McKinsey & Company

Technology, Media & Telecommunications Practice

Beyond the copilot: Scaling the agentic product development life cycle

Only a few software teams are seeing real impact from incorporating AI. The key is launching an AI-driven redesign of the entire product development system, rather than just deploying new tools.

This article is a collaborative effort by Charlotte Relyea, Gautam Lunawat, Janaki Palaniappan, Martin Harrysson, Matt Linderman, and Prakhar Dixit, with Albert Savitzky and Vasu Tekriwal, representing views from McKinsey's Technology, Media & Telecommunications Practice and the Tech & AI group.



August 2026

In brief

- AI and agents are accelerating product and software development, but outcomes remain uneven enterprise-wide.
- Some organizations are pulling ahead, realizing twofold productivity gains across meaningful portions of their teams, while others are seeing little impact or even negative returns.
- Four common themes emerged across organizations pulling ahead: They are rewiring their operating model and processes, redesigning roles and responsibilities, building verification mechanisms and AI operations that keep pace with faster work, and proactively managing and investing in change.
- Organizations that delay moving beyond experimentation risk falling further behind, given that AI costs are increasing, competitive pressures are rising, and many companies are already working in substantially accelerated agentic day-and-night shifts.

Ever since software organizations began to adopt AI in recent years, the technology's impact has been remarkable—and uneven.[1] As more enterprises incorporate AI into their software and product development, the divergence in AI outcomes is rapidly widening. While some organizations are building "agent factories" that ship code 24/7, most organizations have yet to move the needle on productivity.

In our recent survey of 334 product and engineering leaders, only 25 percent of the respondents in director-and-above positions report meaningful (or top) AI acceleration, which we define as more than a quarter of their teams achieving twofold or greater productivity gains. Concerningly, 30 percent reported that team productivity had, in fact, *fallen*.[2]

This gap highlights a challenge that software organizations face when embedding AI into their processes. Technology is not the differentiator. What separates leaders is how effectively they tailor their end-to-end workflows, roles, control systems, and change management to an agentic product development life cycle (PDLC).

Closing this gap is an almost trillion-dollar opportunity. Around 80 percent of the software engineers we surveyed report an average AI productivity acceleration of approximately 3 percent. The top 20 percent of engineers see AI productivity growth averaging 55 percent.[3] If

[1] "Unlocking the value of AI in software development," McKinsey, November 3, 2025.
[2] McKinsey Agentic PDLC/SDLC Survey, May 2026, n = 334; director-and-above analysis, n = 173.
[3] McKinsey Agentic PDLC/SDLC Survey, May 2026; developer analysis, n = 198.

organizations can coax their roughly 30 million engineers globally[4] to catch up to the top quintile—a conservative target in a world where leading teams are already exceeding twofold productivity gains—the potential new value starts to rival the GDP of a midsize economy (about $0.8 trillion, assuming roughly $85,000 average fully loaded cost per engineer). This is before considering that AI productivity applies to non-engineering roles in the PDLC too.

About the research

This article is based on a May 2026 survey of 334 product and engineering respondents, including a subset of 173 in director-level or higher roles, from across Africa, Asia, Europe, Latin America, and North America. Respondents represent a range of industries, including technology, media, and telecommunications; finance and insurance; consumer and retail; healthcare and life sciences; and manufacturing, logistics, and transportation. It also builds on McKinsey's November 2025 baseline research on AI in software development.

This article, based on our second quarter 2026 survey of product and engineering leaders as well as our experience in the market, examines how outperforming software organizations are generating meaningful impact, and the steps others can take to systematically realize similar value from AI (see sidebar, "About the research").

What agentic PDLC success requires

When we studied the organizations that are ahead of the curve in integrating AI into the PDLC and what makes them distinctive, four themes emerged. First, they are reimagining end-to-end processes and workflows, rather than just plugging AI tools into existing ways of working. Second, they are changing team structures, job roles, and responsibilities to focus on human judgment and product intent. Third, they are building verification, control, and measurement systems that can keep pace with new and faster ways of working. Finally, they are investing heavily in hands-on upskilling to drive change and adoption. In short, they are leveraging AI to redesign the whole system of product development—and it is paying off.

Reimagining end-to-end processes and workflows

Workflow or process redesign was identified as one of the top two enablers, according to respondents who reported at least one of their teams experiencing twofold productivity gains

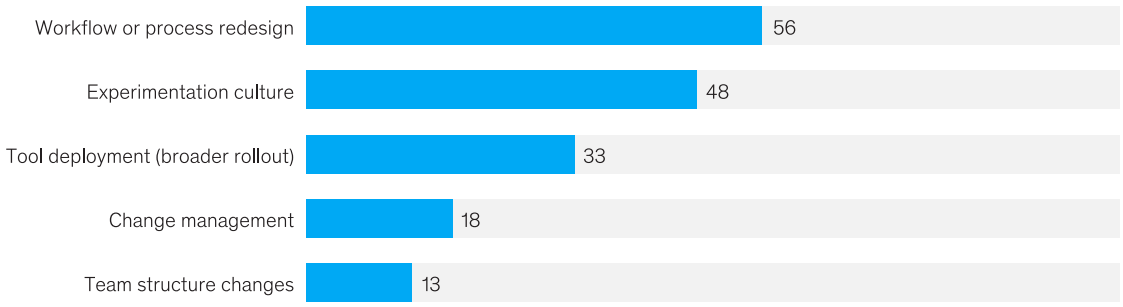
[4] *State of developer ecosystem 2025*, JetBrains, October 2025; "Number of software developers worldwide," Statista, June 3, 2024; SlashData Developer Nation surveys, Q3 2025.

(Exhibit 1). Ninety-three percent of top accelerators—those respondents who reported more than a quarter of their teams achieving twofold or greater productivity gains—embedded AI into their workflows.[5] How they embedded AI mattered just as much: respondents at organizations that redesigned their processes before incorporating the technology were more than twice as likely to report productivity gains of more than 20 percent as those that layered AI onto existing ways of working.[6] By sharp contrast, those organizations that have strong AI adoption and follow most best practices (including AI goals in performance reviews and implementing strong coaching programs) but fail to holistically reshape their workflows around AI tend to miss out on meaningful impact.

Exhibit 1

Software leaders say redesigning workflows or processes around AI has enabled their teams to significantly increase their productivity.

Share of respondents citing enabler as a reason for 2× productivity increase in agentic product development, top two enablers, %



Note: Figures sum to more than 100%, because respondents chose top two enablers.
Source: McKinsey Agentic PDLC/SDLC Survey, May 2026 (director-level and higher respondents; n = 138)

McKinsey & Company

This focus on redesigning end-to-end workflows, rather than automating individual tasks or overemphasizing the technology stack, distinguished top accelerators. They took a clean sheet view of how work should get done, questioning what should be led by humans versus AI, refactoring their software development artifacts, and then deploying agents to power the redesigned processes.

These shifts are increasingly critical as product development evolves and becomes more

[5] McKinsey Agentic PDLC/SDLC Survey, May 2026; n = 334; director-level and higher analysis, n = 138.
[6] McKinsey Agentic PDLC/SDLC Survey, May 2026; director-level and higher analysis, n = 138.

asynchronous. Agents work independently around the clock, so the process no longer must move through a linear chain of human handoffs. They can make progress during a "night shift," while people focus on review and high-judgment decisions during the day. It is a new way of working with little resemblance to agile scrum sequences of the past. Simply speeding up existing cadences won't make a meaningful difference. To keep pace, leaders need to redesign how work gets done around asynchronous, agent-driven execution.

Sonar, a leader in code verification tools, took this approach, rebuilding its path from product insight to shipped code. Three frontrunner teams embedded agents across research, ideation, backlog definition, coding, testing, and remediation. They synthesized customer and market signals, turned discovery into Jira tickets, generated tests and code, and carried a bug report all the way to a draft pull request. As a result, pull-request cycle duration fell by 3.4 times, pull-request throughput improved by 2.2 times, and build activities became 50–80 percent more productive. (To learn more about Sonar's experience, read the full case study.)[7]

Changing roles around human judgment and product intent

While process redesign is important, role redesign is often where top accelerators differentiate themselves. Leading software organizations typically transform teams, roles, handoffs, and decision rights around AI before they adjust staffing models or reallocate capacity.

The survey data reinforces that workflow and tool integration alone are insufficient. Organizations that embedded AI without changing roles or ways of working, which we call adopters, became top accelerators only 27 percent of the time. Those that adopted some AI tools but didn't embed them in the workflow, which we call experimenters, only attained that milestone 17 percent of the time. By contrast, teams that embedded AI and redesigned the roles and work, which we call transformers, became top accelerators 40 percent of the time. Role changes are an underappreciated lever. Team members give more credit to process and tooling overhauls, but impact tends to be minimal unless people's responsibilities change as well (Exhibit 2).

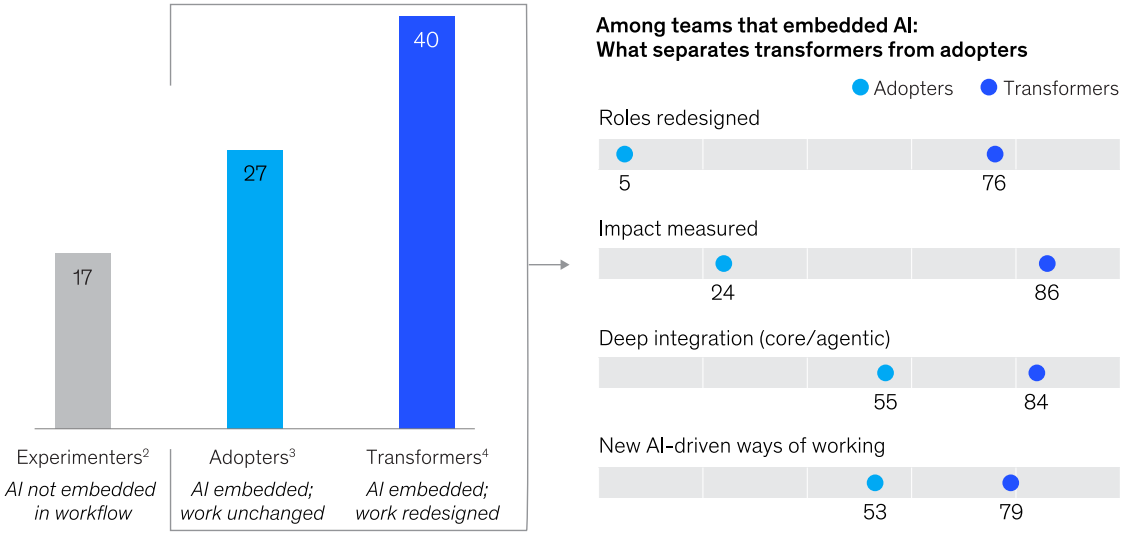
What people spend time on is changing. As AI and agents take on more routine, execution-centric work, human judgment continues to gain importance. Teams need to translate product intent into executable instructions: telemetry-linked requirements, acceptance criteria, definitions of done, guardrails for what agents can change, and clear routing to the right reviewer, test, or approval gate. This matters because ambiguity travels faster in an AI-enabled PDLC and can do more damage: a vague requirement, disconnected from telemetry, can generate plausible but incorrect tickets, prototypes, tests, and code before anyone notices. Specifications and requirements should be clear and detailed enough for an agent to act on, an engineer to challenge, a test suite to validate, and a reviewer to judge. Narratives that an experienced engineer interprets through conversations and tacit context may no longer do the job. In this new model, people spend less time producing every artifact themselves and more time setting direction, making trade-offs, validating outputs, and reviewing the decisions that matter.

[7] "Rethinking software development: An AI-native approach," McKinsey, June 12, 2026; *Sonar blog*, "The future is AC/DC: The agent centric development cycle," blog entry by Tariq Shaukat, March 2, 2026.

Exhibit 2

Embedding AI into software development workflows is important, but redesigning roles and ways of working drives the biggest productivity gains.

Share of software teams reaching top acceleration of AI productivity impact,[1] by category, %



[1]Top acceleration refers to respondents who reported that more than a quarter of their teams achieved 2× or greater productivity gains. [2]Respondents reported using only a stand-alone AI assistant or no AI tools. [3]Respondents are considered to have embedded AI if they have incorporated at least one of the following: integrated development environment (IDE) or copilot integration, integration into core workflows, or agentic automation. [4]Respondents are considered to have redesigned work if the composite score of responses on reshaping roles, adopting AI-native ways of working, and tracking impact metrics is above the median score.
Source: McKinsey Agentic PDLC/SDLC Survey, May 2026 (director-level and higher respondents; n = 138)

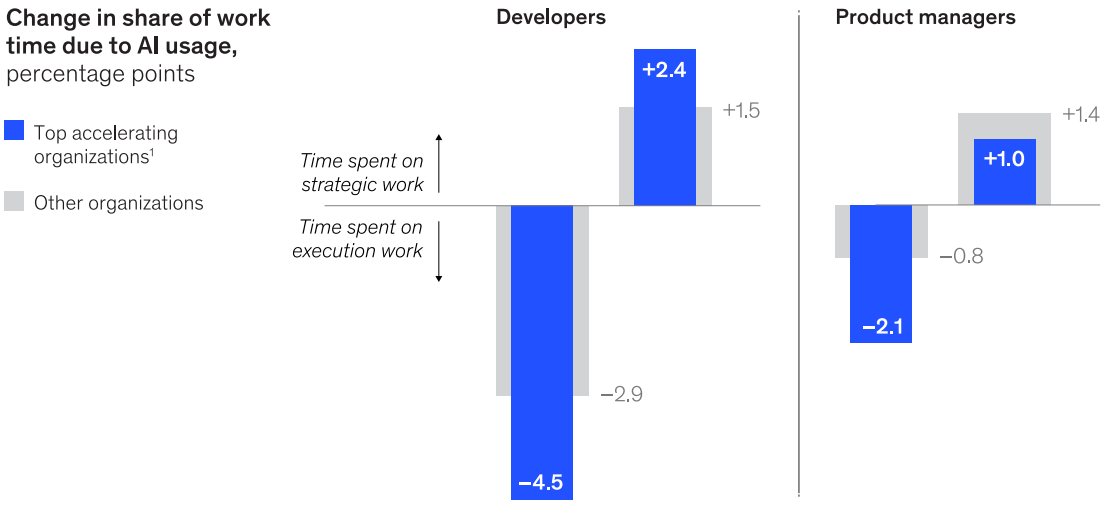
McKinsey & Company

These shifts are already underway. Product managers in top-accelerating organizations report a 24 percent reduction in time spent on execution activities, compared with an 18 percent reduction in strategic activities. The split is sharper for developers, who report a 19 percent reduction in execution activities versus just 6 percent in strategic activities. As AI absorbs routine drafting, translation, testing, and documentation, human work moves up the stack (Exhibit 3).

How teams are designed is also changing. In top-accelerating organizations, 79 percent of respondents say squad sizes have decreased after AI implementation, with the median squad shrinking from about ten people to seven. This compression is likely to continue, creating value by reducing handoffs, broadening ownership, and bringing review, quality, security, and operations closer to the delivery flow. Nearly 20 percent of director-level and above survey respondents at top accelerating organizations are already working on teams of one to four

Exhibit 3

Software teams experiencing the biggest impact from AI are reallocating more of their workday from execution activities to strategic work.



Note: Baselines ("before") reconstructed from per-activity time changes.
[1]Top accelerators = >25% of teams experiencing 2× productivity gains.
Source: McKinsey Agentic PDLC/SDLC Survey, May 2026; developers, n = 87 (26 top accelerators, 61 others); product managers, n = 51 (17 top accelerators, 34 others)

McKinsey & Company

people.[8] Smaller, more autonomous teams are becoming more common.

Finally, how individual roles are defined is changing. In top-accelerating organizations, most respondents report AI-driven changes in software engineering (93 percent), product management (84 percent), design (53 percent), and quality engineering (53 percent) roles. Three shifts stand out. First, repeatable work is moving from people to agents: documentation, test generation, first-draft artifacts, and other routine outputs. Among top accelerators, AI cut the time developers spend writing code by roughly 20 percent, and documentation by even more. Second, roles are expanding as individuals take on a larger share of the path from intent to release and run.

Engineers, for example, are moving upstream into solution design, requirements, and security, and downstream into deployment and operations. Third, role boundaries are blending. Product and design are converging to sharpen product definitions while testing is being split between agents that automate it and engineers who absorb more of what remains; as a result,

[8] McKinsey Agentic PDLC/SDLC Survey, May 2026; director-level and higher analysis, n = 138.

organizations may consolidate some dedicated QA activities. Taken together, these shifts point toward broader role archetypes: definers who fuse product and design, and builders who fuse development, testing, and operations.

These changes point to where role design is headed next: leading organizations are beginning to define roles not only for people but also for agents. As agents run larger parts of sprints and workflows, these organizations are giving agents scoped mandates, permissions, escalation paths, and accountability. Instead of a human organizational chart, the unit of design is increasingly a blended system of people and agents, each with defined responsibilities, organized around a redesigned flow of work.

One global technology company shows how this new approach can work in practice. It rebuilt its teams into lean definer–builder pods, shrinking eight-to-ten-person product pods into four-to-six-person agentic teams, with product managers owning definition, developers owning build, and agents running discovery, delivery, and launch tasks in between. The result: Not only did the smaller pods unlock roughly twofold capacity and 50 to 80 percent shorter development cycles, but user-story quality—as measured by an AI reviewer trained on the company's standards—rose by more than 45 percent.

Building verification, control, and measurement systems that keep pace with faster work

As AI speeds up software creation, teams may find it harder to reliably test or govern their output, raising the risk of an observability gap. In an agentic world, organizations therefore need checks and monitoring tools that can keep up with the new pace of work.

Our survey underscores this risk: Speed is improving faster than quality. Across different use cases that make up the PDLC (development, customer research, et cetera), respondents report average time savings of 11.8 percent and average rework reduction of 6.2 percent; the gap is even wider in development activities, where time savings average 11.2 percent and rework reduction 6.8 percent (Exhibit 4). If leaders prioritize velocity without adequate visibility, they can create technical debt faster than teams can pay it down.

External evidence points in a similar direction. Dora's 2025 research describes AI as an amplifier of the underlying software delivery system, with both positive and negative aspects.[9] Veracode's 2025 gen AI code-security research found that 45 percent of tested AI-generated code samples failed security checks.[10] GitClear reported a sharp increase in duplicated code in repositories with AI-assistant activity.[11] Stack Overflow's 2025 survey found that more developers distrusted AI accuracy than trusted it, with only 3.1 percent reporting high trust in AI outputs.[12]

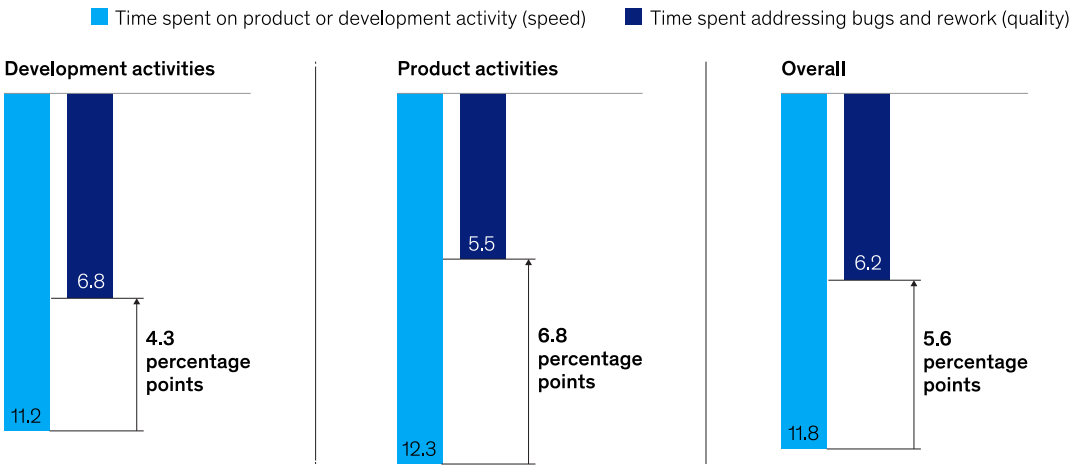
Leaders recognize that faster generation raises the standard for verification, which should be designed into the AI-supported workflow. They use specialized AI tools to help check the work that AI coding and development solutions accelerate. These agents generate tests and security scans, compare outputs against requirements, surface missing acceptance criteria, identify likely defects, and propose fixes. Such checks feed into the normal release path, with

[9] "State of AI-assisted software development 2025," DORA, September 23, 2025.
[10] *2025 genAI code security report*, Veracode, July 30, 2025; "AI-generated code poses major security risks in nearly half of all development tasks, Veracode research reveals," Veracode press release, July 30, 2025.
[11] *AI copilot code quality report*, GitClear, February 2025.
[12] "2025 developer survey: AI," Stack Overflow, July 29, 2025.

Exhibit 4

The speed of agentic software development is outpacing the increases in product quality.

AI-driven reductions in total time spent across product development life cycle use cases,[1] %



Note: Figures may not sum to totals, because of rounding.
[1]Values are averages based on respondents' self-reported reductions.
Source: McKinsey Agentic PDLC/SDLC Survey, May 2026 (n = 334; weighted averages across activities)

McKinsey & Company

auditability and traceability built into the workflow, rather than becoming a separate layer of AI activity that teams assess after the fact.

AI can create more findings than teams can act on, so the same high-performing organizations triage by risk and value. Leaders define which issues agents can fix automatically, which issues—including, in some cases, agent behavior—need expert review, and which issues should stop a release. Low-risk, recurring fixes can be managed with automations and agent-assisted remediation. Higher-risk changes, including customer-facing behavior, regulated data, security-sensitive code, architecture changes, and critical dependencies, should require stronger evidence and accountable human review.

Another key ingredient is a shared AI operations (AI Ops) layer. As agents move across tickets, repositories, test suites, deployment pipelines, and observability systems, organizations need common controls for identity, permissions, tool access, routing, evaluation, logging, escalation, policy enforcement, and cost visibility. Without that, it becomes challenging to monitor and scale agent activities safely.

The survey results suggest this makes a big difference: Many respondents report that their organization has an AI Ops capability, but most staff it lightly and have individual teams improvise key decisions. The outperforming cohort invests more heavily, treating orchestration and evaluation as foundational elements. Among organizations with AI Ops capabilities, respondents at those with five or more mandates (such as established governance standards and agent and platform performance monitoring) report higher positive impact across quality and time savings than those with fewer mandates.

One convenience retail player built the control layer to match its new speed. As AI accelerated code and test generation, the retailer pushed verification into the delivery flow itself, standardizing CI/CD (continuous integration and continuous delivery) and DevSecOps (development, security, and operations) so code coverage, development standards, and release-readiness checks are enforced automatically rather than inspected after the fact, and using agents to generate tests. Front-runner teams achieved velocity increases of up to 30 percent and reported a 50 percent decrease in defect-resolution effort, showing how integrated verification and governance enable work to scale faster without added risk.

Investing heavily in driving AI change

Most software organizations understand that change management is needed to generate substantial value from AI, but fewer have fully defined what it takes to drive this massive organizational shift.

Hands-on training and coaching have proven to be important. Embedded coaching is about 30 percent more common in top-accelerating than in low-accelerating organizations. Teams may well struggle to master the nuances of the agentic PDLC, and what is expected of them, in the abstract. They learn by doing, rewriting requirements, reviewing AI-generated outputs, deciding when to escalate, and changing release routines with an expert close enough to advise. Instruction also needs to focus on entire workflows instead of individual roles, with clarity conveyed about which steps in the process are now AI-supported, which decisions and outputs require human input, and which metrics will be tracked.

This often means experienced AI engineers and coaches must work inside teams long enough to change habits. They help teams adapt workflows, configure and interact with agents, review outputs, and adjust ceremonies. The model is shifting from one-time training to continuous, embedded coaching that can scale.

Top-accelerating organizations also give teams ample practice. Lack of time to learn and experiment is one of the top barriers to realizing the impact of AI, cited by about 38 percent of survey respondents. Teams need protection, scheduled blocks to adapt their workflows, and the ability to test new agent-supported routines, build confidence in verification, and codify what works.

Measurement needs change as well. Many organizations still rely on adoption metrics such as licenses, daily active users, prompt volume, or share of code generated by AI. These show

whether people are using the tools, not whether releases are faster, more reliable, safer, or more valuable. Top-accelerating organizations are much more likely to measure quality, time to market, security, reliability, cost, employee experience, and customer impact. Eighty-six percent of top-accelerating organizations track outcome metrics such as quality, productivity, and speed, while low-accelerating organizations disproportionately rely on tool adoption as their North Star.

Finally, change management should also extend beyond teams to the leaders who manage the organization and allocate resources. As agents take on more work, spending on tokens, compute, infrastructure, and AI Ops can increase to as much as 20 percent of existing labor costs. Leaders therefore need to plan AI capacity, infrastructure, and capital with the same discipline they bring to talent change. The organizations pulling ahead are the ones making AI adoption supported, practiced, measured, funded, and safe to scale.

One global bank experienced what a difference such a comprehensive approach can make. After an early copilot adoption delivered only 10 to 15 percent productivity improvements and limited new value, the bank developed a structured change program built around putting experienced AI engineers inside delivery teams: principal and distinguished engineers became trainers, seasoned practitioners were seconded into squads to rewire ways of working from within, and velocity and throughput targets were written into team objectives and key results. In-scope teams achieved material operating model changes, such as moving towards daily sprints with agent night shifts. Teams went on to achieve efficiency gains of 40 to 80 percent, with one legacy-modernization flow cutting bug-fix time from multiple days to a few minutes.

What organization leaders can do next

Software teams eager to attain the type of AI-driven impact their outperforming peers are experiencing can consider taking the following steps while tracking their progress against the included markers.[13]

1. *Go after the highest-value workflows.* Organizations can pick the highest-value flows in the business and redesign them end to end, mapping each from trigger to outcome: the decisions, roles, handoffs, gates, telemetry, and customer feedback that decide whether value is unlocked. This can prevent teams from automating fragments while the end-to-end flow remains slow or unclear.

   Progress marker: You may know it's working when the redesign starts to fund itself—whole categories of manual work (first-pass market research, test writing, status reporting) have been automated rather than merely sped up, and the freed capacity pays for the tokens and coaching the change requires.

[13] The progress markers in this section are illustrative targets only, not survey-derived benchmarks. They should not be interpreted as recommended benchmarks or universal indicators of success.

2. *Redesign roles for people and agents alike.* As agents absorb more drafting, translation, testing, and remediation tasks, organizations can give them scoped roles while clearly delineating what humans continue to own. Many roles may broaden, with engineers responsible for more of the path from design to release, while product, design, and analytics converge on clearer definitions and outcome validation.

   Progress marker: You may know it's working when at least 70 percent of job descriptions have changed (new roles, new titles, and new people in them) and squads have shrunk to two to four people working alongside agents.

3. *Raise the standard for requirements.* Teams can move to testable specifications that connect the customer problem, outcome, constraints, acceptance criteria, risk, release guardrails, and success metrics. The specifications should be usable by an agent, an engineer, a test suite, and a product reviewer; otherwise, the organization is generating work faster than it can assess and validate.

   Progress marker: You may have met the bar when a requirement can be handed to an agent and come back in a form that is close to shippable. This means intent is explicit enough that an agent can act on it, a test suite can check it, and a reviewer can judge it without a follow-up conversation.

4. *Build verification and a shared AI Ops layer.* Designing verification into workflows—agents generating tests and scans, checking output against requirements—and running it all on one control plane, rather than leaving each squad to build its own, can make a big difference. This can include everything from identity, permissions, tool access, and routing to evaluation, logging, escalation, policy enforcement, and cost visibility. The goal is to make agent work observable, comparable, and safe to scale.

   Progress marker: You may have an effective system in place when agents can work across teams, tools, and knowledge sources without bespoke plumbing, and leaders can see agent activity, cost, risk, and output through a single control plane.

5. *Run adoption as a change program.* It can be critical to pair formal, instructor-led training with the practices that separate top accelerators from the rest: embedded coaching, show-and-tell sessions, office hours, incentives, and visible role modeling.

Find more content like this on the
McKinsey Insights App
Scan • Download • Personalize

Progress marker: You may know it's working when the organization looks and feels verifiably different: for example, 90 percent of human/agent teams working day-and-night shifts, two-day (not two-week) sprints, and 100 percent of teams tracking outcome metrics.

AI is dramatically transforming product and software development. It is shortening the distance from customer signal to product decision, from requirement to code, from defect to fix, and from release to measured outcome. The opportunity is substantial, and leading organizations are already realizing disproportionate value. They are doing so by launching an AI-driven redesign of the entire product-development system rather than just deploying new tools and telling teams to adopt and incorporate them. For the rest, the priority now is to scale AI across all aspects of their software organization fast enough to keep this performance gap from becoming entrenched. Those that simply continue to add new tools to old routines may be more likely to find that the payoff from AI remains limited.

Charlotte Relyea is a senior partner in McKinsey's New York office; **Martin Harrysson** is a senior partner in the Bay Area office, where **Gautam Lunawat** and **Janaki Palaniappan** are partners, **Albert Savitzky** is an associate partner, and **Vasu Tekriwal** is a consultant; **Matt Linderman** is a partner in the Connecticut office; and **Prakhar Dixit** is a partner in the Seattle office.

The authors wish to thank Abdul Wahab Shaikh, Adam Thelwall, Ling Lau, Nandita Bothra, and Nicholas de la Flor for their contributions to this article.

This article was edited by Daniel Eisenberg, an executive editor in the New York office.

Copyright © 2026 McKinsey & Company. All rights reserved.