AI-Assisted Coding: Augmenting Software Development with *Generative AI*

Exploring the Integration of Generative AI in Software Engineering to Enhance Coding and Team Collaboration

Capgemini

CONTENTS

Disclaimer: *This paper presents forward-looking perspectives grounded in the current landscape of Generative AI. It reflects the rapid advancement – both in terms of our understanding and experience of Generative AI technologies. It draws insights from external research, technology leaders as well as our own experimentations with our teams and with clients. As always, the future remains unpredictable, but we can establish and imagine likely trajectories and outlooks.*

Software plays a critical role in modern business, whether it's integrated into business apps or products. Despite its ubiquity, the ultimate challenge for software engineering organizations has always been to release quality software fast enough to keep up with constantly accelerating demand.

Over the years, productivity and quality standards have been progressively enhanced through the adoption of new methods and technologies. Agile and continuous product-centric approaches, software life cycle automation, open-source ecosystems, cloud native and composable software architectures, DevOps continuum, and low-code/no code development have all been introduced. And each has built on the advances of past iterations.

But demand for quicker time-to-value and better cost efficiency is still accelerating. This leaves software organizations struggling to deliver at the expected pace, with the expected quality, while controlling technical debt and costs.

Generative AI is software engineering's latest and greatest evolution.

Generative AI now stirs wonder and excitement across many application fields, not least software engineering. Bill Gates is calling its latest advancement the most revolutionary technological achievement in over 40 years, [1] with the potential to dramatically improve the way organizations meet business and IT challenges.

The intersection of business and technology will be at the heart of generative AI's impact. Developments on one side influence the other. For example, efficient software engineering with high quality can reduce time to market and thus provide business value earlier, giving leading organizations both a software and business-oriented edge.

Our Capgemini Research Institute report shows that 61% of organizations see enabling more innovative work, such as developing new software features and services, as the leading benefit generative AI. Close behind are improving software quality (49%) and increasing productivity (40%). Organizations are utilizing these productivity gains on innovative work such as developing new software features (50%) and upskilling (47%). Very few aim to reduce headcount (4%).[2]

According to another Capgemini Research Institute survey across 800 organizations, 67% of executives see the most potential for generative AI in the IT function to drive innovation and create value[3] Moreover, according to Forrester research, "Off-the-shelf and custom AI software spend will double from $33 billion in 2021 to $64 billion in 2025 and will grow 50% faster than the overall software market, with an annual growth rate of 18%." [4]

In both legacy software modernization and new software development contexts, generative AI gives back time to software engineers. They are freed up to focus on business demand, software quality, security, and the advanced features required by new software.

Generative AI also has a positive impact on software professionals' job satisfaction. 69% of senior software professionals and 55% of junior software professionals report high levels of satisfaction from using generative AI for software development. 78% of software professionals are optimistic about generative AI's potential to enhance collaboration between business and technology teams[5]

Generative AI will supplement bandwidth - using the same capacity - for higher productivity and efficiency, along with faster time to market. But only if organizations and their employees commit to getting on board, keeping in mind that early adopters will take a leading position in the field.

In addition, generative AI presents an opportunity not only to transform but also to standardize and enhance the delivery of software. The benefits brought by using generative AI in software engineering can extend to other areas of the business to create even more value, reduce IT costs and minimize technical debt.

Organizations using generative AI have seen 7%–18% productivity improvement[6] in the software engineering function, compared to their initial estimates. Creating literature and documentation, and writing code and scripts show the greatest timesaving with 35% maximum and 10% average, and 34% maximum and 9% average respectively.[7] Generative AI also has a positive impact on software professionals' job satisfaction. 69% of senior software professionals and 55% of junior software professionals report high levels of satisfaction from using generative AI for software. 78% of software professionals are optimistic about generative AI's potential to enhance collaboration between business and technology teams.[8]

[1] *https://www.gatesnotes.com/The-Age-of-AI-Has-Begun*
[2] *Capgemini Research Institute "Turbocharging software", June 2024*
[3] *Capgemini Research Institute "Generative AI in Organizations", July 2023*
[4] *Global AI Software Forecast, Forrester Research, Inc. September 29th, 2022*
[5] *Capgemini Research Institute "Turbocharging software", June 2024*
[6] *By total productivity improvement we mean overall improvement in the productivity of the individual from all types of tasks accelerated by generative AI.*
[7] *Capgemini Research Institute "Turbocharging software", June 2024*
[8] *Capgemini Research Institute "Turbocharging software", June 2024*

Bringing tomorrow's software engineering paradigm into view: augmented software teams

We believe generative AI will gradually transform the way software is developed. The advent of large language models (LLMs) has introduced a compelling rationale for a paradigm shift to more AI-assisted (augmented) software teams. The integration of generative AI into software engineering offers the promise of significantly elevated productivity and enhanced quality. The foundational principles and methods of Agile and DevOps are retained, including collaboration, adaptability, time to value, product centricity, and continuous feedback loops.

By working with AI assistants powered by LLMs, augmented software teams can automate mundane tasks, and promote more nuanced, data driven decision making. This optimizes the software life cycle and helps to achieve more milestones along the way.

We call this approach conversational software engineering. This is where software teams prompt the AI assistant, asking it to generate snippets of code, troubleshoot issues, or even help in designing and architecting the software. The idea is to make the software development workflow more dynamic and interactive, through continuous conversations with the AI assistant in the development environment. It's a way to streamline tasks and solve problems more efficiently, enabling software teams to focus on more complex and creative aspects that boost overall productivity.

The modern evolution of pair programming with generative AI Augmented software teams can effectively benefit from and optimize usage of generative AI through a revised organization and new ways of working, guided by the following principles:

- **Augmented pair programming:** The basic unit of an augmented software team is one or multiple pairs of software engineers working intermittently with each other or individually with the AI assistant.
- **AI-human collaboration:** Team members work individually with the AI assistant to automate repetitive tasks, understand and solve problems, or brainstorm ideas, leveraging an LLM's speed and knowledge, while ensuring human creativity.
- **Human-human collaboration:** After conversations with an AI assistant, humans review the generated output and enhance its quality.
- **Senior coordination:** A senior lead oversees the augmented team, coordinating efforts and resolving conflicts, while ensuring control and validation to promote a smooth workflow and healthy team dynamic.

According to Capgemini Research Institute, the biggest gap in essential prerequisites is usually access to platforms and tools, including IDEs, automation and testing tools, and collaboration tools. Only 27% of organizations claim to have above average availability of these. Of testing domain professionals, 24% say they have access to these tools, compared to 19% of project and program management professionals, further highlighting gaps within the software engineering function.[9]

[9] *Capgemini Research Institute "Turbocharging software", June 2024*

Deep dive on Coding Assistants

Applying the technology to software engineering will significantly assist software teams in the multitude of tasks they perform across the traditional software development life cycle (SDLC).

For example, software engineers use generative AI when transforming backlog stories into software, through design and coding activities. Generative AI can be harnessed to create design outputs such as user interface mockups, entity models, and application programming interfaces (APIs). This leads to a significant productivity improvement without compromising quality, as design outputs are always reviewed, updated, and validated by software engineers or technology leads.

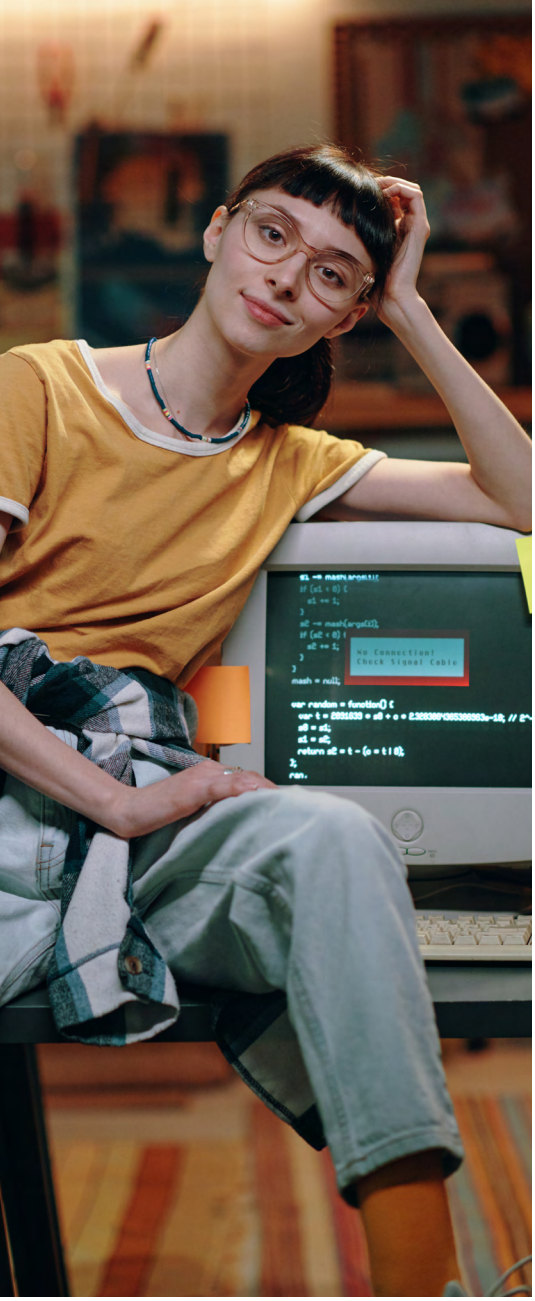
Although adoption of generative AI for software engineering is still in its early stages, functions. Generative AI is expected to play a key role in augmenting software workforce with better experience, tools and platforms, and governance, assisting in more than 25% of software design, development, and testing work by 2026.[10]

Generative AI is also poised to redefine conventional programming practices by shifting the focus from coding to prompt engineering and code proofreading, as confirmed by Andrej Karpathy, an OpenAI computer scientist, who recently said: "the hottest new programming language is English." [11] As an example, using plain language, software engineers can describe the intended functionality of a software feature, then review, update, and validate the generated output. There are many other examples, such as auto-completion of code, generating code for unit testing, (retro) documentation, and code migration from one language to another.

A very important value of generative AI is that it supports developers already during coding. It can either suggest clean code directly or evaluate existing code to improve software quality if it finds issues.

[10] *Capgemini Research Institute "Turbocharging software", June 2024*

[11] *https://twitter.com/karpathy/status/1617979122625712128?lang=en-GB*



How to move forward: A proven method for a software engineering transformation

To start this journey with generative AI, organizations will reap huge benefits from partnering with an experienced and trusted advisor. In addition to having considerable AI expertise, this partner should be ready to oversee experimentations and form collaborative project teams that eventually flourish into joint software houses.

A variety of factors, such as company size, staffing pyramid, industry-specific goals, mean that each organizational unit progresses at a different pace and with a different model. Therefore, each will have distinct needs and challenges to consider before generative AI can become a part of daily operations.

It's important to focus on three areas:

- **Assess the organization's maturity and develop a roadmap.**
- **Run experimentations and measure impact.**
- **Deployment at scale.**

Assess the organization's maturity and develop a roadmap with clear objectives.

Within a company, there are usually many variations of an SDLC. Typically, every program / project / product has its own SDLC version. A meticulous assessment and deep understanding of the SLDC of an IT domain will reveal the current maturity of its software engineering processes. And whether they align with industry best practices. The evaluation will also identify areas for improvement, particularly where bottlenecks or inefficiencies occur.

Based on this assessment, objectives can be defined by selecting the most promising transformation enablers, as well as identifying their associated risks and challenges. Some enablers will be powered by generative AI while, for others, it will be more a matter of applying software engineering best practices. It's important to define key metrics to measure the outcomes of the future transformation and involve stakeholders early to foster a collaborative environment.

Finally, a comprehensive value, accessibility, and risk analysis will serve as a basis for establishing a roadmap for change. All options will be weighted and prioritized to make informed decisions about resource allocation as different domains, teams, and roles are defined for each option.

Run real-world experimentations and measure the impact of *Generative AI*

With the priorities set, it's time to select the best generative AI tools to achieve them. And put-up guardrails to manage any legal and cybersecurity risks, while controlling costs. Measurement and continuous improvement will be pivotal. As generative AI is inserted into more SDLC processes, organizations need to gauge its impact. This means measuring critical aspects such as productivity enhancements, software quality, time-to-market, and developer experience. A feedback loop should be put in place so the pace and scope of the deployment can be adapted to account for inefficiencies. Or respond to evolving needs and specific objectives. Critically, the experimentations and measurements should be relevant to the business context and industry environment of the organization.

But to be truly competitive, organizations need more than internal measurement. At Capgemini, we have developed an industrialized value measurement protocol that evaluates the objective impact of generative AI across an organization's many SDLCs. It is used to measure and compare an organization's metrics against our benchmark, which factors in all our related internal and external generative AI projects. This gives organizations a clear view of how they stack up against their peers.

To put this into perspective, the adoption of generative AI for software engineering is still in its early stages with 9 in 10 organizations yet to scale. 27% of organizations are running generative AI pilots, and 11% have started leveraging generative AI in their software functions.

Three in four (75%) large organizations (annual revenue greater than $20 billion) have adopted (piloted/scaled) generative AI compared to 23% of their smaller counterparts (annual revenue between $1–5 billion).[12]

Deploy *Generative AI* for Software Engineering at scale

After experimenting with generative AI through real-world pilots and full scale delivery, many organizations will want to broaden their applications and possibly involve hundreds or even thousands of developers. For such large deployments, careful consideration must be given to the organizational and HR implications.

Since various roles may need to change, it's important to introduce a generative AI upskilling program that will help shape the new software engineering pyramid, and address the ways of working in respect to the skills required. This will lay the foundation for a strategic plan that gradually integrates generative AI into the software life cycle. A further step may be to offer coaching and assistance to steer employees as they prepare to work alongside a host of new generative AI tools.

Finally, a dedicated team should be set up to define business cases, measure progress, and ensure the outcomes meet expectations. This is essential, as the pace and scope of each deployment will invariably need adjustment.

[12] *Capgemini Research Institute "Turbocharging software", June 2024*

Capgemini's experience in measuring *Generative AI* impact

Our experience working with clients across all sectors tells us that it's not a question of if generative AI-powered software engineering will disrupt and reinvent an organization, but rather when, how quickly, and how radically. There are two questions we hear the most from our clients. First, how will generative AI-based assistance affect software productivity and quality. And second, how will software teams' organization and ways of working change because of it?

At the beginning of 2023, Capgemini started a large-scale global program to experiment with use cases to measure generative AI's impact through experimentations, both internally and jointly with clients. The aim was to see where and how generative AI can augment the many tasks software teams typically perform. We deployed a reliable and consistent measurement protocol and now we are progressively consolidating our results into a repository for internal and external usage / benchmarking.

Measurement protocol and real-world experimentations: designed for actionable and reliable results

Measuring productivity within software development poses inherent complexities due to the multifaceted nature of the SDLC, the dynamic environment it operates in, and the subjective and intangible aspects of many of its components. A comprehensive measurement approach must encompass both qualitative and quantitative factors, tailored to the specific context of the project.

Similarly, assessing software quality presents challenges as it involves various dimensions such as functionality, performance, reliability, usability, maintainability, security, and scalability, each requiring its own set of metrics and criteria.

Moreover, soliciting feedback from software engineers utilizing Generative AI on a daily basis is crucial, considering its impact on the development environment and workflow. Therefore, it's imperative to devise and implement a practical measurement protocol. This should be focused on understanding Generative AI's influence on coding and unit testing within bespoke application development, providing a clear insight into its effects.

The measurement protocol is a combination of several different ingredients and a well-defined process to produce to create comprehensible, actionable and reliable results. This includes a measurement approach, metrics, team, and a well-defined process.

Measurement approach: A robust measurement approach includes steps for measuring progress, including planning, setting baselines, and running an experimentation. The right measurement metrics are key, and we'll explore them in the next chapter. To support the approach, metrics tools like SonarQube, CAST, Jira, and developer surveys are used for collecting and analyzing data. Prerequisites and success factors for a proper and consistent measurement are team stability, duration, technology, legal considerations, and cybersecurity. A normalization process must handle variability during experimentation execution, adjusting metrics to changes in team size, capacity, or complexity.

Metrics: Coding velocity serves as a pivotal metric to measure team productivity, focusing on coding and unit testing activities, typically quantified by implemented story points. Other dimensions must also be evaluated - like velocity per developer seniority and velocity per complexity of user stories. This comprehensive approach offers insights into how Generative AI influences productivity across different developer skill levels and the intricacies of software development tasks.

Besides that, when it comes to testing, the metric for Unit Test Coverage is essential. It helps to assess the quality and reliability of software. To keep it simple we focus on instruction coverage (C0) as this is measured by most of the tools.

On top of those our protocol includes many more metrics like code efficiency, code security, code smells, code duplication.

Team: In the single team approach, one team sequentially executes a backlog of user stories of consistent size or complexity, comparing performance with and without Generative AI assistance. On the other hand, the multiple teams approach involves parallel execution of the same backlog by at least two teams with different tool setups. For example, one with Generative AI tools and the other without Generative AI tools. This allows for simultaneous assessment of Generative AI's effects across different team dynamics.

The seniority or capabilities of a team are important for normalization, and so it's mandatory to know what kind of team mix is working on the defined backlog. We distinguish a senior team of highly skilled individuals representing the gold standard of proficiency. Conversely, the well-balanced team consists of a good mix of seniors and juniors, necessitating some level of coaching typically facilitated by senior members alongside daily work. Finally, the junior team features few seniors, with the primary focus on coaching junior members' development and productivity.

Process: Once all the ingredients have been defined, a process is needed to ensure high quality results and reduce side effects due to estimation inaccuracy and the human factor.

1. Define the team(s) organization and the experimentation scope and timeline.
2. Define the measurement approach.
3. Validate the prerequisites and the success factors.
4. Conduct a baseline for the metrics, without Generative AI assistance.
5. Execute the experimentation sprints with Generative AI assistance.
6. Collect and normalize the metrics and the feedback.
7. Consolidate and report the measured results.

Prerequisites and success factors

Successful experimentation hinges on managing the following prerequisites effectively:

- **Team stability** is vital for achieving comparable results. This includes consistent size, seniority, and unwavering processes.
- **Baselining** necessitates a minimum of three sprints, while Generative AI experimentations benefit from 6-9 sprints. A well-defined backlog with diverse user stories is essential input for the sprints and should be operated accordingly.
- **Measurement tools** must be readily available for an accurate assessment. All team members need access. Compliance with regulations, encompassing legal, security, and privacy aspects, is imperative throughout the process.

Baselining is crucial for understanding the current velocity and quality of the traditional ways of working without Generative AI tools. It's recommended that insights are gathered from the last three sprints, excluding Generative AI assistance. Coding velocity data should be gathered from team backlog management and collaboration tools, as well as code quality reports from static code analysis tools. If data from the last three sprints isn't available, three sprints without Generative AI assistance must be conducted before proceeding to the Generative AI phase. The baseline must be calculated based on this. Additionally, the developer experience survey, based on the experience without Generative AI, is conducted during the baseline phase.

For a successful experimentation, it's important to maintain a constant team size and carefully curate the backlog to include use cases of varying complexities. The team must be equipped with the necessary setup to effectively utilize the Generative AI tool. Additionally, assigning a dedicated tool expert to each team can significantly enhance success and efficiency. Monitoring the team involves ensuring timely task updates on platforms like JIRA or Azure, with team leads responsible for tracking and ensuring compliance. In an augmented team approach, peer reviews are essential, and the Generative AI tool should be utilized as an assistant throughout the process.

Lessons learned

Like all organizations, we are learning from Generative AI in real world and real size conditions. Below are our high-level learnings:

Experimentation Framework: Our experimentations with generative AI in custom software engineering typically involve one to multiple teams, spanning a minimum duration of six sprints (preferably nine), with a diverse backlog of user stories and complexity mix. Different team configurations offer valuable insights. Typical configurations might include, existing teams transitioning to using generative AI. New teams integrating generative AI later. And shadow teams working alongside existing ones.

Baseline Establishment: Solid baselining is crucial to ensure robust and representative comparisons across various metrics. These include velocity, quality, security, time-to-market, and developer experience. Quantifiable results from experimentation showcase productivity improvements.

Tooling Perspective: Experimentations in the coding and unit testing stages of the software lifecycle cover multiple technology combinations. This includes vendor-packaged solutions with generative AI extensions and foundational LLMs enhanced with contextualized prompt engineering techniques.

Prompt Engineering Skill: To harness the full potential of generative AI, developers must acquire the skill of prompt engineering. This involves proper training, playbooks, and prompt libraries. Pair programming can further facilitate learning and optimization of generative AI usage.

Team Stability: Instability within teams can hinder experimentation outcomes. Countermeasures such as normalization processes are necessary to address this challenge effectively.

Developer Experience: Surveys conducted with team members reveal high appreciation for generative AI tools, indicating a positive developer experience. Most team members express a desire to continue using these tools in the future.

Documentation and Packaging: Learning, documenting, and packaging best practices for generative AI adoption is essential. This includes technologies and architecture solutions, prompt engineering handbooks, and prompt libraries. Discovering proven and applicable augmented software team patterns facilitates organizational efficiency and performance.

In summary, experimentation with generative AI in custom software engineering offers valuable insights into productivity improvements, developer experience enhancement, and organizational efficiency. Documenting and packaging best practices are pivotal in fostering widespread adoption and maximizing the benefits of this transformative technology.

We can drive *software engineering transformation* with you

Generative AI is the fastest growing technology we've ever seen. CIOs are eager to explore the opportunity and reap the benefits. For example, greater productivity, better quality, and accelerated time to value. However, they must recognize the confidentiality and IP risks involved. The potential costs of uncontrolled generative AI use. And how the technology can impact the structure, skillset, and ways of working of their software engineering teams.

At Capgemini, we transform the very core of the software development lifeccyle, then iteratively measure the progress to achieve continuous improved outcomes. We don't just implement a one-time solution, leaving clients to navigate the subsequent challenges by themselves.

Our offers look at Generative AI for Software Engineering across three stages:

Kickstart your transformation: Begin your transformation with our Generative AI for Software Engineering blueprint.

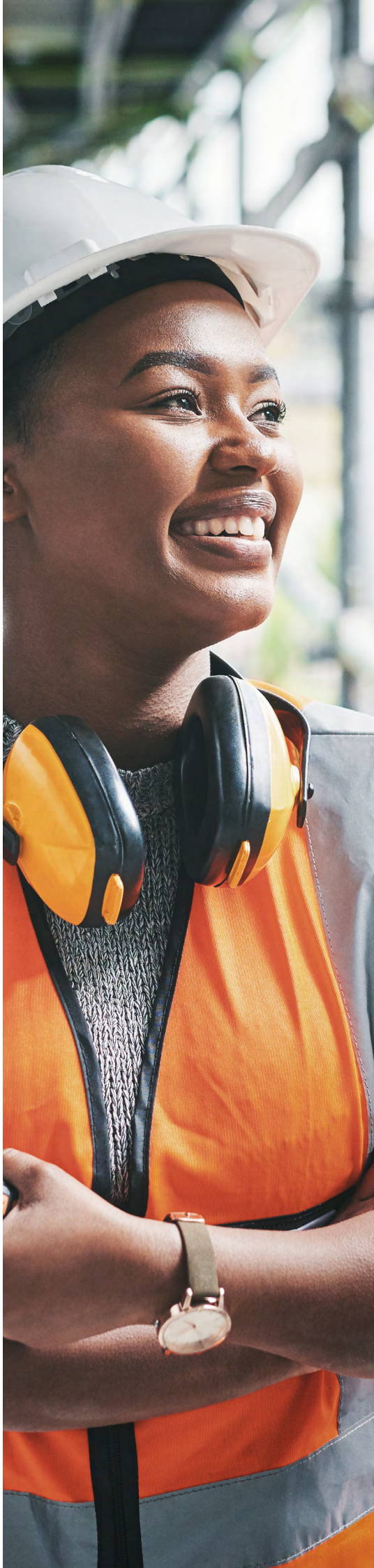
- We assess your current software engineering lifecycle, identifying key opportunities.
- We design and establish secure foundations for deploying generative AI technologies, adapting team structures and skills.
- We develop a progressive plan for integrating generative AI into identified software life cycle stages, starting with coding and unit testing.

Upskill your teams: Upskill and engage your teams through hands-on collaboration.

- Enhance team skills: Collaborate with our generative AI companions, offering coaching to full partnership for premium AI-powered software.
- Ensure AI integration: Our advanced Generative AI for Software Engineering Academy, including a certification program, ingrains AI in team operations.

Measure and improve together: Measure and improve together to set new software engineering market standards.

- We measure the impact of Generative AI across the software development lifecycle
- We focus on productivity, quality, developer experience, and teams' organization
- We drive your transformation with a feedback loop and our industry benchmarks



The dawning of a new era for *software engineering*

As a global leader in software engineering, Capgemini has a proven history of advising clients on how to elevate what is possible. Always with human creativity accenting machine efficiency. We've orchestrated many step changes in this industry, and none has stirred as much excitement and innovative promise as generative AI. It's a giant leap forward that has fundamentally changed the way software is created.

Embracing generative AI is not merely a choice, but a strategic imperative. It's predicted that 70% of professional developers will be using the technology by 2027.[13] And so the faster organizations get this transition underway, the further ahead of competition they will be. Ultimately, the question is whether they will be an early adopter and transform at scale, leading the way. Or a late bloomer, playing catch-up.

Although the path to full adoption of generative AI may seem daunting, it's essential that organizations harness its unprecedented transformative power now.

Are you ready to reshape your software engineering?

For more information on Capgemini's Generative AI for Software Engineering offer, visit:
www.capgemini.com/solutions/generative-ai-for-software-engineering/

[13] *https://www.gartner.com/en/articles/set-up-now-for-ai-to-augment-softwaredevelopment*

About the authors

Pierre-Yves Glever

Executive Vice President, Global Head of C&CA, Capgemini

pierre-yves.glever@capgemini.com

Stephane Girard

CTO, Global C&CA, Capgemini

stephane.girard@capgemini.com

Thilo Hermann

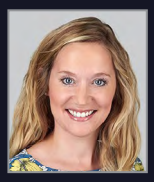
CTO Deputy, Global C&CA, Capgemini

thilo.hermann@capgemini.com

Matthias Sondermann

CTO, C&CA Germany, Capgemini

matthias.sondermann@capgemini.com



Lorna Neville

Marketing Director, Global C&CA, Capgemini

lorna.neville@capgemini.com

About Capgemini

Capgemini is a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It isa responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategyand design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with itsdeep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

www.capgemini.com

Capgemini

Get the future you want