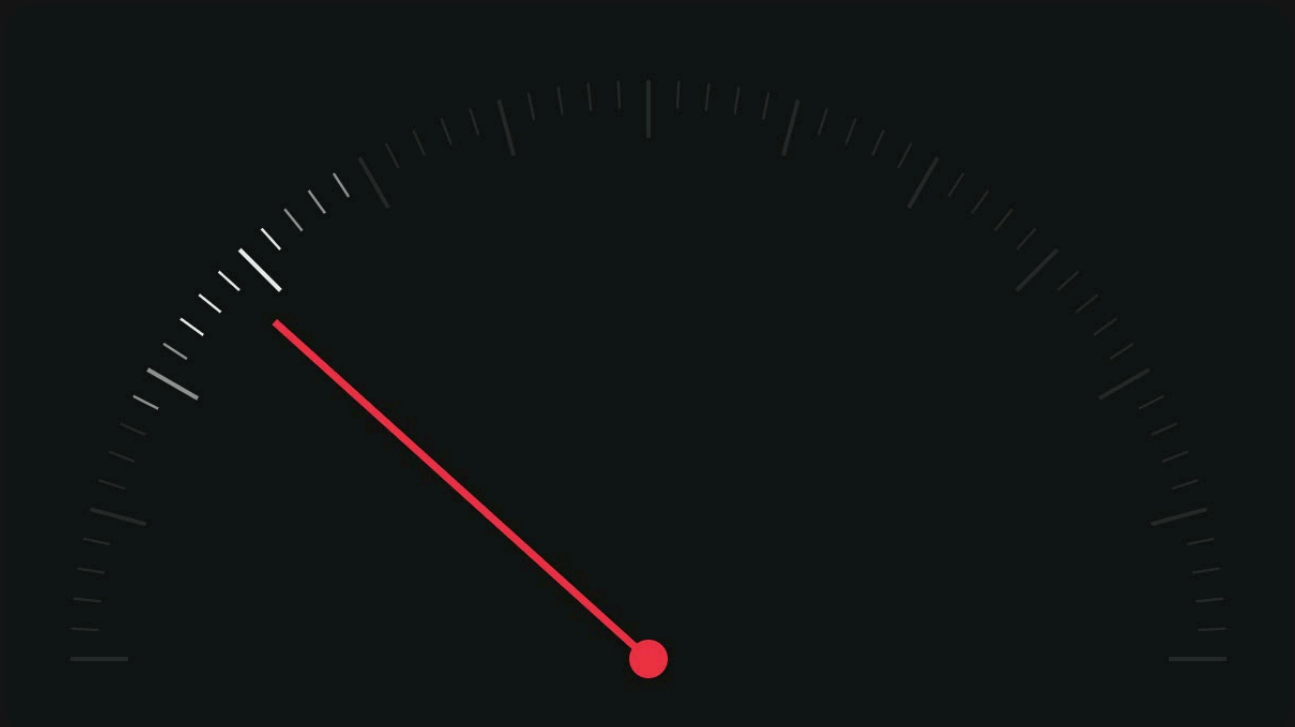


The Invisible Dial

How your engineers' daily AI choices shape what your company can still build.

WRITTEN BY
Tim Van Gehuchten, Director Product & Technology

PUBLISHED
NOV 2026

READING TIME
26 min read

CONTENTS

	The short version	03
01	Executive summary	04
02	Everyone adopted. Nobody decided.	05
03	The invisible dial, and how it compounds	05
04	Why the standard responses fail	09
05	What we do about it	11
06	What this buys you	15
07	Seven tests for any approach, including ours	16
08	Conclusion	19
09	References	21

THE SHORT VERSION

Two minutes, if you don't have twenty–six

01 Every AI–assisted task involves two choices. The first is what AI does for the work: make familiar work faster (an accelerator) or make new work possible (an amplifier).

02 The second is what happens to the engineer's judgment. Did they frame the problem, challenge the output and own the result, or hand that to the tool? We call this the invisible dial, because the diff shows what someone produced but not how much judgment they kept.

03 People misjudge what AI does for their own performance. In a randomized trial, 16 experienced open–source developers were 19% slower with AI while believing they were 20% faster.

04 Policies, code review, training and dashboards each look at an artifact, an output or an event. None of them shows both what AI added and how much judgment the person kept.

05 The paper sets out five habits from November Five's engineering teams and holds them to seven tests. They pass two. The tests ask whether a team keeps its own judgment and skills over time. They don't grade the software it ships. Test 7 is a fail we call useful: a monthly forum that decides which AI tools stay and which go, and had never asked what the team got worse at.

06 You can start all of it this month, without new tooling. The conversations are the harder part.

THE EVIDENCE · EARLY SIGNALS, NOT A VERDICT

90%

of DORA's[5] nearly 5,000 respondents said they used AI at work in 2025.

19%

slower with AI in METR's trial[1] of 16 experienced open–source developers, who believed they were 20% faster.

17

points lower on comprehension for AI–assisted junior engineers in Anthropic's experiment[2].

60%

of 883 LeadDev[8] respondents named the lack of clear AI–impact metrics as a challenge.

01

Executive summary

Faster or further is only half the choice. The other half decides what your team still knows.

Every AI-assisted task comes with two choices. First, what does AI do for the work? It can make familiar work faster, or it can make work possible that was previously out of reach. At November Five we call those accelerator and amplifier.

Then there's the engineer's judgment. Did they frame the problem, challenge the output, and own the result? Or did they hand that judgment to the tool? The first choice changes the work's reach. The second changes the direction of your team's capability. That second choice is the invisible dial. Push that dial too far toward offloading, and you get de-skilling. The team's capability narrows quietly while the work still ships.

Why call it invisible? Your normal workflow doesn't record it. The diff shows what someone produced, but it can't show how much judgment they kept. You can ask the engineer, but even they may misjudge what AI did for their performance. In a randomized trial[1], 16 experienced open-source developers were 19% slower with AI while believing they were 20% faster.

How quickly can skills fade? To be honest, we don't know. The evidence gives us warning horizons, but no universal clock. In one randomized experiment[2], AI-assisted mostly junior engineers scored 17 points lower on comprehension. In another expert domain, an observational study[3] associated roughly three months of routine AI exposure with a lower unassisted detection rate among endoscopists. Neither result predicts what will happen to your engineering team, but both are good reasons to measure before something breaks.

Can policies, code review, training, or dashboards catch this? Each misses part of the picture. They may not record the judgment setting. They may miss how thousands of small choices add up. And they usually tell you something after the fact. In a BCG survey[4] of 70 senior leaders, half said they were already observing de-skilling. I haven't found a validated maturity model for it. What I can share is more modest: five habits our engineering teams practice, one signal you already own, and two problems we haven't solved.

How to read the evidence. This evidence is early. No study yet measures long-term skill atrophy in software engineering. This paper triangulates warning signals. It doesn't provide a single causal proof.

02

Everyone adopted. Nobody decided.

The adoption question is largely settled. In 2025, 90% of DORA's[5] nearly 5,000 respondents reported using AI at work. Separate developer surveys put use or intended use at 84% for Stack Overflow[6] and regular coding use at 85% for JetBrains[7]. Yet 46% of Stack Overflow respondents distrusted AI output. Plenty of people use AI, but that doesn't mean their teams have decided how to use it well.

Boards tend to push in one direction: adopt faster. They ask about adoption rates, license coverage, and whether competitors are further along. Those are fair questions, but what about the next two? Did AI only make existing work faster, or did it make new work possible? And where did human judgment stay? Those questions are easy to skip when nobody makes them part of the work.

Your usual instruments don't help much. The JetBrains survey[7] asks which tasks developers hand to AI. It doesn't ask whether that work was previously out of reach or how much judgment the developer retained. In LeadDev's survey[8] of 883 respondents, 60% named the lack of clear AI–impact metrics as a challenge. The two choices are still being made across thousands of tasks. You simply can't see them yet. They shape what your team can build and what it can still understand.

03

The invisible dial, and how it compounds

Two axes, one invisible dial

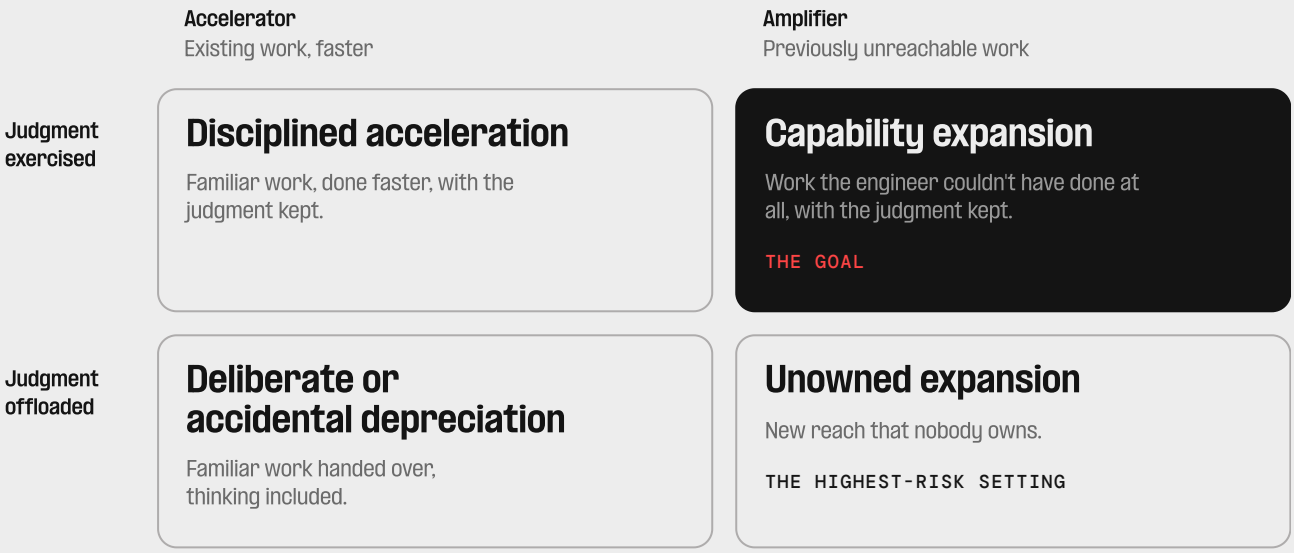
The first axis is about reach. At one end, AI is an accelerator. It does work your engineer could already do, only faster. Think boilerplate, a routine migration, or a familiar CRUD feature. At the other end, AI is an amplifier. It makes work possible that the engineer couldn't have done at all. One example is analyzing a codebase at a scale no single person could reach.

The second axis is about judgment. Does the engineer frame the problem, challenge the output, make the decisions that matter, and own the result? Or does the tool do that thinking while the engineer passes the answer along? That's the invisible dial. The artifact only shows what was produced. Ask while the work is still fresh, and you can surface how judgment was divided. You still need a separate check to know what the engineer can do.

FIGURE 1 · THE DIAL

Two axes. The artifact records neither.

Reach runs left to right: existing work made faster, or work that was out of reach. Judgment runs top to bottom: exercised by the engineer, or offloaded to the tool.



Quadrant names and lines from section 3.

Is acceleration bad? No. Does amplification automatically build capability? No. An engineer can use AI for a familiar migration and still understand every trade-off. Someone else can create an unfamiliar system and be unable to explain why it works.

Offloaded judgment, repeated often enough, is what de-skilling looks like in practice. A capability nobody exercises stops being available when you need it.

We use amplifier for the reach of one task. GitKraken[9] uses the term at the level of the organization and argues that AI magnifies the practices already in place. The two meanings lead to the same questions. What did AI add to the work? And what did the work do to your people?

Keep handing over the same judgment, and you lose chances to practice it. Will that lead to measurable capability loss? We don't know. In a survey of 319 knowledge workers[10], higher confidence in generative AI was associated with less self-reported critical-thinking effort. The study measured what people said about their behavior during AI-assisted work. It didn't measure long-term skill loss.

Unrecorded in the artifact, and easy to misjudge

Can people judge how much AI helps them? Not very well, it seems. In a randomized trial[1], 16 experienced open-source developers completed 246 tasks 19% slower with AI while believing they were 20% faster. Two logical-reasoning studies[11], with 246 and 452 participants, found a

similar gap. AI users overestimated their performance, and higher AI literacy correlated with lower self-assessment accuracy. Asking people how much AI helped is useful. It just isn't enough.

By default, an organization can see very little. An Aalto accounting case[12] found that routine automation hid capability loss until the system stopped. It shows you one way this can go wrong, but it doesn't tell you how common that failure is. A separate cross-sectional study of 666 UK participants[13] associated higher reported AI use and cognitive offloading with lower critical-thinking scores. The researchers selected participants based on availability and specific characteristics. Combined with the study's observational design, that means the findings cannot show that AI caused the differences.

The aggregation runs on three horizons

One choice may leave no visible trace, but thousands of them can add up before anything breaks. How much time that takes, nobody knows yet.

An observational study of 19 experienced endoscopists[3] gives us one warning. Adenoma detection during non-AI colonoscopies fell from 28.4% before routine AI implementation to 22.4% after roughly three months. This was a clinical study, and it was observational. Treat it as a warning from another expert domain. It gives us no engineering decay rate.

Workforce surveys show that people see both missed upside and risk. Among 15,000 employees across 29 countries[14], 88% reported using AI, mostly for basic tasks. Only 5% said they used it to "transform" their work, and 37% worried about skill erosion. In a survey of 70 senior leaders[4], half said they already observed de-skilling. More than 60% expected it to become material within three to five years. These are perceptions. They aren't a measured engineering curve.

Think in three horizons:

HORIZON	WHO	WHEN IT SHOWS	MEASURED
Task	One assignment	Immediately	**Every time**
Signal	One person	After one task, or months	**Two studies**
Organizational	The company	Varies	**Not yet**

The task horizon is immediate. Every assignment divides judgment between person and tool, whether anyone records that division or not.

The signal horizon can be immediate or emerge over repeated use. Anthropic found a comprehension gap after one programming task. The Lancet study observed lower unassisted performance after months in a different expert domain. Neither result predicts an engineering team's rate of change.

The organizational horizon spans repeated planning, hiring, mentoring, and delivery decisions. Its length will vary by capability and context, and it hasn't been measured.

Who each horizon touches, when it shows, and how far it has been measured, from section 3.

So when should you test? We can't give you a magic interval. Establish a baseline now and inspect it regularly. Don't wait for failure to become your first measurement.

The junior pipeline intersects learning and hiring

What about juniors? A small randomized experiment[2] gives us the clearest short-term learning signal. Among 52 mostly junior engineers completing one task, the AI-assisted group scored 17 points lower on comprehension. The gap was widest in debugging. They finished about two minutes faster, though that difference was not statistically significant. Under those conditions, assisted completion and learning moved in different directions. The experiment can't tell us whether that becomes long-term atrophy. In LeadDev's survey[8], 38% agreed that AI had reduced direct mentoring for juniors. Another 28% disagreed.

There may also be fewer juniors entering the work. One study[15] covered 285,000 US firms. Junior employment was 7.7% lower after six quarters among firms classified as GenAI adopters, mainly because of slower hiring. The study used a posted GenAI-integrator role as its adoption proxy. Adoption wasn't random, and the authors call the result suggestive.

SignalFire[16] reported that new graduates made up 7% of Big Tech hires. It also pointed to tighter budgets and post-pandemic overhiring as possible causes alongside AI. Stanford[17] later found employment among

22–to–25–year–olds in AI–exposed occupations 19% below the path of less–exposed peers. The difference was driven mainly by hiring, and the researchers explicitly call the finding descriptive rather than causal.

These studies don't prove one neat causal chain, but they show two risks that may reinforce each other. People may get fewer chances to learn inside the work, while fewer people enter through junior roles.

There's a fair objection. Csaba Okrona[18] argues that search engines didn't erase cognitive work. They moved it toward synthesis, architecture, and judgment. We think he's largely right, but shifted, not shrunk assumes the judgment moved to a new level. Did it? Or did we hand it over? That's what the invisible dial asks.

We don't think an individual engineer can solve this through discipline alone. The artifact hides how judgment was divided, and people can misjudge their own performance. Multiply that across a team, and it becomes a leadership problem. Ask about the dial while the work is fresh, then check the capability separately. Most leadership teams start with one of five familiar tools. Each misses something.

04

Why the standard responses fail

Leaders usually do one of five things. They wait, set a policy, rely on code review, offer training, or watch the dashboards. Why do those responses fall short? They tend to miss at least one part of the problem. The judgment setting isn't recorded. Thousands of individual choices add up, and visible failure arrives late.

Doing nothing

Doing nothing is still a decision. No study can tell you when an engineering capability will cross a visible threshold. Does that uncertainty make waiting sensible? We don't think so. The Aalto accounting case[12] shows how normal operation can hide a loss until the automation stops. And a junior you neither hire nor train won't become your experienced senior later.

Tool mandates and policies

Policies tell people which tool they may use. They can even tell people to challenge the tool or forbid them from delegating certain decisions. Can the policy show you what happened during the work? No. DORA warns[19] that an unclear stance can push developers to hide their usage, known as shadow AI, or avoid the technology altogether. Shopify's published mandate[20] for "reflexive AI usage" has a similar limit. It tells people that they should use AI. It doesn't tell you what AI added or how much judgment the person kept.

Relying on code review

The diff doesn't tell you how it was produced. A fully delegated pull request can look exactly like one the author challenged line by line. Review may catch the "almost right" output that frustrates 66% of Stack Overflow respondents[6], but can it show whether the author understands the decisions that matter? Not always. Okrona calls the gap between generation and validation verification debt[18]. Unexamined delegation is one way that debt can grow. There are others.

FIGURE 2 · SAME DIFF

The diff doesn't tell you how it was produced.

Two pull requests for the same change, both approved. One author challenged the output. The other accepted what the tool generated.

PULL REQUEST 1 — APPROVED

```
  async function charge(req: ChargeRequest) {
-     return gateway.post('/charges', req);
+     return withRetry(
+       () => gateway.post('/charges', req, {
+         idempotencyKey: req.orderId,
+       }),
+       { attempts: 3, backoffMs: 200 },
+     );
  }
```

WHAT THE DIFF DOESN'T RECORD

Framed it first: a retried charge must never bill a customer twice.

Rejected the first suggestion, which retried without an idempotency key.

Can walk a colleague through the decisions that matter with the repo closed.

JUDGMENT EXERCISED

PULL REQUEST 2 — APPROVED

```
  async function charge(req: ChargeRequest) {
-     return gateway.post('/charges', req);
+     return withRetry(
+       () => gateway.post('/charges', req, {
+         idempotencyKey: req.orderId,
+       }),
+       { attempts: 3, backoffMs: 200 },
+     );
  }
```

WHAT THE DIFF DOESN'T RECORD

Asked the tool to add retries to the payment client.

Accepted the first suggestion that passed the tests.

Asked why a retry can't charge twice: not sure.

JUDGMENT OFFLOADED

The code didn't change. Review "may catch the 'almost right' output", but it can't always show "whether the author understands the decisions that matter."

An illustration of section 4, "Relying on code review". The two authors are invented.

Training programs

Training is an event. These choices happen inside the work, day after day. A course can teach people to use AI better. That's useful, but the course can't show you how they divide judgment afterward. In EY's workforce survey[14], 12% of employee respondents said they received sufficient AI training. Close that gap, and then pair the training with observation and some form of capability check.

Productivity dashboards

Output can rise while people exercise judgment. It can also rise while they hand judgment over. Your dashboard can't tell the difference, and it can't distinguish faster work (acceleration) from new reach (amplification).

The evidence around delivery is mixed too. In 2024, DORA[21] associated higher adoption with lower throughput and stability. In 2025[5], it linked higher adoption to better throughput and product performance, while stability remained lower. Faros[22] is an analytics vendor. Its data from 22,000 developers associated higher adoption with larger pull requests, more bugs, and slower review. That telemetry is observational, not randomized.

Neither source can tell you who kept the judgment. If you wait for delivery or quality to deteriorate, the damage becomes your first measure of capability. That may be too late.

The pattern

Do you see the pattern? Each response looks at an artifact, an output, or an event. None can tell you both what AI added and how much judgment the person kept. A useful approach needs to surface those choices, check the capability, and help leaders act before something breaks. I haven't found a framework that has demonstrated all of that. So here's what we do with our teams, and where it falls short.

05

What we do about it

I lead product and technology at November Five, and I've read many of the reports and proposed responses to AI-related de-skilling. I understand the appeal of a maturity model with five stages and a scorecard: it feels like something you can buy and install. We chose not to run one with our teams. General-purpose generative AI is still recent, and I haven't found a playbook that has been validated over the long term.

To be fair, some organizations are trying. BCG documents six strategies[4] and names the organizations behind them. Shell asks juniors to frame the problem, test their assumptions, and produce a baseline analysis before they touch AI. At CNIL, France's data protection authority, managers are assessed on whether their people can challenge AI output rather than simply use it. One Indian bank runs an AI-free working session across every function on the first Friday of every month.

These are serious attempts. They're more structured than anything we run, though they're also recent. Among BCG's 70 respondents, one in ten reported an organization-wide strategy. Roughly a third said the issue

hadn't been discussed. Watch these practices and learn from them. Just don't call them a validated maturity model yet.

What I can share are five habits our engineering teams practice. None needs new tooling, but each one needs a conversation you may be avoiding. After that, I'll be open about the two problems we haven't solved. Pretending otherwise would defeat the point of this paper.

There's one rule above all five. Disclosure has to be safe. If "AI did more of the thinking than I realized" becomes evidence in a performance review, people will stop saying it, and the dial disappears again.

Use these habits for inquiry, coaching, and better ways of working. Don't use them to rank people. That doesn't weaken accountability, in our view. People can only take responsibility for offloaded judgment when they can admit it before a mistake reaches production or a client.

Habit 1: Use both axes, constantly

You cannot manage a distinction your team cannot name. At November Five, accelerator and amplifier are part of our everyday vocabulary. So are exercising and offloading judgment. They come up in 1:1s and reviews. They come up in ordinary questions. Did this only make us faster? Did it make new work possible? And where did human judgment stay?

Most instruments don't ask those questions. The JetBrains survey of 24,534 developers[7] doesn't ask them. Your dashboards don't record them either. If the questions get asked, it's because a leader keeps asking and answering them.

As leaders, we use the same language about our own work: what AI made possible, where we accepted an answer too quickly, and what we checked. Without that reciprocity, the vocabulary starts to feel like an interrogation. DORA treats a clear and communicated AI stance[19] as one of seven organizational capabilities for effective adoption. Communicate a stance once, and you have a memo. We talk about ours every week because the repetition is the point.

Communicate a stance once, and you have a memo.

TIM VAN GEHUCHTEN

Habit 2: Make durable value explicit

When we see AI being used only to make familiar work faster, we say it plainly. Speed is useful, but speed alone isn't much of a moat. If the improvement came from a tool anyone can buy, everyone can buy the same improvement.

Where does durable value come from? Frame a better problem. Use AI to reach work that was out of range before. Keep judgment over the result. This is a conversation about where the team invests its capability. It should never become a conversation about who can be replaced.

Why do we keep raising this? Feeling faster can be a poor guide. The METR and Aalto studies give you good reason to compare that feeling with outcomes and understanding. They don't prove that any particular engineer uses AI badly.

Make it a joint diagnosis. What did we expect AI to change? What actually changed? Which decisions stayed with us? These are direct questions that force us to examine the work. They don't put the person on trial. Once engineers understand that speed, reach, and retained judgment create different kinds of value, they can examine both settings themselves.

Habit 3: You own what you ship

We keep a one-page accountability document called You Own What You Ship. The rule is simple. If you produce something with AI and pass it on, you own it, not the tool. There are no exceptions.

AI makes work cheaper to produce. The bar for ownership stays where it is. Check the work in proportion to the risk. Say what you're uncertain about and ask for help before your uncertainty becomes someone else's problem.

We make each other a simple promise. Nobody else should be the first person to scrutinize my AI output. We don't pass unreviewed output to each other, and we don't pass it to a client.

FIGURE 3 · YOU OWN WHAT YOU SHIP

Four lines

November Five's one-page accountability document comes down to these. Copy them, print them, change the wording for your team.

01 **I read and verify anything before I pass it on to a person.**

02 **I never make a peer or client the first human to read my AI output.**

03 **I can answer for every claim, number, and decision in what I share.**

04 **When something is wrong, it's mine to fix. The tool isn't a place to put the blame.**

AI output is often almost right. Verification is an explicit task and responsibility. When generation moves faster than validation, verification debt builds. That's why ownership matters. Any colleague may ask you to

walk them through the decisions that matter. That's a normal question between people who own the work together. It should never be a surprise test.

A fluent explanation doesn't prove understanding, but if someone can't explain an important choice, slow down. Review it together and decide what practice is missing. Don't turn one awkward explanation into a two-minute diagnosis of atrophy.

Habit 4: Read patterns, not people

We deliberately don't use AI spend per engineer as a productivity or capability score. Why not? A cost line mixes model pricing, the type of work, experimentation, and inefficiency. Sometimes it only shows that one person is doing the evaluation work for everyone else. It can't locate either axis.

We start one level up. Look at patterns by team, initiative, or type of work over time. If spend rises, ask what changed. Did AI make familiar work faster? Did it make new work possible? Where did people challenge and check the result? What happened to quality and delivery? The pattern won't answer those questions, but it tells you which work is worth discussing.

Sometimes the useful conversation is individual. That's fine. Start with the work and its context instead of comparing colleagues. Tell people in advance what data is visible, why you look at it, and how you will use it. Be equally clear about how you won't use it.

Never rank people by spend and don't smuggle that number into a performance review through a side door. If cost control or security requires an individual audit, call it what it is. Keep that process separate from the capability conversation.

Be straight with people. If an engineer discovers that you inspect their spend and nobody explained why, what do you think happens? They spend less or move the usage out of sight (shadow AI). DORA identifies[19] concealed use and excessive caution as two risks of an unclear AI stance. Transparency protects the signal you are looking for.

Habit 5: Decide what to preserve, reframe, and let go

You don't need to defend every skill in its current form. Handing commodity work to AI can be a sensible choice when you make the trade-off explicit. "Keep it or lose it" is too crude for us. We use three decisions instead:

- **Preserve it.** Human capability needs a baseline and deliberate practice because failure without the tool would be unacceptable.
- **Reframe it.** The old execution skill matters less, but judgment moves to a new level and needs to be taught there.

- **Let it go.** The dependency is accepted, recorded, and matched with an appropriate fallback.

When velocity multiplies, nobody knows every line of the codebase anymore. Expecting that is nostalgia. We do expect engineers to understand the system. What's in the codebase? How does it fit together? Why is it shaped that way? Our engineers are becoming more architects and less line-by-line authors. We see that as a reframe, and we're at peace with it.

When velocity multiplies, nobody knows every line of the codebase anymore. Expecting that is nostalgia.

TIM VAN GEHUCHTEN

Use it or lose it still applies to the capabilities you choose to preserve. How quickly does a skill weaken? It depends on the person, the task, and how often they practice. Engineering doesn't have a measured decay curve yet. The Lancet study[3] is why we refuse to assume we have plenty of time. In another expert domain, unassisted performance was lower after months of routine AI exposure.

Some skills will be practiced less as the work changes. That's manageable when you make the choice explicit. Say it out loud. Write it down. We will preserve this, reframe that, and deliberately let this other thing go. Then revisit the choice when the tool, the role, or the cost of failure changes.

What we haven't solved

Juniors. The evidence doesn't form one causal chain, but two risks may interact. People may learn less inside the work, and there may be fewer entry-level opportunities. November Five isn't the exception. We don't hire juniors right now. That removes one route for a junior to become an experienced engineer here.

Every organization that makes the same choice relies on someone else to train its future seniors. We don't have the answer yet.

Measurement. I want answers for every euro we spend. How much faster do we ship at the same or better quality? What new work became possible? And which human judgment remained intact? For an agency like ours, those questions affect pricing, staffing, and what we can promise clients.

We're working toward those answers. We don't have them yet, and we haven't seen anyone who credibly does. We talk to the people doing the work. We remain skeptical of any dashboard that claims it can read either axis directly.

06

What this buys you

What will these habits buy you? I want to keep the claims modest and tell you what we've seen work. You now know two ways our own approach falls short. A solved capability problem is beyond what these habits

can promise.

The habits give your team words for the differences. The JetBrains survey[7] doesn't ask these questions, and 60% of LeadDev respondents[8] named the lack of clear AI–impact metrics as a challenge. Your team can ask them in a regular 1:1. Did this make us faster or take us further? Where did human judgment stay? Leaders answer too.

Does that sound small? It is, but it addresses the blind spot every standard response leaves. Shared words and a timely conversation can bring the settings into view.

You can discuss the dial without policing people. When disclosure is safe and leaders answer the same questions, an engineer can look at speed, reach, and retained judgment without worrying about an individual score. You can steer without tags or surveillance. Both can give people a reason to hide or manage the signal. DORA also treats[19] clarity and psychological safety as conditions for effective AI use.

Responsibility becomes explicit. "The AI wrote it" is no longer a defense. Asking someone to walk you through the work becomes normal when both of you own the result. Okrona's verification debt[18] gives your team a name for the gap in checking the work before a client sees the consequences.

You can get a first signal this month. Use team–level and initiative–level spend patterns as pointers to work worth discussing. They aren't a validated research instrument and cannot place anyone in a quadrant. What became faster? What became possible? What happened to quality? And where did judgment stay?

That gives you a more meaningful conversation with leadership. You can explain what you spend, what became possible, and where you're still checking whether capability grew with it.

Your capability choices become explicit too. Write down what you'll preserve, reframe, and let go. Writing it down makes the decision easier to discuss and revisit.

It's only fair to hold these habits to the same standard as the frameworks above. So let's score them.

07

Seven tests for any approach, including ours

These seven tests come straight from the blind spots we have just seen. Can the approach surface the invisible choices? Can it see how they add up? Can it act before a late signal becomes a failure? Ask these questions of any approach, including ours. This isn't a maturity ladder.

01 **Does it observe both settings, not just adoption or output?** It needs to distinguish acceleration from amplification. It also needs to surface where human judgment stayed. Vendor frameworks[23] separate adoption, output, and quality for the same reason. One cannot tell you what happened to the others.

02 **Does it capture the division of work at the moment it's made?** The artifact cannot tell you who framed, generated, challenged, or checked the work. Ask while the work is fresh. Then check performance separately. Remember METR? Perceived and actual speed differed by 39 points.

03 **Does it establish and repeat a baseline for unassisted capability?** Assisted output cannot show what a person or team can still do without the tool. How often should you test? Match the cadence to the capability's importance and risk. The evidence doesn't support a one-size-fits-all interval.

04 **Does it keep disclosure unscored?** Turn the dial into an individual score, and people have a reason to manage the number or hide their use. Keep the conversation candid. Don't score it.

05 **Does it schedule replenishment as a standing cadence?** People hand over small choices inside their daily work. One training event cannot address what happens afterward.

06 **Does it address learning and the junior pipeline explicitly?** Anthropic gives us a short-term learning signal. The employment studies give us descriptive workforce signals. Neither proves that capability will fade fastest here. Together, they give you enough reason to pay attention.

07 **Does it produce a recurring steering decision at leadership level?** Choose a cadence that is frequent enough to turn what you observe into choices about tools, practice, and the skills worth preserving.

How do our five habits score? They pass two tests. The conversations surface both settings (1), and disclosure stays unscored (4). To be clear, spend patterns earn no credit for that first test. They point to a team or initiative worth discussing. The conversation does the work.

The habits fail to capture the division of work when it happens (2). We have no cadence for measuring unassisted capability (3). Replenishment is ad hoc at best (5). And you already know where we stand on juniors (6).

We thought we passed test 7 until we wrote this section. Every month we decide which AI tools and processes go to the main branch, which stay on a feature branch, and which get killed. Leadership is in the room, and decisions get documented. The meeting has value.

But we talk about which tool. The capability risk sits in where judgment moves. Nobody in that room has ever asked what we got worse at since last month, so that's a fail, if a useful one. We're adding that question from now on. Do you run a similar forum? Add it there. It's one of the cheapest places to start.

FIGURE 4 · SEVEN TESTS

Score your team beside ours

Tick each test your team's approach passes. The November Five column is how November Five scored its own habits in October 2026.

	THE TEST	NOVEMBER FIVE	YOUR TEAM
01	Does it observe both settings, not just adoption or output?	PASS	◯
02	Does it capture the division of work at the moment it's made?	FAIL	◯
03	Does it establish and repeat a baseline for unassisted capability?	FAIL	◯
04	Does it keep disclosure unscored?	PASS	◯
05	Does it schedule replenishment as a standing cadence?	FAIL	◯
06	Does it address learning and the junior pipeline explicitly?	FAIL	◯
07	Does it produce a recurring steering decision at leadership level?	FAIL	◯
	SCORE	**2 of 7**	**__ of 7**

Test wording and November Five's results from section 7. The tests ask whether a team keeps its own judgment and skills over time. They don't grade the software it ships.

Nobody in that room has ever asked what we got worse at since last month.

TIM VAN GEHUCHTEN

We're publishing that scorecard anyway. It only passes two tests, and that's useful information. I haven't found a documented approach that passes every test. If someone tells you their framework does, ask them to show the evidence for each one.

Use these as team prompts, never as surprise individual tests.

FIGURE 5 · SIX QUESTIONS

For your next team meeting

01	Can an engineer walk you through a month–old shipped change with the repo closed and without an AI assistant?
02	What did AI make possible that was previously out of reach, and who retained judgment over it?
03	Which tasks did we silently start delegating this year, and did anyone decide?
04	Where is our preserve–reframe–let go list written down?
05	When did a junior last debug a problem from first symptom to fix without AI, with a senior watching?
06	If your team lost access to its AI tools for two weeks, what work could still continue?

You can begin every test on this list this quarter. Does that prove where your decay threshold sits? No. It gives you a practical place to start. Give it a try. The risky choice is to keep going without knowing which judgments and capabilities are moving.

08

Conclusion

Somewhere on your team today, an engineer used AI to make work faster or take it further. They also set the invisible dial. They exercised judgment, or handed some of it over. The artifact only records the result. Multiply that choice across thousands of tasks, and it starts to shape the team.

How fast does that happen? No study gives us a universal engineering decay clock, but if you wait until output or quality breaks, failure becomes your first measure of capability. Skill erosion deserves more than a tooling discussion or a training program. It's a leadership responsibility. How you let your team use AI shapes what your company can still build and understand.

What should you take with you? Start by naming both settings, and use those words every week. The surveys and dashboards in this paper don't ask whether AI increased reach or replaced judgment.

Next, make disclosure safe and reciprocal. You can't read the dial through surveillance. A team that can talk about offloaded judgment without turning it into a score can learn from it. And own everything you

ship. Look at patterns rather than people. Ownership makes offloaded judgment discussable, while team-level cost patterns point you toward work worth examining.

You can start everything in this paper this month. You don't need new tooling. The mechanics are easy enough. The conversations are harder, and so is seeing your own organization clearly from the inside.

These habits explain why we practice AI-led product development the way we do. We use AI to expand what teams can build while keeping human judgment active where it matters.

If you're working through the same questions, I'd be happy to compare notes.

COMPARE NOTES

How did your team score?

We'd like to hear it, especially if you run a forum like the one that failed test 7. Write to Tim Van Gehuchten, our Director Product & Technology, at hello@novemberfive.co with the subject "The Invisible Dial". Tim replies personally.

NOVEMBERFIVE.CO/THE-INVISIBLE-DIAL

About the author

Tim Van Gehuchten is Director Product & Technology at November Five. He works with clients on the technology choices that matter, from AI and agent platforms to build-or-buy decisions, and leads November Five's security and compliance. He writes about how teams adopt AI, what it changes in the way products get built, and how to keep it secure.

09

References

01 METR. Measuring the Impact of Early–2025 AI on Experienced Open–Source Developer Productivity. 2025.
metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study

02 Anthropic. How AI Assistance Impacts the Formation of Coding Skills. 2026.
anthropic.com/research/AI-assistance-coding-skills

03 The Lancet Gastroenterology & Hepatology. Multicentre observational study of unassisted adenoma detection after routine AI exposure. 2025.
thelancet.com/journals/langas/article/PIIS2468-1253%2825%2900133-5/abstract

04 Boston Consulting Group. When Everyone Uses AI, Companies Risk Critical Skills. 2026.
bcg.com/publications/2026/when-everyone-uses-ai-companies-risk-critical-skills

05 DORA. Announcing the 2025 DORA Report. 2025.
cloud.google.com/blog/products/ai-machine-learning/announcing-the-2025-dora-report

06 Stack Overflow. 2025 Developer Survey: AI. 2025.
survey.stackoverflow.co/2025/ai

07 JetBrains. State of Developer Ecosystem 2025. 2025.
blog.jetbrains.com/research/2025/10/state-of-developer-ecosystem-2025

08 LeadDev. The AI Impact Report 2025. 2025.
leaddev.com/wp-content/uploads/2025/08/THE-AI-IMPACT-REPORT-2025-download__LDMO__.pdf

09 GitKraken. AI Is an Amplifier, Not a Shortcut. 2026.
gitkraken.com/blog/ai-is-an-amplifier-not-a-shortcut

10 Microsoft Research and Carnegie Mellon University. The Impact of Generative AI on Critical Thinking. 2025.
microsoft.com/en-us/research/publication/the-impact-of-generative-ai-on-critical-thinking-self-reported-reductions-in-cognitive-effort-and-confidence-effects-from-a-survey-of-knowledge-workers

11 Aalto University. AI Makes You Smarter, but None the Wiser: The Disconnect Between Performance and Metacognition. 2026.
research.aalto.fi/en/publications/ai-makes-you-smarter-but-none-the-wiser-the-disconnect-between-pe

12 Aalto University. Researchers Warn That Skill Erosion Caused by AI Could Have a Devastating and Lasting Impact. 2025.
aalto.fi/en/news/researchers-warn-that-skill-erosion-caused-by-ai-could-have-a-devastating-and-lasting-impact-on

13 Michael Gerlich. AI Tools in Society: Impacts on Cognitive Offloading and the Future of Critical Thinking. 2025.
mdpi.com/2075-4698/15/1/6

14 EY. Companies Are Missing Out on Up to 40% of AI Productivity Gains Due to Gaps in Talent Strategy. 2025.
ey.com/en_gl/newsroom/2025/11/ey-survey-reveals-companies-are-missing-out-on-up-to-40-percent-of-ai-productivity-gains-due-to-gaps-in-talent-strategy

15 Harvard/SSRN. Working paper on GenAI adoption and junior employment across US firms. 2025.
papers.ssrn.com/sol3/papers.cfm?abstract_id=5425555

16 SignalFire. State of Talent Report 2025. 2025.
signalfire.com/blog/signalfire-state-of-talent-report-2025

17 Stanford Digital Economy Lab. Canaries in the Coal Mine? Six Facts about the Recent Employment Effects of Artificial Intelligence. 2025; revised 2026.
digitaleconomy.stanford.edu/publications/canaries-in-the-coal-mine

18 Csaba Okrona. Verification Debt. Leadership Garden, 2026.
leadership.garden/verification-debt

19 DORA. Clear and Communicated AI Stance. 2026.
dora.dev/capabilities/clear-and-communicated-ai-stance

20 Tobi Lütke. Internal memo on reflexive AI usage at Shopify, reported by Business Today. 2025.
businesstoday.in/technology/news/story/ai-use-is-no-longer-optional-at-shopify-declares-ceo-tobi-lutke-in-internal-memo-471211-2025-04-08

21 DORA. Accelerate State of DevOps 2024. 2024.
dora.dev/research/2024/dora-report

22 Faros AI. AI Acceleration Whiplash. 2026.
faros.ai/research/ai-acceleration-whiplash

23 DX. AI Measurement Framework Guide. 2026.
getdx.com/blog/ai-measurement-framework-guide

NOVEMBER FIVE

Most products work. Few win. Almost none are remembered. That gap is our job.

November Five builds products that move the number your business depends on, and that people come back to.