MIT Technology Review Insights

Produced in partnership with

SoftServe

How agentic AI will change the way software is developed and managed.

Redefining the future of software engineering

Preface

“Redefining the future of software engineering” is an MIT Technology Review Insights report sponsored by SoftServe. This report, which is based on survey research and executive interviews, seeks to understand how organizations intend to adopt and deploy agentic AI in software engineering. Denis McCauley was the author of the report, Virginia Wilson was the editor, and Nicola Crepaldi was the publisher. The research is editorially independent and the views expressed are those of MIT Technology Review Insights.

We would like to thank the following executives for their time and insights:

Nathaniel Barnes, Portfolio Chief Technology Officer, Hg Capital

Serge Haziyev, Chief Technology Officer, Advanced Technologies, SoftServe

Rick Kazman, Danny and Elsa Lui Distinguished Professor of Information Technology Management, University of Hawaii

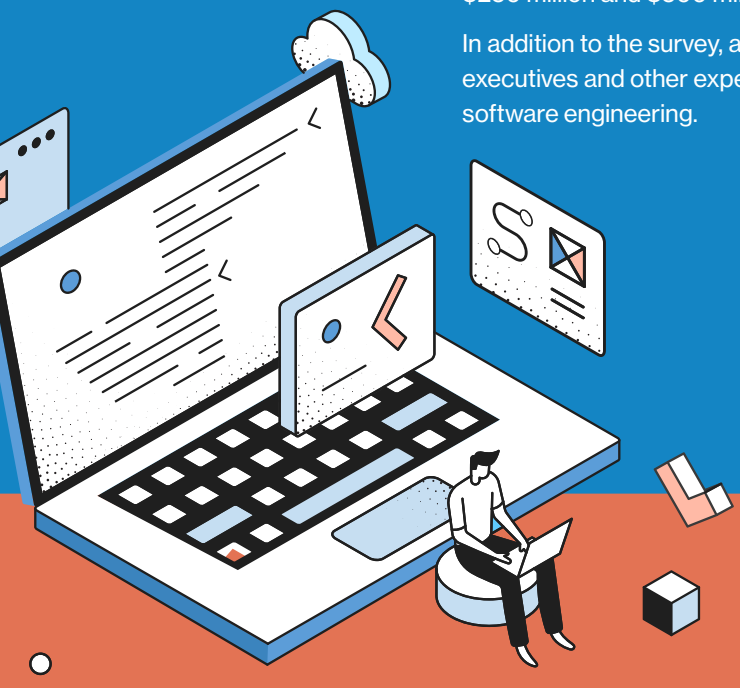
Shashi Prabakhar, Director, Americas Partner Solutions Architect, Amazon Web Services

Martin Vesper, Managing Director and Chief Digital Officer, Pfeifer & Langen

Methodology

In December 2025 and January 2026, MIT Technology Review Insights conducted a survey of 300 chief information officers, chief technology officers, chief data and analytics officers, chief AI officers, and other senior software engineering, data, and technology executives. The survey respondents work in organizations across seven industries, each headquartered in one of six countries. Most of the organizations earn $500 million or more in annual revenue, while a minority (13%) earn between $250 million and $500 million.

In addition to the survey, a series of in-depth interviews with senior technology executives and other experts offer firsthand insights into how agentic AI will impact software engineering.



CONTENTS

Foreword

The next era of software development

2025 has marked a steep acceleration in how AI is used in software engineering, evolving from vibe coding experiments to more disciplined agentic engineering.

For the software industry, the implication is profound: Productivity no longer scales linearly with headcount. It scales with how clearly we define the product vision, engineer context, and onboard agents.

But technology alone will not deliver the promised leap. As this report highlights, organizations recognize both the opportunity and the barriers ahead, from establishing AI-native roles to fundamentally redesigning the software development lifecycle. This report captures where that shift stands today, and what separates the companies already pulling ahead from those still finding their footing.

The turning point will come from rethinking people, processes, and tools in concert – not in isolation.

Agentic engineering is ultimately about dreaming bigger. If teams use the productivity dividend merely to ship the same backlog faster, they will miss the moment. If instead they channel it into deeper user value and innovation, the business impact will extend far beyond gains in speed and quality.

The history of technological progress suggests that the future of software is not human versus machine. It is human ambition amplified by autonomous systems. The organizations that understand this shift and act decisively will redefine what modern software engineering looks like for the next decade.

Serge Haziyev, CTO, Advanced Technologies, SoftServe

01 Executive summary

Software engineering has experienced two seismic shifts this century. First was the rise of the open source movement, which gradually made code accessible to developers and engineers everywhere. Second, the adoption of development operations (DevOps) and agile methodologies took software from siloed to collaborative development and from batch to continuous delivery. Now, a third such shift looks to be taking shape with the adoption of agentic AI in software engineering.

Thus far, engineering teams have mainly used AI to assist with coding, testing, and other individual tasks, within tightly designed parameters. But with agentic capabilities, AI agents become reasoning, self-directing entities that can manage not just discrete tasks but entire software projects – and do so largely autonomously. If adopted and fully embraced by engineering teams, agentic AI will usher in end-to-end software process automation and, ultimately, agent-managed development and product lifecycle automation.

This report, which is based on a survey of 300 engineering and technology executives, finds that software engineering teams are seeing the potential in agentic AI and are beginning to put it to use, but so far in a mainly limited fashion. Their ambitions for it are high, but most realize it will take time and effort to reduce the barriers to its full diffusion in software operations. As with DevOps and agile, reaping the full benefits of agentic AI in engineering will require sometimes difficult organizational and process change to accompany technology adoption. But the gains to be won in speed, efficiency, and quality promise to make any such pain well worthwhile.

Key findings include the following:

- **Adoption momentum is building.** While half of organizations deem agentic AI a top investment priority for software engineering today, it will be a leading investment for over four-fifths in three years. That spending is driving accelerated adoption. Agentic AI is in (mostly limited) use by 51% of software teams today, and 45% have plans to adopt it within the next 12 months.

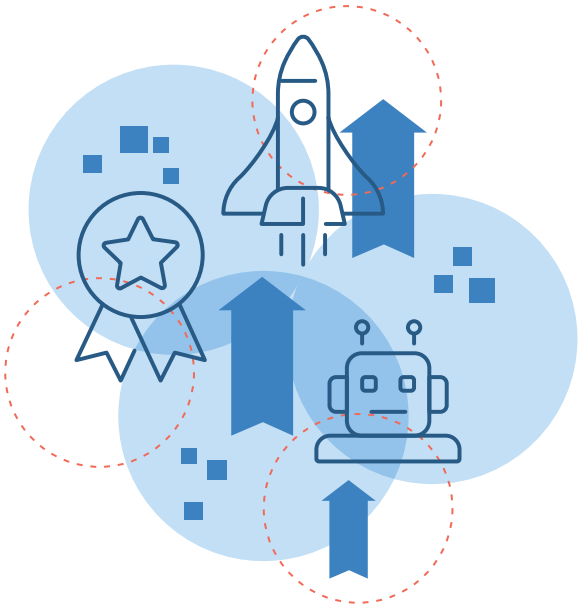
- **Early gains will be incremental.** It will take time for software teams' investments in agentic AI to start bearing fruit. Over the next two years, most expect the improvements from agent use to be slight (14%) or at best moderate (52%). But around one-third (32%) have higher expectations, and 9% think the improvements will be game changing.

- **Agents will accelerate time-to-market.** The chief gains from agentic AI use over that two-year time frame will come from greater speed. Nearly all respondents (98%) expect their teams' delivery of software projects from pilot to production to accelerate, with the anticipated increase in speed averaging 37% across the group.

- **The goal for most is full agentic lifecycle management.** Teams' ambitions for scaling agentic AI are high. Most aim for AI agents to be managing the product development and software development lifecycles (PDLC and SDLC) end to end relatively quickly. At 41% of organizations, teams aim to achieve this for most or all products in 18 months. That figure will rise to 72% two years from now, if expectations are met.

- **Compute costs and integration pose key early challenges.** For all survey respondents – but especially in early-adopter verticals such as media and entertainment and technology hardware – integrating agents with existing applications and the cost of computing resources are the main challenges they face with agentic AI in software engineering. The experts we interviewed, meanwhile, emphasize the bigger change management difficulties teams will face in changing workflows.

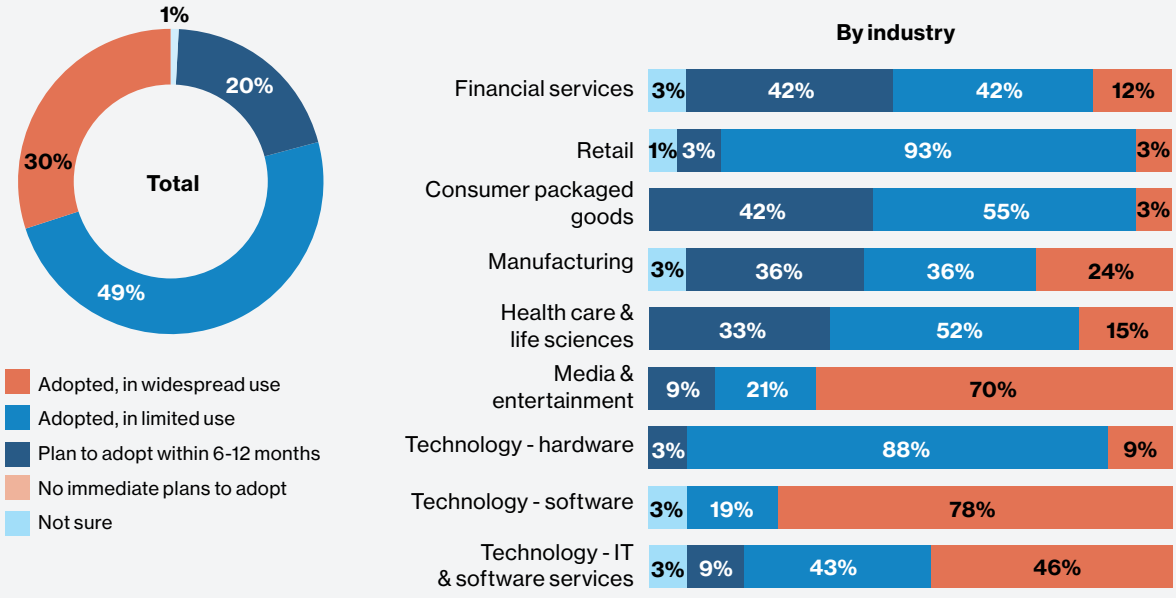
02 From performance-enhancer to game-changer



Software engineering is no stranger to agentic AI. Developers and engineers have been using AI tools known as AI assistants for over two years. Nearly four-fifths (79%) of the respondents in our survey say their engineering teams have adopted such tools, and almost one-third (30%) are using them widely.

Adoption of AI assistants in two industry verticals – the technology sector, particularly software producers, and media and entertainment – is much more advanced than in others (see Figure 1). This industry divergence in the development of AI capabilities is a pattern that holds through much of the survey.

Figure 1: Nearly four in five engineering teams have adopted AI assistants for specific software tasks while 30% say they use them widely



Source: MIT Technology Review Insights survey, 2026

Such tools have helped developers become more productive in executing discrete software tasks. Examples include Claude Code, Cursor, and GitHub Copilot for coding, and Applitools and Testim for quality assurance.

While a boon to software development, the very term "AI assistants" hints at these tools' limitations in terms of their overall impact. They assist developers in specific tasks rather than taking on the execution of those tasks in their entirety. The work of those assistants is guided by the natural language prompts written by developers for the underlying large language models (LLMs).

Enter agentic engineering

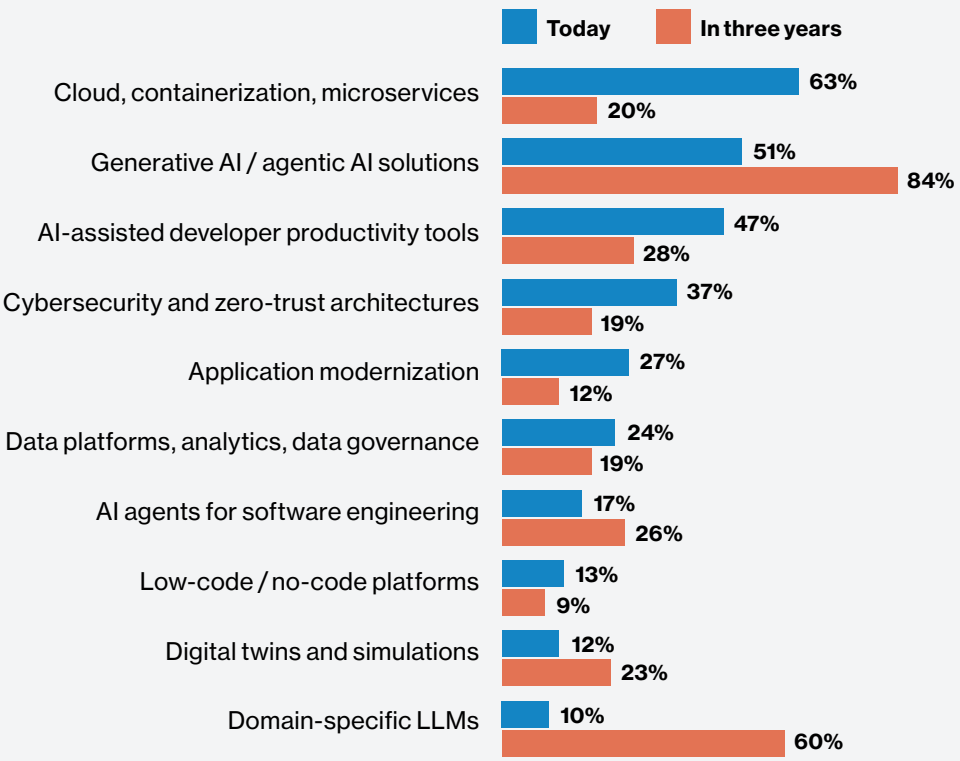
Agentic AI systems deliver a step change in capabilities when applied to software engineering. "What we call agentic engineering will bring a revolutionary shift in software management," says Serge Haziyev, chief technology officer of advanced technologies at SoftServe. "While AI assistants have helped individuals boost their own productivity, agentic engineering has potential to boost team productivity by an order of magnitude."

Thanks to considerable advances in LLM intelligence, AI agents can now execute more complex development tasks far more reliably than was possible just a year ago. Agentic engineering marks a shift from prompt engineering to context and objective engineering, moving toward a specification-first approach. This enables learning from previous executions, planning, acting, and refining in a largely autonomous manner.

With these capabilities, AI can automate entire tasks or lightweight user stories (brief, high-level descriptions of software requirements) across the SDLC, rather than just individual steps. "In time, this will lead to more end-to-end software product management, in which autonomous agents handle the majority of routine project work – freeing human teams to focus on higher-order decisions and innovation," says Shashi Prabakhar, director and Americas partner solutions architect at Amazon Web Services (AWS).

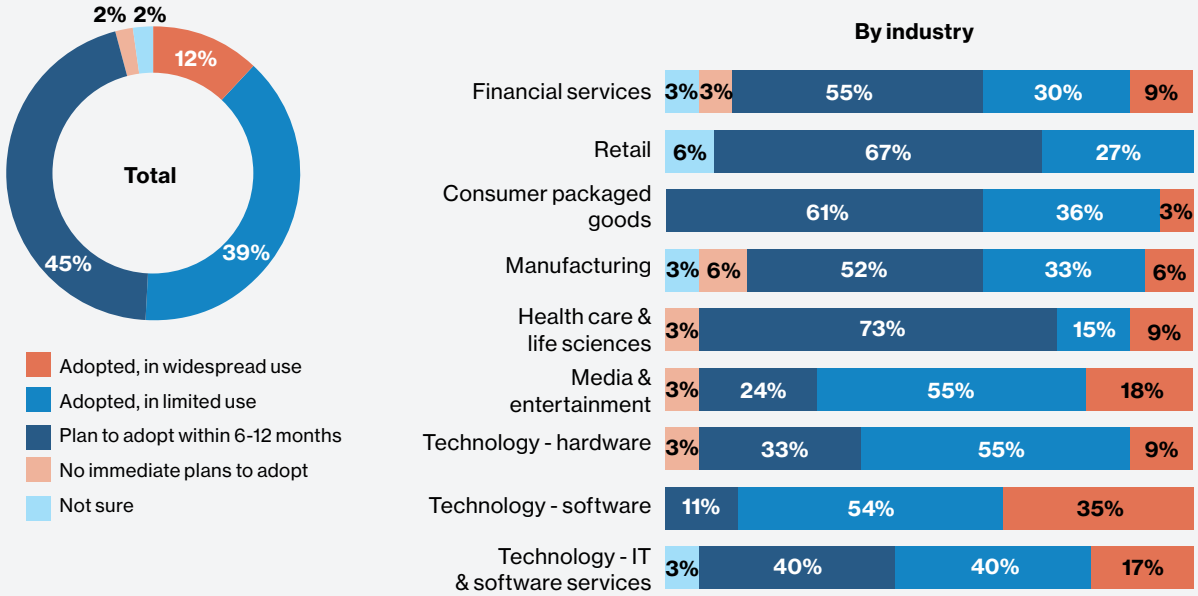
Judging by their technology investment intentions, the respondents' organizations understand agentic AI's

Figure 2: Half of software engineering teams prioritize investment in agentic and generative AI today with 84% expecting to do so within three years



Source: MIT Technology Review Insights survey, 2026

Figure 3: Just one in 10 say agentic AI is in widespread use in software engineering operations today while four in 10 say it is used in limited ways



Source: MIT Technology Review Insights survey, 2026

potential to deliver gains. Asked about their investment plans for software engineering, 51% list agentic AI among their top three priorities today, surpassed in frequency of response only by spending on cloud, containerization, and microservices. Three years from now, it will be a top priority for 84% of organizations, ranking far ahead of other tech spending objectives (see Figure 2).

That investment will drive accelerated adoption. Having become available only in the past year, agentic AI systems are in more limited deployment today than AI assistants. Just 12% of respondents report the widespread use of agentic AI today in software engineering operations, while 39% say it is in limited use. But 45% report plans to adopt it within the next six to 12 months. Here, too, software and media and entertainment businesses are the earliest adopters (see Figure 3).

The longer-term outlook

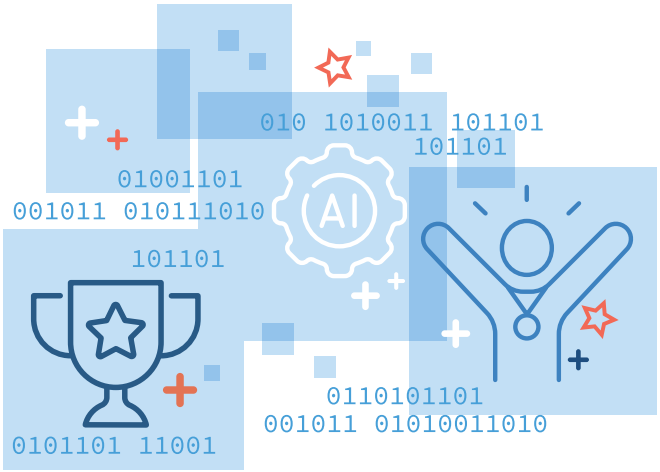
The experts we interviewed expect it will take much longer for agentic AI to diffuse throughout software engineering, but all are convinced that it will eventually prove a game changer. "I see it coming into widespread use around two years from now," says Rick Kazman, Danny and Elsa Lui distinguished professor of information technology management at the University of Hawaii. "In areas like code generation, testing, refactoring, and deployment, the changes that agentic AI brings will be revolutionary."

Nathaniel Barnes, portfolio chief technology officer at Hg Capital, is similarly bullish about agentic AI's long-term impact in software engineering. "I think it will eventually be ubiquitous in this field," he says. "The gains are potentially so transformative that software teams can't afford not to adopt it."

"I see agentic AI coming into widespread use around two years from now. In areas like code generation, testing, refactoring, and deployment, the changes that agentic AI brings will be revolutionary."

Rick Kazman, Danny and Elsa Lui Distinguished Professor of Information Technology Management, University of Hawaii

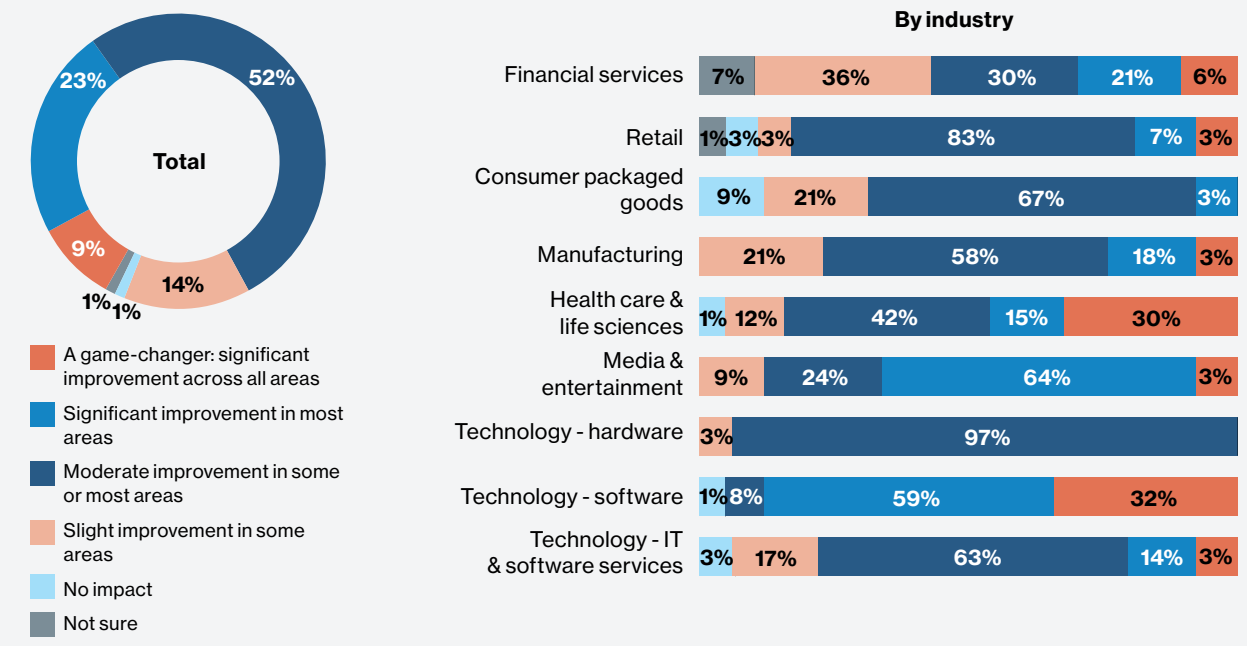
03 The gains to be won



It will take time for software teams' investments in agentic AI to start bearing fruit. For most in the survey, the expectations for software engineering improvements from agentic AI use are moderate (52%) or slight (14%) over the next two years. That said, nearly one-third (32%) believe the improvements will be bigger, and for a handful (9%) even game changing.

Software producers, media and entertainment companies, and health and life sciences organizations are most likely to expect major improvements (see Figure 4). Technology executives at the smallest organizations in the survey (earning between $250 million and $500 million in annual revenue) have higher expectations of a significant early impact than the largest ones (earning over $10 billion).

Figure 4: Most organizations expect agentic AI to moderately or slightly improve software engineering over the next two years while one third anticipate significant or even game-changing advances



Source: MIT Technology Review Insights survey, 2026

Many of these are software producers themselves, which arguably have greater incentive to deploy agentic AI than other types of organizations, given that software delivery is at the very core of their business model.

In what areas will those early gains manifest? At AWS, Prabakhar is seeing especially significant gains on the operations side of software engineering. "That's where the most monotonous work is concentrated," he says. "If it's monotonous, you can automate it." According to Kazman, one of those monotonous areas is documentation. "Nobody loves this," he says. "But agents do an amazing job of documenting their decisions and their chain of reasoning."

From coding to lifecycle management

The areas of greatest early improvement vary widely by industry, according to the survey. In verticals where early agentic AI adopters are more prevalent – software production, media and entertainment, and to some extent retail – coding comes top of the list. For consumer goods producers and manufacturers, quality assurance – the prevention of defects – is the most frequent response. And financial services providers and

> "Agentic AI is enabling us to accelerate prototyping. The pace of that has increased dramatically, so we're able to do more iterations of product requirements. The more iterations we do in the same time frame, the better the product quality will be."
>
> **Martin Vesper**, Managing Director and Chief Digital Officer, Pfeifer & Langen

Hg Capital: Experimenting more for better outcomes

Within its portfolio, Hg Capital has 58 companies in a diverse range of industries from services to technology to manufacturing. The software engineering teams at each of those companies can therefore differ widely in size, capabilities, and the roles they play in delivering on the overarching business strategy. That helps explain the different approaches those teams are taking toward the use of agentic AI.

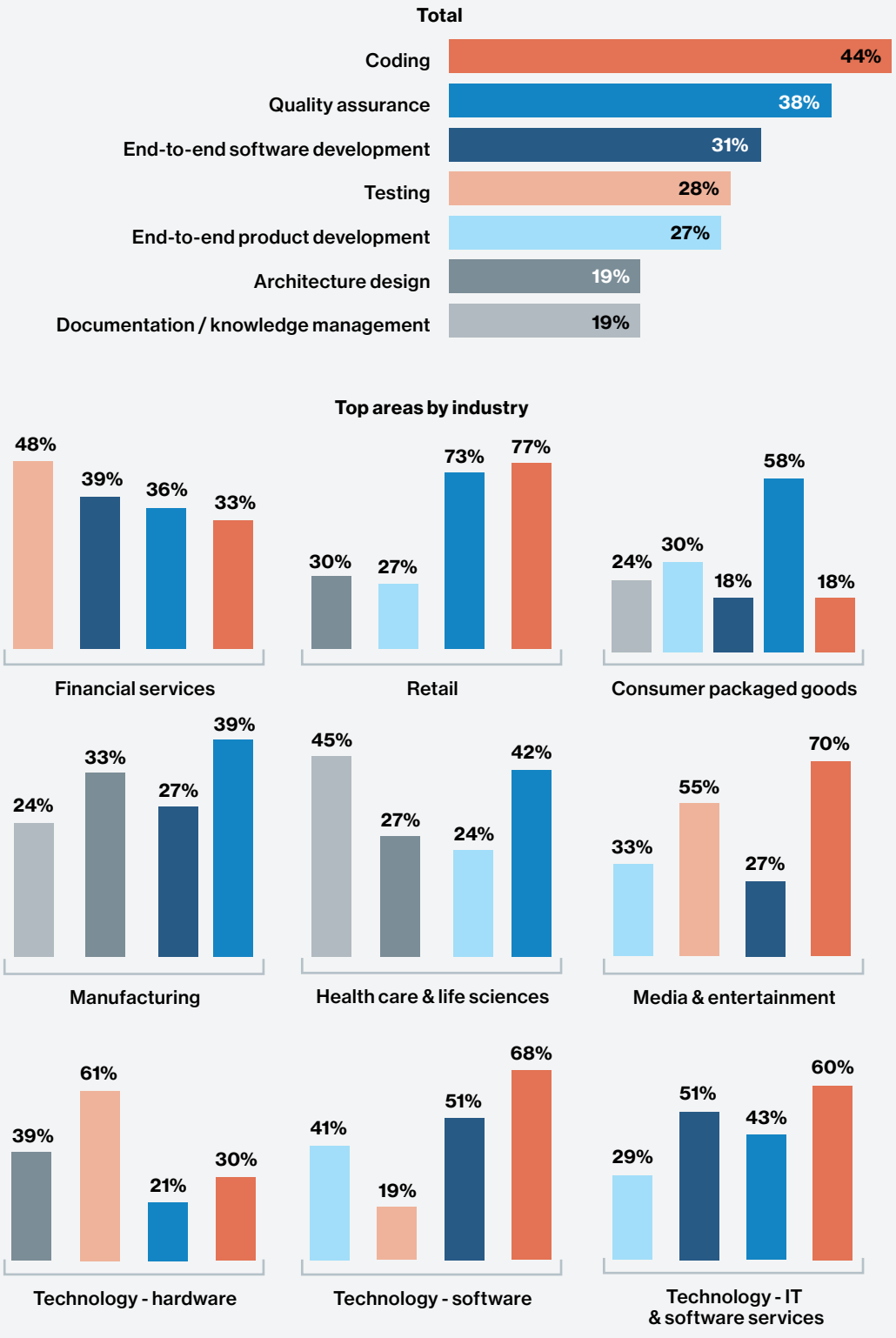
"We have some engineering teams that are highly invested in agentic AI and others that don't even want to look at it yet," says Nathaniel Barnes, portfolio CTO at Hg Capital. "The teams that are highly invested in it are getting big returns already, with productivity growth more than doubling in some cases."

One area of significant uplift for the engineering teams is experimentation, says Barnes. "We're seeing teams producing more but also experimenting more at the same time. That usually leads to better results."

Barnes provides an example: "I'll often see a developer get a story and cut it into specs for agents in different horizontal tiers, such as APIs, database migrations, or the front end. Now that developer writes a spec in different ways and sends those to a few different agents. That way, they can pick the spec they like best at the end of that process. So they're not only getting more raw output, they're also able to experiment better with different solutions in order to find the best result at the end."

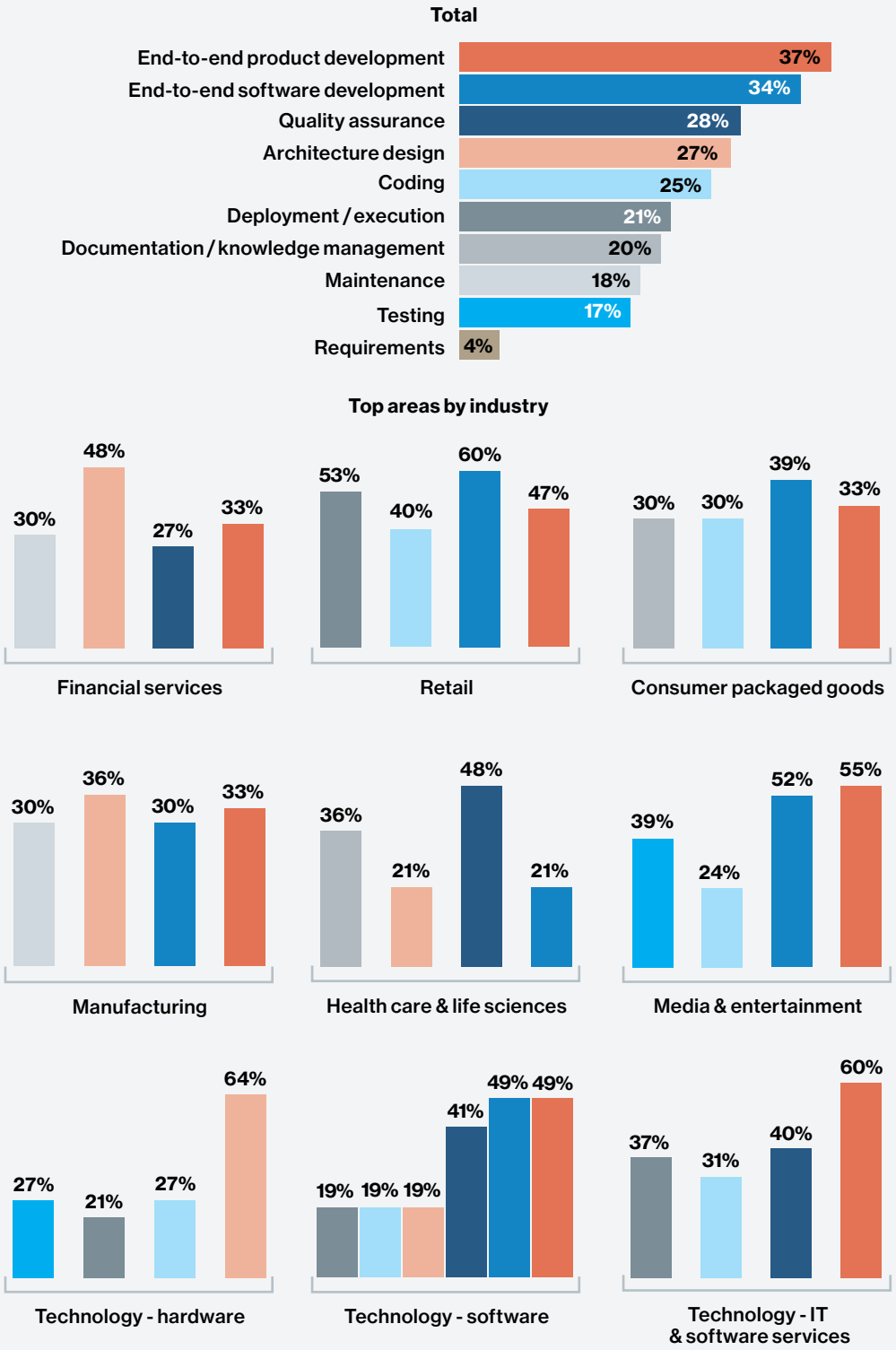
> "One area of significant uplift for the engineering teams is experimentation."
>
> **Nathaniel Barnes**, Portfolio Chief Technology Officer, Hg Capital

Figure 5a: Coding and quality assurance are the software engineering areas where respondents see the greatest early benefits from agentic AI in year one



Source: MIT Technology Review Insights survey, 2026

Figure 5b: Respondents expect to see the greatest benefits from agentic AI in overall lifecycle management by year three



Source: MIT Technology Review Insights survey, 2026

technology hardware producers emphasize the gains they expect in testing (the detection of defects). Health and life sciences organizations, meanwhile, highlight early benefits in documentation and knowledge management (see Figure 5a).

Further out (in three years), teams in early-adopter industries (software, media and entertainment) expect agentic engineering to be delivering its greatest value in overall lifecycle management. Around half or more of respondents from these verticals point to the speed, quality, and other gains their teams will be realizing in end-to-end software and product development (see Figure 5b). This is the point at which agentic AI use starts to become game-changing for software engineering.

Over the long term, speed and time to market will be the biggest gains agentic engineering delivers to the business. "It's the ability to build more features and more software products in the same time frame," says Haziyev. That view is supported by the survey results. Almost all respondents (98%) expect the speed of software engineering projects from pilot to production to increase over the next two years, with the average increase of 37% across the survey group (and considerably higher in the technology and media and entertainment sectors).

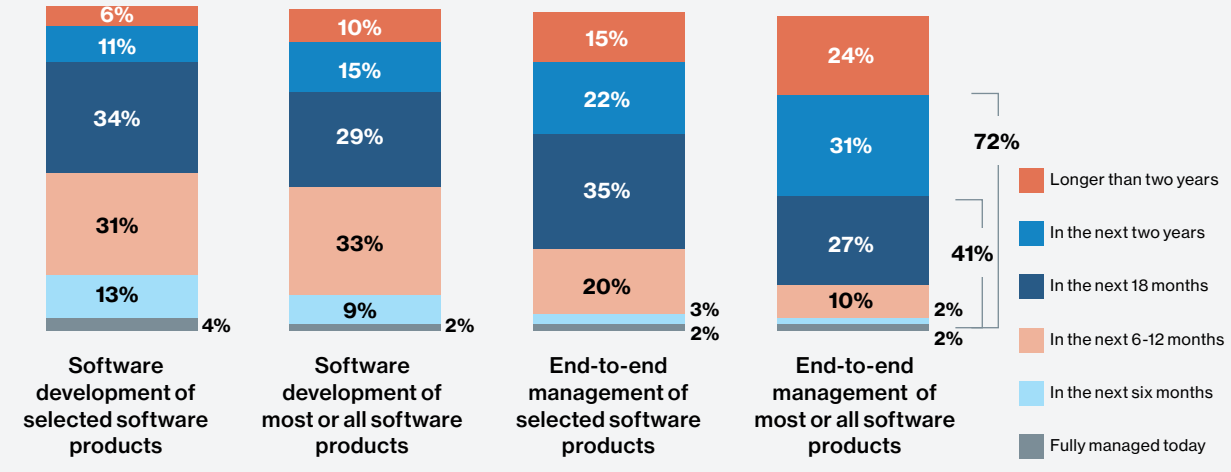
For Martin Vesper, chief digital officer at Pfeifer & Langen, a Germany-based food company, speed and quality gains go hand in hand. "For example, agentic AI is enabling us to accelerate prototyping," he says. "The pace of that has increased dramatically, so we're able to do more iterations of product requirements. The more iterations we do in the same time frame, the better the product quality will be."

In the long run, cost savings are also likely to prove a major benefit of agentic engineering. "Senior management is putting enormous pressure on project teams not just to be more productive, but also to cut the cost of software development," says Haziyev. "They want the same scope of work but with smaller teams. Agentic engineering has strong potential to enable this."

At present, the surveyed executives see the cost of agentic AI use as a significant challenge. But as agents assume end-to-end management of entire product lifecycles, project teams are likely to become smaller and less cost-intensive to support.

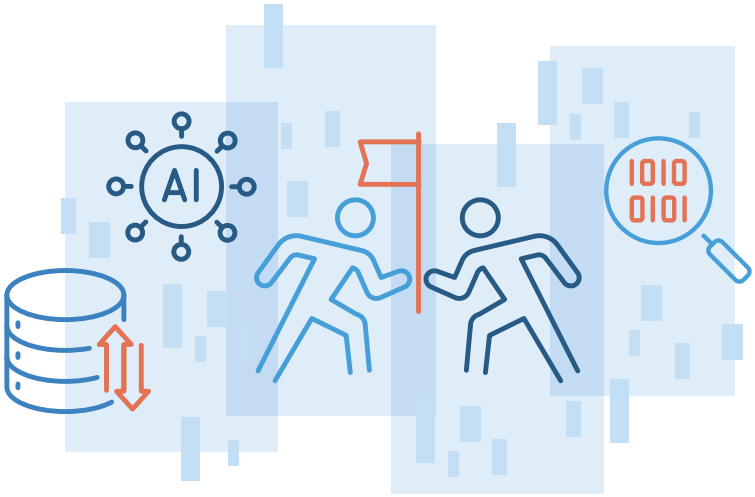
The software teams in our survey have ambitious plans to scale agentic AI to achieve full lifecycle management of their products. If their expectations prove accurate, in 18 months from now, AI agents will be managing most or all of those products end to end at 41% of organizations. Two years from now, agents will be doing this at 72% of organizations (see Figure 6). But software teams face significant challenges in achieving that objective in these relatively short time frames.

Figure 6: Four in 10 respondents expect agentic AI to be fully managing aspects of their software engineering operations in 18 months while seven in 10 foresee this in two years



Source: MIT Technology Review Insights survey, 2026

04 Challenges of change



Neither the survey respondents nor the experts we interviewed are under the illusion that scaling agentic AI in software engineering will be straightforward. The executives point to a wide range of factors that block their organizations' way to agentic engineering.

Industry-specific challenges

In some of the early-adopter industries – particularly media and entertainment and technology hardware – the foremost challenges are the cost of computing resources and agent integration. Elsewhere, governance issues like limited transparency and auditability of agents pose big concerns to financial services providers and IT services firms. Respondents in the software and health sectors say agent reliability is a key challenge to be addressed. And for consumer goods producers, human and process-related challenges – in talent and skills development and change management – are the chief concerns.

Notwithstanding the differing levels of emphasis, organizations across industries need to monitor and, if necessary, address all these issues to advance agentic engineering. For example, the costs of computing resources can easily escalate when teams are learning how and where to apply AI agents. Token usage is a potential blind spot. "You have to control for errors," says Vesper. "If you're paying as you go for cloud-based compute power and teams are doing things incorrectly, you can use up a lot of money fast." Barnes agrees: "This is a common source of cost overruns. A team might apply a high-level automation widely that worked once elsewhere and then find there are bugs in it. They'll burn through a lot of tokens without a lot of outcome."

As with any new technology capability, integrating AI agents with existing applications, especially legacy ones, is a major headache. "It's essentially a challenge of orchestration," says Prabakhar. "In large enterprises like ours, there are hundreds of development teams, thousands of APIs, and hundreds of third-party and homegrown applications. That creates enormous complexity."

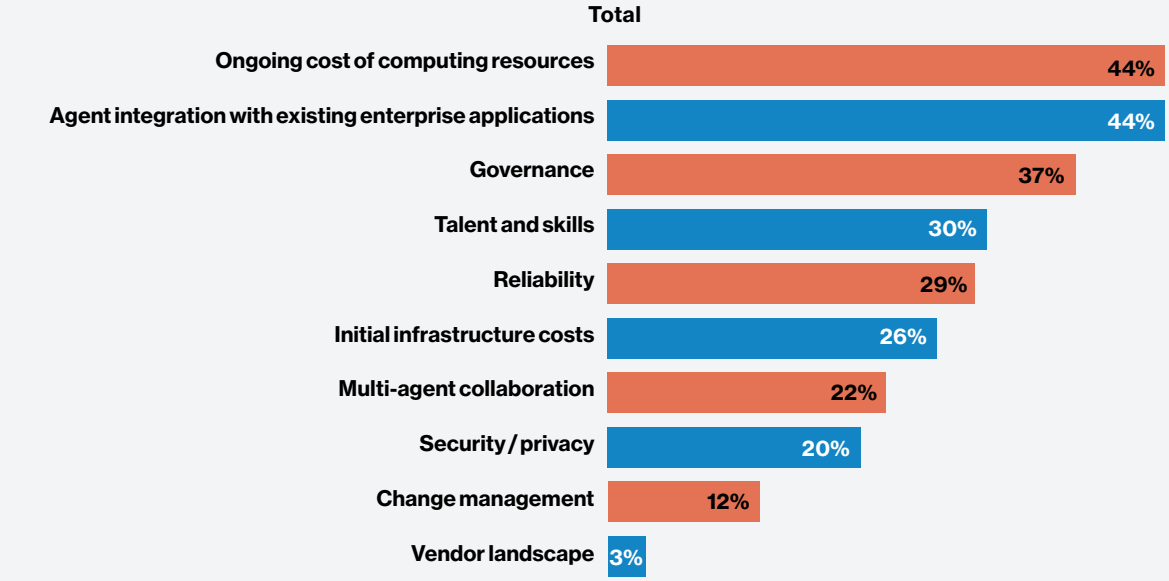
Multi-agent orchestration – agents collaborating with one another – is a related integration challenge. Orchestration agents exist to manage this but are not yet able to manage lifecycles end to end, partly because of that complexity. Haziyev concurs but also points out that agentic AI can simultaneously help to reduce integration challenges, as software modernization is one of its major use cases.

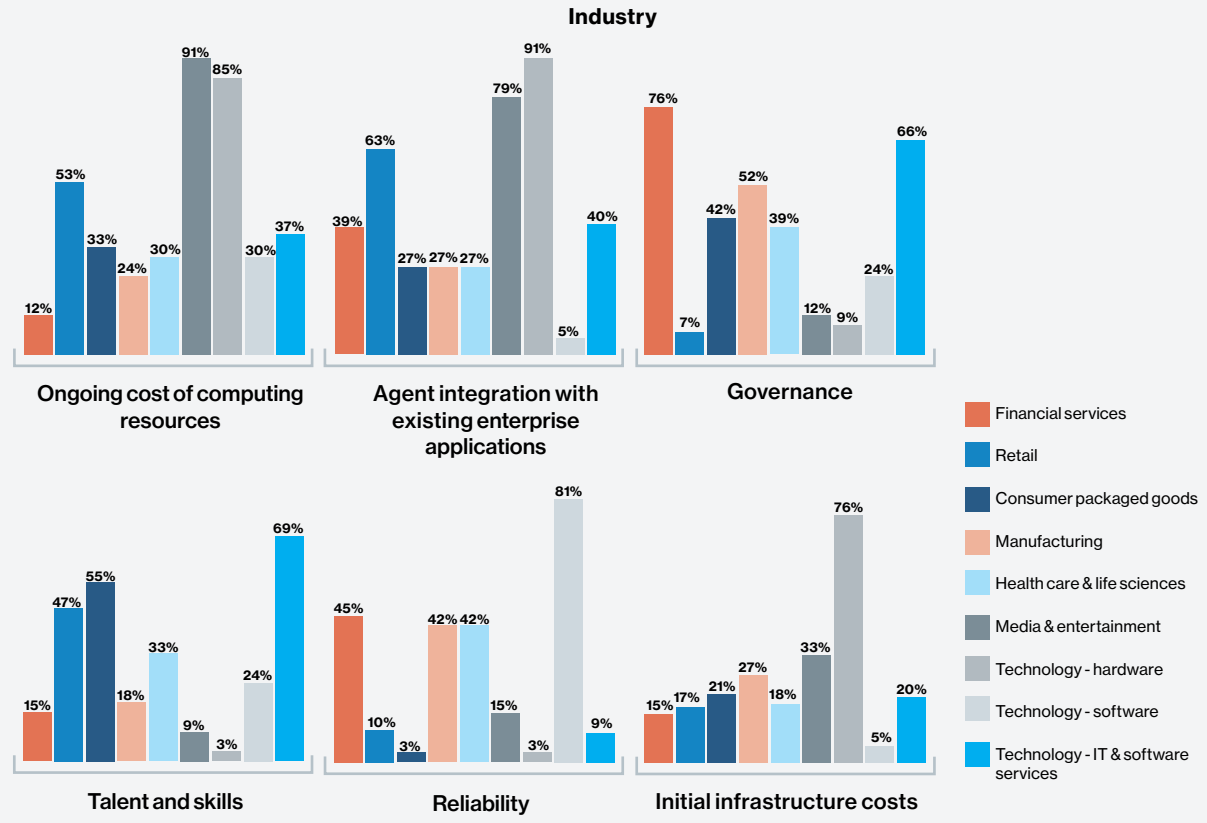
Governance and reliability

Governance of AI is not a new challenge, but the advent of reasoning AI agents adds another layer of complexity to it. "Most agents are currently unsuitable for autonomous operation," Kazman explains. "That means human oversight is a must for now, and there need to be rigorous and repeatable governance processes."

Limited auditability is by far the biggest concern of survey respondents from the financial services industry, where compliance and fiduciary responsibilities are heavier than in

Figure 7: Agent integration and the cost of computing resources are the biggest challenges organizations face with agentic AI in software engineering





Source: MIT Technology Review Insights survey, 2026

others. “As with any AI model, things will go wrong at times,” says Kazman. “There must be ways to find out the reasons and to put controls in place to avoid future occurrences.” Agent governance must, of course, be tried, tested, and proven before agents are allowed to run autonomously.

The surveyed executives are clearly worried about agent reliability and things going wrong, no more so than in the software industry itself. “Doubts about agent reliability, especially hallucinations, is a big challenge,” confirms Haziyev. “Any errors can be propagated widely during automated code generation. Reliability today is the biggest factor we need to improve in agentic engineering.”

Kazman points out that reliability concerns are a brake on wider agentic AI adoption, especially in safety-critical use cases. “If you’re designing air traffic or railway-switching control software, for example, the idea that agents could hallucinate is scary,” he says.

> “The change coming this time is transformational. We’re not talking about the need to adopt new tools. Software engineers have to rethink how they’ve been working as a workforce.”
>
> **Shashi Prabakhar**, Director, Americas Partner Solutions Architect, Amazon Web Services

Change management: What is different this time

Change management is not high on the list of agentic AI challenges for survey respondents (see Figure 7). Exceptions are the consumer goods and retail industries, where 48% and 27% of executives respectively, tag it as a major concern. It is foremost on the minds of the experts we interviewed, however, which suggests that the survey respondents may be underestimating the organizational difficulties that lie ahead in this area.

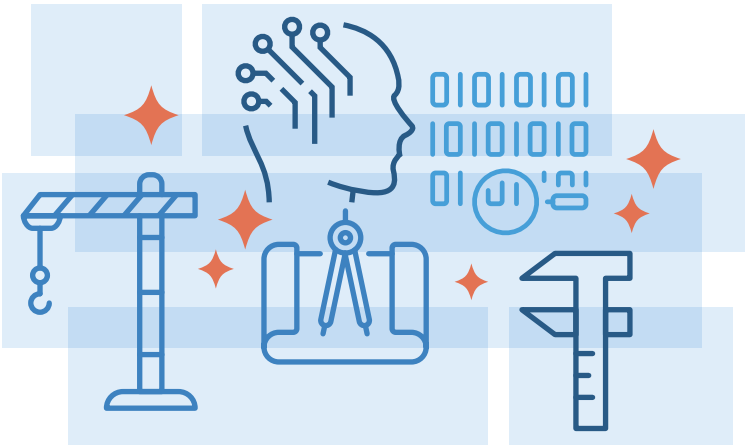
The change management challenge comes in the form of resistance in the software team to redesigning the workflow and the SDLC. “Scrum was a big process redesign,” says Serge Haziyev, chief technology officer, Advanced Technologies at SoftServe, of the agile team collaboration framework, “and that took 15 years to become mainstream. It will take a few years, possibly more, to make the process changes needed for agentic engineering.”

Nathaniel Barnes, portfolio chief technology officer at Hg Capital, agrees: “Much like the early DevOps movement, there is a fundamental change that has to happen at ground level in how people write software, and that’s not an easy thing to do. This is a big change management problem.” Barnes emphasizes that it is fundamentally an organizational and people management challenge, and not one of technology. “The companies in our portfolio who are good at agentic AI don’t necessarily have the newest tech stack. But they’ve fully embraced the change management aspect of it.”

Taking advantage of leaps in enterprise technology (think cloud computing and software as a service (SaaS), big data, containerization, and generative AI) has always demanded a rethinking of workflows and processes. For software engineering, however, the changes needed will be even more profound. According to Haziyev: “The complexity of change management with agentic AI is greater than with AI assistants, for example, in that we need to redesign processes so that the team, not just individuals, can be more effective.”

“The change coming this time is transformational,” says Shashi Prabakhar, director, Americas partner solutions architect at Amazon Web Services. “We’re not talking about the need to adopt new tools. Software engineers have to rethink how they’ve been working as a workforce.”

05 Roles rearchitected



There is no escaping the near certainty that agentic AI, as an advanced form of intelligent automation, will eventually replace human developers and engineers in the execution of several software-related tasks. But it is a nascent technology capability that will take some years for software teams to fully embed and scale.

Humans will remain in the loop, whether in governance, orchestration, or other roles, at least until teams are ready to allow agents to operate without supervision, if not longer. Multi-agent orchestration – the ability to organize the agentic workflow – will be in high demand. Especially at early lifecycle stages, agents need input, context, and guidance from engineers about the tasks they need to achieve. Meanwhile, new engineer roles are taking shape that aim to harness the potential of AI agents and expedite their adoption.

Wanted: AI-native engineers

For several years, software teams have placed a premium on hiring and developing engineers who are knowledgeable about DevOps, cloud platforms, and how to manage application development across the entire technology stack. The growth of AI is changing that.

For the past two years, organizations have prioritized the recruitment and development of DevOps, cloud, and full stack engineers. Looking ahead, however, organizations' focus will shift markedly toward engineers who are proficient in working with AI. Over half (51%) of survey respondents cite this as a priority role to fill in the next two years. The roles of software architect and data engineer will also grow in importance.

Haziyev offers his company's vision of AI-native engineering roles. "What we call the intelligence engineer is an AI-native role," he says. "The individual performing it is an operator of agents, able to configure them, feed them with proper context, evaluate and improve the agents' work. They will also be the human in the loop providing oversight and can also ensure that others in their teams are trained in this role."

"What we call agentic engineering will bring a revolutionary shift in software management. While AI assistants have helped individuals boost their own productivity, agentic engineering has potential to boost team productivity by an order of magnitude."

Serge Haziyev, Chief Technology Officer, Advanced Technologies, SoftServe

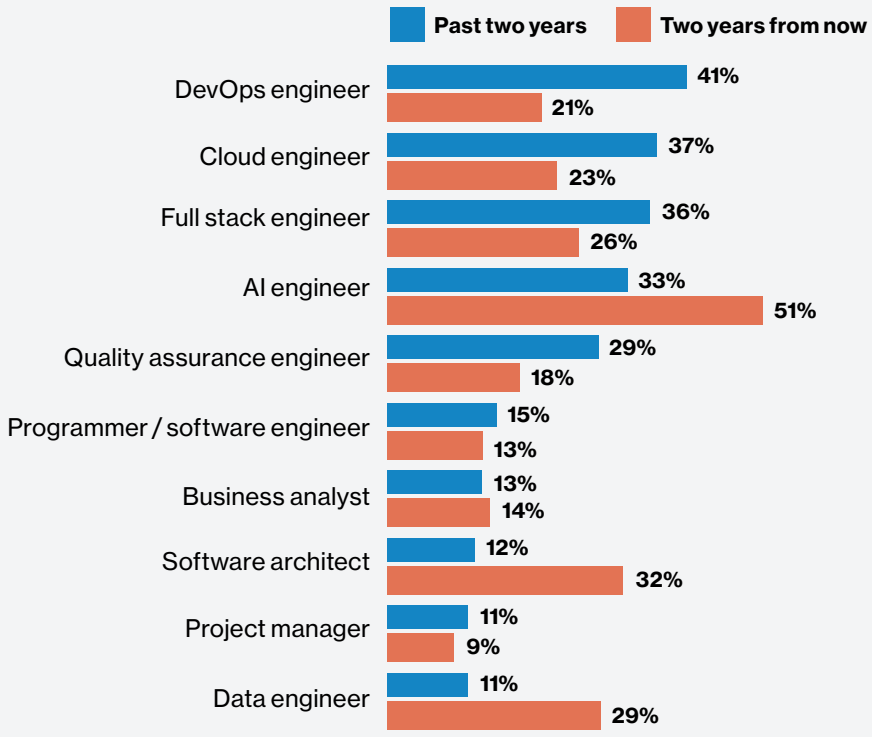
Reorienting roles towards ideation and creativity

If AI agents are assigned to the most monotonous and repetitive software tasks, engineer activity will likely gravitate toward the front end of software projects, believes Kazman. "This is where requirements engineering happens," he says, "with the emphasis on figuring out what the project team should build, specifying what's needed in a way that is clear and unambiguous. There will be less emphasis on programming and more on these and other skills like quality assurance."

Reorienting existing engineers toward roles that emphasize ideation and creativity should not be too complex, as roles in design and product management, for example, already exist within the engineering function. But AI-native engineers and software architects, among others, are newer roles that companies will find hard to fill easily, internally or externally. That may be especially difficult for companies in traditional industries. According to Vesper at Pfeifer & Langen, which produces and refines sugar: "We have plans in place to help staff upskill so that they can use agentic AI effectively, but they're for our entire workforce, not specifically for software engineers."

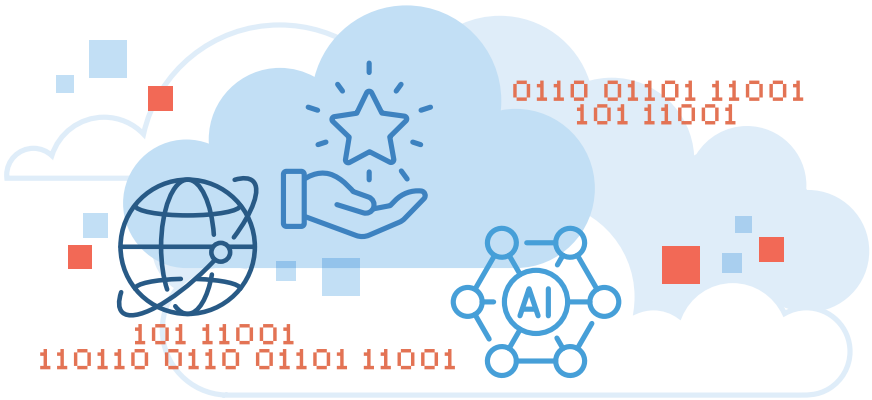
This is a challenge not just for organizations, but also for universities and other training institutions, says Kazman. "For decades, institutions' computer science and software engineering training has focused mostly on programming. We need to put more emphasis now on requirements engineering, prompt engineering, and quality assurance, for instance. Those are going to be key skills for software professionals."

Figure 8: AI engineer is the software role expected to be the highest priority in the next two years



Source: MIT Technology Review Insights survey, 2026

06 Dreaming bigger



Ultimately, agentic engineering is about more than faster and increasingly cost-efficient software delivery, as desirable as those gains are for the business. It is also about better software products. These are products that meet or exceed users' expectations for functionality and value, doing so reliably at consistent performance levels. They are also products that can potentially have a sizable impact on the results of a team, department, or even line of business.

Agentic AI has the potential to free up the time and space for developers and engineers to create high-quality products by removing the burden of monotony from their day. The extra productivity that agentic AI generates in software teams must be put to competitive advantage for the business. As Kazman says: "Agentic AI must empower teams to dream bigger."

Where organizations best leverage the capabilities of agentic AI, teams can dream bigger through:

Exploration of alternative approaches. Engineers should follow the precepts of design thinking, which involves making time to study and experiment with multiple design ideas, challenging assumptions, and iterating before converging on one tested and validated solution.

More and faster iteration of ideas. This involves reducing the time required for iterations by, among other things, breaking down tasks into small components and putting tight limits on feedback loops. Agents themselves can help by automating testing.

> "Agentic AI must empower teams to dream bigger."
>
> **Rick Kazman**, Danny and Elsa Lui Distinguished Professor of Information Technology Management, University of Hawaii

Wider, deeper collaboration. Engineers and developers must deepen collaboration with users. In-person discussions with users and in situ observation of their interaction with software can do more than lists of requirements to foster deeper understanding of user needs.

Learning from non-software domains. Fields such as graphic design, architecture, finance, aviation, manufacturing, and medicine offer lessons to software engineers on ways of ensuring product reliability, scalability, auditability, and other attributes of high quality.

Critical thinking. This is the soft art of asking well-crafted questions and challenging the assumptions of both team members and users alike. It also involves asking big questions, such as what success means, not just for the product, but also for the team or department using it.

These are the types of activities that, in a multitude of realms, lead to better innovation and enduring competitive advantage.

About MIT Technology Review Insights

MIT Technology Review Insights is the custom publishing division of MIT Technology Review, the world's longest-running technology magazine, backed by the world's foremost technology institution – producing live events and research on the leading technology and business challenges of the day. Insights conducts qualitative and quantitative research and analysis in the US and abroad and publishes a wide variety of content, including articles, reports, infographics, videos, and podcasts. This content was researched, designed, and written by human writers, editors, analysts, and illustrators. This includes the writing of surveys and collection of data for surveys. AI tools that may have been used were limited to secondary production processes that passed thorough human review.

About SoftServe

SoftServe is a digital engineering and technology consulting company specializing in AI, data, and cloud solutions. We decode complexities and reveal technology's full potential to rapidly unlock new possibilities for businesses, industries, and the communities they serve. Our global expertise is anchored in local understanding. From strategy to execution, we elevate technology into a force for growth, progress, and lasting impact. Visit the SoftServe website, blog, LinkedIn, and contact pages for more information.

SoftServe

Designed by Shultz Design Collaborative, LLC. Illustrations provided by Adobe Stock.

To cite this report, please use: "Redefining the future of software engineering," MIT Technology Review Insights and SoftServe, April 2026.

MIT Technology Review Insights

www.technologyreview.com

insights@technologyreview.com