



How generative AI can revolutionize the software development lifecycle

Companies that make the transition see enormous advantages

Introduction

Generative AI can revolutionize how software developers work. In this paper, we explore the various generative AI solutions available to developers and offer insights about the benefits developers can derive from them throughout the entire software development lifecycle (SDLC). To make the most of generative AI and large language models, a new field of IT expertise has emerged: Prompt engineering. This specialized role determines the depth of queries required to obtain the needed outputs. Engineers can provide simple prompts to gain generalized information quickly, use a multi-prompt approach for greater specificity, or take time to fine-tune the model using granular details most pertinent to a business model.

Our new research into how executives across industries expect to use generative AI is based on findings from an exclusive KPMG survey conducted in March 2023 of 300 global executives across a wide range of industries. (Where relevant, we compare the results with findings from our follow-up survey of 200 US executives three months later.) Our March data indicated that 73 percent of the respondents saw AI and machine learning as crucial skill sets for their organizations and are therefore a hiring priority. In June, we found that 19 percent of companies polled already use generative AI for writing code or other development documentation, and that an additional 24 percent are currently researching and piloting ways to maximize generative AI in their application coding efforts.¹ Those that made the transition are already seeing an enormous advantage in streamlining their processes and shortening their development timeline. A study conducted by GitHub noted that developers who used code generation tools, such as GitHub, reported feeling 88 percent more productive.²

Developers using tools like ChatGPT, DALL-E, and GitHub Copilot derive multiple benefits that change how they work. Generative AI can accelerate

¹ KPMG Generative AI Survey, March 2023 and June 2023

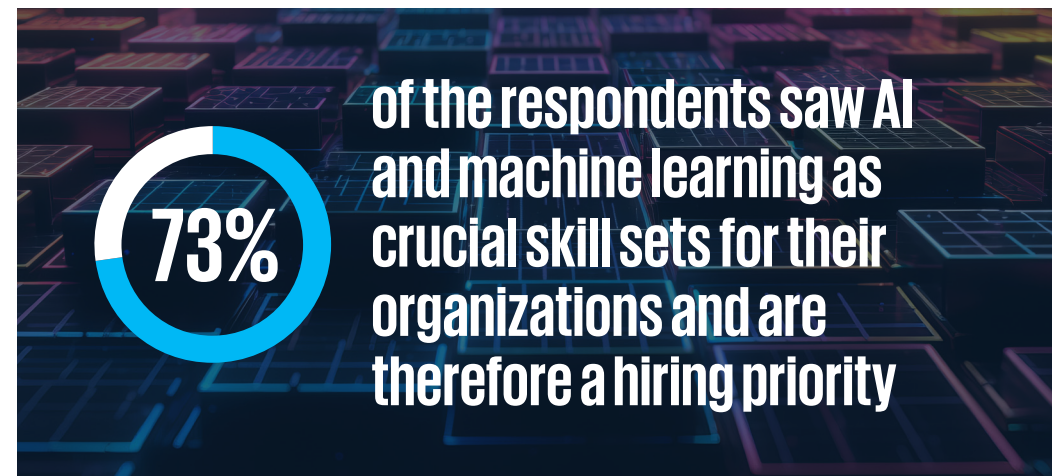
² "How generative AI is changing the way developers work," GitHub, April 14, 2023

³ Ibid.

⁴ "A Software Developer's Guide to Generative AI," Built In, April 13, 2023

the application creation process and allow coding specialists to focus on higher-level, creative, more complex activities. In Github's study, for example, developers using Copilot coded up to 55 percent faster than those who did not.

Today, rules-based logic and basic machine learning assist some developers because these tools can rapidly predict coding sequences. However, generative AI can support developers in more extensive and automated ways. Because generative AI can handle more repetitive and tedious portions of the coding process, developers can focus their time and skills in more valuable ways.³ For instance, developers can use ChatGPT to automatically find potential security issues within an application or code specific functions. While these tasks could take humans several hours, generative AI can accomplish them in less than a minute.⁴ There also need to be considerations to the experience of the developers that are leveraging these code generation tools. A more seasoned developer would be expected to identify code generated inefficiencies and make the required changes versus leverage everything that is produced.



Impact across the entire software development lifecycle

In the information technology space, a system’s software (application) development lifecycle (SDLC) guides the process of planning, creating, testing, and deploying an information system. The SDLC concept applies to a range of hardware and software configurations to support a hardware-only, software-only, or combination system.

The SDLC generally contains these five elements:

Planning and analysis of requirements: In this stage, software engineers frame a new application’s functionality and capability based on user needs and other pertinent details.⁵

Design: During this phase, developers must choose the hardware systems their application requires and the programming languages to facilitate the coding process. Sometimes software engineers also create a prototype before diving into the development process.

Testing: This portion of the SDLC focuses on tasks like software functionality testing, scanning for security issues, and finding ways to optimize performance.⁶

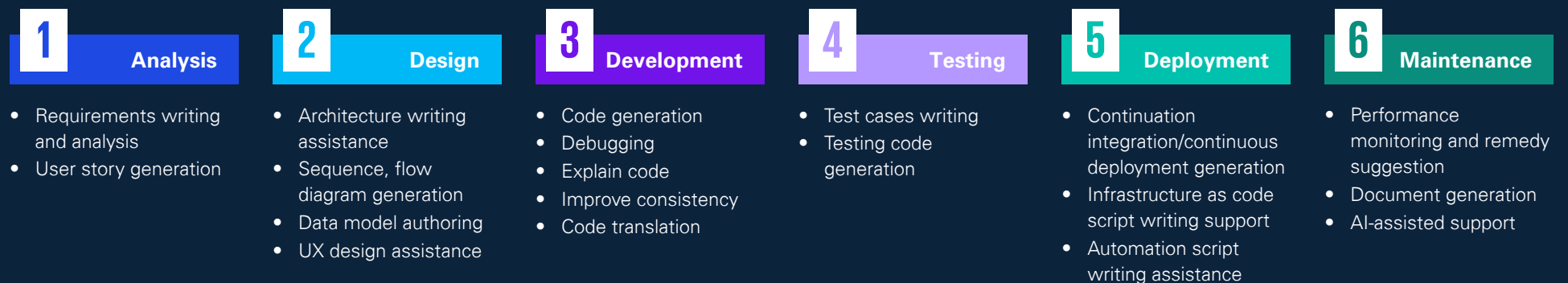
Deployment and implementation: Once an application passes the testing and vetting process, developers bring the software online for user training and use in a production environment.

Maintenance and evaluation: Rarely is software perfect during its initial deployment. Monitoring the application, collecting user feedback, and bug-hunting help developers identify and fix any unforeseen functionality issues.

Integrating generative AI into SDLC

The SDLC process has served engineers for decades—and still does today. However, generative AI makes the process easier and faster while reducing the number of potential errors needing diagnosis later (Exhibit 1).

Exhibit 1: Generative AI offers developers several vital forms of assistance:



⁵ “Software Development Life Cycle: A Step-By-Step Process,” Top Developers, September 1, 2022

⁶ “What is the Software Development Lifecycle?” Developer.com, April 11, 2023

Benefits and challenges

Generative AI brings many benefits to the SDLC, including more than just code generation. It will become even more reliable and accurate as it matures over time. Software engineers using it in the coding process can expect several benefits that can free up their time and expertise for higher-level tasks:

- It can help with various phases** like requirement analysis, design, development, testing, deployment, and maintenance.
- It assists in applying new concepts, new programming languages, and new packages.**
- Developer productivity can increase** since generative AI can guide lifecycle steps and assist with scaffolding code and code snippets.
- It helps alleviate some repetitive tasks** like testing and accumulating test data.
- It provides alternative ideas** and helps with problem-solving.
- It generates functional prototypes** for developers so they can more quickly test and iterate on ideas, thereby reducing an application's time to market.

These factors indicate generative AI is excellent at many things today. However, businesses planning its implementation in their SDLC must consider some challenges, too. They need expertise, infrastructure, and procedural safeguards to ensure trustworthy, compliant, transparent, and secure output.⁷

Legacy applications: In our recent study, nine percent of respondents cited the inability to pivot legacy applications and systems as the biggest generative AI implementation challenge (based on weighted average). Twenty-seven percent of the respondents consider it among the top three barriers.

Data privacy: When forming partnerships or investing in other companies to support their generative AI initiatives, companies must have confidence that the data and training input their generative AI system requires does not infringe on individuals' privacy. Enterprises must also ensure adherence to regional data governance policies.⁸

Intensive computational requirements: The latest versions of generative AI algorithms require high computational resources and significant infrastructural investment to train and run models. These concerns create a substantial barrier for many enterprises.

Ethical considerations: Depending on its training process, generative AI has the potential for bias or use of misleading information.

Reliance on accurate code outputs: AI-generated code can contain bugs, security issues, or inefficiencies. If the developer is not experienced or trained properly these issues can move into production.

Prompt injection: Without proper guardrails, would-be attackers could inject prompts and control the software model unexpectedly.

Need for specialized and new skills: Developing and implementing generative AI systems requires technical skills and expertise that may not be readily available. The coding experience will be different, and developers need to be trained in new ways. In the past, developers learned through making mistakes and figuring out the best solution; in the future, they will need to identify the machine-generated mistakes. This will be a harder problem to solve.

⁷ "Game changer: The startling power generative AI is bringing to software development," KPMG LLP, 2023

⁸ "Generative Artificial Intelligence and Data Privacy: A Primer," Congressional Research Service, May 23, 2023

Approaches to leverage generative AI

In the real world, we observe several approaches to leveraging generative AI. In all cases, though, generative AI models require training to deliver the needed outputs. Different prompting techniques produce results that vary from general insights to highly specific information. Below we outline how various forms of prompting, retrieval, and fine-tuning can help an organization obtain the insights it needs:

Prompting:

Sometimes called “zero-shot” prompting, this initial step involves making a simplified, general query of the generative AI language model. Zero-shot prompts can provide rapid results when working with a widely adopted model trained with generic information.

One-shot and few-shot prompting:

When developers need more honed output from a large language model, they must use more advanced prompting techniques. One-shot and two-shot approaches give the model more context and examples to build from. For instance, if a developer wants an output in the form of a table, they can ask the system model to return data in a comma-separated format and provide a relevant example. This greater level of detail helps the larger language model follow guidance and generate more applicable output.



Prompting and retrieval:

The prompt-and-retrieval approach refines results to make them much more targeted. When a large language model trains on public data, it is not specific to one domain. Therefore, it lacks insights that could come from your proprietary data or other critical data points. By providing this depth of context, the generative AI model can produce more meaningful, customized output grounded in a specific context.

Fine-tuning:

Fine-tuning is a more advanced technique to make the most of generative AI and large language models. In many cases, gathering data and extending it into the training environment requires expertise from both the engineering and the data center teams. Therefore, the fine-tuning process involves greater technical complexity, setup time, and cost. In the banking industry, for example, a model can benefit from input about regulations, policy, and intrinsic knowledge about the company’s domain. For fine-tuning, a few techniques can help reduce the cost of training:

- Parameter-efficient fine-tuning using LoRa and QLoRa
- Increase training speed with multi-node/multi-GPU training frameworks like accelerate, Deepspeed and PyTorch Fully Sharded Data Parallel.

Below is a summarized comparison of the prompting approaches:

Task	Complexity	When to use	Speed to adoption
Prompting	Low	Simple, generic responses required	Rapid
One-shot/ few-shot prompting	Low	More complex, specific responses required	Quick but more involved process of honing prompts and introducing examples for the needed outputs
Prompting and retrieval	Medium	Complex, specific responses required from private data	Slower process of gathering and introducing business-specific data for more detailed output
Fine-tuning	High	Cannot use online models due to privacy concerns or want to adapt a foundation model to a specific domain-like financial	Significant investment of time and cross-team resources



Prompt engineering

The process of overcoming generative AI's challenges while reaping its advantages has sparked a rapidly growing field known as prompt engineering.⁹ Simply put, prompt engineers use natural language processing (NLP) to provide generative AI systems with a description of the task they must accomplish and the output needed. Developers can give those instructive elements to a generative AI system in the form of written sentences, examples, a question,

or even a block of code.¹⁰ This versatility makes it much easier for engineers to encourage accurate outputs without providing overly implicit instructions. However, there is a balance. If an engineer's prompts do not give enough detail for a generative AI system to interpret the request, it could produce irrelevant results.¹¹ In that case, prompt engineers must hone their input to derive an ideal outcome.

Best practices for prompt engineers:

To gain the greatest level of coding assistance from generative AI, developers should consider four best practices:¹²

- 1 Guide the model:**
 - Use a high-level task description.
 - Give high-level contextual information.
 - Provide specific directions for input data accompanied by output examples.
 - Specify undesired output.
- 2 Break down complex tasks:** Use smaller, more granular steps to guide generative AI to get the results needed.
- 3 Take it in steps:** Start with zero-shot, then one-shot, then few-shot learning for the generative AI model.
- 4 Iterate:** Experiment with different variations of prompts, analyze the results, and refine the prompts based on the feedback.

What generative AI tools are available today?

Today, software engineers have several generative AI tools to assist them in application development, like GitHub Copilot, Cursor.so, and AutoGPT. While some tools are more generalized, others have specific strengths. Depending on the desired output, most generative AI tools will help the end user to construct better prompts by adding additional information to the initial prompt. Below is a partial list of generative AI systems and their unique benefits for developers undertaking various phases of the SDLC:

- **Writing general user stories and software requirements:** AI chatbots like OpenAI, ChatGPT, and Google Bard
- **Coding and testing support:** Github Copilot, Visual Studio Intelllicode, Cursor, Tabnine, Replit, Codium, and Mutable.AI
- **Facilitating deployment:** Github Copilot and ChatGPT
- **Streamlining DevOps:** Robusta ChatGPT bot and k8sgpt

⁹ "The AI Job That Pays Up to \$335K—and You Don't Need a Computer Engineering Background," Time, April 14, 2023

¹⁰ "Weird New Job Alert: What Is an AI Prompt Engineer?" PC Magazine, May 22, 2023

¹¹ "What is Prompt Engineering – The AI Revolution," Geeks for Geeks

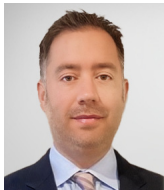
¹² "Best practices for prompt engineering with OpenAI API," OpenAI

How KPMG can help

An early and enthusiastic advocate for the power of artificial intelligence, KPMG can help your organization leverage generative AI to transform software development and maintenance. Drawing on our deep expertise in machine learning and natural-language processing, we can guide your organization through strategy development, platform selection, and implementation—and then provide ongoing support to help you optimize your investment in this transformative technology. We understand the promise of generative AI and the process and cultural changes—including the embrace of responsible AI practices—required to realize its full potential.

Contact us

To learn more out more about general artificial intelligence and how it can accelerate your company's software development process, please contact:



Kevin Martelli

Principal, Advisory
Lighthouse
kevinmartelli@kpmg.com



Hongfei Cao

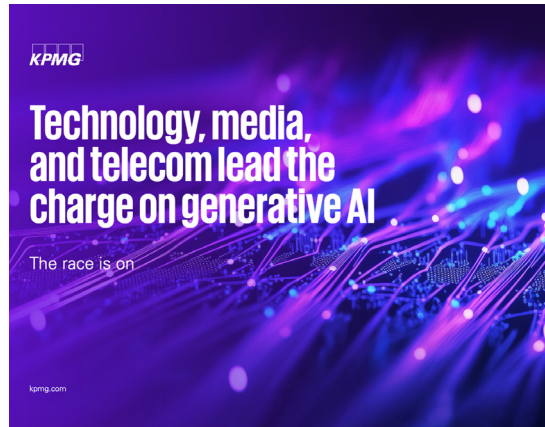
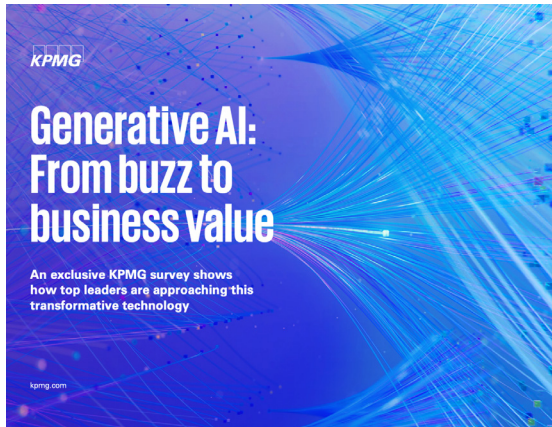
Director, Data Engineer
Lighthouse
hongfeicao@kpmg.com



Bill Cheng

Manager, Data Engineer
Lighthouse
xcheng1@kpmg.com

Related thought leadership:



Learn how KPMG can help make your [generative AI implementation](#) successful, and explore how we can help you [adopt AI](#) in a safe, trustworthy, and ethical manner.

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act upon such information without appropriate professional advice after a thorough examination of the particular situation.

© 2023 KPMG LLP, a Delaware limited liability partnership and a member firm of the KPMG global organization of independent member firms affiliated with KPMG International Limited, a private English company limited by guarantee. All rights reserved.

DASD-2023-13415

September 2023

kpmg.com/socialmedia

