

Generative AI can dramatically accelerate developer productivity, but enterprises need the right tools to ensure safety and security.

Accelerating Software Development with Generative AI: An IT Leader's Guide

August 2023

Written by: Michele Rosen, Ph.D., Research Manager, Low-Code, No-Code, and Intelligent Developer Technologies

Introduction

IT leaders have faced many changes in the past few years. The COVID-19 pandemic led to a radical reconfiguration of workspaces, which are no longer bound to workplaces. Organizations accelerated their transformation into digital businesses to meet customers' needs in this new environment, while seeking ways to cut costs in light of economic and geopolitical instability.

Through all these changes, one thing has remained the same: the shortage of technical talent. Organizations continue to struggle to hire and retain developers. IDC estimates that the global demand for developers will exceed supply by 4 million in 2025. According to IDC's February 2023 *Worldwide C-Suite Survey: CIO Perspectives*, achieving developer productivity is one of the top concerns for CIOs.

IT leaders are increasingly looking to generative AI to help them overcome this persistent obstacle to digital transformation. As a result of the technology's potential to solve time-intensive, real-world problems, business leaders have already moved from the question of "What is generative AI?" to "How do I use generative AI?" IT leaders are experimenting with and investing in this technology to ensure their organizations do not fall behind their competitors. Technology leaders and decision makers are looking for strategic guidance on how to leverage generative AI to accelerate developer productivity safely and securely.

Generative AI in the Software Development Life Cycle

Generative AI software creates new content (text, images, audio, video, code, etc.) in response to short prompts by leveraging large language models (LLMs) trained on vast amounts of previously created content. Google researchers introduced the Transformer architecture that enabled the creation of LLMs in the 2017 paper *Attention Is All You Need*. LLMs have several advantages over earlier AI models, including:

- » Parallelizable training, enabling a significant increase in the scale of model parameters and training data sets

AT A GLANCE

KEY STATS

- » Among global organizations, 45% are currently exploring generative AI use cases and 32% are investing significantly in generative AI technologies this year (source: IDC's *Future Enterprise Resiliency and Spending Survey*, July 2023).
- » Generative AI will have the greatest impact on software development and design in the next 18 months, beating customer engagement, marketing/PR, sales, and other business areas (source: IDC's *Future Enterprise Resiliency and Spending Survey*, March 2023).

- » Unsupervised training, eliminating the need for data labeling
- » The ability to fine-tune models for specific use cases using a small amount of additional training data
- » A wide range of capabilities compared with previous AI models, which targeted narrow use cases

During the past five years, these models have become increasingly capable of a wide range of human-level cognitive tasks, including writing code. Conversational interfaces also allow the use of LLMs to provide technical assistance using natural language. Because of their broad capabilities, LLMs are now often referred to as foundation models. These foundation models can also be adapted or fine-tuned using custom data to make them even more applicable for specific use cases.

Given recent improvements in foundation models, IDC predicts that by 2027, AI will be capable of automatically generating code to meet functional business requirements for 80% of new digital solutions in development and early deployment. Moreover, developers will increasingly interact with a team of AI assistants, making it easier for less experienced developers to get up and running and for more experienced developers to focus on more challenging tasks, including incorporating intelligent features into their digital solutions.

According to IDC's survey about developer attitudes toward the adoption of generative AI (see Figure 1), developers expect the technology to increase their productivity (60.9%), give them the opportunity to spend more time on higher-value development tasks (50.0%), and accelerate developer velocity (39.8%).

FIGURE 1: **Developer Attitudes Toward AI**

Q How do generative AI or intelligent development suggestions contribute to developer happiness?



n = 128

Base = respondents who indicated generative AI and intelligent development suggestions have the most positive impact on their happiness as developers

Notes:

This survey is managed by IDC's Global Primary Research Group.

Data is not weighted.

Multiple responses were allowed.

Use caution when interpreting small sample sizes.

Source: IDC's U.S. Generative AI Survey, April 2023

Figure 2 shows that developers are particularly keen to use generative AI–driven tools for automated software quality and testing, security testing, and vulnerability management even more than for writing code (see *Generative AI Adoption and Attitudes: A Survey of U.S. Developers*, IDC #US50655123, May 2023).

FIGURE 2: **Where Generative AI Tools Can Aid Development**

Q What application development process do you think has the most potential to benefit from generative AI?



n = 200

Base = all respondents

Notes:

This survey is managed by IDC's Global Primary Research Group.

Data is not weighted.

Use caution when interpreting small sample sizes.

Source: IDC's U.S. Generative AI Survey, April 2023

This Technology Assessment Guide explores the leading use cases for AI-assisted development and DevOps and the critical questions that IT leaders must consider when adopting this technology.

Key Use Cases for Generative AI in the Software Development Life Cycle

Generative AI can write poems, paint portraits, and mimic the sound of a person's voice. However, its inherent natural language capabilities make it especially suitable for providing technical assistance. LLMs fine-tuned to answer questions are the basis for the chatbots that have drawn so much recent public attention. With additional training and tuning, these conversational interfaces can become specialized assistants for tasks across the software development life cycle.

AI-driven technical assistants can improve productivity for all developers by eliminating repetitive tasks, answering questions, and providing an informed interlocutor for brainstorming and learning. These capabilities help developers by reducing cognitive load and the need for context switching. Psychologists define cognitive load as the amount of working

memory being used at a given time. Heavy cognitive load leads to errors and overgeneralizations, particularly for more difficult tasks.

Research shows that multitasking, such as switching to a browser to look up information while coding, reduces productivity. Vendors can help developers reduce or eliminate the need to context switch by integrating an AI-powered conversational interface into development tools.

AI-driven technical assistants can also help developers by automating tasks in real time. Organizations are already using process automation tools to automate tasks throughout the software development process, but generative AI-driven technical assistants empower developers to offload tasks on an ad hoc basis during the development process where appropriate, decreasing human errors and accelerating processes.

AI-driven technical assistants also embody the human-in-the-loop (HITL) approach to AI, which emphasizes solutions that assist and augment human capabilities instead of replacing them. HITL-oriented tools are more likely to produce aligned output because they are being guided through an iterative development process in a feedback loop.

AI-Assisted Development

Ever since the launch of digital computers, humans have had to communicate with them using formal languages that are convenient for the computer but difficult for humans to learn, read, and remember. Today, foundation models can generate much larger blocks of code than traditional code completion tools, which augment a developer's memory by providing context-sensitive lists of terms and parameters as the developer is typing. Generative AI-based tools can recommend full functions and blocks of code in response to comments or by evaluating context as the developer types. These tools can also explain code in natural language, making it much easier to update existing code and providing a learning tool for less experienced developers in conjunction with a chat assistant.

Because of their versatility, generative AI-driven code assistants and code generation tools can improve developers' productivity throughout the software development life cycle:

- » Professional developers expect generative AI to help them design and develop user interfaces (see *Generative AI Adoption and Attitudes: A Survey of U.S. Developers*, IDC #US50655123, May 2023).
- » Interacting with a code assistant through natural language will empower line-of-business developers to create digital solutions that they could not have otherwise created, particularly in combination with no-code development environments.
- » Infrastructure developers can also benefit from generative AI-powered tools. Deployment automation is one of the top development tasks that developers expect generative AI-driven tools to augment (see *Generative AI Adoption and Attitudes: A Survey of U.S. Developers*, IDC #US50655123, May 2023).
- » Data engineers, scientists, and analysts can benefit from generative AI-driven tools to add this technology to their existing tools and development interfaces, including notebooks.

By augmenting existing skill sets, generative AI-driven tools can facilitate collaboration, upskilling, and cross-skilling, all of which contribute to productivity.

To evaluate generative AI–assisted development tools, IT leaders should determine which of the following questions are the most important considerations for their enterprise use cases and then evaluate vendor offerings based on those questions:

- » What kind of data was used to train the model?
- » If the training data included public code, is the source license of the code identifiable?
- » Can a private model be trained using curated code?
- » Can the model be fine-tuned to generate specific programming languages or to use selected programming design patterns?
- » Does the tool rely on a single model or multiple fine-tuned models?
- » How accurate is the generated code for a range of use cases?
- » How fast is the tool? Latency above 100ms can interrupt developer flow.
- » Are both the developer's input and the model's output supplemented by complementary tools, such as security scanning or software quality tools?
- » Does the tool integrate with developer IDEs? Because developers have a high degree of choice with respect to developer tools, it is very difficult to convince them to change their primary development environment.
- » If the tool will be used by line-of-business or junior developers, will it integrate with no-code or low-code development tools to further increase productivity?
- » Does the tool provide enterprise-level monitoring and control over developers' use of generative AI features?

AI-Assisted DevOps

Software Testing

Enterprises have been using AI to define and execute test cases for several years. However, despite advances in test automation, DevOps teams continue to struggle with software testing, which impacts application quality and code release deployment frequency. As a result, more than 50% of DevOps teams are expanding their AI/ML use or are using, or piloting the use of, AI/ML to augment software testing (source: IDC's *DevOps Practices, Tooling, and Perceptions Survey*, November 2022). According to the survey, the top 5 use cases for AI to augment the testing process are:

- » Test prioritization
- » Root cause analysis for failed tests
- » Automated test case creation
- » Self-healing test case maintenance
- » Test process improvement insights

Because generative AI models are far more flexible than previous AI models, they can augment existing testing tools, improving their efficiency and leading to higher-quality results. For generative AI-based testing tools to contribute to developer productivity, they will have to be based on models trained on high-quality, domain-appropriate training data and integrated with tools that enable enterprise-level governance and administration.

DevSecOps

Enterprises continue to face a growing application threat landscape. As a result, many companies have adopted DevSecOps practices that involve shifting security further to the left — as far as the planning and design phases — so that all enterprise applications are secure by design. Key priorities include ensuring security across multiple cloud environments, security policy management, and software supply chain security (source: IDC's *DevSecOps Survey*, January 2023).

However, IT leaders say that developers need more security knowledge and training to facilitate DevSecOps practices. To address this skills gap, AI-powered assistants can help developers and DevOps professionals model threats, analyze scan results, detect anomalies, recommend remediation, and generate management reports.

AI-powered assistants can also help developers and DevOps professionals find the information they need to make decisions. Foundation models are exceptionally good at finding patterns in and summarizing unstructured data, conversations, and policies. Conversational assistants that use models tuned on custom enterprise data can query logs and scans to guide remediation, explain security scan results and attack paths, and make security recommendations.

Next Steps

IT leaders seeking to leverage generative AI in the software development life cycle can take the following steps to move forward:

- » Identify use cases and evaluate their applicability to pain points in the enterprise development process.
- » Evaluate the risks of using generative AI for the identified use cases. Determine how the use case would be impacted by:
 - Hallucinations
 - Unvetted training data
 - Legal challenges to training data use
 - Risk of data leakage
 - Data bias
- » Identify key tooling, staffing, and resourcing required to operationalize and manage generative AI for each use case.
- » Enable iterative experimentation and adoption by providing a sandbox environment with the appropriate generative AI capabilities for each prioritized use case.
- » Determine which use cases can be served with off-the-shelf generative AI solutions and which require the use of enterprise data and additional governance and security capabilities. Further:
 - Evaluate the enterprise's data resources to determine the most useful data for few-shot training and fine-tuning.

Worksheet Section

Organizations can choose generative AI tools/vendors based on the frameworks provided in Tables 1 and 2.

TABLE 1: *Use Case Assessment*

Application Development Processes with the Potential to Benefit from Generative AI	How Well the Current Solution Addresses This Need (Very Well, Not Well, Don't Know)
Software quality and testing	
Security testing and vulnerability management	
Writing code (text to code)	
Understanding code (code to natural language)	
User interface design	
Refactoring (language translation, modernization)	
Prototyping	
Deploying code	
Application monitoring	
Requirements, user stories, and value stream management	

Source: IDC, 2023

TABLE 2: **Solution Evaluation Criteria**

	Yes/No
Does the platform provide appropriate developer experiences for all personas related to your key use cases?	
Is the tool or platform capable of scaling to meet all potential use cases?	
Does the tool or platform enable customization with enterprise code and data?	
Does the vendor provide the ability to fine-tune the foundation model to meet specific use cases or vertical industry requirements?	
How does the vendor ensure the security and privacy of enterprise data?	
Does the vendor have tools to manage the enterprise data that will be needed to fine-tune models for relevant use cases?	
Does the vendor provide tools to ensure responsible AI governance?	
How has the vendor addressed generative AI-specific risks, such as hallucinations and bias?	
Does the vendor provide a model card or other observability features?	
How does the vendor ensure the quality of generated code?	
Does the vendor enable integration of generative AI tools with other tools in the software development life cycle?	

Source: IDC, 2023

About the Analyst



Michele Rosen, Ph.D., Research Manager, Low-Code, No-Code, and Intelligent Developer Technologies

Michele Rosen is research manager for IDC's Low-Code, No-Code, and Intelligent Developer Technologies practice. Dr. Rosen's research focuses on tools for application development that enable the democratization of the development process using low-code, no-code, and intelligent developer technologies and on the impact of AI on the development process. She directs research on how these developer tools accelerate digital transformation and allow professional and line-of-business developers to build apps that solve business problems and drive competitive differentiation.

MESSAGE FROM THE SPONSOR

Powered by Google's state-of-the-art generative AI foundation models, Duet AI in Google Cloud is an always-on AI collaborator that provides generative AI-powered assistance to cloud users of all types where they need it.

It embeds the power of generative AI across Google Cloud products and services to provide intelligent AI assistance to cloud users — developers, operators, data engineers and analysts, security professionals, and even citizen developers — so they can get more work done faster while navigating Google Cloud with more confidence and ease.

With Duet AI, we're on a mission to deliver a new cloud experience that's personalized and intent-driven and can deeply understand your environment to assist you in building secure, scalable applications while providing expert guidance.

Learn more about <https://cloud.google.com/duet-ai> and start your generative AI journey [here](#).



The content in this paper was adapted from existing IDC research published on www.idc.com.

This publication was produced by IDC Custom Solutions. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2023 IDC. Reproduction without written permission is completely forbidden.

IDC Research, Inc.
140 Kendrick Street
Building B
Needham, MA 02494, USA
T 508.872.8200
F 508.935.4015
Twitter @IDC
idc-insights-community.com
www.idc.com