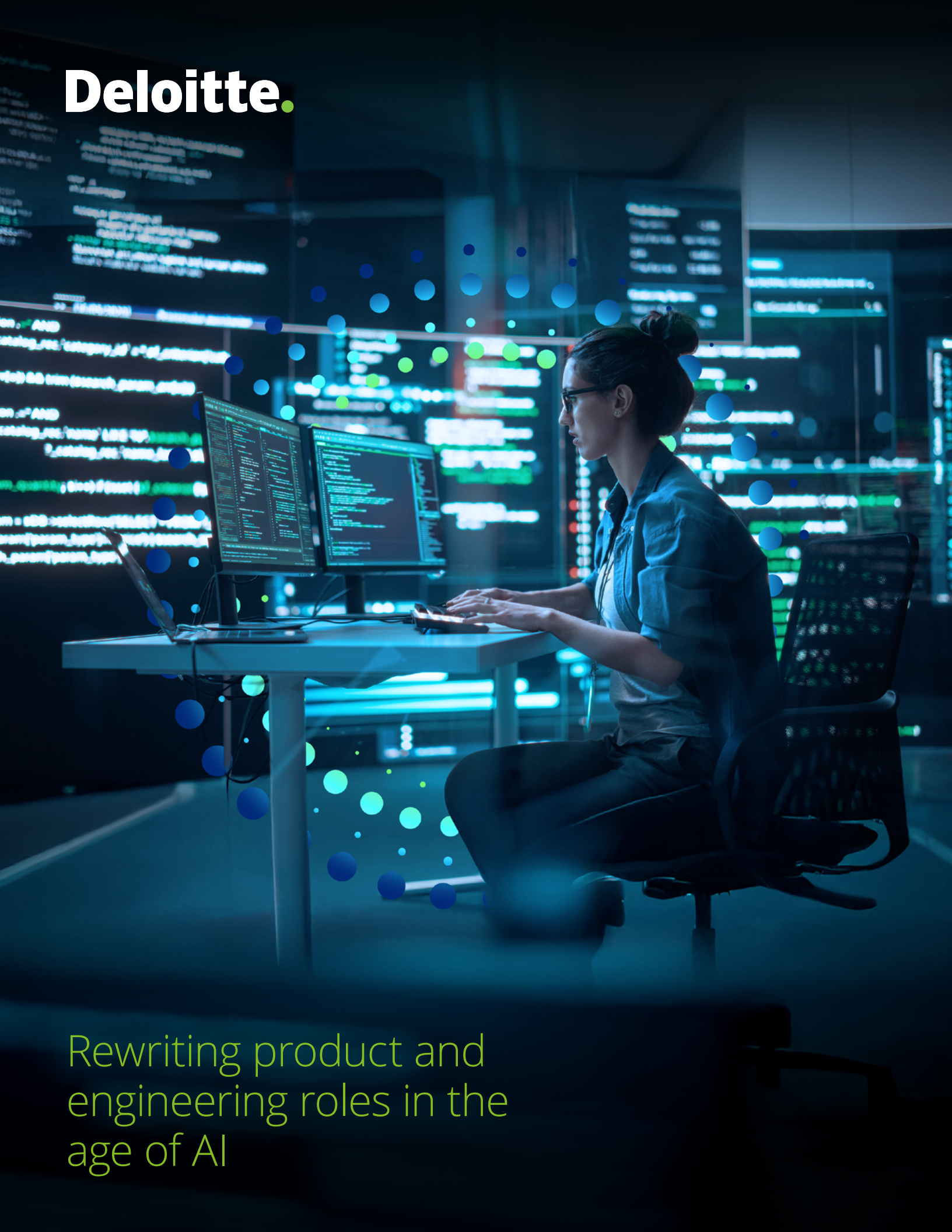
Deloitte.



# Rewriting product and engineering roles in the age of AI

Writing user stories. Troubleshooting code. Translating outdated software. Converting wireframes. All these tasks are central to the day-to-day lives of product managers, software engineers, and others involved in the software development life cycle (SDLC)—and are increasingly being augmented and accelerated through the use of artificial intelligence tools.

Generative artificial intelligence (GenAI) has the potential to disrupt workflows across the enterprise, but use cases at each stage of the software development life cycle are largely more advanced and have achieved greater adoption than most use cases in non-engineering functions. A recent GitHub survey of 2,000 globally distributed developers found that more than 97% of respondents had used AI coding tools at work, and 88% in the US indicated at least some company support for AI use.[1] Compared to broader adoption—where fewer than 60% of surveyed workers who have access to GenAI actually use it daily[2] —GenAI adoption has accelerated in software development.

The impact of this accelerated adoption is likely to be most profoundly felt in the tech industry. Nearly half of all developers work for tech companies, and leaders at these organizations are significantly more optimistic than those in other industries that GenAI will transform their organization within the next 12 months.[3] Within the tech, media, and telecommunications industry, scaled GenAI initiatives are most advanced in IT (34%), product development (17%), and cybersecurity (12%).[4] The reasons for this adoption are multifold: product and engineering talent are likely more tech literate and comfortable with new technologies; Generative AI is uniquely applicable to generating content; most models have been trained on vast amounts of software code; and the software engineering talent market remains constrained, to name a few.

As the evolution of AI tools and use cases continues to accelerate, a fundamental transformation of the work done by product managers, software engineers, and others involved in adjacent workflows is underway.

# The revolution is here: GenAI is already transforming the SDLC

The value proposition of GenAI in the enterprise will come from many potential channels, including value creation, enhanced customer or workforce experience, and efficiency or velocity gains. But as of the writing of this article, efficiency gains are the earliest path to return on investment, driven in part by market conditions and a tight engineering labor market. Many tech leaders have made bold claims about efficiency gains from AI, particularly in the software development life cycle, with varying implications for the size and shape of future engineering workforces. The increasing demand for software output and technical progress suggests this also presents a lot of opportunity to redesign or reallocate work, not cut out roles entirely.

As AI-enabled solutions enter the market, the greatest value has been realized when GenAI is embedded throughout the entire SDLC, rather than focusing solely on coding.[5] The following effects on roles across the SDLC have emerged:

### Product managers

At the top of the process flow, product managers are responsible for translating business objectives and requirements into actionable development work. They use GenAI to quickly transform business requirements into leading practice user stories, including acceptance criteria and tasks, built on context from prior stories and broader knowledge stores. Moving forward, humans will still need to be in the loop to review and edit the stories, but GenAI can significantly reduce the time it takes to draft and edit them.

### Architects

Next, software architects can easily edit, update, and publish diagrams generated by large language models (LLMs) and create standardized views for flow diagrams and architecture assets. These activities are often done manually (at a whiteboard or on a collaboration platform), so the ability to rapidly visualize, validate, and update diagrams and flow charts is a huge time-saver.

### Developers

The most mature solutions in the market exist for software developers, who are writing, reviewing, and improving code. GenAI accelerates development and improves quality by creating initial designs and code directly from stories, generating code that complies with enterprise standards, generating unit tests, assisting in troubleshooting and compliance checks, and migrating older code into new tech stacks. Perhaps just as importantly, LLMs are being used to research business context or support ideation of new approaches.

### Data Engineers

Data engineering is the backbone of AI and analytics, ensuring seamless data integration, transformation, and governance. Generative AI is revolutionizing this field by automating extract, transform, and load (ETL) processes; optimizing data pipelines; and enhancing data quality through anomaly detection. It assists data engineers in exploratory data analysis, interpreting data models, inferring attribute relationships, and generating complex queries. Additionally, GenAI tools are improving metadata management, automating documentation, and helping to make sense of real-time pipeline monitoring data.

### DevSecOps engineers

DevSecOps engineers are turning to GenAI to accelerate secure, efficient software delivery by building security into development and operations. Generative AI is automating threat detection, vulnerability assessments, and security policy enforcement. It also aids in code security analysis, compliance monitoring, infrastructure as code (IaC) script writing and validation, and real-time anomaly detection. By streamlining security workflows and reducing manual effort, GenAI is strengthening application security while accelerating development cycles.

### Quality assurance

While humans will perhaps always play the biggest role in quality review,[6] GenAI tools can reduce manual labor by generating synthetic data and test scenarios and publishing recommended results directly to Jira or similar tools.

**Figure 1.**

| Product engineering | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Product manager** | **Architect** | **Data** | **Developer** | | **Dev/SecOps** | **Quality engineering** | **Operate** |
| **Requirements gathering & project assets** | **Design & architecture** | **Data & analytics** | **Code implementation** | **Code transformation** | **Dev/SecOps** | **Testing & quality assurance** | **Support, maintenance, & operations** |
| 1. Creating a PRD | 7. Application and system analysis (code mining) | 15. Data mapping | 20. Authentication setup (use context) | 28. Code language/ version conversion | 30. Creating config file | 37. Test data creation | 43. L3 - Code generator |
| 2. Process flow creation | 8. Sequence/ Architecture diagram creation | 16. Data validation/ reconciliation | 21. Writing reusable utility methods | 29. API reconstruction | 31. Docker file for serverless development | 38. Test case creation | 44. Code change recommendation |
| 3. User story & acceptance criteria creation from epics | 9. Cloud architecture creation | 17. Script or query generation/ optimization | 22. API creation | | 32. CI/CD template | 39. Setting up automation framework | 45. Issue insight generation |
| 4. Tech & functional spec doc creation | 10. Initial data model | 18. Script or query conversion | 23. New code generation | | 33. Setting-up cloud/ infra-as-a-code | 40. Test automation | 46. L2 - Resolution recommendation |
| 5. Automated MOM & RAID log | 11. DDL creation (LLD/HLD) | 19. Data migration/ conversion | 24. Code documentation | | 34. Modules for GitHub actions | 41. Test environment setup documentation | 47. Ticket categorization and logging |
| 6. Epic and sprint status reporting | 12. Adding table relationships | | 25. Refactoring (optimization, linting) | | 35. Infrastructure automation | 42. Test result analysis | 48. L1 - Customer support chat |
| | 13. Defining table constraints | | 26. Unit test creation | | 36. Log analysis | | |
| | 14. Inferring schema from natural language | | 28. Defect fixing | | | | |

0 - <5% Efficiency gain  5 - 10% Efficiency gain  10 - 20% Efficiency gain  +20% Efficiency gain

Early proof points with clients and broader research signal estimated efficiency gains (see figure 1, derived from the early implementation of Deloitte's AI Assist tool at a global consumer brand).

It should be noted that these gains are not (yet) broad enough for GenAI to replace any one role—although the promise of AI agents might challenge that assumption.[7] Instead, these applications will likely lead to shifts in resource allocation or the combination of previously separate roles (e.g., as product managers can increasingly "vibe code," their roles may begin to converge with engineering), as individuals can accomplish more faster. The talent models of the past will likely not apply to this new world, requiring new approaches to organization design, hiring, career pathing, and upskilling. As AI-enabled solutions enter the market, the greatest value has been realized when GenAI is embedded throughout the entire SD me pipeline monitoring data.

# The future of software engineering: Three eras

Given this degree of disruption and model evolution just a few years into GenAI adoption, one thing is for certain: *Nothing is for certain.* It is therefore essential for tech leaders to engage in scenario planning to seek answers to fundamental questions about their product and software teams' potential size, shape, workflows, and ways of working.

**Key uncertainties to consider in planning include:**

1. **Is the pace of GenAI model maturity sustainable?**

   As of the writing of this article, Generative AI model development and advancement of agentic capabilities are evolving at a dizzying pace. The degree to which this is sustainable is perhaps the most significant uncertainty; AI companies are optimistic, but there are several hurdles that may hinder development.

2. **What will the GenAI ecosystem look like—more fragmented or integrated?**

   It remains to be seen to what degree the GenAI ecosystem (models and applications) for product engineering will be fragmented or integrated, closed or open source, etc. Speed of adoption across the entire development life cycle will likely depend on how easy it is to leverage tools within existing technology ecosystems and how fast it is to build necessary integrations.

3. **How will the AI regulatory environment evolve?**

   The regulatory environment regarding technology, and especially AI, continues to come into focus but is far from finalized. While these decisions remain in flux or at odds, certain use cases are likely to be avoided, especially by enterprises (for example, those relying on personally identifiable information or generating recommendations with life-or-death implications). Where companies choose to draw the line in the grey area may have as much to do with risk tolerance, culture, and competition as it does regulatory pressure.

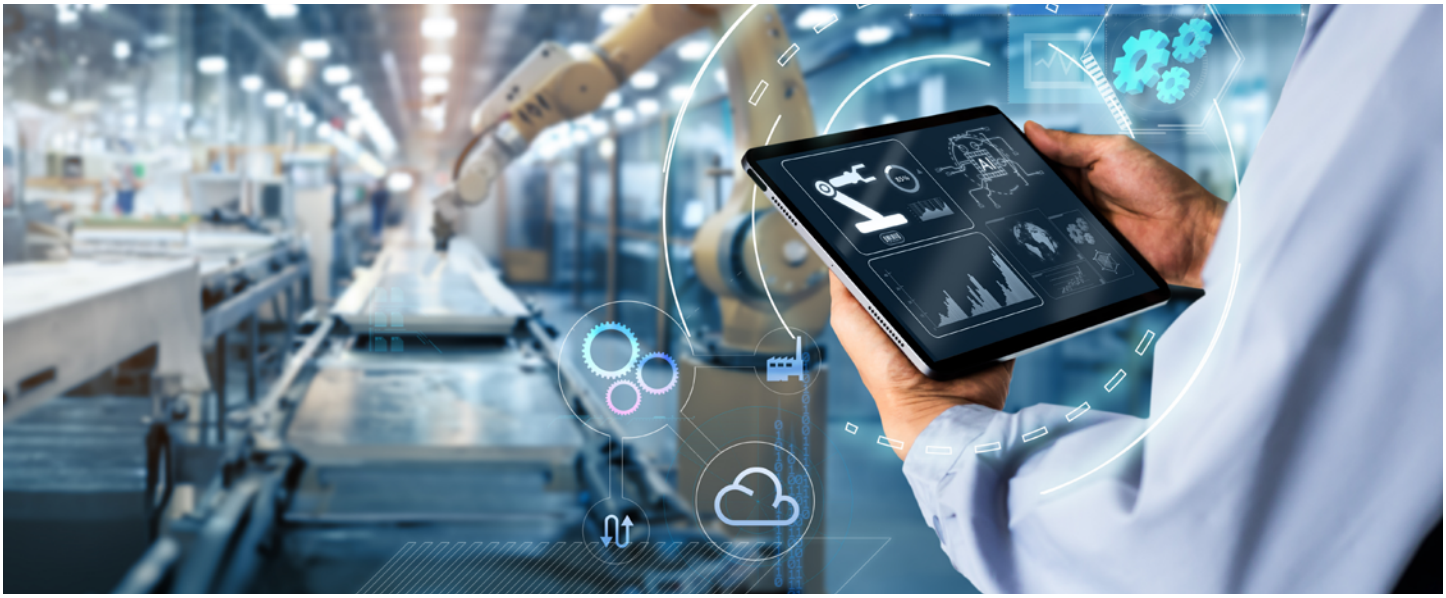
4. **Will infrastructure development keep up with demand?**

   The need for AI chips, energy, and other natural resources, as well as training data, is likely to continue to grow as model development and adoption grows. These resources are constrained and may cause a slowdown in development.

Along with these uncertainties, leaders should also consider the following framework to describe three phases of AI adoption in product engineering. They are not mutually exclusive; rather, it's likely that we will progress through most (if not all) of them over time. The time horizon to reach the later stages of maturity, however, could vary substantially—as a society, as individual organizations, and even within an organization.

**Figure 2.**

| | Era 1: Stable evolution | Era 2: Transformative automation | Era 3: Disruptive paradigm |
|---|---|---|---|
| | **Roles remain largely unchanged** | **Roles transform significantly** | **The entire paradigm of knowledge work is flipped** |
| **Role of AI** | • Enhances efficiency in discrete tasks across the SDLC<br>• Agents can automate or orchestrate a simple series of tasks, but without fully replacing roles<br>• Workflow-specific tools like GitHub Copilot accelerate code development | • Fully automates routine tasks<br>• Handles complex, multistep tasks<br>• Increases velocity of output<br>• Leveraged regularly to cocreate strategic artifacts | • Operates highly complex workflows autonomously<br>• Manages and evaluates its own performance<br>• Iteratively improves workflows<br>• Requires minimal human intervention |
| **Role of humans** | • Decision-makers<br>• Responsible for managing and delegating to AI tools<br>• Ensure ethical use of GenAI tools<br>• Heavy quality control | • Focus on strategic oversight and management of outcomes<br>• Cross-functional coordination<br>• Quality assurance<br>• Oversee end-to-end subprocesses | • Set strategy and prompt senior AI agents, which manage the rest of the process<br>• Conduct ethical reviews<br>• Work in tandem with AI on cutting-edge development<br>• Focus on customer relationships and stakeholder management |



### Era 1: Stable evolution

In this era—which more or less reflects the state of tech companies with mature GenAI adoption today—AI is an ever-present assistant that enhances capabilities but does not replace professionals, and humans remain the decision-makers across the SDLC. Product and software teams' role descriptions and responsibilities remain largely unchanged. While GenAI acts as a powerful tool to enhance efficiency by augmenting discrete tasks in the SDLC, it does not fundamentally alter workflows or automate tasks wholesale. The introduction of agentic AI leads to a degree of task orchestration and more complex automation, but with limited decision-making power. Some externalities that may sway society in the direction of a sustained stable evolution phase include significant government regulation, an inability to reduce hallucinations in foundation models, a lack of training data, and/or chip or power shortages.

An example of previous "stable evolution" in the tech industry occurred in 1989. Before the widespread adoption of integrated workplace tools, workers relied on discrete tools for creating documents, spreadsheets, and presentations. The introduction of Microsoft Office brought together powerful tools like Word, Excel, and PowerPoint, which greatly improved the efficiency of office workers and shifted approaches to communication and collaboration. However, roles were largely just redefined rather than redesigned or eliminated.

In the GenAI context, tools would accelerate code development and review—and free up capacity for workers to deliver more of their endless product roadmap faster—but not fundamentally shift the way work gets done.

In this era, leaders should focus on upskilling the existing workforce on the application of GenAI tools and effective methods for ethical use.

When defining a workforce strategy for this era, leaders should consider the following:

1. How can we quickly and iteratively upskill our teams within the context of their work?
2. How should we redesign processes to integrate AI into our workflows while preserving critical human oversight and recognizing that the technology will constantly evolve?
3. How should we measure the impact of AI use on productivity, velocity, and value creation?
4. How might our ways of working need to shift to take advantage of AI evolution?
5. What policies should we implement to ensure ethical and responsible AI use?

### Era 2: Transformative automation

In this era, the roles and responsibilities of product and software development teams transform significantly, but humans still play a significant role "in the loop." Routine and well-defined tasks like generating boilerplate code or test cases become fully automated. And autonomous multi-agent systems take over large portions of more complex, open-ended, multistep tasks like defining and performing integration, functional, and user acceptance testing. This evolves existing roles into hybrid positions who will take on broader sets of responsibilities and focus further on high-priority capabilities such as strategic oversight, cross-functional coordination, and quality assurance. The premium for human judgment and creativity goes up.[8]

While roles become more strategic and new roles are created, this will likely result in either a need for smaller teams overall or a far greater velocity of output by similar-sized teams—with entry-level or routine roles the first to become less critical. This era likely occurs if the current mathematical models and assumptions underpinning today's GenAI hit a ceiling and are unable to evolve into artificial general intelligence or superintelligence.

An example of previous "transformative automation" in the tech industry can be seen with the advent of cloud computing. Before the widespread adoption of cloud services, companies relied on managing their own physical servers and infrastructure, which required substantial labor and resources. The introduction of cloud computing brought together powerful services that greatly improved the efficiency of operations and shifted approaches to resource management and application development. This transformation allowed engineering professionals to focus on more strategic tasks—such as optimizing cloud resources, ensuring data security, and developing innovative applications—while creating entirely new roles and responsibilities dedicated to cloud computing and management.

In the GenAI context, introducing agents into the way work gets done would fundamentally change, with significant workforce implications. For example, consider a user interface flow where users are getting stuck and need to backtrack. With current processes, this would require manual interpretation of application monitoring to identify the suboptimal user behavior, review UI/UX options to correct the pattern, develop and test the fix, and queue the update in an upcoming release.

When defining a workforce strategy for this era, leaders should consider the following:

1. How should we redesign roles and restructure teams to effectively combine human capabilities with the increasingly autonomous capabilities of multi-agent systems?
2. What new skill sets will be required? How should our talent acquisition, learning and development, and performance management systems evolve to support them?
3. How might we structure governance and decision rights to realize the benefits of automation while ensuring humans are in the loop for critical decisions?
4. What risks might arise from a reduced need for certain workforce segments? How do we prepare our workforce for these shifts?
5. How do we address regulatory and compliance challenges that may emerge as core workflows are further automated?

### Era 3: Disruptive paradigm

In this era, artificial general intelligence (AGI), defined by one tech company as the "hypothetical intelligence of a machine that possesses the ability to understand or learn any intellectual task that a human being can,"[9] comes to pass. This may sound like science fiction, but it is a future that leaders from tech to government increasingly believe could become reality in a matter of years.[10] This era does not easily lend itself to predicting how many roles may change, let alone software engineering. The entire paradigm of knowledge work could be flipped on its head.

However, it is likely that in this era, AI can operate highly complex workflows autonomously and with greater efficiency and velocity than human workers today. AI would also be able to manage itself and evaluate its own performance, iteratively improving workflows and outputs with limited human intervention. This would suggest three priority roles for humans: first, setting strategy and prompting the "most senior" AI agents who would then design and delegate a majority AI "workforce"; second, reviewing processes and outputs for ethical outcomes where human intervention is legally required or ethical evaluation is particularly difficult; and third, highly specialized engineers and researchers working in tandem with AI on the cutting edge of AI development.

For example, an agent may realize that the cost of software is becoming too high and destroying profitability, initiating a workflow to compare the cost of refactoring code to negotiating new contracts with vendors. The agent may then determine that the cost of refactoring limits potential return on investment, leading to the initiation of negotiation with the organization's cloud vendor. Assuming the organization has entrusted the agentic system to make strategic decisions because it can do so better than, or almost as well as, humans for a much lower price, it is empowered to do so.

This era does likely result in significant workforce transformation (both in engineering functions and across the enterprise), requiring continuous reevaluation of enterprise business and operating models. To prepare for this era, it's important to actively work toward the two previous, and put in place the appropriate workforce programs and ethical standards to progress in a manageable way.

In defining a workforce strategy for this era, leaders should consider the following:

1. How can we create meaning and continue to support our human workforce in a world nearing or embracing AGI?
2. What role will human leadership play when AI can independently optimize and even self-improve operational processes?
3. How might we redesign operating models, teams, roles, and skill sets when AI systems assume responsibility for managing complex workflows? Where do humans need to be in the loop?
4. How can we ensure responsible and ethical choices when decision-making is increasingly driven by AI systems? What safeguards should be in place to ensure accountability and transparency?
5. How can we develop reskilling programs and alternative career paths for employees?

# Workforce recommendations for product and engineering leaders

So what does this all mean for today's product and engineering leaders? Regardless of where you are on this journey, the most important step is to continue moving in the right direction, at a pace that's right for your organization, with the awareness that GenAI is evolving rapidly and being adopted most quickly within the SDLC.

- **Encourage AI use:** First, and most importantly, create fluency and adoption programs within product and engineering teams to ensure the workforce feels comfortable using approved AI tools in the flow of work. This should happen in tandem with initiatives to create a culture of innovation, encourage risk-taking (within reason), create safe spaces for experimentation, and reward innovative behavior.

- **Elevate human capabilities:** In all the scenarios detailed above, the skills that make us essentially human will become more and more important, given that AI is particularly good at automating more routine tasks. These skills include management, critical thinking, creativity, ethical reasoning, collaboration, and empathy. While these skills can be cultivated in your existing workforce, it's also important to bake these into your hiring profiles—to the extent they aren't already.

- **Evaluate AI role impact:** Get a head start on considering how specific roles in your organization might be impacted by GenAI and begin redefining those that are likely to be most impacted in the near term. This also informs hiring needs, upskilling, and career pathing, and may help not only with evolving the workforce experience but realizing efficiencies through effective workforce planning.

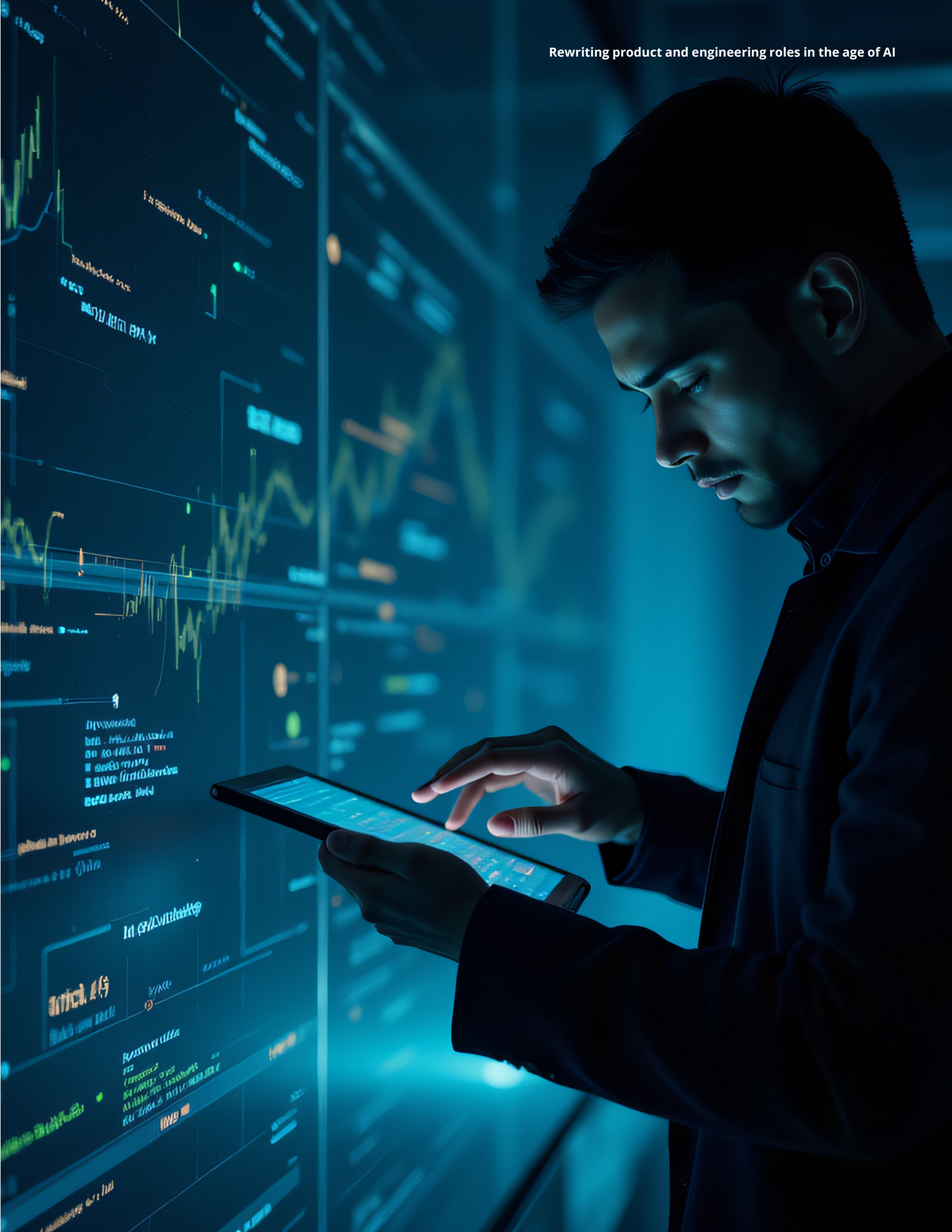
- **Cocreate alternative career paths:** As roles change and merge over time, employees will likely be looking at potential futures that are very different than how they entered the workforce. Creating a skills-based talent marketplace, redefining career paths, and providing visibility to the workforce can create tremendous value; and giving your workforce the agency to cocreate those futures is a win-win.

With great power, as they say, comes great responsibility. Leaders in the tech industry are likely to be the first to face widespread task augmentation and automation from AI, and they should take seriously the responsibility to prioritize the workforce experience.

The promise of Generative AI in the software development life cycle is significant and, in many cases, already proven. Leaders should begin taking action as soon as possible to prepare employees to take advantage of this tremendous opportunity, and to design an intentional future that can benefit all.

# Endnotes


1. Kyle Daigle and GitHub Staff, "Survey: The AI wave continues to grow on software development teams," GitHub, August 20, 2024.
2. Deloitte, *State of Generative AI in the Enterprise: Quarter four report*, January 2025.
3. Faruk Muratovic, Duncan Stewart, and Prashant Raman, "Generative AI for software development: Tech companies leading the way," Deloitte's *CFO Journal for the Wall Street Journal*, November 25, 2024.
4. Deloitte, *State of Generative AI in the Enterprise: Quarter four report*.
5. Deloitte, "AI Assist™: GenAI for software development," accessed June 2025.
6. Faruk Muratovic et al., "How can organizations engineer quality software in the age of generative AI?," Deloitte, October 28, 2024.
7. Sarah Hoffman, "Will AI agents join the workforce this year?," AlphaSense, February 19, 2025.
8. Sequoia, "From software engineers to AI word artisans: Filip Kozera of Wordware," *Training Data* podcast (episode 35).
9. Google Cloud, "What is artificial general intelligence (AGI)?," accessed June 2025.
10. Ezra Klein, "The government knows A.G.I. is coming," *New York Times*, March 4, 2025.



Deloitte.

**About Deloitte**
As used in this document, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.


11105286