



Generative AI in the Software Development Lifecycle (SDLC)

BY

Abhishek Gupta¹, Jerish Sam David², Achraf Golli³, Venkata Sai Manoj Pasupuleti⁴

^{1,2}Cisco Systems

³Quizard AI, USA

⁴Lead Software Engineer, USA



Article History

Received: 15/01/2025

Accepted: 28/01/2025

Published: 31/01/2025

Vol – 4 Issue – 1

PP: - 26-31

Abstract

Generative AI is revolutionizing the Software Development Life Cycle (SDLC) by streamlining processes, enhancing productivity, and improving software quality. As its adoption accelerates across industries, generative AI is becoming an indispensable part of software development workflows. This paper explores the integration of generative AI into SDLC phases, including requirements gathering, design, coding, testing, deployment, and maintenance. It highlights the transformative potential of generative AI, analyzing its applications and addressing the challenges associated with its incorporation. Furthermore, the paper examines the long-term impact of generative AI on key development tasks, emphasizing its capacity to reshape software engineering methodologies and influence all SDLC participants. By providing insights into both current applications and future potential, this paper underscores the pivotal role of generative AI in driving efficiency, innovation, and progress within the software industry.

INTRODUCTION

The Software Development Life Cycle (SDLC) is a structured framework guiding software application development through distinct phases. Traditionally, these processes have been heavily reliant on human expertise, often resulting in inefficiencies and errors. Generative AI, an advanced branch of artificial intelligence dedicated to creating content and solutions, has emerged as a transformative force in SDLC. By improving efficiency, accuracy, and innovation, generative AI is rapidly becoming integral to modern software development practices. Organizations are integrating generative AI into various applications, including customer-facing interfaces such as shopping platforms and banking services. Additionally, businesses leverage generative AI internally to automate routine tasks, equipping employees with intelligent tools like developer co-pilots and AI-assisted interfaces to enhance productivity.

GENERATIVE AI OVERVIEW

Generative AI is capable of creating diverse content, including text, video, audio, and images. These models learn patterns from vast datasets, such as online repositories and social media, to generate contextually relevant content.

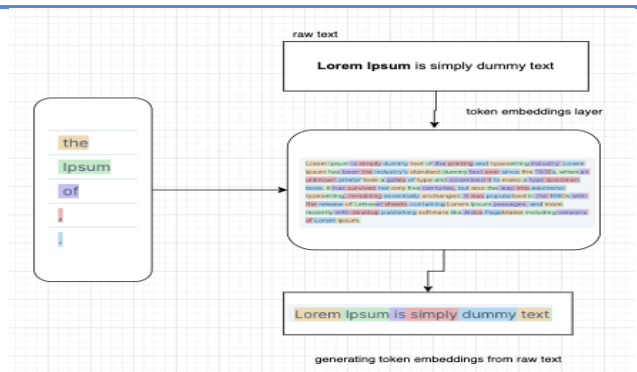


Figure 1: Tokens generated from embeddings

The above figure depicts the next token prediction, also known as the standard language modeling objective, is a fundamental concept behind causal language models.

What is a Token?

A token is a fundamental unit of text, either a word or a sub-word, that forms the basis of AI language processing. When raw text is input into a language model, the first step involves tokenization, which breaks the text into discrete tokens. For instance, the phrase "Understanding Next Token Prediction" might be tokenized into ["Understanding", "Next", "Token", "Prediction"].

Tokenizing a Sequence of Raw Text

A tokenizer is responsible for converting raw text into tokens. It is trained on a diverse corpus of text to establish a fixed vocabulary that the model recognizes. Ensuring that the training



data aligns with the model's intended application is essential for effective performance.

Token Embeddings

Once tokenized, each token is mapped to a corresponding vector representation using an embedding layer within the language model. This transformation enables the model to interpret textual data numerically, facilitating deeper contextual understanding.

Comprehending next-token prediction requires a strong grasp of how text is tokenized and subsequently transformed into meaningful embeddings.

Generative AI in SDLC

Generative AI is reshaping SDLC by automating repetitive tasks, minimizing human errors, and optimizing workflow efficiency. Various AI-powered tools facilitate different phases of software development.

Phase	Tools
Requirements Gathering	IBM Engineering, WriteMyPrd
Design	AlphaCode by Google, GitHub Copilot, cursor, Duet AI by Google, Claude.ai by Anthropic, Amazon Bedrock, Google Vertex AI
Code Generation	AlphaCode by Google, GitHub Copilot, cursor, Duet AI by Google, Claude.ai by Anthropic, Amazon Bedrock, Google Vertex AI
Automated Testing	Functionize, Katalon, Applitools, Cypress, Ranorex
Implementation	AlphaCode by Google, GitHub Copilot, cursor, Duet AI by Google, Claude.ai by Anthropic, Amazon Bedrock, Google Vertex AI
Code Review	IBM watsonx Code Assistant, GH co-pilot, cursor
Project Management	watsonx BI Assistant, Asana, ClickUp, Monday, Basecamp, Trello, Motion, OneCal, Slack, Loom
Maintenance	restack, AlphaCode
Support	chatgpt, bard, scribe, Cohere Generate
documentation	chatgpt, bard, scribe,

Table 1: Popular gen AI tools in SDLC
Let's dive into each of these phases.

Requirements Gathering

Generative AI (GenAI) has the potential to significantly enhance product feature development by automating the creation of detailed and contextually relevant features using standardized templates. The following section provides an in-depth explanation of how this process works.

Inputs for GenAI

For GenAI to generate product features effectively, it requires the right inputs, which may include

User Requirements

Detailed descriptions of what users need or want from the product.

Market Research Data

Insights into current market trends and competitor analysis.

Product Specifications

Technical details and constraints of the product. Historical Data: Information from previous product versions or similar products. Generating Relevant, Detailed Features

Given these inputs, GenAI can perform the following tasks:

Analyzing User Requirements

GenAI processes natural language inputs to understand the needs and preferences of users. It identifies patterns and common themes in user requirements to generate relevant features.

Synthesizing Market Research

GenAI analyzes market trends to ensure that the generated features align with current market demands. It compares features with those offered by competitors to ensure competitiveness.

Adhering to Product Specifications

GenAI ensures that the generated features are feasible within the given technical constraints. It suggests innovative features that push the boundaries of current product capabilities.

Standardized, Easy-to-Use Templates

One of the significant advantages of using GenAI is its ability to produce features in standardized templates, making them easy to use and implement. GenAI ensures that all generated features follow a consistent format, making it easier for teams to understand and utilize them. Each feature is described in detail, including its functionality, benefits, and implementation guidelines.

Benefits of Automatic Feature Generation

Automating the feature generation process saves significant time and effort, allowing teams to focus on other critical tasks.

DESIGN AND CODE GENERATION

Generative AI solutions revolutionize software development by facilitating code generation from simple text prompts, modernizing legacy systems, and translating code across programming languages. By incorporating AI, developers gain access to high-quality code suggestions, enhancing productivity and streamlining workflows. Additionally, these tools help detect coding errors and security vulnerabilities, enabling the



creation of more efficient and secure software. The advancement of large language models (LLMs) and natural language processing (NLP) has made generative AI for coding possible. These technologies utilize deep learning algorithms and large neural networks trained on extensive datasets from open-source projects.

How Generative AI Enhances Coding?

- **Training Data:** AI models are trained on extensive datasets sourced from open-source repositories, enabling them to recognize diverse coding styles, structures, and patterns.
- **User Prompts:** Developers input plain text descriptions of the desired functionality, which serve as guidelines for the AI to generate relevant code.
- **Code Suggestions:** Generative AI tools provide code snippets or entire functions based on the given prompts, automating repetitive tasks and minimizing the need for extensive manual coding, ultimately streamlining the development process.

Expanded Capabilities of Generative AI in Coding

- **Code Translation:** Generative AI facilitates seamless translation between programming languages, making it a valuable tool for code conversion and modernization. For instance, it can convert legacy COBOL code into modern Java, simplifying the process of updating outdated applications.
- **Quality, Review, and Accuracy:** While AI-generated code continues to improve in accuracy, it is not entirely error-free. Human developers must review, edit, and refine the generated code to ensure quality, correctness, and adherence to best practices.
- **Automated Unit Testing:** Certain generative AI tools can automatically generate unit tests, helping identify and resolve potential issues early in the development cycle.

By leveraging advancements in Large Language Models (LLMs) and Natural Language Processing (NLP), generative AI streamlines and automates various aspects of the coding workflow, enhancing efficiency and productivity.

Key Success Factors

Leading tech companies are rapidly adopting generative AI, with Gartner reporting that two-thirds of businesses are either piloting or deploying AI-driven coding tools. Furthermore, projections indicate that by 2028, 75% of engineering professionals will be utilizing AI coding assistants to enhance development efficiency and innovation.

Currently, 82% of developers using AI tools primarily utilize them for writing code. Among those interested in AI tools but not yet using them, 46% are most curious about AI's potential in testing code. Experienced developers may rely on AI tools to assist with initial code writing, while recognizing that the complexity of testing is best addressed through traditional

methods.

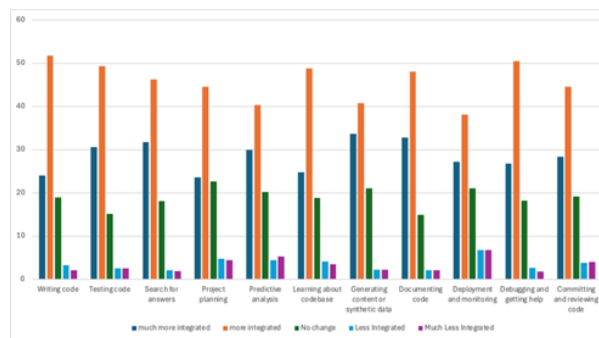


Figure 2: Survey on using AI tools by next year (courtesy: stack overflow)

In the coming year, most developers anticipate greater integration of AI tools, particularly in documenting code (81%), testing code (80%), and writing code (76%).

Interest in generative AI has highlighted a broader range of AI capabilities. For the past six years, AI adoption in organizations has remained around 50 percent. However, this year's survey reveals a significant increase to 72 percent. The interest in AI is global, with more than two-thirds of respondents in nearly every region reporting AI usage in their organizations. The professional services industry has seen the most substantial increase in AI adoption.

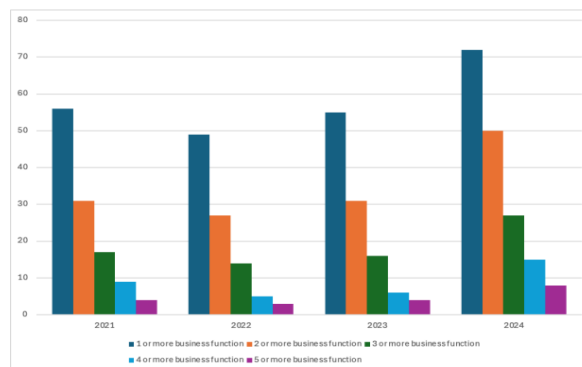


Figure 3: Companies using AI in more parts of the business

Generative AI is reshaping industries by revolutionizing workflows, much like the personal computer and the Internet once did. It is driving the emergence of new business models and redefining enterprise operations. The scope of AI-driven innovation continues to expand, with creative tools like Adobe's Firefly enhancing design processes. Even historically resistant sectors, such as energy and chemicals, are integrating AI for critical functions like production scheduling and sales optimization, exemplified by DuPont's adoption of advanced language models.

Use Cases of Generative AI

- **Predictive Forecasting:** Generating accurate forecasts for complex scenarios.
- **Data Analysis:** Processing and interpreting large volumes of complex, unstructured data.
- **Performance Metrics:** Aggregating essential metrics across production systems for better insights.

- Automated Reporting: Creating text, charts, and graphs dynamically for reports.
- Operational Optimization: Analyzing key operational factors, such as inventory management and workforce planning.
- Dynamic Pricing Optimization: Enhancing pricing strategies based on market trends and demand.
- Automated Document Processing: Generating and adapting contracts, purchase orders, and invoices efficiently.
- User Behavior Analysis: Understanding customer preferences, behaviors, and contextual patterns.
- AI-Driven Customer Support: Automating and personalizing customer service interactions.
- Support Ticket Insights: Learning from past and unresolved tickets to improve response strategies.
- Intelligent Agent Assistance: Providing optimized scripts for customer service representatives.
- Personalized Marketing Strategies: Crafting tailored marketing campaigns based on user data.
- Advanced Ad Targeting: Improving ad precision and effectiveness through AI-driven insights.
- Cybersecurity Enhancement: Detecting security threats and preventing account takeover attempts.
- Fraud Prevention: Analyzing communications for phishing attacks and social engineering threats.



Figure 4: Generative AI tools usage, 2023–24, by industry

In most industries, organizations are equally likely to allocate more than 5 percent of their digital budgets to both generative AI and analytical AI.

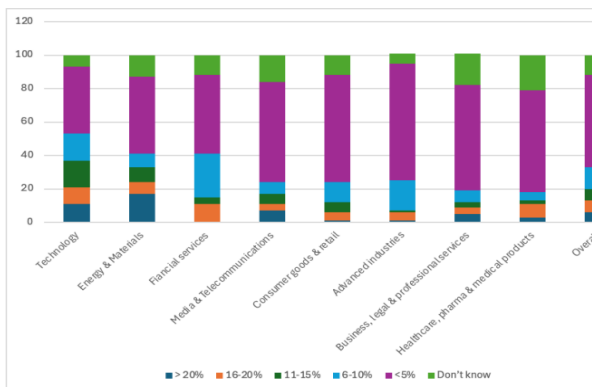


Figure 5: Share of organization's digital budget spent on generative AI

Evolving Team Dynamics with AI Agents

AI agents are sophisticated software programs powered by machine learning algorithms that execute tasks traditionally performed by human developers. These intelligent systems can analyze context, identify patterns, and make informed decisions, significantly enhancing the software development process. The global AI market in software development is projected to reach **\$11.2 billion by 2025**, growing at a **34.7% compound annual growth rate (CAGR) from 2020 to 2025**.

As AI continues to integrate into agile methodologies, the way teams estimate work is evolving. Today, a user story may require **eight human workdays** to complete. In the near future, the same task could be distributed among human developers and AI-powered tools. For instance, the effort could be reduced to **four human workdays**, supplemented by **two AI coding agents, one AI testing agent, and one AI-driven CI/CD agent** to achieve completion efficiently.

AI-powered tools are transforming software development, maintenance, and deployment by driving efficiency and innovation. A **Deloitte survey** found that **74% of IT and business leaders** expect AI to be integrated into all enterprise applications within the next three years. This rapid adoption underscores the potential of AI agents to enhance productivity, streamline workflows, and improve code quality across all phases of the **software development lifecycle (SDLC)**.

AI-Powered Coding Assistant

AI coding assistants, such as **GitHub Copilot**, have revolutionized software development by transforming how code is written and maintained. These tools are gaining widespread adoption within the developer community, significantly enhancing productivity. According to a **2022 GitHub survey**, **77% of developers** using AI coding assistants reported increased efficiency, with an average **55% reduction in coding time**.

AI coding assistants can generate **code snippets**, complete **function definitions**, and suggest **entire algorithms** based on the project's context. This automation minimizes time spent on repetitive tasks, allowing developers to focus on complex problem-solving and creative software design. Developers leveraging tools like GitHub Copilot reported completing tasks **55.8% faster on average**, with efficiency gains reaching **61.7% in languages like JavaScript and Python**. By handling boilerplate code and common programming patterns, these assistants empower developers to devote more time to **architectural decisions and innovation**, ultimately accelerating development cycles and improving job satisfaction.

AI-Driven CI/CD Assistant

AI integration into **Continuous Integration and Continuous Deployment (CI/CD)** pipelines is proving to be a game-changer for enterprises, driving automation and efficiency in software deployment. According to a **MarketsandMarkets report**, the **global AI in DevOps market** is projected to grow from **\$2.6 billion in 2022 to \$10.4 billion by 2027**, with a



compound annual growth rate (CAGR) of 32.2%. This rapid expansion underscores the growing importance of AI in optimizing **CI/CD workflows**, reducing deployment errors, and ensuring seamless software delivery.

Benefits of using AI augmented teams:

1. Accelerated Coding: AI agents help speed up coding by generating code and prompting code based on the current module and intent. It also helps developer to write unit test, integration tests and end to end test like behavioral driven test
2. Debugging: AI agents can automatically detect errors, bugs, and anomalies in the code by analyzing patterns and comparing them with known issues. This helps in quickly identifying potential problems without manual intervention. AI can analyze extensive log files to identify patterns and anomalies that indicate the presence of bugs.
3. Code Quality : AI tools can perform deep code analysis to understand the underlying logic and dependencies. This helps in pinpointing the exact location and cause of bugs, making the debugging process faster and more accurate.
4. Automated Tests: AI can generate test cases automatically, ensuring comprehensive coverage and helping identify edge cases that might be missed by human developers. This leads to more robust and error-free code.
5. CICD process: CICD AI tools like harness, Spinnaker, DeepCode help in building CICD pipelines. Spinnaker is an open-source continuous delivery platform that can be enhanced with AI capabilities for smarter deployment strategies. Deep Code provides AI-driven code reviews and suggestions, integrating with CI/CD tools to ensure code quality.

Agentforce, recently launched by Salesforce is an augmented team of AI agents with real humans. It is the culmination of all Einstein products, formerly known as Copilot, designed to create generative "agents." The agent builder features a simple UI based on prompt engineering and natural language processing.

The Large Language Models (LLMs) interpret natural language inputs (mapping chat to topics and instructions), while the Large Action Models (LAMs) act as deterministic engines that utilize all available data (chat history, knowledge base, prompts, data Cloud, etc.) to determine the best course of action to address a question or problem.

Agents can be customized for roles like sales coaches or customer service reps, serving as Level 1 support and escalating to human agents when needed. They enhance capabilities by accessing knowledge data. Google reports increased productivity with AI in software development, where AI assists in completing 50% of code characters, allowing developers more time for code design. The base builder's natural language accessibility simplifies use, while complex actions utilize additional tools.

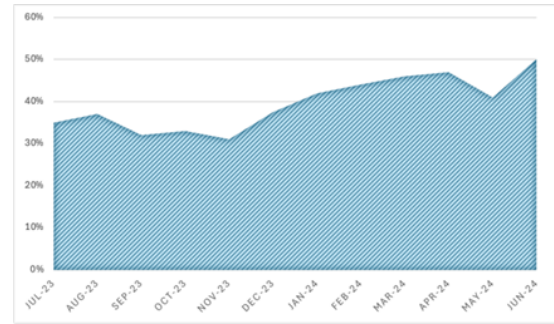


Figure 6: Increase of code created with AI assistance at Google

CHALLENGES OF INTEGRATING GENERATIVE AI INTO EXISTING WORKFLOWS

While generative AI presents numerous advantages, its implementation is accompanied by several challenges:

- **Selecting Valuable Backlog Items:** Organizations must carefully assess backlog items based on potential business impact, feasibility, and return on investment. Frameworks such as the Value vs. Effort matrix can aid in prioritization.
- **Balancing Short-Term vs. Long-Term Benefits:** While generative AI offers immediate productivity gains, organizations must ensure that its deployment aligns with sustainable business models.
- **Resource Efficiency:** The implementation of generative AI can be costly. Companies should begin with targeted pilot projects to demonstrate its value before scaling up.
- **Alignment Gaps:** A lack of synchronization between business objectives and AI initiatives can hinder success. Clear communication and alignment with strategic goals are crucial.
- **Ethics & Compliance:** AI models may reinforce biases inherent in training data. Establishing an ethical AI oversight board and conducting routine audits are essential to maintaining fairness and compliance.
- **Scaling AI Solutions:** Transitioning from pilot projects to full-scale AI adoption necessitates meticulous planning and cross-functional collaboration.
- **Skills & Expertise:** Upskilling employees and fostering AI proficiency within teams is vital for effective AI integration.
- **Cognitive Abilities:** Over-reliance on AI may stifle human creativity and critical thinking. AI should be used as an augmentative tool rather than a substitute for human decision-making.
- **Hallucination & Accuracy:** AI models occasionally generate plausible but incorrect outputs. Human oversight remains necessary to verify accuracy.
- **Plagiarism & Intellectual Property:** AI-generated content must be reviewed to prevent unintended plagiarism and protect intellectual property rights.



- Privacy & Transparency: Organizations must ensure transparency regarding AI training data and decision-making processes to foster trust and compliance.

REFERENCES

1. G. M. Amdahl, G. A. Blaauw, and F. P. Brooks, "Architecture of the IBM System/360," *IBM J. Res. & Dev.*, vol. 8, no. 2, pp. 87–101, 1964. (journal)
2. Bazzan, T., Olojo, B., Majda, P., Kelly, T., Yilmaz, M., Marks, G., & Clarke, P. M. (2024, September). Analyzing the Role of Generative AI in Software Engineering-Results from an MLR. In the European Conference on Software Process Improvement (pp. 163-180). Cham: Springer Nature Switzerland.
3. Şimşek, T., Gülşeni, Ç., & Olcay, G. A. (2024). The Future of Software Development With GenAI: Evolving Roles of Software Personas. *IEEE Engineering Management Review*.
4. "2024 CEO Study: 6 hard truths CEOs must face," IBM, <https://www.ibm.com/thoughtleadership/institute-business-value/c-suite-study/ceo>
5. "ArtificialIntelligence (AI) Solutions | IBM." <https://www.ibm.com/artificial-intelligence>
6. "The great acceleration: CIO perspectives on generativeAI" https://www.databricks.com/sites/default/files/2023-07/ebook_mit-cio-generative-ai-report.pdf
7. T. Bazzan et al., "Analyzing the Role of Generative AI in Software Engineering - Results from an MLR," in *Communications in computer and information science*, 2024, pp.163–180.doi: 10.1007/978-3-031-71139-8_11.
8. "GPTtokenizerplayground." <https://gptforwork.com/tools/tokenizer>
9. PricewaterhouseCoopers, "10 ways GenAI improves software development," PwC. <https://www.pwc.com/us/en/tech-effect/ai-analytics/generative-ai-for-software-development.html>
10. P. Khare, "GenAI in Test Automation - 5 best AI-based tools," May 06, 2024. <https://www.linkedin.com/pulse/genai-test-automation-5-best-ai-based-tools-prasoon-khare-rvlwe/>
11. "The state of AI in early 2024: Gen AI adoption spikes and starts to generate value," McKinsey & Company, May 30, 2024. <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>
12. "AI | 2024 Stack Overflow Developer Survey." <https://survey.stackoverflow.co/2024/ai>
13. "AI in software engineering at Google: Progress and the path ahead." <https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/>
14. "Overcoming Challenges in Generative AI Implementation" <https://www.signitysolutions.com/blog/challenges-in-generative-ai-implementation>
15. Duhaylungsod, A. V., & Chavez, J. V. (2023). ChatGPT and other AI users: Innovative and creative utilitarian value and mindset shift. *Journal of Namibian Studies: History Politics Culture*, 33, 4367-4378.
16. N. A. V. Duhaylungsod and N. J. V. Chavez, "ChatGPT and other AI Users: Innovative and Creative Utilitarian Value and Mindset Shift," *Journal of Namibian Studies History Politics Culture*, vol. 33, Mar. 2023, doi: 10.59670/jns.v33i.2791.
17. Abd-Alrazaq, A., et al. (2023). "The Role of AI in Enhancing Software Development Practices." *Journal of Software Engineering*.
18. Koos, T., & Wachsmann, M. (2023). "Challenges and Opportunities of Integrating AI in SDLC." *International Journal of Computer Science*.
19. Dergaa, J., et al. (2023). "AI-Driven Solutions for Automated Software Testing." *Proceedings of the Software Engineering Conference*.