

From AI-Assisted Coding to AI-Native Delivery

What the emerging model of software development means for technology organizations

Executive Summary

Generative AI has moved from an emerging developer experiment to a mainstream part of software development in remarkably little time. Leading technology organizations are now demonstrating what becomes possible when AI moves beyond coding assistance toward increasingly autonomous development, testing and delivery.

Yet AI adoption does not automatically translate into faster technology delivery. As AI accelerates software creation, constraints can shift elsewhere—to how work is defined, how teams are organized, how quickly work moves through enterprise controls, and how technology outcomes are measured.

A new model of software delivery is emerging: humans provide product intent and technical judgment while increasingly capable AI agents perform more of the execution. This model also requires an enterprise environment that enables teams to operate with greater autonomy through platforms, engineering context, automated controls and clear guardrails.

The question for technology leaders is therefore no longer simply whether our developers are using GenAI tools. It is whether the organization as a whole is changing fast enough to capture what GenAI makes possible.

This paper examines the emerging model of AI-native delivery and introduces READY, five practical markers that technology leaders can use to assess whether their organizations are moving toward it: **Ready to Adopt AI** · **Executable Work** · **Automated Flow** · **Directed Teams** · **Yield Outcomes**

1. AI Has Made Code Abundant

The transformation of software development by generative AI has happened remarkably quickly. DORA's 2025 research¹ found that 90% of technology professionals were using AI at work. OpenAI reported in June 2026² that more than five million people were using Codex every week, reflecting the rapid shift from coding assistance toward increasingly agentic software development and broader knowledge-work workflows.

¹ <https://dora.dev/dora-report-2025>

² <https://openai.com/index/codex-for-knowledge-work>

The frontier is moving even faster. Google reported³ in April 2026 that **75% of its new code was being generated by AI and approved by engineers**, up from 50% only months earlier. Google is also moving toward increasingly agentic development workflows.

And there are now examples of radically different engineering economics. AWS⁴ described a six-engineer team that delivered in 76 days work it estimated would previously have required roughly 30 developers over 12–18 months. The team used AI agents not simply to generate code, but to code, test, identify and fix bugs, and continue development around the clock.

These examples do not represent every technology organization. But they demonstrate how rapidly the frontier is moving. Software development has progressed quickly from AI-assisted coding to AI-enabled software development to increasingly agentic software development. So the question is no longer whether GenAI will change software development. It already has. The more important question is: **Is the technology organization getting faster?**

2. The GenAI Value Paradox

The early evidence is compelling. Yet the experience of many traditional enterprises is much less clear. Technology organizations have invested in AI coding tools. Developers are experimenting with increasingly capable models. AI usage is growing. Token consumption is rising. But leaders often still struggle to answer:

- Is our technology organization actually delivering faster?
- Where is GenAI creating value?
- Where is that value being lost?
- Are we producing more meaningful software—or simply more code?

The emerging research suggests that these are legitimate questions. DORA's research finds that AI can increase software delivery throughput while also introducing instability. Its broader conclusion is that AI tends to **amplify the underlying software delivery system** rather than substitute for it.

That helps explain the paradox. AI may make the **creation of software** faster without necessarily making the **delivery of software** faster.

The unit of AI investment is increasingly the token; the unit of business value is not.

Knowing how many developers use AI, how much code AI generates or how many tokens are consumed tells us something about adoption. It does not tell us whether the organization is delivering better outcomes.

³ <https://blog.google/innovation-and-ai/infrastructure-and-cloud/google-cloud/cloud-next-2026-sundar-pichai>

⁴ <https://aws.amazon.com/blogs/machine-learning/how-frontier-teams-are-reinventing-ai-native-development>

3. A New Model of Software Delivery Is Emerging

Across recent research and the experience of leading technology organizations, a common direction is emerging. AI is moving beyond code assistance toward increasingly agentic execution. Humans are moving from performing individual development tasks toward directing, reviewing and governing AI-enabled workflows. Work needs to be structured so that agents can execute it effectively. Delivery pipelines need to become increasingly automated. And organizations need to rethink how they measure technology performance.

- Microsoft's 2026 Work Trend Index⁵ describes a progression from humans creating work themselves toward **directing and orchestrating increasingly capable AI agents**.
- McKinsey's 2026 research⁶ describes leading organizations redesigning software delivery around near-continuous execution by humans and AI agents, with McKinsey reporting examples of **3-5x improvements in productivity and up to 60% reduction in team size**. Importantly, McKinsey attributes these gains not simply to deploying AI agents, but to rewiring the operating model so humans and agents can work together effectively.
- AWS provides a practical example of the potential: a small team working with AI agents can perform work that historically required a much larger engineering organization.

Taken together, these developments suggest a fundamentally different model for the technology team.

The AI-Native Team

The team has always been a fundamental unit of software development. Historically, a team needed multiple human specialists to turn an idea into working software: product analysis, development, testing, architecture and other engineering capabilities.

GenAI changes the economics of that model. The core human capabilities may increasingly center around **product intent** and **technical orchestration**, surrounded by AI agents performing more of the execution.

Product Owner

The Product Owner translates intent into **meaningful, shippable stories** i.e., owns ‘the what’ and ‘the why’: customer and business intent, priorities, desired outcomes and whether the resulting product meets the intended need.

Technologist

Owns the **how and quality**: technical direction, architecture and tradeoffs, orchestration of AI agents, engineering standards, testing, security, performance and overall production readiness

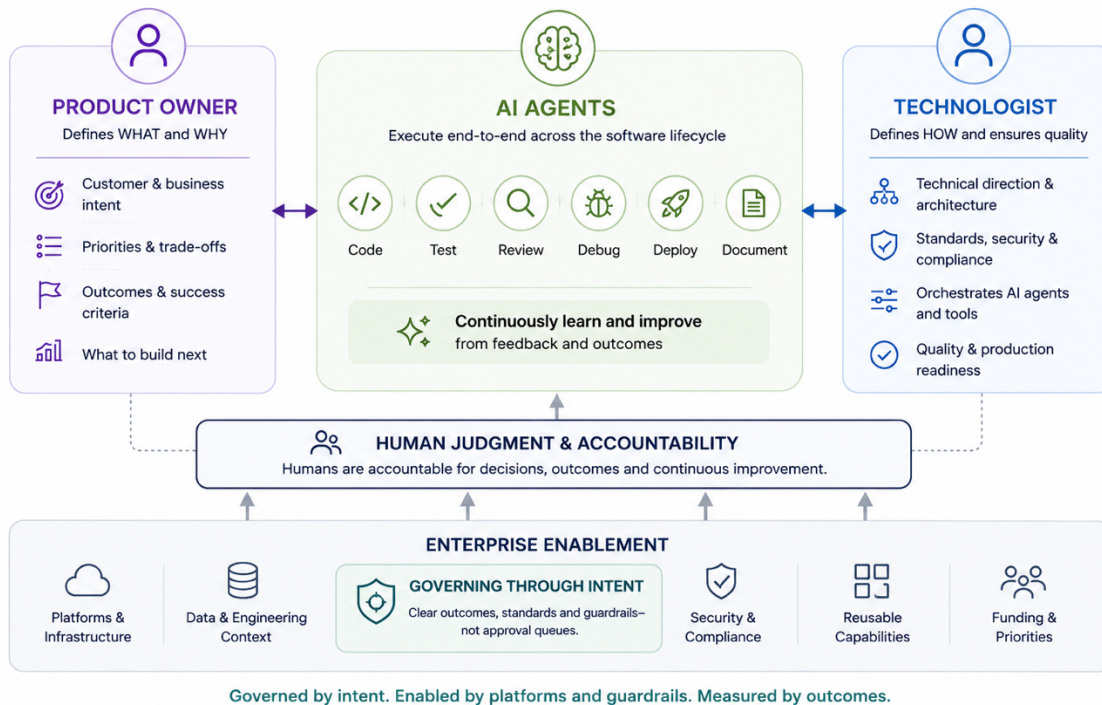
⁵ <https://www.microsoft.com/en-us/worklab/work-trend-index/agents-human-agency-and-the-opportunity-for-every-organization>

⁶ <https://www.mckinsey.com/capabilities/mckinsey-technology/our-insights/rewiring-software-delivery-for-the-agentic-era>

In some organizations, these capabilities may reside in one person. In larger enterprises, they may remain distinct. Around them, increasingly capable AI agents perform more of the execution: **coding** → **testing** → **review** → **documentation** → **remediation** → **deployment**

The AI-Native Team

Humans provide intent, judgment and accountability. AI agents execute.



The result is not simply a smaller version of today's software team - it is a different kind of team. This changes not only what the team does, but also **how humans, AI and the enterprise interact**.

The Human in the Middle

As AI becomes more autonomous, the human does not disappear. The human role changes. Instead of manually performing every step of software creation, the team increasingly:

- defines intent,
- establishes priorities,
- evaluates alternatives,
- makes tradeoffs,
- directs AI agents,
- manages exceptions,
- validates outcomes,
- and learns from what reaches production.

Microsoft's research describes this shift as humans moving toward directing and orchestrating AI rather than simply executing individual tasks.

This makes persistent teams particularly important. A persistent team accumulates product knowledge, customer context, technical judgment and organizational memory. It can learn continuously rather than repeatedly starting over as projects are formed and dissolved.

The more autonomous the engineering system becomes, the more important stable human ownership of intent, context and outcomes becomes.

The Enterprise Around the Team

The team cannot operate in isolation. If AI agents can generate and test software in hours, but environments are difficult to obtain, infrastructure requires manual intervention, security controls create queues, architecture decisions require repeated escalation or releases require multiple approval layers, the organization may simply move its bottleneck.

The enterprise therefore has an important role—but that role changes. It needs to provide:

- Platforms and environments
- Data and engineering context
- Security and quality controls
- Architecture standards
- Reusable capabilities
- Funding and priorities
- Risk guardrails

And governance itself needs to evolve. The traditional model often relies on control through approval. An increasingly autonomous technology organization requires more **control through intent, standards and guardrails**. The enterprise establishes what must be true and the boundaries within which teams can operate. The team determines how to achieve the intent. AI increasingly performs the execution.

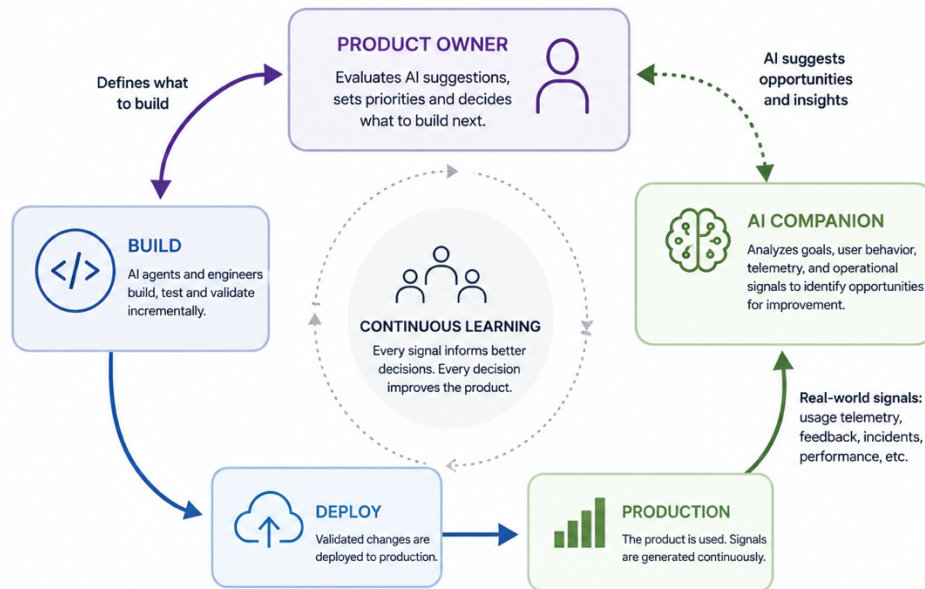
From Delivery to Continuous Learning

The future-state model can extend beyond software development itself. Today, a Product Owner may gather customer feedback, analyze production data and determine what should be built next. Increasingly capable AI could participate in that process as well. Recent research⁷ is beginning to describe AI systems using product goals, user behavior, telemetry and operational signals—not simply source code—as inputs to software development.

⁷ https://www.thoughtworks.com/content/dam/thoughtworks/documents/looking-glass-2026/tw_looking_glass_2026_accessible.pdf

From Delivery to Continuous Learning

AI turns real-world signals into improvement suggestions, acting as a companion to the Product Owner.



This is an emerging possibility, not a claim that every organization is operating this way today. But it illustrates the larger shift: **The future of software delivery may be less about humans performing each step of development and more about humans directing an increasingly autonomous system that continuously learns from production.**

4. The READY Framework

We have described a future in which a small number of human capabilities provide product direction and technical judgment, increasingly capable AI agents perform more of the execution, and the enterprise creates the environment in which that team can operate effectively. The question for technology leaders is: **Are we already building the capabilities required to move toward it?**

Many of the individual characteristics are already being discussed across the emerging research: AI adoption, agentic workflows, small batches, automated delivery, platform capabilities, human orchestration and outcome measurement. DORA, for example, identifies AI-accessible data, small batches, quality internal platforms and team performance among the capabilities that amplify AI's benefits.

TalentSavvy has synthesized these ideas into five practical markers that executives can use to assess whether their technology organization is moving toward AI-native delivery.

Ready to Adopt AI

Do your people, AI capabilities and technology environment enable meaningful AI adoption? AI-enabled development requires more than access to a coding assistant. Teams increasingly need AI across the development lifecycle—from code assistance and generation to code review, testing, debugging and increasingly autonomous coding agents.

Tools alone are not enough. Team members need the skills and working practices to direct AI effectively, evaluate its output, manage exceptions and understand where human judgment remains essential. The technology foundation also matters: environments, engineering context, build and test infrastructure, and appropriate security and quality controls need to support the new way of working.

Executable Work

Is work structured for AI-enabled delivery? AI can help developers execute work faster. It cannot make a large, tightly coupled project independently shippable. Demand therefore needs to be translated into small, meaningful increments that can be understood, tested and, where appropriate, deployed independently.

DORA similarly identifies⁸ **working in small batches** as an important capability for realizing AI's benefits. The issue is not whether an organization calls its work "stories." The issue is whether a team can take meaningful intent and move it independently toward production.

Automated Flow

Can validated work move rapidly into production? As AI accelerates software creation, downstream controls can become the new constraint. Testing, security, performance and compliance controls remain essential. But increasingly capable engineering teams cannot realize their potential if every control becomes a manual queue.

The goal is not zero governance. It is **governance without unnecessary friction**. AI is already expanding beyond code generation into testing, review and other parts of the SDLC.

Directed Teams

Do persistent teams have the ownership and capability required to direct AI-enabled execution? Increasingly autonomous development requires clear human accountability for intent, priorities, technical decisions and outcomes. Teams need to be persistent enough to accumulate context, cross-functional enough to act, and empowered enough to make decisions without unnecessary escalation.

⁸ <https://dora.dev/ai/capabilities-model/report>

This is consistent with the emerging human-agent models described by Microsoft and McKinsey: humans increasingly direct and orchestrate AI execution rather than simply perform individual tasks themselves.

Yield Outcomes

Can the organization tell whether the new model is actually better? AI usage, token consumption and code volume are measures of activity—not measures of organizational value. Technology leaders need to understand whether their organization is delivering:

- **Faster:** Lead time from work-item creation to production deployment.
- **More:** Meaningful work items deployed into production.
- **Better:** Net production defects.
- **Capacity:** The delivery capability available to the organization.

The objective is not to measure whether AI is busy - it is to understand what the organization delivered with the capability AI created.

5. What This Means for Technology Leaders

The transition to AI-native delivery will not happen all at once. Some organizations will need to strengthen their technology foundation. Others will discover that their teams are using GenAI effectively but that work is too large or tightly coupled to move quickly. Others may have strong teams and engineering pipelines but lack the measurement needed to demonstrate impact.

Every organization will start from a different point. But the direction is becoming clearer. The organizations that benefit most from GenAI may not simply be those with the best AI models or the highest AI usage. They may be those that can best combine:

- AI-enabled execution
- Human judgment
- Enterprise enablement
- Continuous learning
- Outcome measurement

About TalentSavvy

TalentSavvy helps technology organizations **build better capabilities, understand how their organization operates, and continuously improve technology delivery.** Our approach combines **advisory services** with **technology organization intelligence.**

Transform

Our advisory services help organizations build the capabilities required for modern technology

delivery—including effective team structures, ways of working, decision-making, demand management and enterprise enablement. The goal is not simply to redesign the organization, but to **develop the capabilities needed to sustain and continuously improve the new way of working.**

Manage

ManageTeam provides visibility into the technology organization—its teams, people, skills, workforce investment and organizational evolution—giving leaders an operational view of how their technology organization is structured and resourced.

Improve

ImproveTeam provides objective delivery intelligence to understand how teams and initiatives are performing, where delivery is constrained, and whether changes such as GenAI are improving technology outcomes.

Together, these capabilities help technology leaders **build stronger technology organizations, understand how they are operating, and continuously improve their ability to deliver.**