McKinsey & Company

McKinsey Center for Future Mobility

From engines to algorithms: Gen AI in automotive software development

As the automotive and industrial sectors undergo major technological advancements, gen AI–driven operating models can allow companies to develop software more easily, safely, and innovatively.

This article is a collaborative effort by Dominik Hepp and Martin Harrysson, with Lukasz Maslaniec, Mateusz Wozniak, and Michael Amroudi, representing views from the McKinsey Center for Future Mobility.

January 2025

Companies in the automotive and industrial sectors are rewiring to become software-enabled enterprises. As in-vehicle software emerges as a critical differentiator, companies have started to reevaluate the role of software and overhaul their development approaches. Today, software serves as the backbone for advanced features and safety-critical functions while enhancing operational efficiency and propelling innovation. But software can also introduce organizational challenges, because underdeveloped software capabilities can result in start-of-production delays and budget overruns.

Generative AI (gen AI) is disrupting the ongoing software transformation by introducing new opportunities and challenges. Companies are still in the process of changing their software operating models, for example, by setting up and maturing dedicated software development and delivery units while adapting the collaboration model with their suppliers. Additionally, organizations are actively recruiting new talent with specialized software expertise while simultaneously reskilling their existing hardware-focused workforces to adapt to software-centric roles. As gen AI advances, organizations must be able to capture the substantial productivity potential reported across various domains while balancing the nondeterministic challenges of this technology and the criticality and security requirements of their systems.

A recent McKinsey survey of automotive and manufacturing executives revealed that more than 40 percent of respondents are investing up to €5 million in gen AI research and development, and more than 10 percent are investing more than €20 million.[1] Leading automotive and industrial companies have become even stronger competitors by effectively experimenting with gen AI and leveraging its potential for software-defined hardware.[2] The gap between top performers and others is likely to widen further as these companies implement gen AI effectively and derive value from it.

This article discusses the potential for gen AI to improve software development processes in the automotive and industrial industries, as well as the change management and strategic approaches required to properly integrate it into current procedures. The insights presented are drawn from McKinsey's work with leading automotive and industrial organizations, including initial gen AI impact pilots, organization-wide rollout strategies, and holistic R&D transformations.

Overcoming challenges to enhance software operating models

In the automotive and industrial sectors, gen AI can transform how software is created and used, greatly increasing productivity across the software development life cycle. Software engineering is expected to be the area most affected by gen AI.[3] It could boost the productivity of developers by reducing the time they spend on various software engineering activities, such as generating initial code drafts, correcting or refactoring code, and creating new system designs. These capabilities

[1] "Automotive R&D transformation: Optimizing gen AI's potential value," McKinsey, February 9, 2024.
[2] Ali Rizvi, Ani Kelkar, Philipp Kampshoff, and Sarthak Vaish, "Software-defined hardware in the age of AI," McKinsey, January 3, 2025.
[3] "Where business value lies," in *The economic potential of generative AI: The next productivity frontier*, McKinsey, June 14, 2023.

About the McKinsey Center for Future Mobility

These insights were developed by the McKinsey Center for Future Mobility (MCFM). Since 2011, the MCFM has worked with stakeholders across the mobility ecosystem by providing independent and integrated evidence about possible future-mobility scenarios. With our unique, bottom-up modeling approach, our insights enable an end-to-end analytics journey through the future of mobility—from consumer needs to a modal mix across urban/rural areas, sales, value pools, and life cycle sustainability. Contact us if you are interested in getting full access to our market insights via the McKinsey Mobility Insights Portal.

have led to rapidly rising adoption of gen AI, with most companies currently experimenting with at least one gen AI application.

Succeeding with gen AI requires more than just technology, however; it necessitates the right operating model. Executives perceive significant barriers to implementing and rolling out gen AI, largely due to the organizational and cultural changes required to integrate gen AI into existing operating models. The value of gen AI will not be realized by merely adding new gen AI tools—it hinges on an organization's ability to adapt to new ways of working and embrace a transformative operating model. For example, if an organization is underperforming in software development, gen AI alone is unlikely to resolve the issue.

In addition, applying gen AI to critical embedded software is often perceived as more challenging than traditional software development. To embed gen AI, for example, organizations have to maintain a highly optimized code that works within their computational resource constraints, ensures low latency, and interacts with hardware interfaces. In safety-critical applications such as automotive, software must undergo extensive testing and validation to meet safety standards. Ensuring that gen AI meets stringent certification processes, is checked and approved by multiple reviewers, and uses specific languages and models requires additional effort and oversight.

While gen AI can support software transformations, it is not a perfect solution; it cannot solve all the software challenges an organization might face. Developing the wrong product will not lead to innovation or customer satisfaction, and without an adequate tool chain and database, the applicability of gen AI will be significantly limited. Previous McKinsey research outlines what successful software operating models at automotive companies should look like.[4] With the introduction of gen AI, the steps for properly developing and embedding software take a new shape. Gen AI has important implications for each dimension of the software operating model (Exhibit 1).

How can gen AI help with product management and defining requirements? It's important to consider how gen AI can help product managers define software requirements for developers. It should reduce complexity, increase modularity, and ensure development of the right features for the customer.

How can gen AI be embedded in the software development process? Considering how gen AI can be embedded into software development processes is important in incorporating it at scale. Selecting the right use cases for gen AI, with consideration for the technology's limitations and the criticality of specific systems, is essential.

What capabilities are required to use gen AI end to end along the software development cycle? To manage the risks of applying gen AI in safety-critical software, investing in capability building is essential. In addition, effective training programs can address concerns about gen AI limiting productivity in the automotive and industrial sectors.

What tool chain is required to maximize value from gen AI in software development? Companies can decide what tools are required to maximize the value from gen AI in their software development. Due to the complexity of embedded systems in the automotive and industrial sectors, gen AI tools and models require customization, and fewer off-the-shelf solutions are applicable.

Ensuring that the processes, capabilities, and tools are in place to help usher in the gen AI transformation is imperative. Organizations that neglect these foundations may drive individual, isolated use cases without prioritizing and aligning them across the business, ultimately hindering their progress or creating safety risks.

Mastering software development with gen AI

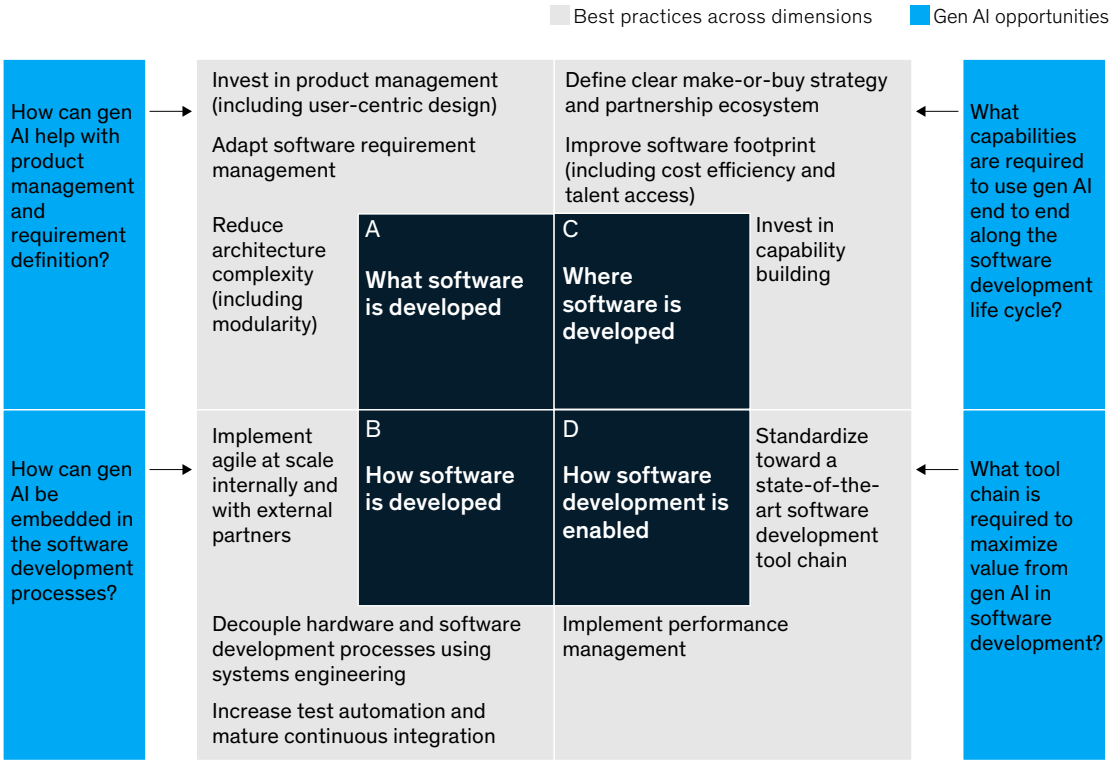
Integrating gen AI can be an opportunity for automotive and industrial companies to advance their software development capabilities. The automotive and industrial sectors can accelerate

[4] "When code is king: Mastering automotive software excellence," McKinsey, February 17, 2021.

Exhibit 1

Software development excellence requires distinctiveness across four main dimensions.

Generative AI (gen AI) opportunities across dimensions



McKinsey & Company

transformations across operating models, tooling, process changes, ways of working, and upskilling with these tools. They can also greatly increase productivity, helping organizations close the gap between the complexity of new digital tools and ways of working while managing costs, supporting talent, and improving innovation. What's more, these advancements can improve customer experience by, for example, offering more sophisticated voice assistants that enhance the human–machine interface, bolstering safety through advanced driver-assistance systems, and elevating the overall driving experience with increased personalization.

How gen AI can support product managers

Gen AI can support product managers (PMs) in various use cases. It is especially helpful in supporting PMs as they refine business case assumptions, generate objectives and key results or KPIs, or compile a summary of new feature development. It's also adept at helping to create product artifacts, including summarizing product requirement feedback from technical experts, drafting user stories , and creating and refining press releases, frequently asked questions, and product requirement documents.

In the development of safety-critical embedded systems, gen AI can significantly assist PMs by ensuring comprehensive coverage of safety aspects. For example, gen AI can analyze regulatory standards and historical project data to automatically generate detailed and compliant requirement specifications, acting as a quality controller to ensure that no critical element is overlooked. Advanced models can parse and understand complex regulatory documents, aiding

PMs in meeting safety requirements. Additionally, gen AI can automate the creation of detailed documentation, continuously checking for compliance throughout the development process, flagging any deviations from safety standards, and suggesting corrective actions. It is important that rigorous validation and verification processes are employed, including human-in-the-loop systems for which AI outputs are reviewed before being acted upon.

We estimate that PMs could save between 10 and 30 percent of their time on these tasks when they get access to the right gen AI tools and receive proper training. The most impactful use cases can achieve a time savings of 40 percent or more. One of our recent surveys of PMs tested the time-saving potential of gen AI in ten different use cases. The results indicated that PMs with access to gen AI could save up to 39 percent of the time they take to create and refine product requirements and user stories. Survey respondents also indicated a productivity improvement of 44 percent when using gen AI with quality assurance measures, such as creating and automating tests to then enhance efficiency and code reliability.

Where gen AI can advance the software development processes

Once PMs outline the requirement definition of gen AI use cases, developers can integrate these tools across the whole software development life cycle (Exhibit 2).

Incorporating gen AI into software development processes could save developers substantial time when innovating in the automotive and industrial sectors.

In the beginning of the software development life cycle, developers can use gen AI to understand business requirements and design architecture. For embedded systems, such as those in automotive control units or industrial machinery, the design should capture functional requirements while adhering to stringent hardware constraints and real-time performance needs. Gen AI analyzes extensive data sets to generate insights, helping developers capture and translate business needs into technical specifications more accurately, which reduces miscommunication. It assists in creating multiple software architecture designs, generating draft diagrams, and offering market solution comparisons, which speeds up the design phase and ensures robust, scalable architectures that meet the specific demands of embedded environments.

In the development stage, gen AI helps to write, translate, refract, and document code. It can draft code, autofill existing code, generate code from pseudocode prompts, and accelerate the coding process. For critical embedded software, such as advanced driver-assistance systems, gen AI can produce code optimized for limited memory and processing power while ensuring proper hardware interfacing; however, its impact in these settings tends to be more limited. Gen AI also translates code between programming languages, aiding in the modernization of legacy systems and enabling developers to address backlog initiatives without rewriting existing code bases. It automates the refactoring process by identifying and improving code areas, enhancing maintainability and readability while reducing technical debt. Moreover, gen AI automates the creation of documentation, including user manuals, API documentation, and inline comments, ensuring consistency and accuracy for easier comprehension by new developers.

Last, when it comes to finalizing products, gen AI can help write unit, integration, and acceptance tests. Note, though, that testing can also be an initial step in methodologies such as test-driven development, in which tests guide the coding process from the outset. For embedded systems, gen AI can help create hardware-in-the-loop and software-in-the-loop test environments to simulate real-world operating conditions, which can augment real-world data to reduce the amount of data that needs to be collected, further enhancing the testing process. In addition, it can generate test cases that consider hardware interactions, timing constraints, and real-time performance, identifying high-priority events and anomalies. These tests provide performance insights that enhance the reliability and stability of software applications, which is necessary for embedded systems that often operate in mission-critical environments. In such settings, gen AI can also assist in automating compliance testing to meet industry standards and regulatory requirements, further ensuring the robustness and safety of the final product.

Exhibit 2

Generative AI can support how software is developed across the whole software development life cycle.

Generative AI (gen AI) use cases across software development process

McKinsey & Company

Across use cases, companies need to manage the risks associated with using gen AI in critical and safety-relevant systems. While powerful, gen AI is still prone to nondeterministic behavior and hallucinations, which can pose significant risks in safety-critical applications. Nondeterministic behavior refers to the AI's tendency to produce different outputs despite being given the same inputs, which can lead to unpredictable and unreliable results. Hallucinations occur when the AI generates information that appears plausible but is factually incorrect or nonsensical. To manage this risk, companies can implement redundancy and cross-validation mechanisms, in which multiple AI models independently analyze the same input and their outputs are compared to ensure consistency. In addition, companies should employ human-in-the-loop systems in which AI outputs are reviewed and validated by experts.

Across use cases, our research revealed that tasks requiring writing and understanding code benefit most from incorporating gen AI into the integrated development environment and the ability to work with the code base . First pilots for writing, translating, and documenting code revealed time-

saving potential of up to 40 percent. Conceptual tasks that involved using gen AI as a brainstorming partner or assistant (in understanding business requirements, for example) also see time savings, although less significant, with early pilots indicating an impact potential of between 15 and 30 percent.

The capability building required to use gen AI along the software development cycle

Recent McKinsey research outlines the array of new skills engineers and product managers will need to master to be successful with gen AI.[5]

To capture the full potential of gen AI, it is important that users are properly trained. Capability-building programs should cover foundational gen AI skills, such as prompt engineering techniques, setting context, and risk management, and should include practice opportunities with use cases, such as code generation, reviews, and documentation. Executives are often concerned about data protection and related legal issues when using gen AI, so these issues need to be proactively addressed by an adequate risk-management process for introduction and usage. Trainings should also cover topics that are more advanced, such as code translation and refactoring. For embedded software development, these trainings should emphasize the unique constraints and requirements of embedded systems, such as optimizing for limited memory and processing power, real-time performance, and hardware-software integration.

Once capability-building programs are rolled out, it's important to track their progress and impact by using a set of predefined metrics. For example, one automotive organization rolled out an AI developer tool across about 10,000 developers by distributing licenses but without including a dedicated training program. By tracking the usage of this tool, they saw that licenses were being underutilized: only 20 percent of developers were actively using the tool, and less than 10 percent of that share were effectively using key features, such as chat functionality.

McKinsey conducted a pilot program with a European industrial software company with more than 40,000 developers. The study measured how gen AI improved selected software development use cases on a weekly basis, and it launched a capability-building initiative concurrently to ensure that the new tools were used effectively. As a result of these trainings, developers used gen AI 60 percent more often per week than when they didn't have these programs. Not only did engagement improve, but after the trainings, 95 percent of developers also reported that gen AI has a positive impact on their developer experience (Exhibit 3).

The tool chain required to maximize value from gen AI in software development

The strategic application of gen AI in software development typically involves a phased approach, starting with standard use cases and advancing to tasks that are more complex and specialized. To effectively harness the potential of gen AI tools in software development, organizations can consider four distinct stages.

The first stage, and a typical starting point, involves using off-the-shelf models and tools for standard use cases, such as code generation and documentation for popular programming languages such as Java, JavaScript, Go, and Python. This stage allows organizations to quickly realize the benefits of gen AI with minimal initial investment and complexity.

The second stage, which is less commonly pursued, involves fine-tuning these models on the organization's own code base and potentially self-hosting them for standard use cases. For embedded software development, this stage might involve adapting models to understand and generate code for lower-level programming languages such as C or assembly. While this can offer solutions that are more tailored, it often requires significant effort and resources, making it less appealing for many organizations at the early stages of their AI journey.

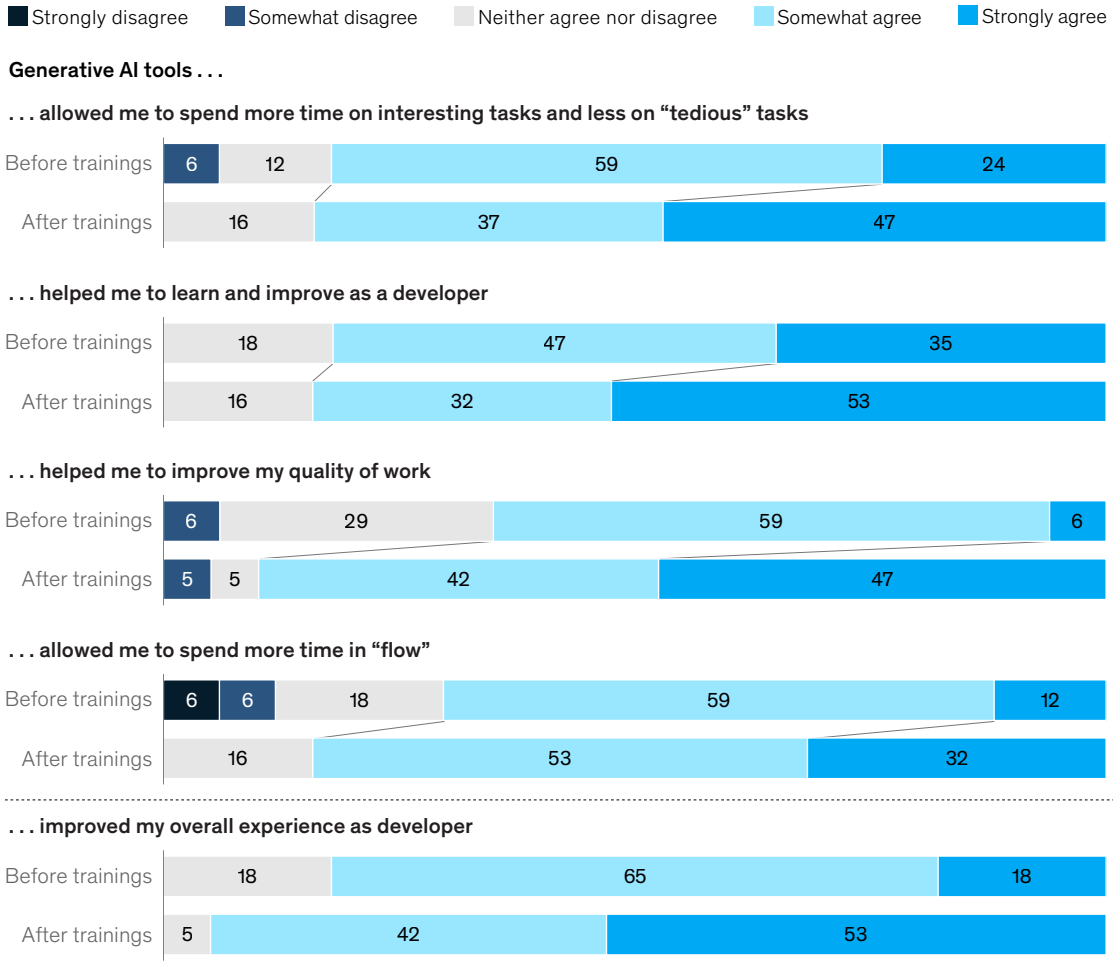
The third stage involves using off-the-shelf models and tools for use cases that are more advanced. This could include complex tasks such as advanced code refactoring or integrating AI into embedded systems

[5] "The gen AI skills revolution: Rethinking your talent strategy," *McKinsey Quarterly*, August 29, 2024.

Exhibit 3

Developers see a positive impact of generative AI tools on their developer experience, especially when they are offered a proper capability program.

Share of respondents considering their developer experience, before and after trainings, by level of agreement,[1] %



Note: Figures may not sum to 100%, because of rounding.
[1]Question: With regard to your experience last week, please share the extent to which you agree or disagree with the following statements about gen AI tooling.
Source: McKinsey weekly developer surveys (n = 32), 2024

McKinsey & Company

to manage signal transmissions between sensors. In the context of embedded software, this stage might also involve optimizing code for real-time performance and memory constraints and ensuring seamless hardware-software integration.

The final and most advanced stage is using fine-tuned models on proprietary code bases or self-hosting these models for highly specialized tasks. For example, in requirements engineering, tools can assist product managers with general tasks, but more companies are developing their own AI tools to produce better results, such as generating entity-relationship diagrams.

The right applications are crucial for maximizing the benefits of gen AI, whether in enterprise or embedded software development. Overall, gen AI can be applied to both enterprise and embedded software, albeit with some distinctions. While the integration process involves similar strategies, tools, and risk management processes, the impact tends to be higher in enterprise software because it has fewer constraints compared to critical embedded systems. Embedded software development often requires specialized testing environments such as hardware-in-the-loop and software-in-the-loop to ensure reliability and performance. Additionally, embedded software development benefits less from generic off-the-shelf AI tools and typically necessitates more-specialized solutions and custom setups to meet specific hardware and performance constraints. Lower-level programming languages used in embedded systems may also require additional fine-tuning or skill embedding to fully capitalize on gen AI's capabilities.

Moreover, certain gen AI tools and large language models may be more suitable than others for developing embedded software. For instance, models that can handle the intricacies of real-time systems and provide support for low-level programming languages will be particularly valuable. These tools need to be adept at generating code that is not only functional but also highly optimized for the limited resources typical of embedded systems.

Setting the foundation for success

For automotive and industrial leaders to properly integrate these capabilities, they must be intentional about how they roll out gen AI. To successfully integrate gen AI into software development, organizations need to establish three foundational steps : capability building and change management, impact measurement and value capture, and a multilever transformation approach.

The first step focuses on equipping developers with necessary skills and knowledge. Organizations should start with high-impact, easily implementable use cases, supported by a centralized governance structure for data and tooling. Training a group of coaches to drive the enablement program further ensures the sustainable growth of gen AI capabilities.

The second step involves defining and tracking metrics to measure the impact of gen AI initiatives. Critical metrics such as code quality and development velocity should be established to create a baseline to track progress against. Tools that track and analyze the performance of gen AI and monitor usage patterns across teams can provide valuable insights. Developing programs to realize financial value from the capacity unlocked by gen AI is essential for demonstrating tangible benefits.

The third step focuses on catalyzing a comprehensive transformation of the software development operating model. Remember, the gen AI integration is never just about the technology, so this stage involves changing processes, organizational structures, capabilities, and tools. Fostering a DevOps culture that emphasizes collaboration, communication, and shared responsibility between development and operations teams is also crucial for creating an environment conducive to allowing gen AI to thrive and for achieving higher developer velocity.

A logical progression is to consider the financial ramifications of this effort. Three methods are most often employed. First, organizations can accelerate their backlog by integrating banked efficiency gains into their planning processes, thereby enabling

teams to handle more work with the same capacity. This approach requires meticulous alignment between efficiency improvements and project planning. Second, organizations can schedule future hiring efforts to coincide with the measured impact of gen AI. This approach would allow them to capitalize on the increased productivity of gen AI to maintain or even expand operations without additional head count. Last, organizations can reallocate the extra capacity generated by gen AI toward launching new initiatives. By ring-fencing or redistributing this capacity across various teams, companies can undertake new or incremental strategic projects with clear benefits, such as enhancing resilience and reducing technical debt. This approach not only leverages time savings for immediate operational improvements but also strategically positions the organization for long-term growth and innovation.

The integration of gen AI into automotive and industrial software development represents a significant leap forward in enhancing productivity, innovation, and operational efficiency. As companies navigate this transformation, it is crucial for them to adopt a holistic approach that includes capability building, strategic implementation, and continuous impact measurement. By fostering a culture of collaboration and adaptability, organizations can fully leverage the potential of gen AI to drive growth and maintain competitive advantage in an increasingly digital landscape. The journey toward mastering gen AI is complex, but with the right strategies and tools, the automotive and industrial sectors can achieve remarkable advancements in their software development processes.

Dominik Hepp is a partner in McKinsey's Munich office, where **Michael Amroudi** is a consultant; **Martin Harrysson** is a senior partner in the Bay Area office; and **Mateusz Wozniak** is an associate partner in the Warsaw office, where **Lukasz Maslaniec** is a consultant.