In this BCG Executive Perspective, we articulate the potential of AI-enabled engineering excellence

Introduction – AI's potential in the software development lifecycle and the need for engineering excellence

We studied the current state and sentiment of how generative AI (GenAI) is transforming the software development lifecycle (SDLC) – through engagements with CIOs and CTOs, and through a market survey[1][2].

While everyone is experimenting, most organizations struggle to realize impact.

To unlock full value, GenAI must go beyond code copilot tool deployments: It must be embedded into core engineering strategies. This means tackling platform, tooling, process, and talent bottlenecks, not just deploying tools.

The opportunity: Treat GenAI as a catalyst for a broader engineering transformation – to boost productivity, accelerate delivery, and reinvent developer workflows.

Key questions for leaders:

- Are you seeing measurable impact from GenAI across the SDLC?
- Are you solving core bottlenecks – or just layering on another tool?
- Are you evolving fast enough to keep up with GenAI's rapid advancements?

This document provides a practical roadmap for CIOs and CTOs to cut through the hype and focus on what drives real value – today and tomorrow.

1. 60% CIO; 23% CTO; 12% Senior Leadership; 5% others, survey conducted January 2025
2. 27% SW Engineers; 24% IT Leadership; 16% ML Engineers; 14% other Engineers; 19% others, survey conducted January 2025

Executive summary | AI-enabled engineering excellence – from hype to 2X capacity

Current state	**Industry turning point:** >80% of companies now use GenAI for coding, yielding early gains (~5–10% cost savings, ~15% performance boost). However, **value remains spotty and limited to pockets** **Not a silver bullet:** GenAI isn't plug-and-play. Six challenges – a) the "toil paradox" (AI automates fun tasks but leaves grind work), b) low adoption, c) unclear ROI, d) tech debt, e) rapid tool churn, and f) org resistance – **are holding back full value**
Proposed approach	**Holistic transformation needed:** The promise is real, but leaders must **elevate their strategy** beyond deploying tools. A **holistic GenAI-driven engineering transformation** is required to unlock the full potential **Target weakest links:** Use GenAI gains as a **laser focus on bottlenecks** – e.g., modernize platforms, streamline processes, address talent gaps. By boosting the weakest areas, organizations can approach **2x engineering capacity**
Path forward	**From rollout to excellence:** Shifting mindset from a one-off "GenAI rollout" to an ongoing **"AI-enabled engineering excellence"** journey requires getting **five things right (5Rs)** – ways of working, tech stack, org setup, workforce skills, and continuous improvement loops

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Industry sentiment | AI in engineering – widespread adoption, shallow impact

GenAI code generation tool is increasingly becoming "table stakes"

>80% of companies have **paid / enterprise GenAI code generation** tool beyond POC stage

>75% of companies using GenAI to **write new code**

Early proof points show improvements in quality and satisfaction and freeing up capacity

5-10% typical **cost savings** for CIOs thanks to GenAI adoption[1]

15-20% improvement in **velocity, quality,** and **developer satisfaction**

2x ambition for future impact of GenAI in **next 3 years**[1]

However, most companies are still early in the journey to drive adoption

20% only ~**20%** of companies have widespread adoption (>75% developer access)

>6 mos. ~80% of slower-adopting companies will take **another 6+ months** to achieve 75% adoption

Key hurdles are unclear ROI, lack of trust, and general inertia

50% of CIOs see **unclear ROI** as a top three challenge for GenAI adoption

40% of engineers see **hallucination & lack of trust in output** as a top three challenge

60% of CIOs and engineers see **general inertia to change** as a top three challenge

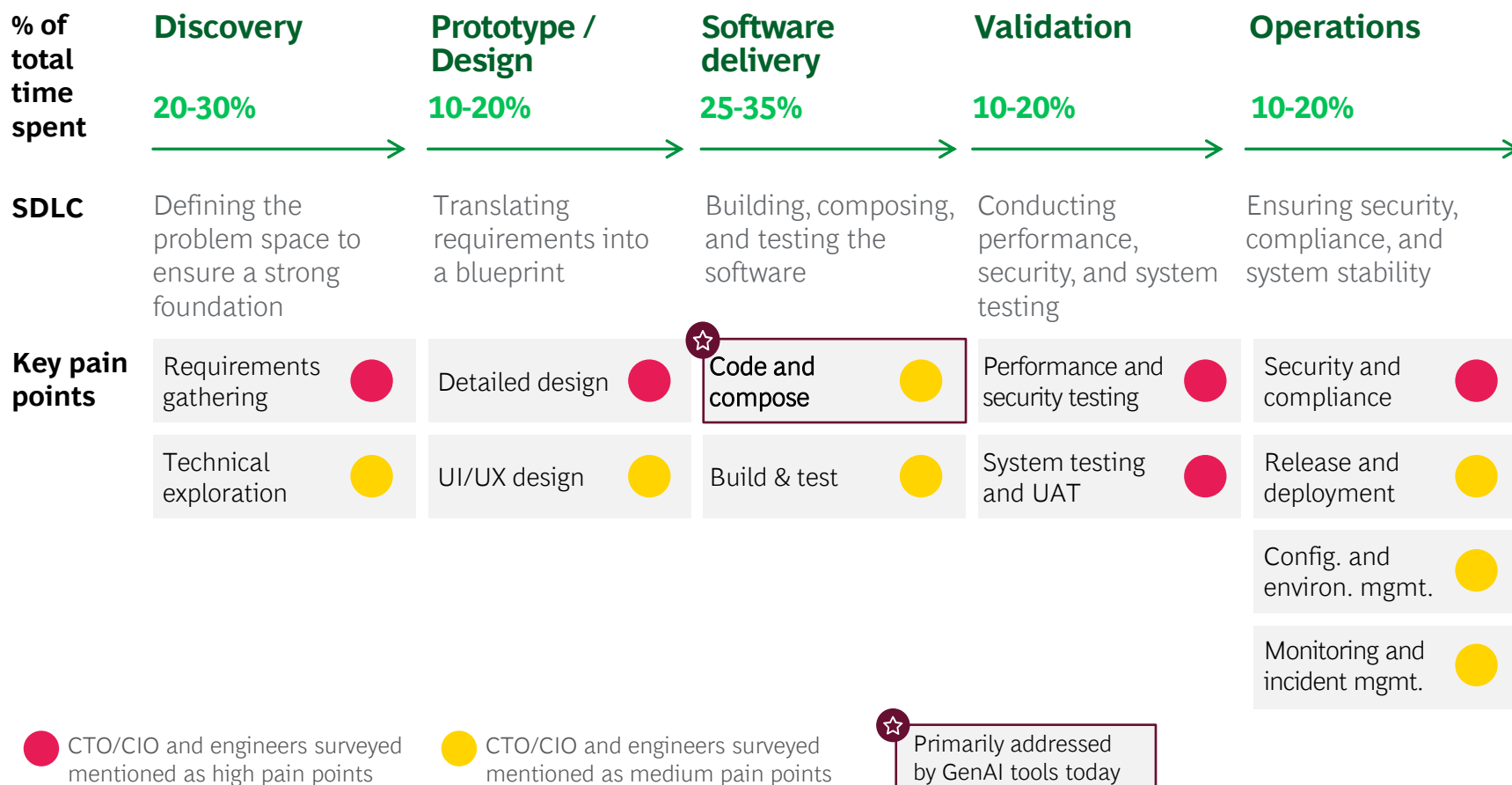## Security, testing, & reqs gathering are promising areas for GenAI beyond coding

10% of engineers state that **security and compliance, testing, and reqs. gathering** create most friction across SDLC

>33% of engineers desire a solution in **testing / debugging** if they could fix one area with GenAI

>60% of CIOs are at minimum exploring **GenAI agents** – many beyond coding

a	**The "toil paradox"**	**GenAI automates "fun" coding tasks** but leaves testing, security, and requirements as bottlenecks – limiting productivity gains. Unlocking its full potential requires both fresh innovation and a mindset shift
b	**Adoption lags**	**Only 20% of enterprises report >75% adoption.** Many treat GenAI as a tool rollout vs. new way of working – resulting in poor sustained uptake. A multi-pronged change approach is required
c	**Unclear path to measuring value**	**50% of CIOs struggle to quantify GenAI's impact**, reinforcing the imperative for end-to-end value tracking
d	**Tech debt dilemma**	**Outdated systems and poor DevOps severely dampen GenAI's impact.** Modern practices are needed to realize gains, and GenAI can help with the modernization
e	**Keeping up with speed of innovation**	GenAI tools evolve rapidly. Engineers get change fatigue with "tool churn" unless a **stable adoption flywheel is in place**
f	**Uncertainty in organizational structure**	**Traditional roles and team set-ups don't fit an AI-assisted world.** Leaders are uncertain how to reorganize for GenAI – but preparation must start now

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Pain points │ AI mostly tackles the "fun" coding tasks, but leaves critical steps as major bottlenecks – demanding innovation and a mindset shift

Engineering pain points: GenAI needs to evolve across the SDLC

% of total time spent	Discovery	Prototype / Design	Software delivery	Validation	Operations
	20-30%	10-20%	25-35%	10-20%	10-20%
SDLC	Defining the problem space to ensure a strong foundation	Translating requirements into a blueprint	Building, composing, and testing the software	Conducting performance, security, and system testing	Ensuring security, compliance, and system stability

Key pain points

Discovery	Prototype / Design	Software delivery	Validation	Operations
Requirements gathering 🔴	Detailed design 🔴	☆ Code and compose 🟡	Performance and security testing 🔴	Security and compliance 🔴
Technical exploration 🟡	UI/UX design 🟡	Build & test 🟡	System testing and UAT 🔴	Release and deployment 🟡
				Config. and environ. mgmt. 🟡
				Monitoring and incident mgmt. 🟡

🔴 CTO/CIO and engineers surveyed mentioned as high pain points   🟡 CTO/CIO and engineers surveyed mentioned as medium pain points   ☆ Primarily addressed by GenAI tools today

Engineers also need to evolve how they think of their role

❝ Engineers today find joy in writing code, but mental models need to evolve to enjoy the thinking part and let GenAI write the code."
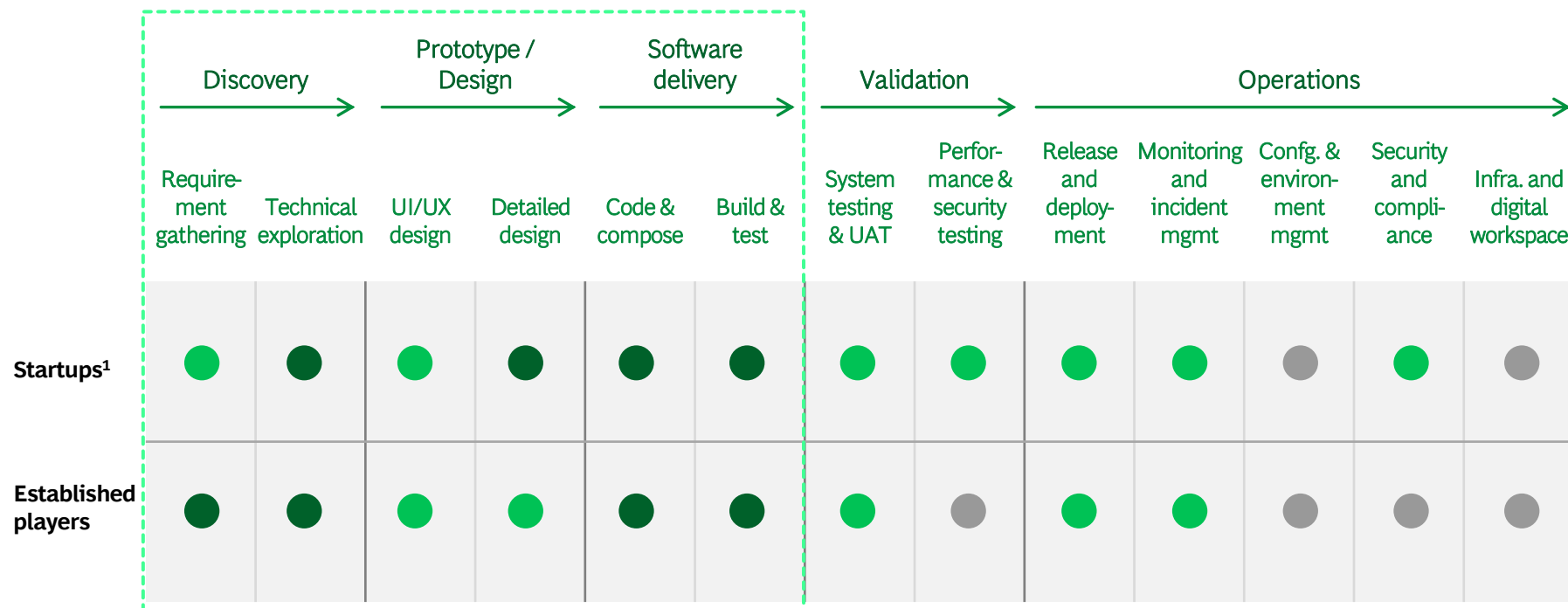
❝ An engineer's greatest skill now is knowing when to let the GenAI do the heavy lifting—and when to step in."

85% of companies expect GenAI to impact engineering jobs

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025
Note: Pain point allocation based on survey question "We would like to understand which phases of the software development lifecycle create the most and least friction for your enterprise"

Investments | Innovation hotspots: design, code, test – AI everywhere

	Discovery		Prototype / Design		Software delivery		Validation		Operations				
	Require-ment gathering	Technical exploration	UI/UX design	Detailed design	Code & compose	Build & test	System testing & UAT	Perfor-mance & security testing	Release and deploy-ment	Monitoring and incident mgmt	Confg. & environ-ment mgmt	Security and compli-ance	Infra. and digital workspace
Startups[1]	Moderate	High	Moderate	High	High	High	Moderate	Moderate	Moderate	Moderate	Minimal	Moderate	Minimal
Established players	High	High	Moderate	Moderate	High	High	Moderate	Minimal	Moderate	Moderate	Minimal	Minimal	Minimal

SDLC areas with current high innovation and different tool offerings (Discovery through Software delivery)

Level of innovation

- ● Minimal
- ● Moderate
- ● High
- ⬛ SDLC areas with current high innovation and different tool offerings

	Minimal	Moderate	High
Startups (investment in $, # of players)	< $50M, <10	$50-$250M, 10-20	$250-$500M+, >30
Established players (# of players)	<10	10-25	25-40

Source: BCG analysis as of Nov '24
Note: Effort is measured as # of companies and/or amount of investment flowing to companies addressing each SDLC area
1. Considering only privately held companies with fewer than 500 employees and less than $1B in revenue. Additionally, for funding, we are considering only last financing round if happened in the last 2 years. If a startup raised $100M and covered 4 different SDLC areas, the $100M is split between them, with $25M attributed to each phase

Code co-pilots are the hottest area of innovation - revolutionizing how developers write and review code

Code co-pilots are expanding upstream actively (incl. via agents) - integrating into planning, design, and agent-driven automation

Downstream expansion is slower - opportunities are in the space of testing and security

Too early days to pick winners - in some cases, enterprises that picked established solutions are now considering emerging solutions

Not all stacks are equal - existing tech stack (code repo, etc.) impacts efficacy of GenAI solutions

Adoption conundrum | Adoption remains uneven, with only 20% of teams reporting >75% uptake – current change efforts don't lead to stickiness
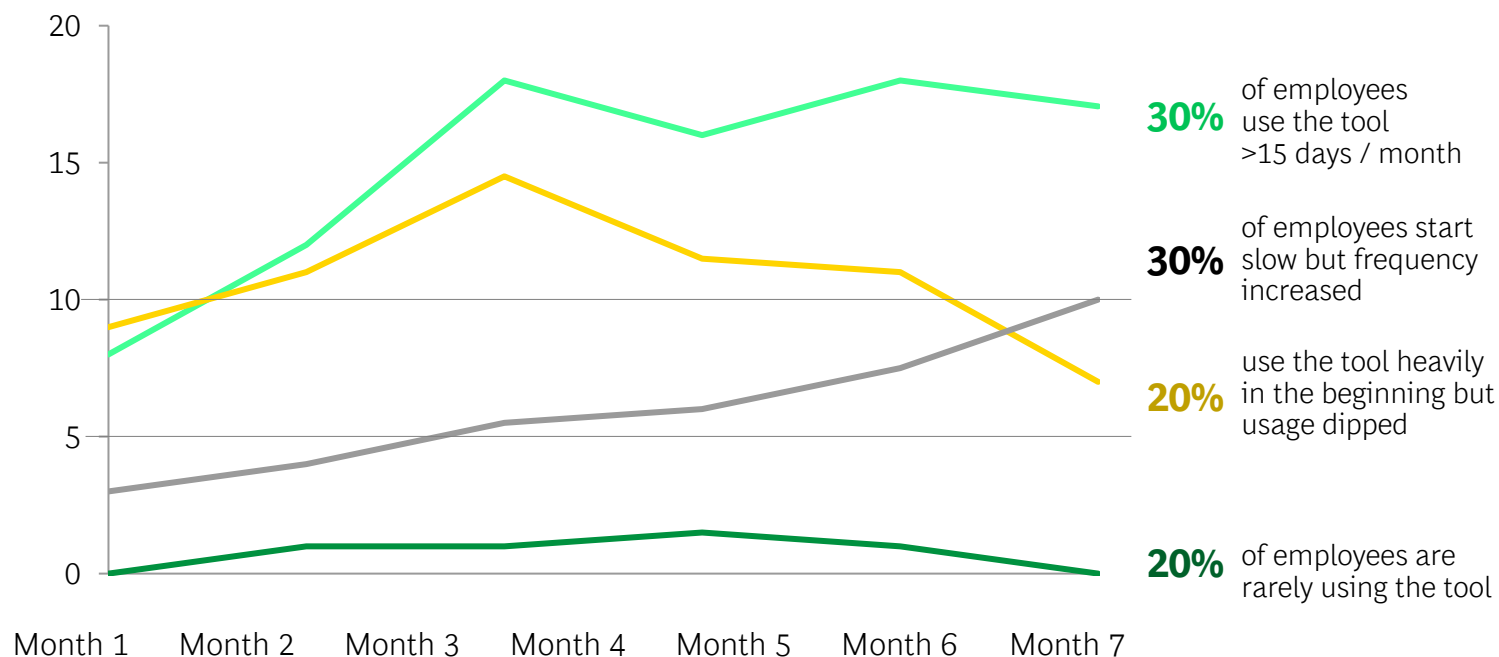
Level of GenAI adoption
% of CIO & engineers by FTE size, N=257

Only 21% report >75% adoption of GenAI

21%	
12%	
22%	
25%	
20%	

Companies with >1K IT FTEs
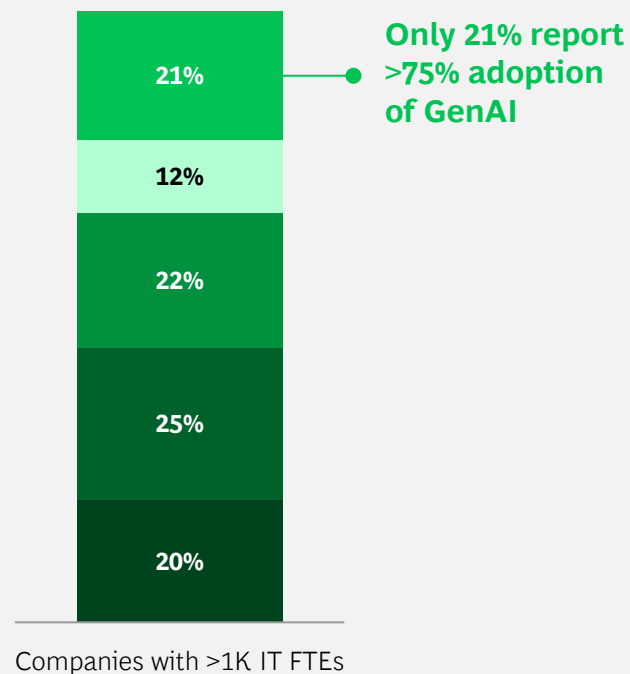
- ■ >75% (e.g., fully scaled to all engineers)
- ■ 51% - 75%
- ■ 26% - 50%
- ■ 10% - 25%
- ■ <10% (e.g., proof of concept)

Why is adoption hard? | Looking under the hood reveals pockets of adoption and a lack of stickiness despite change management efforts

Days of GenAI use per user per month (avg. by cohort)

Case example

30% of employees use the tool >15 days / month

30% of employees start slow but frequency increased

20% use the tool heavily in the beginning but usage dipped

20% of employees are rarely using the tool

GenAI adoption is not just a technology shift – it's a behavior shift
Sustained impact requires structured enablement, ongoing reinforcement, and a programmatic approach to drive consistent usage and long-term change

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Learning curve | Instead of simply rolling out tools, establish a continuous learning process that actively guides engineers up the learning curve

Learning curve of GenAI user adoption

1 The Cautious Beginner

Has heard the hype but skeptical about GenAI-powered coding

Occasionally uses chat, but **avoids GenAI-generated code; prefers traditional coding methods** and sees GenAI as a supplementary suggestion tool rather than an assistant

2 The Curious Experimenter

Dabbles in GenAI assistance but still prefers manual control

Uses code suggestions but manually pastes GenAI-generated code; prefers GenAI assistance on short code snippets over full-code generation

3 The Frequent User

Embraces GenAI-driven coding as a core part of workflow

No longer manually writes most code; accepts GenAI-generated code with light modifications; sees GenAI as a productivity booster, using it for rapid iteration and development

4 The Automator

Uses GenAI to generate, refine, and optimize code at scale

Uses **GenAI to edit and optimize multiple files** in a repository; leverages prompting to generate entire modules or refactor large sections of code
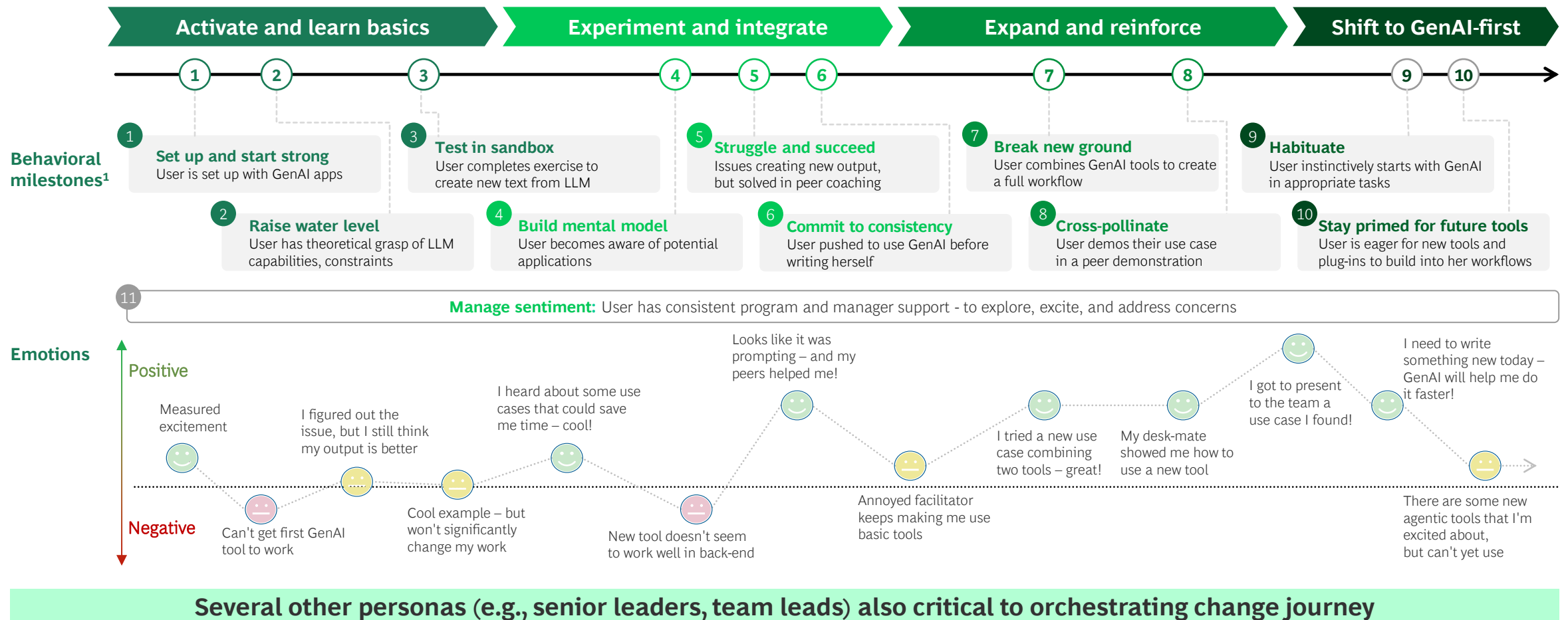
5 The GenAI Orchestrator

Leverages AI agents for complex tasks with minimal intervention

Shifts to agent lead with human support, where **GenAI agents drive test writing, test, and debug code** without manuals edits; emphasizes prompt curation instead of direct code generation

8

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Learning journey | Sustained AI use requires intentional behavioral interventions across the learning journey

Activate and learn basics	Experiment and integrate	Expand and reinforce	Shift to GenAI-first

1   2   3   4   5   6   7   8   9   10

Behavioral milestones[1]

1 Set up and start strong
User is set up with GenAI apps

3 Test in sandbox
User completes exercise to create new text from LLM

5 Struggle and succeed
Issues creating new output, but solved in peer coaching

7 Break new ground
User combines GenAI tools to create a full workflow

9 Habituate
User instinctively starts with GenAI in appropriate tasks

2 Raise water level
User has theoretical grasp of LLM capabilities, constraints

4 Build mental model
User becomes aware of potential applications

6 Commit to consistency
User pushed to use GenAI before writing herself

8 Cross-pollinate
User demos their use case in a peer demonstration

10 Stay primed for future tools
User is eager for new tools and plug-ins to build into her workflows

11 Manage sentiment: User has consistent program and manager support - to explore, excite, and address concerns

Emotions

Positive

Negative

Measured excitement

Can't get first GenAI tool to work

I figured out the issue, but I still think my output is better

Cool example – but won't significantly change my work

I heard about some use cases that could save me time – cool!

New tool doesn't seem to work well in back-end

Looks like it was prompting – and my peers helped me!

Annoyed facilitator keeps making me use basic tools

I tried a new use case combining two tools – great!

My desk-mate showed me how to use a new tool

I got to present to the team a use case I found!

I need to write something new today – GenAI will help me do it faster!

There are some new agentic tools that I'm excited about, but can't yet use

Several other personas (e.g., senior leaders, team leads) also critical to orchestrating change journey

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025
1. Behavioral milestones are not necessarily sequential

9

Holistic change program | A holistic change program requires multiple levers to drive adoption and sustain impact

Measurements and tracking	**Adoption** Measure number of users at each level of GenAI proficiency (e.g., beginner vs. frequent user)	**Productivity** Measure uplift in individual output (e.g., PR rate) & attribute / link to GenAI usage	**Quality** Measure accuracy and usability of output (e.g., number of bugs)	**Developer satisfaction** Measure user sentiment with the tools
Communications and purpose	**Strong mandate from C-suite** Reinforce GenAI as an urgent competitive need, not just an experiment	**Accountability** Link and communicate GenAI adoption metrics with team OKRs and budget		**Change ambassadors** Identify "micro-influencers" to share success stories and approach
	Communication cascade Coordinate series of messages across channels to reinforce sense of purpose	**AB testing** Continuously test and refine effective approaches to reinforce adoption		**Nudges** Deploy frequent (e.g., 3x weekly) nudges to encourage habit building & share useful tips
Education and enablement	**GenAI acceleration sprints** Intensive, multi-week program to apply GenAI to real workload & ingrain new habits	**Learning paths** Provide structured training paths, from beginner use cases to advanced agentic AI		**Scalable "train the trainer" model** Enable internal experts to become AI advocates running future workshops
	Hackathons Teams to try new tools, way of work & share knowledge while solving real problems	**GenAI knowledge hub** Build central repository with best practices, how-to guides, troubleshooting resources		**Community support** Form mentorship program and community channels (e.g., Slack) for knowledge sharing
Incentives	**Manager targets** Tie funding and business performance targets to GenAI adoption & impact	**Individual contributor incentive** Make GenAI adoption part of review; award special bonuses for top users	**Team gamification** Introduce GenAI adoption leaderboards and friendly competition between teams	**Scarcity** Reduce engineering budgets or headcounts to incentivize GenAI usage

Successful GenAI adoption requires a multi-faceted approach, integrating measurement, communication, enablement, and incentives

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Measuring value | Value tracking is a major hurdle to AI adoption – exemplars deploy an end-to-end value steering and capture framework

1c

~50% of CIO/CTOs see unclear ROI as key hurdle to GenAI adoption in their enterprise

~50% of respondents report redeploying capacity gains

" *What are we doing with 30% time savings? At this point nothing - our devs are just finishing work 30% sooner."*

E2E value tracking and steering framework

Icon	Category	Description
%	**Balanced scorecards**	Balancing adoption, productivity, quality, and developer satisfaction
↗	**Adoption maturity levels**	Definition and measurement of AI maturity by cohort (vs. tool usage)
💼	**Business case with lineage to metrics**	Alignment on metrics and ROI by each "using" division
🤝	**Handshake (and targets) by divisions**	ROI method backed by metrics (e.g., hard to measure % of time coding - commonly used in business cases)
📋	**Link to budget process**	Mechanisms to identify and capture value (e.g., reduce current and/or open roles, clear additional backlog, and commit to faster time-to-delivery)
💰	**Incentives and rewards**	Monetary and non-monetary nudges to align behaviors

Tech debt dilemma | AI in code development with mature DevOps significantly enhances velocity and capacity savings

GenAI velocity gains
based on DevOps maturity[1]
in **traditional** tech stack
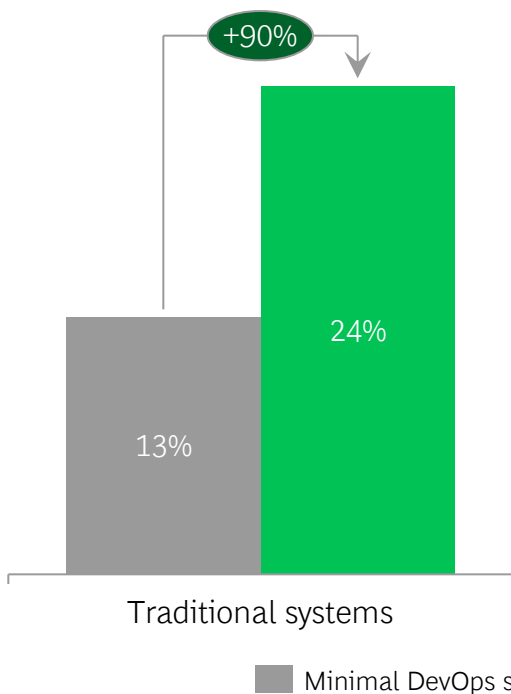
GenAI velocity gains
based on DevOps maturity[1]
in **modern** tech stack

+90%

24%

13%

Traditional systems

+49%

21%

14%

Modern systems
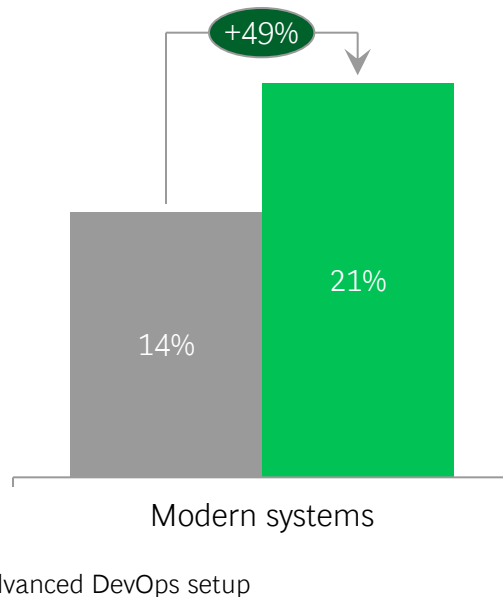
■ Minimal DevOps setup   ■ Advanced DevOps setup

GenAI-ready tech stacks and ecosystems maximize gains

GenAI-ready codebases,
e.g., standardized coding practices, modular design, and trunk-based branching strategy

Robust DevSecOps, e.g., testing with AI-generated artifact scans and vulnerability detection

Platform & data simplification and modernization, e.g., unified infrastructure and streamlined release pipelines

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025
1. At least two DevOps automations utilized (e.g., automated testing, CI/CD tools)

Adoption flywheel | To drive adoption and faster innovation loop – establish a central engineering enablement team

Exemplars are creating a central engineering enablement team to orchestrate the adoption flywheel to:

Take a user-centric vs. tool-centric approach to adoption

Build a robust measurement and tracking approach

Drive behavioral change for 1000s of engineers in a scalable and cost-efficient manner

Rapidly enable tool evolution through a tight product-to-user feedback loop

Promote continuous upskilling in response to evolving tooling

Central teams   User teams

1 Latest features / products

2a Adoption sprints

2b Reinforcement + new patterns / features

AI and non-AI tooling

GenAI tools

DevSecOps tooling

Vendors/tooling sandbox

Infra/Cloud

Engineering enablement NEW

Evangelists/Adoption success

Facilitators & trainers

Integrated metrics & dashboards

Broader organization
Teams across geos., divs.

Team leads | Identified "champions"

Developers | Identified "champions"

Ongoing P2P programming

3

Feedback on roadmap

5

Feedback, patterns, requests

4

13

Upskilling | AI will reshape the engineering function, eliminating some roles, creating new ones, and redefining skill sets – upskilling is essential

>85%[1]

of CIO/CTOs and engineers expect AI will impact jobs

43%[2] new required skills and competencies

27%[2] elimination or reduction of existing roles

20%[2] emergence of new roles

Type of impact by AI

- Increase demand for skills in AI tool proficiency, prompt engineering, ethics, and critical thinking
- Eliminate or reduce roles associated with repetitive, low-value tasks such as junior developers and testers
- Create new roles focused on AI integration, governance, and interdependencies.

Example skills required/roles impacted

- AI literacy
- AI ethics and governance
- Analysis of AI output and oversight

- Junior/entry level developers
- Traditional developer roles (focused solely on coding)
- Testers

- Prompt engineers
- AI governance and ethics specialists
- AI-augmented developers

Expectations to learn new technical skills, but also "softer" ways of work
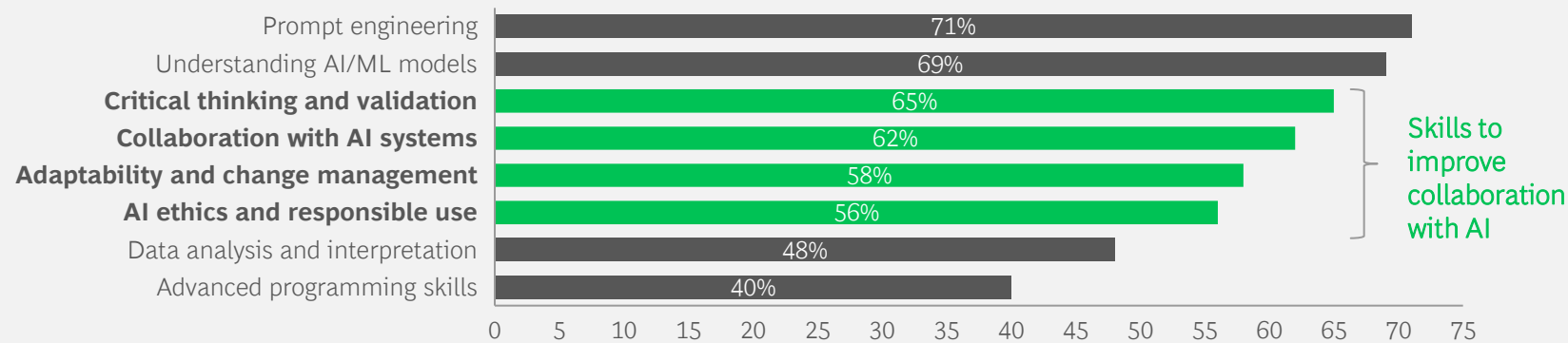
Skill	%
Prompt engineering	71%
Understanding AI/ML models	69%
Critical thinking and validation	65%
Collaboration with AI systems	62%
Adaptability and change management	58%
AI ethics and responsible use	56%
Data analysis and interpretation	48%
Advanced programming skills	40%

Skills to improve collaboration with AI

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025
1. % of survey responses by CIO/CTOs and engineers if they expect a GenAI impact on jobs; 2. % of survey responses on specific impact of AI on jobs

14

Future state | AI will reshape development organization structure – flatter hierarchies, faster ramp-up, fewer silos

Future state of software Dev Org

Pyramid diagram with levels top to bottom: Executives (1), Upper management (2), Mid management (3), Sr. SWE, Mid-level SWE (4, 5), Entry-level SWE.

Legend: Traditional workforce · GenAI-enabled workforce · AI agents
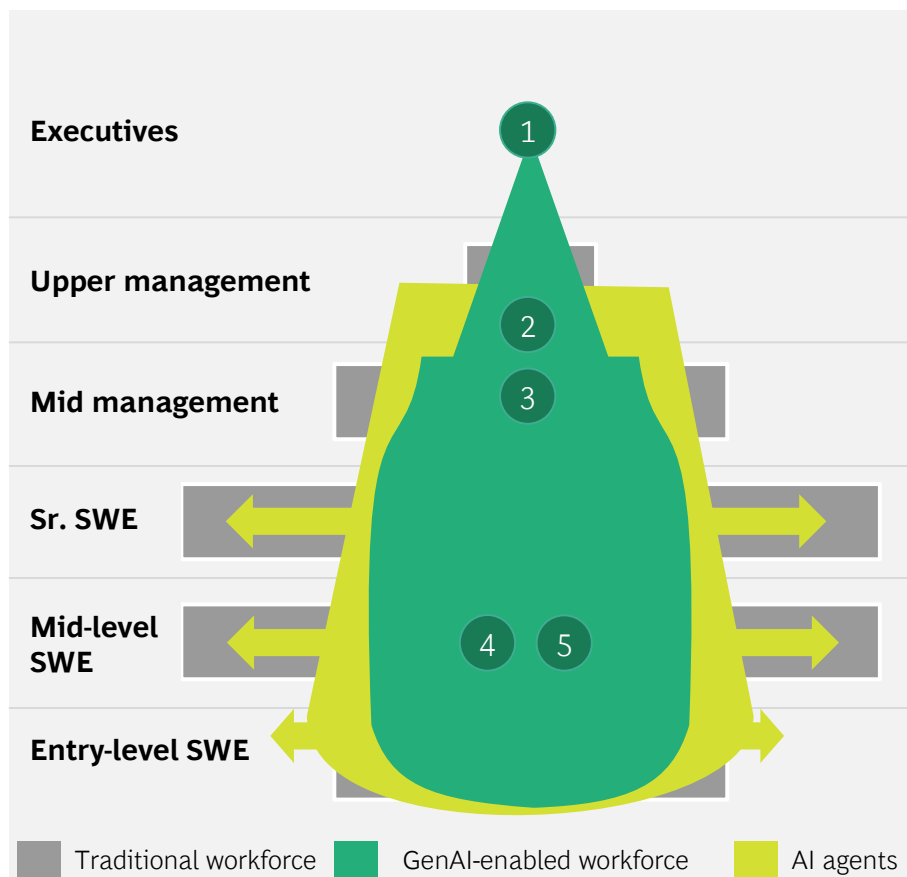
Key implications of GenAI on software Dev Org structure

1 **Leaner, high-velocity organizations:** AI boosts productivity across the board. Fewer layers, faster releases, leaner headcount

2a **Skills shift – less coding, more critical thinking**: Development work moves up the stack. Success now depends on system design, AI collaboration, and problem framing

2b **New roles reshape the organization:** AI creates demand for roles like AI stack architects, prompt engineers, agent orchestrators, and ethics/governance leads

3 **Middle management thins out:** "Democratization of knowledge" – AI reduces the need for hand-holding. Less time spent guiding juniors, more time on strategy

4 **Entry-levels level up faster:** Juniors ramp-up quicker, focus on logic and product thinking – AI handles boilerplate & testing

5 **QA & support roles shrink or vanish**: Agents absorb traditional testing, documentation, and some DevOps – reducing adjacent team needs

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

Imagine a world | Human-AI collaboration doubles SDLC productivity

Discovery & prototype/design

Gathering & refining requirements

Rapid prototyping

Adaptive architecture

Delivery & validation

AI-augmented coding

Self-optimized CI/CD

Deployment

Intelligent testing

Operations

Predictive monitoring & maintenance

"Always on" governance & oversight

Continuous feedback

Reduced time to clarifying objectives & validate concepts
- AI-assisted solution design (fewer missed edge cases)
- Self-reconfiguring platform and algorithms for performance

Faster coding & implementation
- Automated repetitive coding tasks and structural improvements
- Streamlined deployment process and lowered time-to-production

2X Productivity

Better quality & testing
- AI-driven test suites & rapid bug identification and resolution
- Proactively reduced security & vulnerabilities risks

Self-healing system & continuous improvement
- Agentic-maintained system stability and minimized downtime
- Auto. analyzed usage patterns to guide future development

Accelerated time to market and reduced total cost of ownership

Delivery of higher quality and cultivation of resilient systems

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

How to get it right | Full AI value requires holistic engineering transformation

Singular roll-out of code-generating AI tools

Off-the-shelf tools provide **easy access to AI capabilities** but **long-term transformation potential limited to 30%** as outdated practices can largely eliminate impact from deploying GenAI

>30% Value from GenAI

Most commonly followed approach today …

GenAI-enabled engineering excellence transformation

A holistic approach that fundamentally rethinks how teams build, ship, and maintain software **(including GenAI, agents, traditional DevOps, and underlying technology modernization)**

>2X[1] Value from GenAI

… recommended way forward

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025
1. 30% from maximizing use AI code generation tools, 20% impact from extending tools to other non-coding stages (including via agents) and 2x from impact if working on a modern tech stack with modern ways of working

What is required | Key requirements for a holistic AI-enabled engineering transformation

Inputs for a successful AI-enabled engineering organization

Measures

Right ways of working within teams
- Interaction models, cadence, & flow
- AI-first mindset (e.g., reviewer vs. builder)
- Tech Team best practice adoption (e.g., TDD)
- Iterative tech debt mgmt. & decoupled architecture

Right tech stack and ecosystem
- AI-embedded workflows (e.g., coding assistants, agents)
- Integrated toolchains & platforms
- Reusable assets, APIs, & InnerSource
- Tech standards & architectural vision

Right organizational conditions
- Product/Platform Org & Op Model
- Digital, customer, & innovation mindset(s)
- Agile Funding and Governance
- Adaptive Risk, Responsible AI & Cyber Controls

Right reskilling of workforce
- Ongoing reskilling & upskilling
- Future-ready organizational structure ("Pyramid of the Future")
- Next-gen talent practices (skills, recruiting & rewards)

Right human enablers and improvement loop

Improvement loop focused on biggest bottleneck

Tech enablement team(s) identify top bottlenecks and propagate best practices

Measurement framework to track & drive efficiency across inputs, outputs, outcomes

Source: State of GenAI Across Engineering Survey (N=100 CIOs/CTOs and 300 engineers), January 2025

18

What is required | Short-term actions and long-term initiatives drive end-to-end transformation

Short-term actions

Long-term initiatives

Right ways of working within teams

- Train developers on **effective prompting techniques**
- Implement **AI-assisted code review processes**
- Pilot **test-driven development (TDD)** with GenAI

- Standardize **AI-augmented SDLC workflows** with clear guidelines for AI-human collaboration
- Embed AI into agile sprint planning

Right tech stack and ecosystem

- Deploy **AI coding assistants in IDEs**
- Fix gaps in **testing, CI/CD, and security scans**
- Integrate AI into **collaboration & DevOps tools**

- Build a **scalable AI-ready development toolchain**
- Establish **AI-embedded workflows** (e.g., agents)
- **Form strategic AI vendor partnerships**

Right organizational conditions

- Define a **clear AI operating model**
- Identify & address cultural **AI adoption barriers**

- Optimize **team structures for AI collaboration**
- Refine **funding model** to capture gains

Right reskilling of workforce

- Introduce **prompt-engineering workshops, learning sprints**

- Develop **role-specific AI training programs**
- Integrate AI skills into **career development**
- Conduct long-term skills & **workforce planning**

Right human enablers and improvement loop

- Form an **AI enablement team and establish tight feedback loop**
- Establish monitoring and measurement

- Establish an engineering excellence function (including driving adoption of AI solutions)

Risks | As companies explore AI, understanding and mitigating risks are critical for responsible adoption

Non-exhaustive

Traditional AI risks | **GenAI-specific risks**

Biased outputs
Models might suggest default or stereotyped logic flows, impacting fairness in user-facing features

Lack of explainability
Tracing why the model proposed certain code or test scenarios can be difficult

Leaks of proprietary data
Companies and employees may transmit proprietary data when using GenAI tools

Undesired outcomes
Model suggestions may appear highly optimized but could contain hidden logic flaws

Privacy violations
Private data may be inadvertently logged or stored within build/test environments

Insecure code generation
Models might be unable to effectively address edge cases or tolerate changes to inputs

Informal use of AI
Employees use GenAI tools without adequate guidance/ supervision or bypass governance and security reviews

Hallucinations
GenAI may convincingly present misinformation as fact

Lack of human controls
Insufficient human oversight can propagate errors and vulnerabilities, especially in context of agents

Unintended side effects
If AI introduces unconventional patterns or dependencies, it could lead to new vulnerabilities

Copyright infringement
GenAI may inadvertently produce code segments that replicate copyrighted or licensed material

High energy consumption
GenAI models require significant compute resources, increasing carbon footprint

Additional third-party risks: vendor models require extra validation checks

Level of urgency and impact for organizations integrating GenAI in the SDLC | min ◔ ◑ ◕ ● max

To mitigate these risks, companies should implement:

- A holistic change program
- An engineering enablement team

∨

- Identify all risks, assess mitigations, and create a solid plan to implement mitigations

Getting started | An organization's approach to an AI transformation is highly dependent on its current tech state

A company can be in more then one category

Current tech state	Transformation approach	Value[1]	Description
In-house dominant	**Three-speed approach:** accelerate adoption, strengthen foundations, enable future innovation	★★★	• Immediately **pilot GenAI coding tools** within agile teams to boost productivity • Invest in **foundational capabilities** (e.g., training, infrastructure) to scale rapidly • Build readiness for future AI-driven innovation by **integrating GenAI** deeply into **developer workflows**
Vendor-driven / Outsourced	**Define clear GenAI standards and expectations with vendors**	★★	• Set explicit **AI-driven standards** for code quality, security, and collaboration in vendor contracts • Execute joint **GenAI pilots** to align vendors and **capture shared productivity gains**
Legacy systems / DevOps maturity	**Modernize foundations with targeted GenAI investments**	★★★	• Focus initial GenAI use cases on **addressing tech debt, legacy refactoring**, and automating foundational DevOps tasks • **Establish a stable, modernized base** to fully unlock future GenAI value
SaaS / Platform-driven	**Maximize GenAI features from leading SaaS providers**	★	• Prioritize SaaS **providers** proactively **advancing GenAI capabilities** aligned to your needs • Quickly **integrate proven, pre-built GenAI solutions into existing processes** for rapid wins

Workforce gravity (covers In-house dominant and Vendor-driven / Outsourced)

Technology estate (covers Legacy systems / DevOps maturity and SaaS / Platform-driven)

1. Degree of value generated through a GenAI transformation

Low ★   Medium ★★   High ★★★

Getting started | CIOs/CTOs can take these practical next steps to prepare for their AI SDLC transformation

1 Activate your engineers

- ❑ **Develop a clear understanding of SDLC pain points** while recognizing the differing priorities of CTO/CIO and engineers (e.g., testing, requirements analysis)
- ❑ Identify a team of **skilled and motivated engineers** to lead the group and act as **change agents**

2 Build a holistic ~3 year roadmap across the ~5Rs[1]

- ❑ **Develop a comprehensive transformation roadmap** that integrates the core principles of the 5-R[1] framework for engineering excellence
- ❑ Create a **measurement framework to track and drive efficiency** across inputs, outputs, and outcomes

3 Start building a central enablement and orchestration function

- ❑ **Establish a centralized team** to orchestrate drive **tooling standardization, AI adoption, developer productivity, security compliance, and cross-team alignment** to ensure scalability and efficiency
- ❑ **Elevate workforce planning** by rethinking **roles**, updating **capabilities**, and aligning skills with the future of software engineering

4 Establish beacons of success

- ❑ **Identify pilot areas to prove the new normal in action** – exemplars start with teams, but then expand to deliver a large project in an AI-first
- ❑ Make an informed, strategic choice between **buying and building solutions**

BCG experts |
Key contacts for AI transformation

EMESA

- **Marc Schuuring** — Senior MDP — Amsterdam
- **Julien Marx** — MDP — Paris
- **Tom Martin** — PLA Director — London
- **Miriam Seiler** — Principal — Zürich
- **Michael Grebe** — Senior MDP — Munich
- **Nicolas de Bellefonds** — Senior MDP — Paris
- **Sylvain Duranton** — Senior MDP — Paris
- **Jan Ittner** — Partner & VP — Munich
- **Eike Kock** — PLA MD — Cologne
- **Nicolas Levillain** — VP — Paris
- **Steven Kok** — Partner & AD — London
- **Arne Weiner** — PLA Director — Cologne
- **Sahithya Vemana** — Partner — London

NAMR

- **Matthew Kropp** — Senior MDP — Bay Area
- **Benjamin Rehberg** — Senior MDP — NY
- **Julie Bedard** — MDP — Boston
- **Joppe Bijlsma** — MDP — NY
- **Vikram Sivakumar** — Partner — New York
- **Geoffrey Sipperly** — PLA Director — Seattle
- **Vladimir Lukic** — Senior MDP — Boston
- **Vanessa Lyon** — Senior MDP — NY
- **Sesh Iyer** — Senior MDP — Washington
- **Pinaki Roy** — MDP — NY
- **Djon Kleine** — MDP — Los Angeles
- **Stephen Robnett** — MDP — Denver
- **Vishwas Sugla** — MDP — New Jersey
- **Helen Han** — MDP — New York
- **Liz Lucero** — Partner — Boston
- **Lisa Krayer** — Principal — Philadelphia
- **Tue Tran** — Project Leader — New York
- **Tauseef Charanya** — Senior Director — Austin

APAC

- **Rajiv Gupta** — Senior MDP — New Delhi
- **Julian King** — Partner & VP — Sydney
- **Gaurav Jindal** — MDP — Mumbai
- **Sreyssha George** — MDP — Bengaluru
- **Sambhav Jain** — MDP — New Delhi
- **Shavi Gandhi** — Partner — New Delhi

○ Authors of Executive Perspectives and experts
○ Experts