

Generative AI

The transformative potential to revolutionize
the software engineering landscape in 2024.

Editorial

Generative AI and its rapid adoption has transformed the way we interact with technology, allowing machines to create, design, understand and generate solutions and content with significant implications for sectors, companies and consumers - a clear example was the launch of ChatGPT, which accumulated more than a million users in five days and one hundred million in two months, managing to democratize AI in a way never seen before and becoming the fastest growing application in history.

In the field of software development, **the artificial intelligence revolution translates into significant increases in productivity, quality and agility** since it has an impact on various processes related to the software development cycle. Broadly speaking, it produces an increase in productivity, greatly increasing the effectiveness of routine and automatable development tasks, which can mean, depending on the project and the software, freeing up between 30% and 50% of daily time from a developer. On the other hand, AI accompanies and stimulates creativity in solution design and development tasks, increasing the ability data efficiently and allows articulating monitoring strategies that allow errors to be anticipated. As the development and implementation of

Generative AI systems progresses, a new value chain is emerging to support the training and use of this powerful technology.

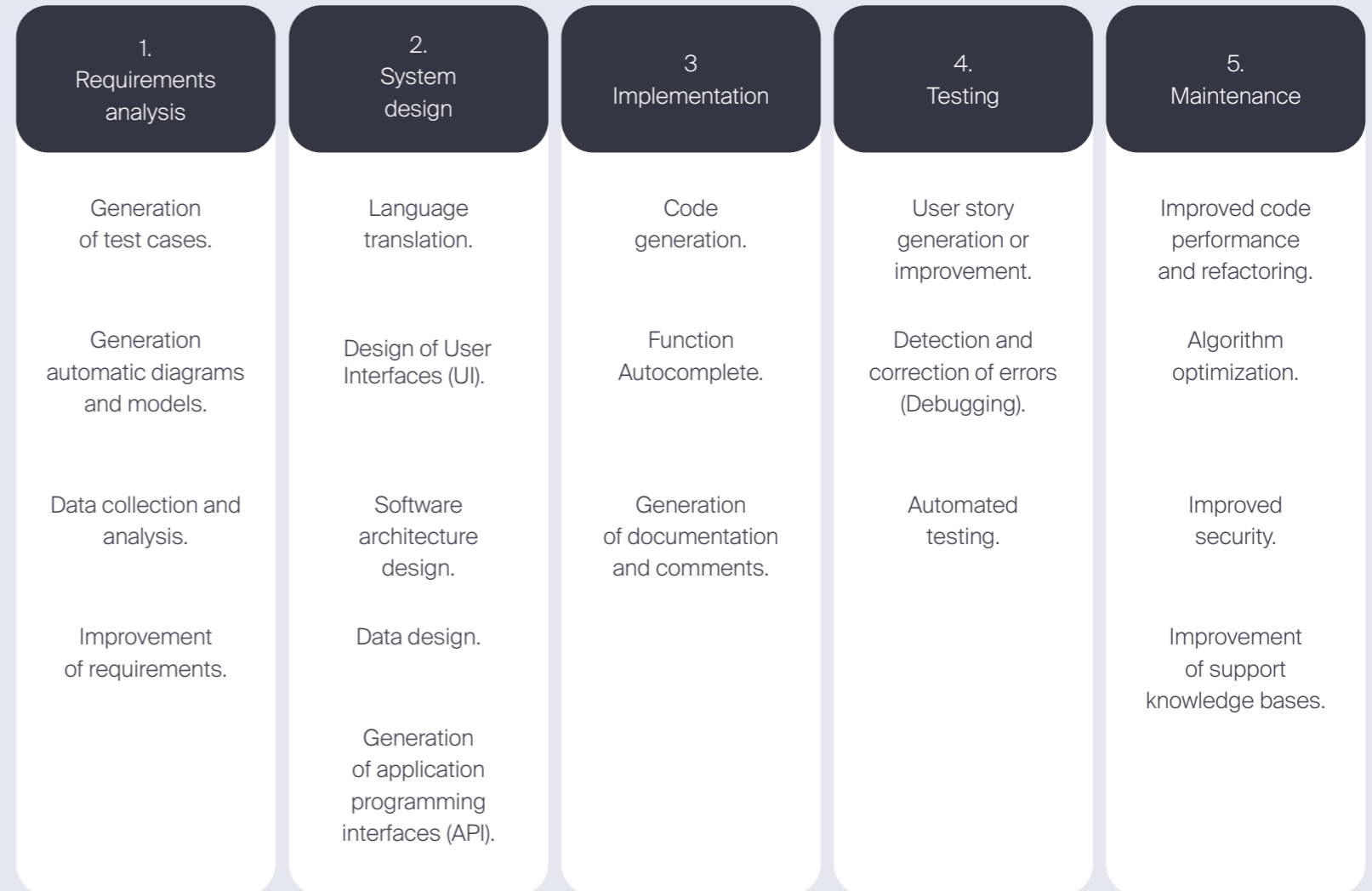
Organizations and their leaders must hurry to adapt and implement the potential of Generative AI to, on the one hand, increase the value of their products and services and, on the other, improve processes and even increase the creative capacity of their teams. The potential of technology to generate value in the economy and society is proportional to its capacity to unlock new scenarios and accelerate procedures and transformations. A trained AI model can correctly generate value across corporate functions and areas (human resources, operations, business development, etc.) and the rewards are up for grabs.

Generative AI & Software

Generative AI and its impact on software development

The rapid evolution of different intelligent models could bring us closer to achieving the so-called Artificial General Intelligence (AGI), acquiring capabilities until now attributable only to the human brain and allowing new levels of cognitive automation. In the field of software development, beyond the hype unleashed after the launch of ChatGPT, generative AI is positioning itself as **a promising innovation, thanks to its ability to improve the productivity and efficiency of developers.**

Use cases of generative AI for the different stages of software development



The ability to automate routine tasks provided by Generative AI allows professionals to focus on more strategic activities, freeing up time and improving efficiency in three main areas of work:

1. Automatic generation of parts base code.

2. Collaboration in writing code, writing down suggestions and noticing errors.

3. Review and control of code quality.

The direct impact of Generative AI on Software Engineering productivity could range between 15% and 45%, depending on each project and the technology used. According to an analysis by Microsoft's GitHub Copilot, **in 2022 developers who used this tool completed tasks 29.73% faster; in 2023, this figure has increased to 88%.**

Technology companies are already driving the following use cases through Generative AI:



Planning: Helps analyze and catalog large volumes of data.



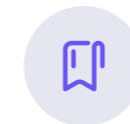
System Design: Allows creation of multiple architecture designs and to test their potential.



Coding: assistance in tasks coding and analysis.



Testing: improving testing functional and performance.



Maintenance: Provides performance data to diagnose problems and predict areas for improvement.



Software Engineering: Enables dual programming, augmented coding, and trains LLMs to develop code-generating applications.

Typologies and models of Generative AI for developers

With more than 100 million users, **ChatGPT is just one example of how generative AI is transforming code writing.** The conversational model has surprised by its ability to create basic lines of code from prompts and analyze patterns within a code, generating new lines optimized for readability and error-free execution. It is also capable of testing, identifying potential bugs and helping developers maintain code quality.

Among the great diversity of Generative AI models and systems, with application in different areas of software development, We find **specialized solutions** in the following categories:

- **Conversational platforms:** ChatGPT, Bard or Perplexity AI.
- **Autoprompting (autonomous models):** Auto GPT, Baby-AGI, Pinecone or JARVIS
- **Code generation:** Copilot, Codey, Replit Ghostwriter, Tabnine or Mutable.AI.
- **Text to SQL conversion:** AI Query, AI 2sql, Seek, CodeSquire or AirOps.
- **Construction of web apps:** Debuild, Enzyme or Durable.
- **Generation of documentation and comments / Refactoring:** Mintlify, Stenography, Figstack, What the Diff or Metabob.
- **Debugging and bug fixes / test cases:** Adrenaline or Sofysense.
- **Image generation:** DALL-E, Midjourney, Craiyon, Stable Diffusion or OpenArt.
- **Generation of UI designs/interfaces:** Diagram, Vizcom or Uizard.



The Generative AI Application Ecosystem

Text

Marketing

copy.ai Jasper Writesonic Ponzu frase
 copysmith MidJourney Moonbeam Bertha.ai anyword
 Hypotenuse AI Clickable letterdrop Simplified Peppertype.ai
 Omneky CONTENTA

Knowledge

glean
 mem
 YOU

General Writing

Rytr wordtune Subtxt
 LEX sudo write LAIKA
 NovelAI WRITER COMPOSE AI
 OTHERSIDE AI

AI assistants

Andi
 Quickchat

Sales

LAVENDER
 Smartwriter.ai
 Twain
 Outplay
 Reach
 regie.ai
 Creatext

Support (Chat/email)

Cohere
 KAIZAN™
 Typewise
 CRESTA
 XOKind

Personalized videos

character.ai
 AI DUNGEON
 KEYS

MODELS: OPENAI GPT-3 DEEPMIND GOPHER FACEBOOK OPT HUGGING FACE BLOOM COHERE ANTHROPIC AI2 GPT-NEOX GPT-J AI21 ALIBABA YANDEX ETC.

Image

Image generation

MidJourney OpenArt
 Craiyon Playground
 WEMBO PhotoRoom
 ROSEBUD AI alpaca
 Lexica Nyx + gallery
 mage.space artbreeder
 KREA

Costumer/ social

MidJourney

Media/ advertising

THE CULTURE DAO

Design

Diagram uizard
 VIZCOM IIA maket
 Poly Aragon
 INTERIOR AI CALA

OPENAI STABLE DIFFUSION DALL-E 2 CRAIYON

Code

Code generation

GitHub Copilot tabnine
 replit GhostWriter
 mutable.ai

Text to SQL

AI 2sql™
 seek

Web app builders

Debuild
 Enzyme
 durable

Documentation

Mintlify
 Stenography
 Other
 excel/formulabot

OPENAI GPT-3 TABNINE CODEGEEEX

Speech

Voice Synthesis

RESEMBLE.AI broadn
 WELLSAID coqui
 podcast.ai descript
 FLIKI Listnr
 REPLICIA VOICEMOD

OPENAI

Other

Music

splash MuseScore
 Aiva Technologies boomy
 Endel SONIFY
 Harmonai by stability.ai SARCASIS

Gaming

AI DUNGEON

RPA

Adept
 māyā

AI characters/ Avatars

character.ai
 inworld
 OASIS

Biology/Chemistry

Cradle

Vertical Apps

Harvey.

OPENAI JUKEBOX

Video

Personalized videos

tavus synthesis Hour One Rephrase.ai
 Colossyan Mario

Editing / Generation

runway FLIKI
 Diverse OpusClip

MICROSOFT X-CLIP META MAKE-A-VIDEO



GPT-4

The OpenAI model has meant a true revolution in the field of AI Generative and is getting most of the attention. Beyond the adoption of the ChatGPT conversational tool, and the large investment received by the technology giant Microsoft for its integration into the Bing browser, another reason that sparked this level of interest is the rapid and continuous evolution that the model has followed in just a few years. This model, with more than 175 billion parameters, has been trained with a very high volume of text to learn and perform language-related tasks, so its algorithms can understand your interlocutor accurately and respond coherently.

The latest of them, GPT-4, launched in mid-March 2023, incorporates a series of features that make the model even more creative and collaborative than its predecessor, GPT-3. This new version is trained based on a much greater volume of knowledge and allows developers to solve more complex problems.

Although previous versions offered grouped responses, **GPT-4 is more iterative and breaks down solutions to a query step by step.** This makes the tool more didactic and understandable, while allowing for a greater level of depth in the answers. Moreover, GTP-4 latest multimodal version can talk, understand spoken languages, and see and generate images (this last when integrated with Dall-E 3).

When it comes to generating code, GPT-4 has not only improved the complexity but also the level of understanding of the prompts and the precision of the responses, enabling the solutions provided to eliminate noise and to be more efficient than in the previous version. The GPT-3 model autogenerated complements to the answers offered without the user requesting them and often altered the final result. In GPT-4 the model has been refined to avoid this type of unnecessary supplementary information.

The new version improves precision when choosing the content of the responses and selects only the necessary data, unlike the previous version that responded to all possible data without previously filtering it.

Finally, the reasoning capacity of the model has also increased and it can now vary the answers depending on the conversation it has with the end user, being able to identify the feeling or intention underlying the question and vary the answer accordingly.

These new features allow users developers communicate in a way more fluent with AI to ask questions or make requests relating to the writing of code bringing us even closer to what a human interaction would be (the long-awaited Artificial General Intelligence or AGI).

Differences between GPT-4 and GPT-3/3.5

	GPT-3 / 3.5	GPT 4
Languages	40	40+24 improved
Contextualization	Limited	Interprets broader and more complex contexts. Generate more accurate and relevant responses
Formats	Texts	Multimodal: Text, web-browsing, image input, plugins, image generation, audio and voice
Data date	Trained with data until 2021	Trained with data until 2021
Emotions	Limited interpretation	Able to detect emotions with a higher level of precision
Maximum tokens	2.049 (GPT-3) / 4.096 (GPT-3.5)	8.192 / 32.768 (GPT-4-32K)
Parameters (millions)	1.600	175.000



GitHub Copilot

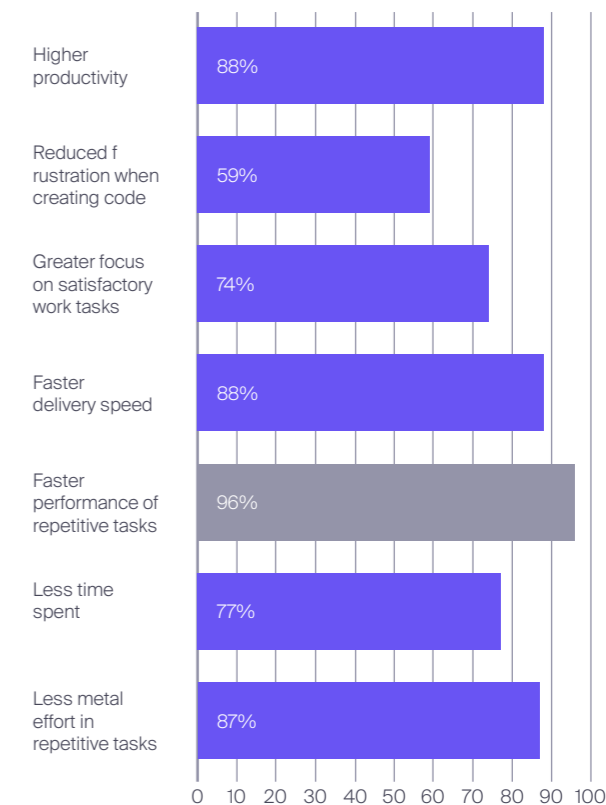
Copilot X (GitHub)

GPT's integration capabilities also offer a whole range of opportunities. One of the most representative examples is the integration with the popular project hosting solution GitHub.

GitHub Copilot **was one of the pioneering tools that raised the possibility of programming based on AI.** In just two years since its launch, Copilot has become a very popular service among developers who use the tool to carry out repetitive tasks much faster and easier.

From GitHub, they assure that in a few months after its launch in October 2021, Copilot was already generating 30% of the new codes produced on its platform, rising shortly after to 40% and with the forecast that it will reach 80% in five years. As for the benefits obtained by programmers, GitHub studies reveal that 88% of users claim to be more productive using the tool, 96% that they can perform repetitive tasks more quickly and 87% that they can do it with less mental effort.

Benefits perceived by GitHub Copilot users



Source: Softtek, based on data from GitHub

The arrival of GPT-4 has occurred in parallel with the launch of Copilot X:

- **Integrates into various editors code and IDEs** making workflows simpler and more comfortable.
- **Integrated chat:** its user experience is the same as that offered by ChatGPT, but integrated within the editor or development environment that the programmer is using, such as Visual Studio or VS Code, being able to interact with AI conversationally within these same tools.
- **Voice commands:** interprets requests through voice, which even allows programming without using the keyboard..
- **Generate descriptions of the pull requests:** This function allows you to manage repositories more quickly and easily, as well as add tags to requests quickly.
- **GitHub Copilot for Docs:** incorporates a new experimental tool specifically focused on offering answers related to the documentation of projects also through a conversational experience and chatbot interface. For now, it works in React, Azure Docs, and MDN.
- **CLI:** is also integrated into the command line interface which allows you to execute the commands provided by the AI directly from a terminal.
- **Also integrates with other tools** such as Excel, allowing users to developers increase speed workflows and create more complex functions in the processor.





AutoGPT (Autoprompting)

One of the most disruptive advances that, in all likelihood, will define the future evolution of generative AI is autoprompting. The starting signal for this trend has been marked by the launch of **AutoGPT**, an experimental Open Source application developed in Python, whose code is hosted in the GitHub repository, and which is designed so that, instead of the user, AutoGPT itself makes requests to the AI in the form of prompts: the model itself feeds itself based solely on the initial request.

This application is open to use, although, at the moment, it is only available to paying subscribers in the **ChatGPT Plus option**. And although it supports the GPT 3.5 API, it was designed to be used with a GPT-4 API.

Another great advantage of this new system is that it has the capacity to access the Internet and collect data with expanded memory both in the short and long term, thus surpassing a limitation of current systems such as ChatGPT whose training is restricted to data prior to 2021.

AutoGPT also integrates the **11 Labs** system, which generates a synthesized voice from the text so that the user can speak directly with the AI through **voice commands**.

In response to this launch, other new similar applications have emerged, such as **Baby AGI, Pinecone or JARVIS**, which add to the autoprompting trend, with the aim of reaching a next level of automation in Generative AI, letting their own models generate the requests to complete tasks, plans or goals.

Autoprompting systems act as a controller, combining other language models and various tools so that they determine the best path to respond to the query made by the user in their initial entry.

These new autonomous models are not exempt from an important risk: the possibility of making inappropriate decisions, so for now they can only be used as a suggestion or recommendation for human professionals.



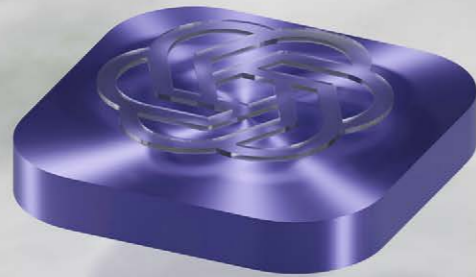
Bard (Google)

Google Bard is an artificial intelligence chatbot developed by Google based on the **LaMDA family**, a powerful experimental language model designed specifically for dialogue applications. With this proposal, Google hoped to compete with GPT and generate mass adoption of new features, such as **direct access to Google to update information in real time**. However, given the meteoric rise of GPT-4 Google has ramped up efforts and is expected to release a groundbreaking model codenamed Gemini.

The competitive advantage it presents over AI chat systems like ChatGPT is that **Bard is constantly connected to the Internet through a button they have called “Google it”, which redirects the user to the search engine, which provides constantly updated information**. Bard is being integrated as a search function within Google, so it will not be independent of the search engine. What is intended is for users to continue entering Google to give free rein to their creativity and make requests to Bard. However, this advantage will last for a short time as Microsoft has already launched Bing with GPT-

4 capabilities, and OpenAI Plus members can already enable Bing's browsing capabilities.

This Bard functionality allows developers to check the latest updates the languages they are generating code based on and ensure they are applying the latest standards. However, Google has warned that, although Bard is capable of accessing updated information, this does not mean that it cannot share wrong or biased information.



Azure OpenAI (Microsoft)

Language models have become an essential platform for developers to innovate and apply AI to solve problems. Azure OpenAI Service **provides REST API access to powerful OpenAI language models**, including the GPT-3, Codex, and Embeddings model series, plus the new GPT-4 and ChatGPT model series. These models are programmed for content generation, summarization, semantic search or translation from natural language to code.

Azure OpenAI Service provides enterprises and developers with high-performance AI models at scale. This is the same production service that Microsoft uses to power its own products, including GitHub Copilot, Power BI, and Microsoft Designer. Along with Google, they are the only ones offering a global public cloud offering AI supercomputers with massive scale-up and scale-out capabilities. Its architectural design combines GPU and networking solutions, delivering the best performance and scalability for the most compute-intensive AI training and inference workloads.



LLaMA (Meta):

LLaMA, Large Language Model Meta AI, is the artificial intelligence system created by Meta, the company specialized in social technology led by Mark Zuckerberg made up of Facebook, WhatsApp and Instagram.

It is an AI language model that seeks to serve as a competitor to ChatGPT, initially focused on scientific community. It relies on sequences of words as inputs, predicting the next one to generate the text recursively. This is a basic model designed to be versatile.

The differentiating element with respect to other solutions on the market is that it can be used in more different use cases, instead of focusing on specific tasks such as the current ones.



Perplexity AI

One less Generative AI solution well-known, but which has already positioned itself as a direct competitor to ChatGPT and Bard. It is a **free conversational search engine model connected to the Internet without time limitation and designed especially for mobile use**. At the operating level, its operation is similar to that of ChatGPT and allows the user to ask the AI any type of question.

As with Bard, the main advantage of this system over ChatGPT is its greater updateability, since its knowledge is not limited to 2021. However, another extra differentiation is that they have developed an application, available for Android phones, with which they seek to position themselves as a leading AI within the mobile channel.



DALL-E

Another OpenAI initiative is DALL-E, a generative AI-based system that specializes in creating **images from textual descriptions**. In its most recent version, DALL-E 3, it allows users to create new and completely original images from the guidelines specified in the text prompts.

The **neural network** that underlies this system is also based on GPT and is the evolution of a project that Open AI initially called **Image GPT**. Developers can use an **OpenAI API** to integrate the DALL-E 3 imager directly into the applications or programs they are building, although use of this API is still subject to limitations, authentication requirements, and various associated fees.



Midjourney

A competitor to DALL-E in the automatic imaging space is Midjourney, which also represents an alternative to other similar systems like **Stable Diffusion, DALL-E Mini or Craiyon**. Midjourney is an independent laboratory, based in the United States, dedicated to research in Generative AI and that bears the same name that they have given to their **“text to image”** model, whose latest version Midjourney 5.1 is still in beta phase.

Midjourney's AI provides **extra realism** to the images generated by DALL-E and is capable of producing images of greater size and quality (up to 1,792 x 1,024 pixels). However, for now Midjourney is **only accessible through a bot on the Discord channel** of the project (where it has already achieved more than 13 million registered users). The integration possibilities are therefore even more limited than those of its competitor DALL-E 3.



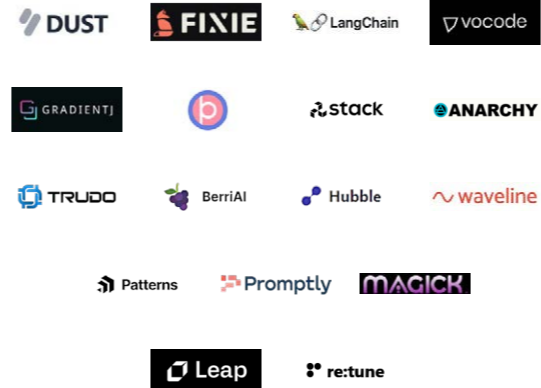
BedRock (Amazon):

Bedrock is a fully managed service that makes FMs from top AI startups and Amazon available through an API. **Allows you to privately customize FMs with your own data and easily integrate and deploy them into your applications using the AWS tools and capabilities** you're familiar with (including integrations with Amazon SageMaker ML features such as Experiments for testing different models and Pipelines for manage your FMs at scale) without having to manage any infrastructure.

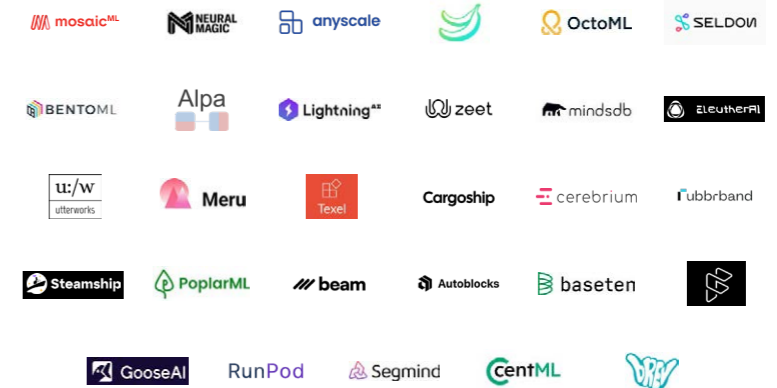
Amazon proposes a complete platform with which developers and companies can launch new solutions using its own infrastructure. The goal is to become an intermediary for LLM models, chatbots and startup APIs. The platform is designed to automate the development process and improve application results, while allowing hyper-personalization with a high level of AI.

In parallel to these Big Tech initiatives, a new ecosystem of Open Source solutions designed for specific tasks within software development is emerging: orchestration, deployment and training, integration and observability.

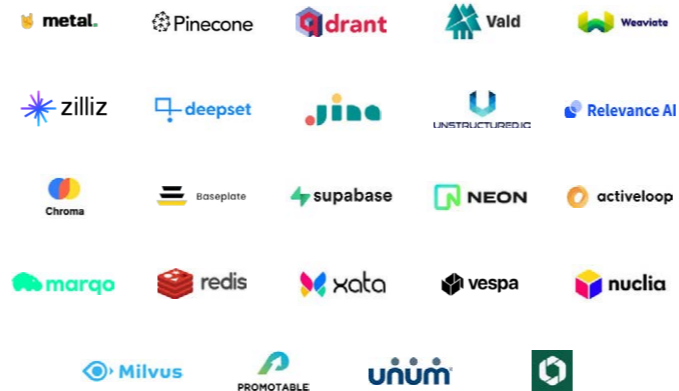
Orchestration



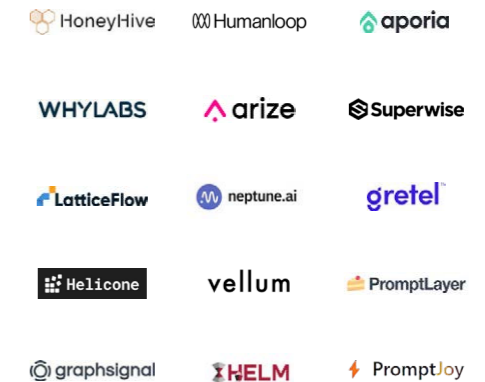
Deployment, Scalability, & Pre-Training



Context & Embeddings



QA & Observability

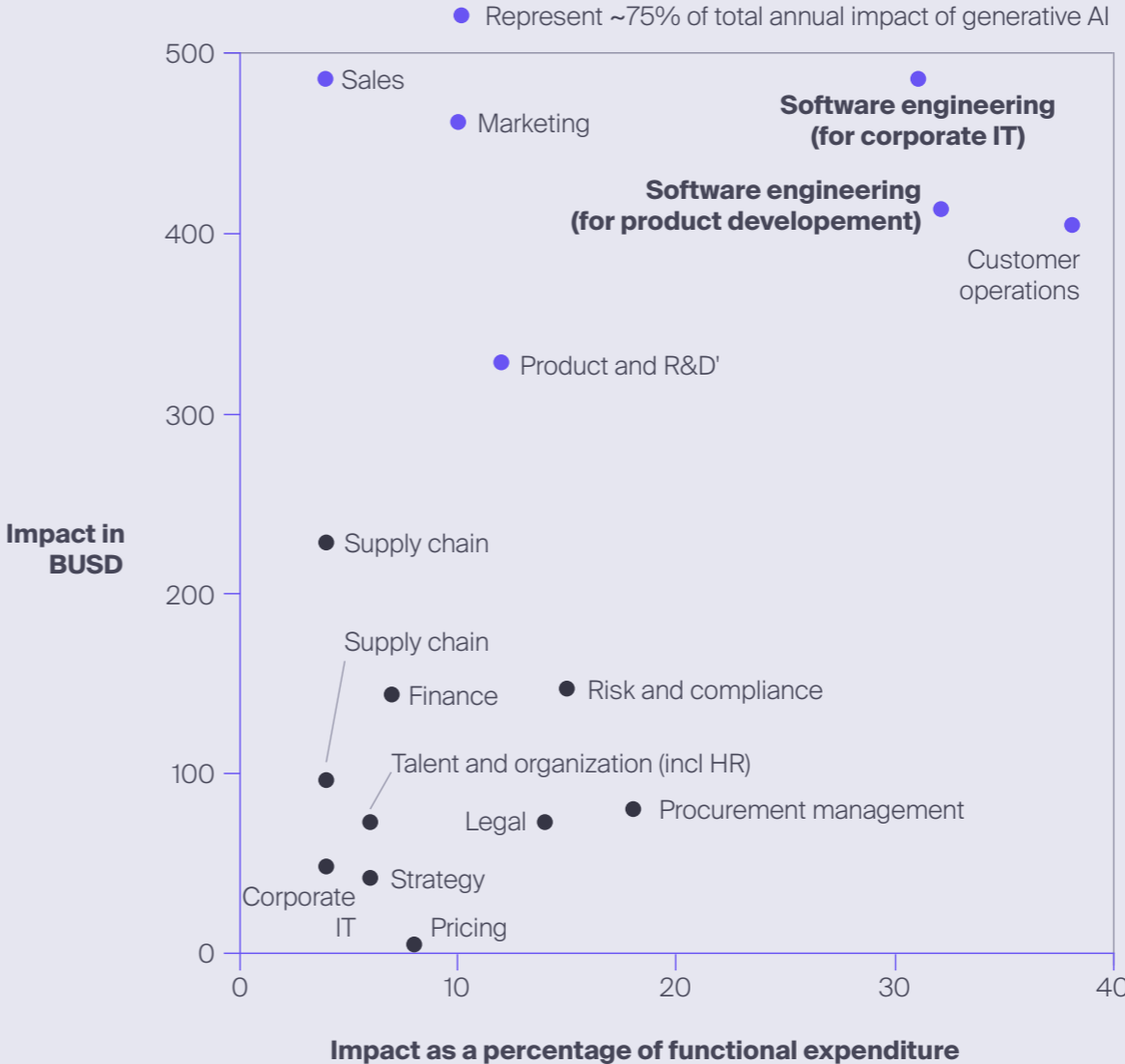


Adoption of Generative AI in software development

Deep learning has driven many of the recent advances in AI, but the basic models that drive Generative AI applications are a radical evolution within Deep Learning, since **they can process extremely large and varied sets of unstructured data and perform more than one task.**

In terms of economic impact, software engineering is estimated to lead most other corporate functions:

Areas of greatest impact



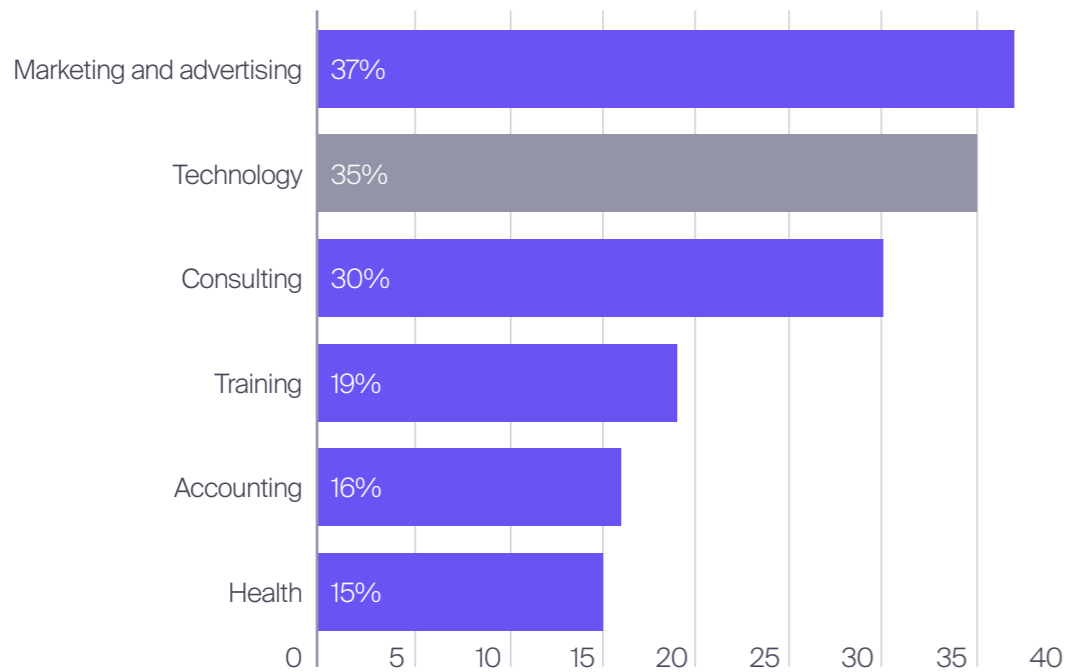
Source: McKinsey

In the last year, adoption has grown strongly among technology companies. Statistics say that, **in the United States, 35% of companies in the technology sector are already applying models of Generative AI to carry**

out daily work processes, which places it as the second sector of activity with the highest adoption rate, only behind the marketing and advertising industry:

Adoption rate of Generative AI in the United States, by industry

% of companies that use it in their work processes



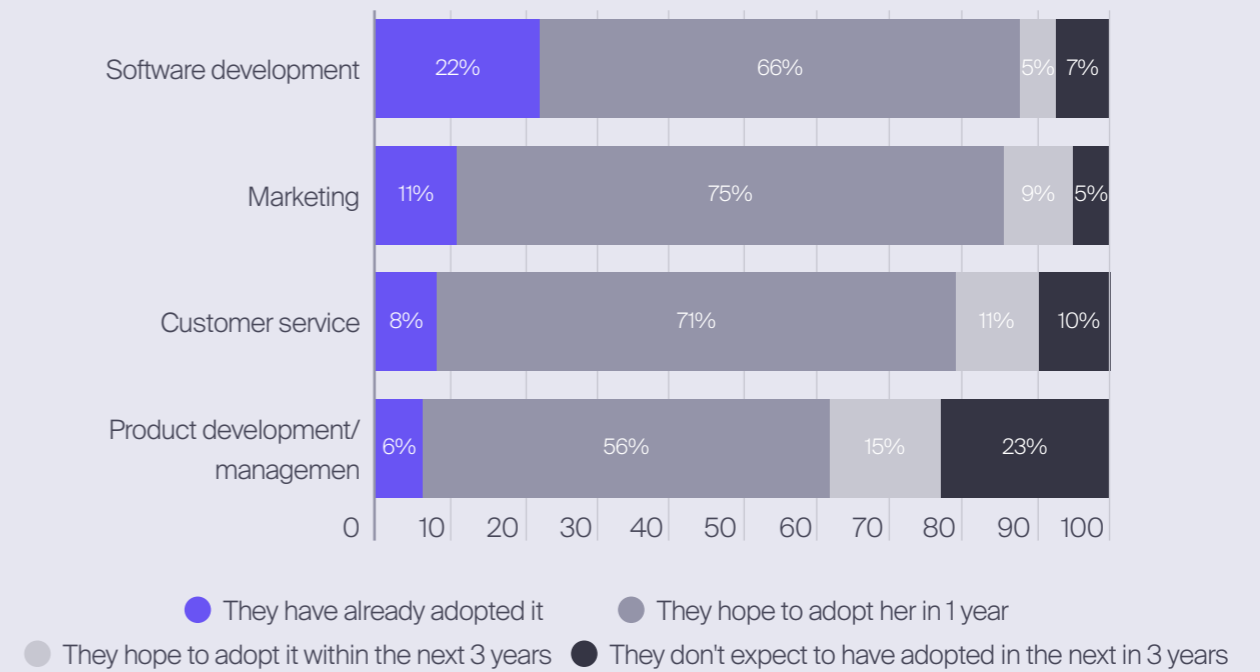
Source: Softtek based on data from Fishbowl

It is estimated that **22% of technology companies already use Generative AI to boost their software development efforts and 88% plan to do so within a year.**

Looking ahead to next year, it is also expected that its adoption will also grow in the rest of the main business areas:

Adoption of Generative AI in the United States technology sector by business area

% of technology companies*

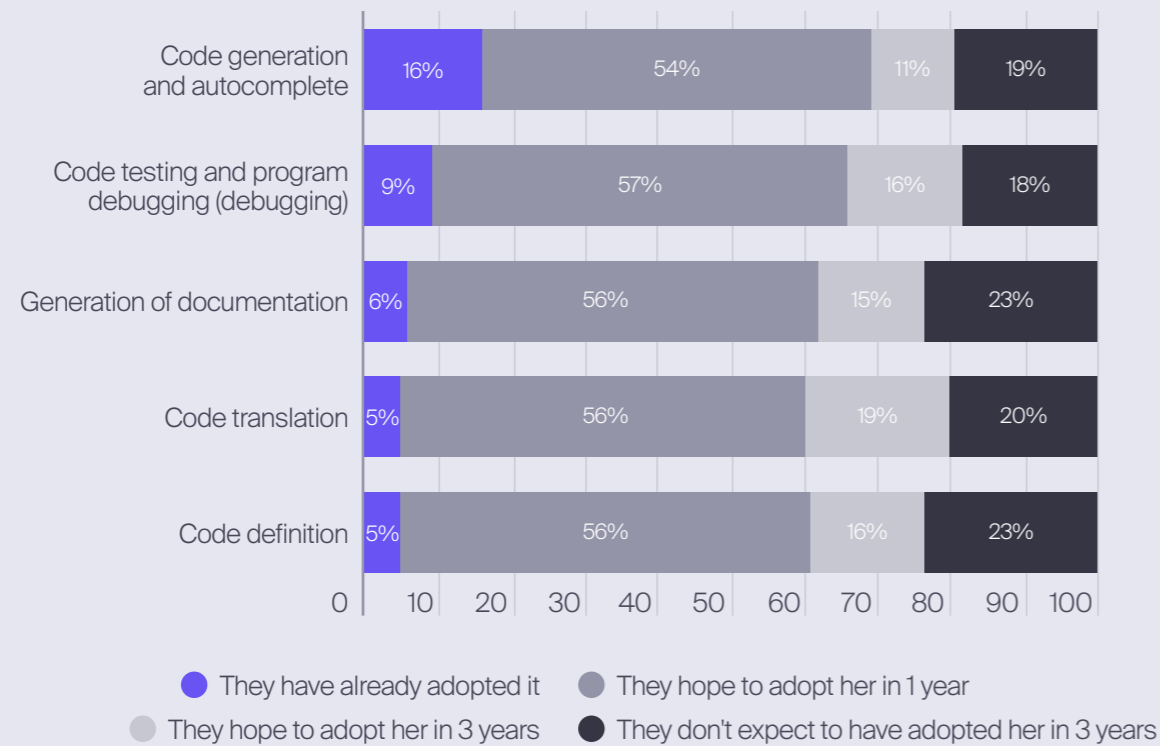


* survey based on a sample of 292 senior executives from technology companies. Source: Softtek based on data from Altman Solon

Among the use cases, the most widespread so far are code generation and the autocomplete function, followed by testing and documentation generation.

Adoption of generative AI in software development, according to use case

% of technology companies in the United States*



* survey based on a sample of 292 senior executives from technology companies.

There is no doubt, therefore, that Generative AI in software development is here to stay, and that its disruptive impact will only increase in the years to come.

Although we are still in a very early adoption phase, and there is a long way to go in terms of regulation and experimentation, **as the technology continues to advance, it is essential that companies, regardless of their sector, understand and fully realize its potential.**

Those organizations that pioneer this technology, and reap its benefits, will be best positioned to meet market challenges and lead the next era of innovation in software development.



Use cases

Analysis of requirements

Generation of test cases

Generative AI can be used to **automatically analyze software requirements** and understand the functionality and expected behavior of the system. Then, it allows you to automatically generate **test cases** covering different scenarios. This process includes the review of requirements documents, diagrams, specifications and any other relevant information. In this field, AI also serves to define the criteria that will guide the generation of test cases, including:

- The different scenarios to cover.
- Identification of borderline cases.
- Prioritizing certain features
- Any specific considerations.

Another possibility is to ask the AI to evaluate the quality of the selected test cases, to provide suggestions and recommendations regarding the quality of the coverage terms, their relevance or their possible effectiveness. To do this, developers can ask the AI to automatically apply predefined quality metrics and criteria so that it certifies that, indeed, the selected test cases meet the established objectives.

Finally, it saves developers from having to carry out all the refinement and tuning processes manually. This includes:

- Add specific cases.
- Elimination of redundancies.
- Modification of existing cases to improve the quality of the test suite.

Sofy.

*Sofysense combines **artificial intelligence and no-code automation** to help developers ensure software quality. Using the tool's generative chatbot, called Sofybot, developers in charge of testing can free up a large amount of time spent on research and test design, to perform other higher-value activities instead. Sofybot relies on OpenAI's language model, GPT, to generate automated responses to specific test queries posed by human developers. SofySense also offers a facility to convert manual tests into automated tests. This option allows you to generate information from the test activities requested on the platform. The tool is also capable of reading Confluence, and other documentation tools, and generating test cases.*



Automatic generation of diagrams and models

When creating diagrams, developers can give the AI text with manually set prerequisites and ask it to extract relevant information from them, including entities, relationships, actions, and properties.

Once said text is analyzed, **the AI applies a layer of natural language processing and machine learning to understand meaning and capture the underlying semantics.**

With the extracted information, it builds possible models of the system in the form of visual diagrams. This proposal includes:

- **Use case diagrams.**
- **Class diagrams.**
- **Flowcharts.**
- **Sequence diagrams.**
- **Block diagrams.**

The answer will always depend on the nature of the requirements indicated in the prompt and the type of model to be generated. Additionally, AI can generate the diagrams based on predefined templates, transformation rules, or algorithms that fit the specified style and needs.

Applying Generative AI in this use case **allows you to accelerate the requirements analysis process to start the implementation more quickly, which substantially improves time-to-market.**

It also facilitates communication between the different work areas of the development team by having a system that graphically represents an architecture and allows the beginning of the software construction to be validated impartially, as well as to more quickly understand and identify possible problems or inconsistencies in the requirements.



*DiaChat uses a natural language processing model that helps **generate flowcharts and architectures instantly** through requests written in text mode. The tool also allows manually edit of diagrams generated by the AI without having to leave the web, as well as exporting the graphs in different image formats.*

Data collection and analysis

During the **data collection stage** through multiple sources (documents, reports, user logs, and any other information that is relevant to the project) regardless of its format (text, audio, video and others), AI can help by preparing said information for analysis .

This feature allows you to avoid routine and low-value tasks that require a lot of work time, such as:

- The **normalization** of the text.
- Elimination of redundant information.
- Error correction.
- The transformation of data to a **suitable format** for processing.

Generative AI applies machine learning algorithms to **discover hidden patterns, correlations and relationships between data**, helping to identify common requirements, user preferences or needs that had not been previously considered.

Finally, **it generates reports, summaries, and visualizations** that synthesize the results of requirements data analysis, including tables, graphs, diagrams, or any other format that facilitates understanding and decision-making by developers and other interested parties.

This feature eliminates a huge amount of prior work needed to define requirements and greenlight implementation.

On the other hand, it helps developers gain a deeper understanding of user needs, identify usage patterns, predict future requirements, and make more informed decisions.

airOps

*This new **Cloud platform** works as an extension of Google Chrome, and can also be accessed through the Web App and API, facilitating multiple integration options. AirOps has a SQL assistant which allows developers easily automate tasks **database analysis**, writing queries and identifying and resolving bottlenecks. It also allows automatic document table schemas, as well as **Generation of Python** scripts and YAML dbt files.*

Improvement of requirements

In cases where the team has already completed a manual requirements identification process, before moving to implementation the AI can be asked to provide an assessment and suggest possible changes or corrections to be made. AI analyzes the existing requirements for problems, inconsistencies, ambiguities or deficiencies. To do this, it applies natural language processing and machine learning techniques in such a way that it **detects patterns or anomalies** that require improvements.

Based on the analysis performed, the AI generates proposals that include changes, additions or deletions of requirements, as well as suggestions to clarify, disambiguate or refine existing requirements. Generative AI therefore contributes to improving the quality, coherence and understanding of requirements and provides feedback with suggestions for improving performance and accuracy in future proposals.

System design

Translation of programming languages

A promising use is automated translation between different programming languages, as it allows for greater **interoperability** between technologies and platforms.

Generative AI models analyze the source code in a given programming language and automatically generate the equivalent code in another language, which facilitates the **adaptation** and **integration** of systems developed in **different environments**.

In addition, it improves efficiency in software development as models perform faster and more accurate conversions, saving developers time and effort by eliminating the need to do laborious and error-prone manual translations.

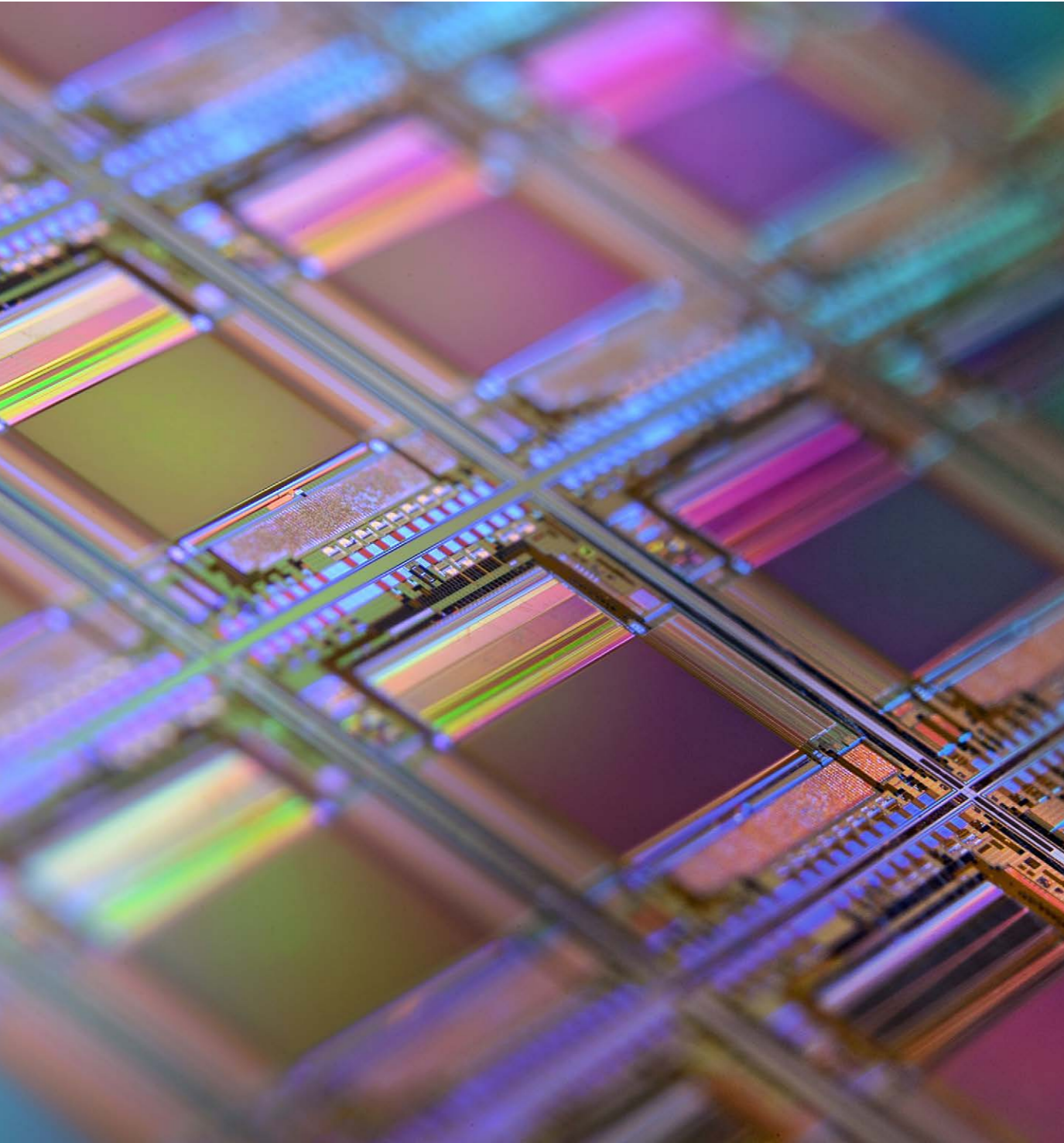
This practice thus reduces barriers to collaboration and reuse of software components and libraries, facilitating code sharing between teams using different programming languages and more fluid collaboration between different team members.

This emerging use case also focuses on **efficiently creating visual interfaces** for applications and programs and reduces the development time required by automating much of the interface design work. The AI models automatically analyze:

- Graphic **patterns**
- User **preferences**
- Specific **requirements** for each project

Through this analysis, they quickly generate proposals for adaptable interfaces, also facilitating iteration and experimentation during the interface design process. AI models thus allow multiple **visual options** to be obtained that meet established requirements, so that developers and designers can then explore different possible approaches.





In this aspect, Generative AI models are capable of adapting visual elements, such as colors, fonts and arrangement of the different elements and components of the interfaces, allowing improvement of the user experience and, ultimately, increasing satisfaction with the product final. This is an especially useful feature in large-scale development projects, where maintaining visual consistency can be a challenge.



Figstack is a web platform that helps developers understand code in any language, **automatically translate programming languages**, and automate documentation generation. This solution based in the cloud integrates with the most popular code editors, such as **VS Code**, allowing the developer to access its features while using other tools while working on the project. Its **no-code interface** makes the tool accessible to developers of all types of level and experience.

Diagram

Diagram is one of the **UI design solutions** that has implemented Generative AI features. One of its products, **Magician**, consists of a collection of AI-based tools that solve problems that until now had been difficult to address, such as the assignment automatic naming at layers, and adds new features that allow you to automatically generate images and create prototypes to quickly bring them to market.

Another product, **Genius**, uses an AI Generative model to offer developers an **intelligent assistant** experience that allows you to carry out autocompletion functions in an integrated way, within other design tools like **Figma**.

Design of software architectures

Creating more **efficient and scalable software architectures** is one of the biggest challenges for IT teams. development when addressing a new project, and Generative AI can be of great help to optimize and streamline this entire design process, as it allows developers to **receive proposals for possible architectures**, based on good practice models and a greater variety of program operating patterns. , such as functional and non-functional requirements, technical limitations, user preferences and pre-established objectives.

This is another emerging use case that is already beginning to be applied by development teams with the aim of accelerating the initial project design processes and ensuring more effective architectures.



*One of the tools that is already being evaluated as a possible solution for the **intelligent generation of software architecture diagrams** is MidJourney, in which a developer provides the tool with a text description of the desired software architecture, which includes detailed information about the front-end, microservices architecture, database components, etc.*

MidJourney then generates a diagram for said architecture that can then serve as a basis for development teams to refine and polish the idea.

*Solutions like this still have many limitations and cannot replace the work of visual representation and architectural design, but they are beginning to emerge as a possible **support tool**, to provide ideas, suggestions or recommendations.*



Database design

Database design is another of the critical tasks within this stage, since a well-designed database structure is essential in ensuring efficient storage, fast data retrieval, and proper data integrity.

Generative AI is starting to play an important role in helping developers generate database models that fit the specific needs of their applications. Automatically analyzes system requirements, available data, and established constraints to generate an optimal database design. This includes:

- Identification of **entities that are relevant**.
- The **relationships** between entities.
- The **definition of the tables**.
- The **definition of the fields and keys**.

In addition, it is also of great help to optimize the design of existing databases. The models are used to analyze database performance and utilization in real time, identify problem areas, and propose optimization solutions. This may include reorganizing tables, creating additional indexes, query optimization, and other techniques to improve database performance and efficiency.

These features offer development teams:

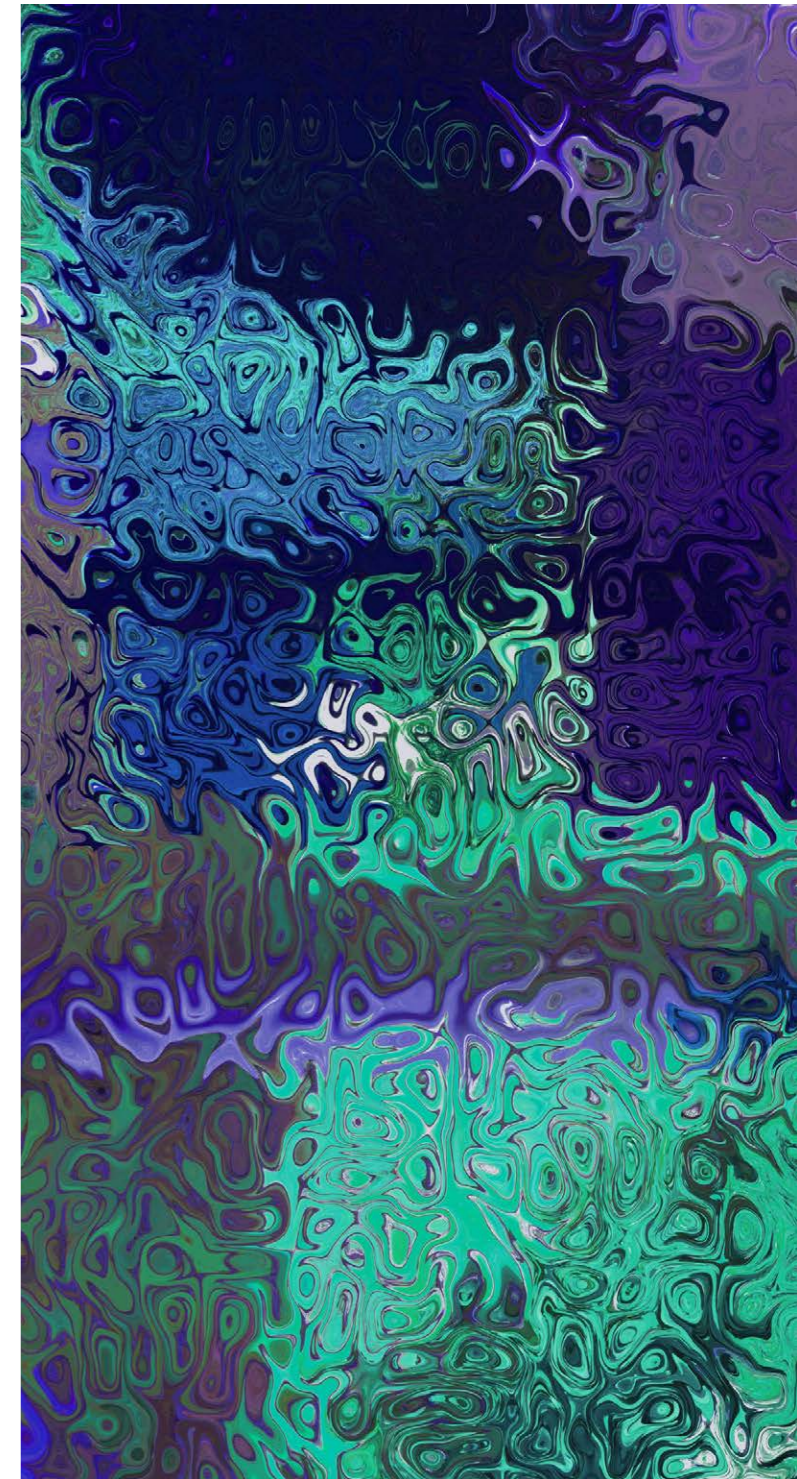
- A greater capacity to manage the complexity and scalability of projects.
- Reduce the probability of error.
- Improve the quality of designs.
- The possibility of considering a greater number of variables.
- Easier to take into account normalization and referential integrity restrictions.
- More precise identification of possible inconsistencies, redundancies or referential integrity problems in the design.
- A more structuring database consistency.

- Greater compliance with project standards and requirements.

Query.ai

*This NLP tool allows developers **to interact with their databases through prompts**. The solution automatically translates prompts into **SQL queries**, being able to understand and generate queries with a high level of complexity, although sometimes requiring manual refinement. Its user interface is **no-code** and supports multiple types of data and schemas, and is also capable of generating explanations and comments within them.*

*It is compatible with **PostgreSQL, MySQL, MariaDB, and SQL Server**.*





API generation

As the adoption of open business models has grown, the creation of APIs has become an increasingly important element in most new program designs, as they allow communication and interaction between different components and services of an application. **Automated API generation** can help overcome some common challenges in this process, such as design complexity, managing different versions, or compatibility between multiple platforms.

Generative AI serves in this case to analyze the context of the system, the functional and non-functional requirements, as well as the available data, and **create a possible structure and definition of the API**. This includes identifying appropriate resources, methods, parameters, and responses, as well as generating associated documentation and testing.

Additionally, the AI can be asked to automatically generate **API adaptation** logic for different environments such as web, mobile or IoT networks.

This allows developers to create compatible and optimized programming interfaces for each platform without having to manually design different versions of the API.

It enables developers to quickly obtain a basic structure of the API, which saves time and effort in the initial design phase and helps maintain consistency in the design, avoiding possible errors and inconsistencies in the programming interfaces.



*PaLM API is a tool designed by Google to help developers to build cutting-edge applications, providing them with APIs and tools that allow them to integrate Generative AI into their projects and allowing them **to experiment with Google LLMs**. Along with the PaLM API, Google has also developed a tool called **MakerSuite** that allows developers to prototype more quickly and easily.*

Implementation

Code generation

Source code generation is probably **the most widespread use case so far**. Starting from existing examples and patterns, Generative AI is capable of creating functional code proposals that meet the requirements specified by the user.

This streamlines the development process by providing an initial base of code that can then be refined and customized by developers based on their particular requirements.

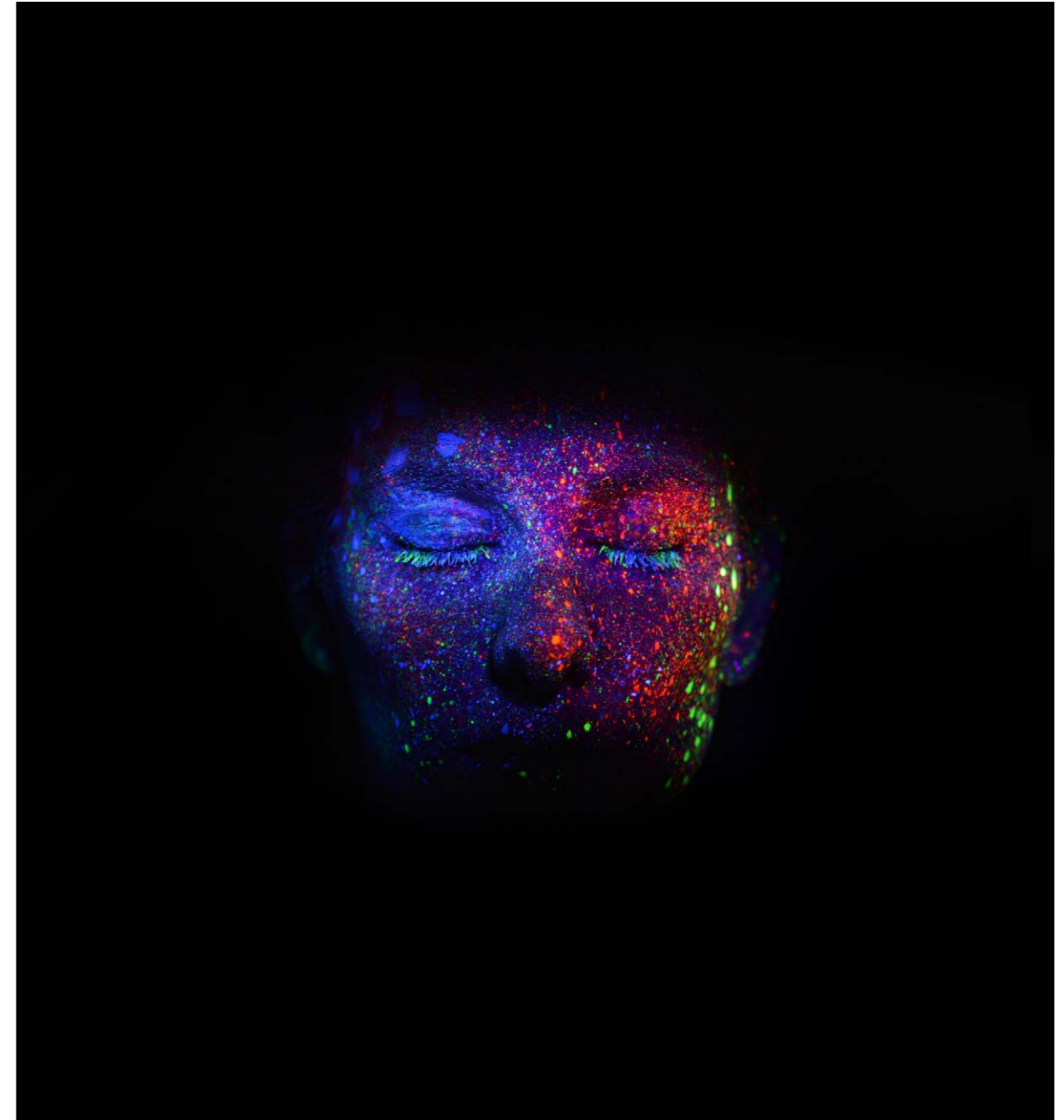
When writing code, Generative AI uses advanced machine learning techniques, such as neural networks and sequence generation algorithms, to create functional code from contextual information and programming rules.

AI models are trained using huge data sets of existing source code, allowing them greater than human ability to identify patterns, structures and dependencies. On the other

hand, by receiving project-specific information, such as business requirements or business rules, models can generate code with a high level of relevance and customization.



*One of the most popular solutions in the field of code generation is Replit, a Machine Learning- based tool designed to complement the developer's knowledge and help reduce the time spent searching or studying examples on websites such as **Stack Overflow**. It is also very useful for quick refactoring and translating code into different programming languages. The solution automatically saves the generated code and stores it in the cloud.*



Migration processes

Within this use case, one of the most common and valuable practices is the generation of code for migration processes. For example, when you need **to migrate a database** or system from one state to another, or **upgrade to a new version or switch platforms**.

Migration is a complex process that involves making changes to existing data and structure to adapt the code to a new environment or new requirements. The generation of migration code with Generative AI seeks to simplify and accelerate this entire process by **automating the creation of scripts** and the procedures necessary to carry out the change. Generative AI analyzes the structure and source data, along with established requirements and constraints, to automatically generate the corresponding migration code. This includes creating SQL statements, data transformation scripts, and any other code necessary to perform the migration as effectively and accurately as possible.

Autocomplete function

Code autocompletion is an essential tool to improve productivity and efficiency of developers. The introduction of Generative AI has led to this functionality to a whole new level by allowing the system to automatically generate suggested code based on context and established conditions.

Generative AI models can identify patterns, structures, and dependencies in code and automatically generate highly relevant and accurate code suggestions. This practice significantly reduces the time needed to write repetitive or commonly used code and speeds up development processes by allowing professionals to focus on more complex and strategic aspects of the project.

On the other hand, it promotes consistency and adherence to best coding practices, allowing for more consistent and readable code and reducing the probability of bugs occurring in the code.



*Focused on code autocompletion through deep learning algorithms in **Java, Python and C++**, among other languages, TabNine automatically indexes code submitted by the developer and **creates customized** suggestions based on learning the user's writing patterns. It is also used to generate **automatic tests** and detect or highlight syntax issues. It stands out for its **integration capabilities, is compatible with the main IDEs** and supports more than **20 languages and 15 different editors**, including VS Code, IntelliJ, Android Studio and Vim.*



This tool applies deep learning algorithms to provide suggestions and complete code blocks in real time. ACT-1 has been trained to integrate with web browsers, and currently works as a Chrome extension. From the browser, the tool observes browsing activities and carries out various automatic actions such as writing text and clicking or scrolling. It stands out for its multitasking functions since it allows performing tasks that require the simultaneous use of several programs.

Document and comment generation

Another of the use cases with the greatest adoption today is the automatic generation of documentation and comments with the aim of improving the quality and understanding of the developed code. This technique uses advanced machine learning models to **analyze source code and generate descriptive documentation and useful comments**.

Generative AI models can analyze the structure, logic, and patterns of code, and generate clear descriptions and comments that explain the functionality and purpose of different sections.

This makes it easier to understand both for the developers who create the code and for those who review or maintain it in the future. In this way, instead of having to manually write detailed documentation and comments for each part of the code, **Generative AI models create descriptive and explanatory texts automatically**, speeding up the development process and facilitating team collaboration.

Another notable benefit is the consistency in the documentation and comments. By following the predefined guidelines and standards in the prompts, Generative AI ensures that the generated documentation is following a coherent structure and that it provides uniformly relevant information.



*Mintlify is a backend tool specialized in **documenting code**. It not only allows you to create documentation automatically, from the code, but also compile multiple sources to keep the documentation constantly updated. It has two main **products**:*

- **Mintlify Connector:** a dashboard that allows you to connect sources and documentation with the project code.
- **Mintlify AI / Writer:** supports Generative AI through which new documentation is generated. It supports the main languages (Python, Java, C++, TypeScript and others) and is available in the most popular editors, including VSCode and IntelliJ, as well as in a web version. It has the ability to analyze the code and understand the context and purposes, then creating automatic descriptions in the form of docstrings.



Testing

Detection and correction of errors (Debugging)

Identifying and fixing bugs early in development is crucial as it prevents these issues from propagating and negatively affecting the functionality and stability of the software. Generative AI can be of great help in automating these tasks, increasing the precision and speed with which human teams can do them. This results in a higher quality final product and reduces the need for costly fixes later in the development cycle.

Generative AI is used to obtain **automatic error correction suggestions**. These models learn through historical examples of fixes previously made by human developers and apply that knowledge to deliver solutions. This not only speeds up the remediation process, but is also helpful for less experienced developers, who can benefit from AI-generated recommendations.

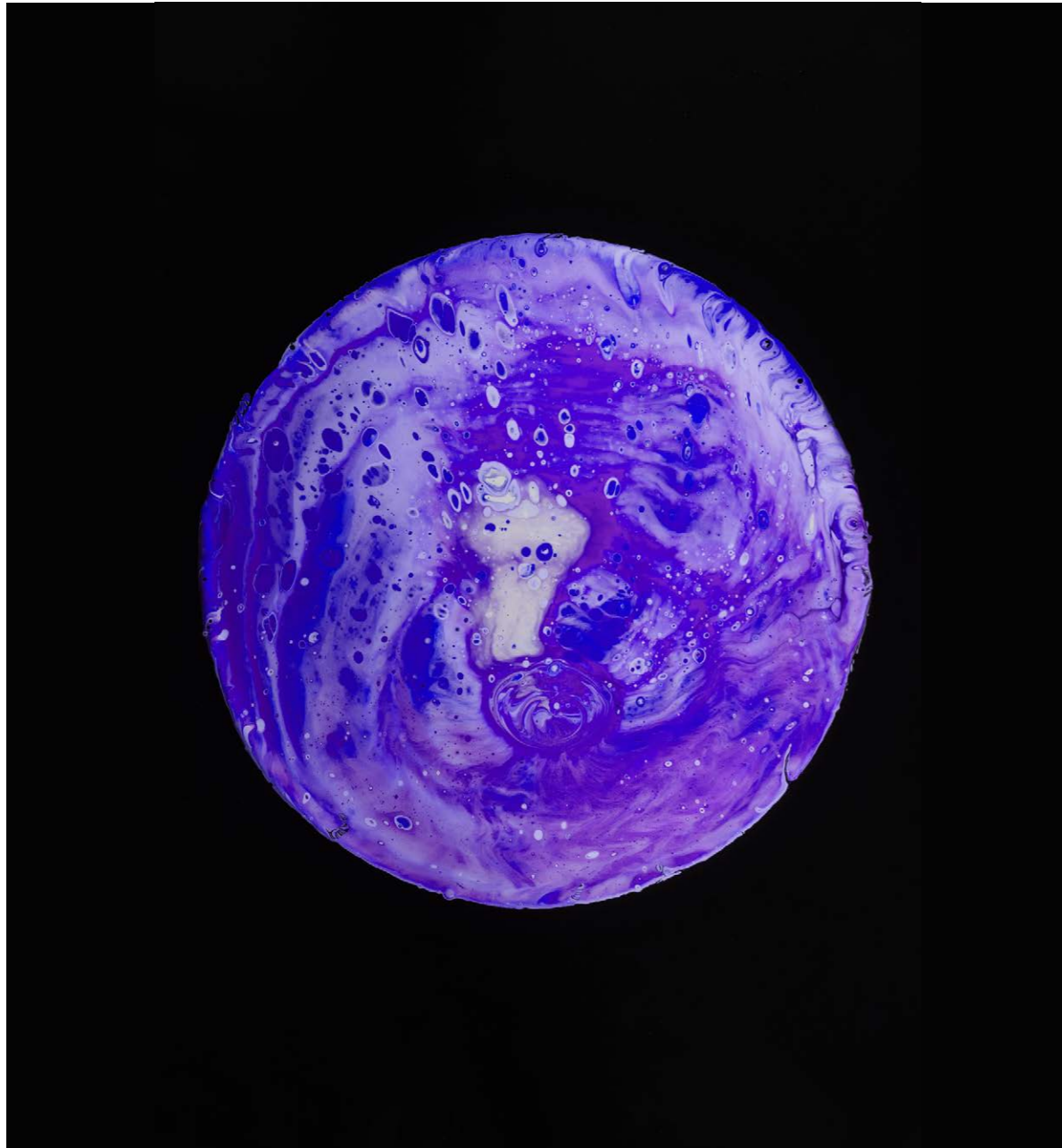
Another key benefit of using Generative AI in debugging is your ability to deal with complex problems and cases that are difficult to debug. AI models understand the relationships and dependencies between different parts of the code, allowing them **to detect subtle errors or problematic interdependencies** that may go unnoticed in a manual debugging approach.

One of the problems for which Generative AI is used most commonly is for resolution of conflicts of dependencies and **incompatibilities**. This is one of the most common challenges in software development when working on projects that involve multiple modules or libraries.

Generative AI benefits is in addressing this very common problem, analyzing the dependencies and restrictions established in the project and automatically generating possible solutions to resolve conflicts. The AI models determine the optimal combinations between the different versions and also directly perform the necessary modifications in the code or in the development environment configurations, greatly reducing the time developers have to spend solving these errors manually.

Adrenaline

*Adrenaline is a software that works as a **debugging assistant** and uses a combination of Machine Learning algorithms with programming analysis and statistical probability to help developers identify and **resolve bugs** much more quickly and efficiently. It automates the entire hypothesis generation process to determine what could be causing the bug. It also provides **real-time feedback** on the actions developers are taking in order to test and refine hypotheses on the fly. The system is integrated through the OpenAI API and uses the GPT model to explain errors in plain text.*



Automated testing

Automatic test generation is another booming feature. For this process, Generative AI uses advanced learning models that automatically generate test cases evaluating the functionality, security and robustness of the software.

The application of Generative AI in **testing** offers numerous benefits. It allows for exhaustive coverage of the different paths and scenarios possible in the software. AI models are capable of analyzing code, understanding its logic and structure, and generating tests that cover a wide range of situations, including extreme situations and cases that are unlikely but whose impact could be critical.

Additionally, automated test generation with AI helps **identify bugs** and **security vulnerabilities** that might go undetected in manual testing. The models can even simulate attacks and adverse situations, detecting possible weak points and proposing security improvements.

Generative AI thus facilitates more decision-making for development teams, informed regarding the quality and stability of the software.

What The Diff

*What The Diff can do pull requests and write a descriptive comments about the changes to be made, without the need to store the code. To obtain the diff, it uses the **GitHub API** on which it applies its Generative AI model, which is capable of understanding the context and offering information about what changes have been made and why. It has the ability to highlight the semantic differences that exist between each line of code, allowing quick identification of problems. You can also identify and ignore differences that are not relevant to code changes, such as whitespace.*

Code performance improvement and refactoring

Generative AI is not only useful during the process of creating a new project but also **to improve the performance of code generated** in previously completed projects. Developers can use it to:

- Identify and correct **bottlenecks**.
- Reduce **algorithmic complexity**.
- Improve **overall code efficiency**.

Generative AI is used to analyze existing code and system features, as well as performance data and relevant metrics. Through this analysis, the AI models can identify patterns and trends that indicate problem areas and opportunities for improvement. Generative AI can then propose alternative solutions and make **automatic changes to the code** to optimize its performance.

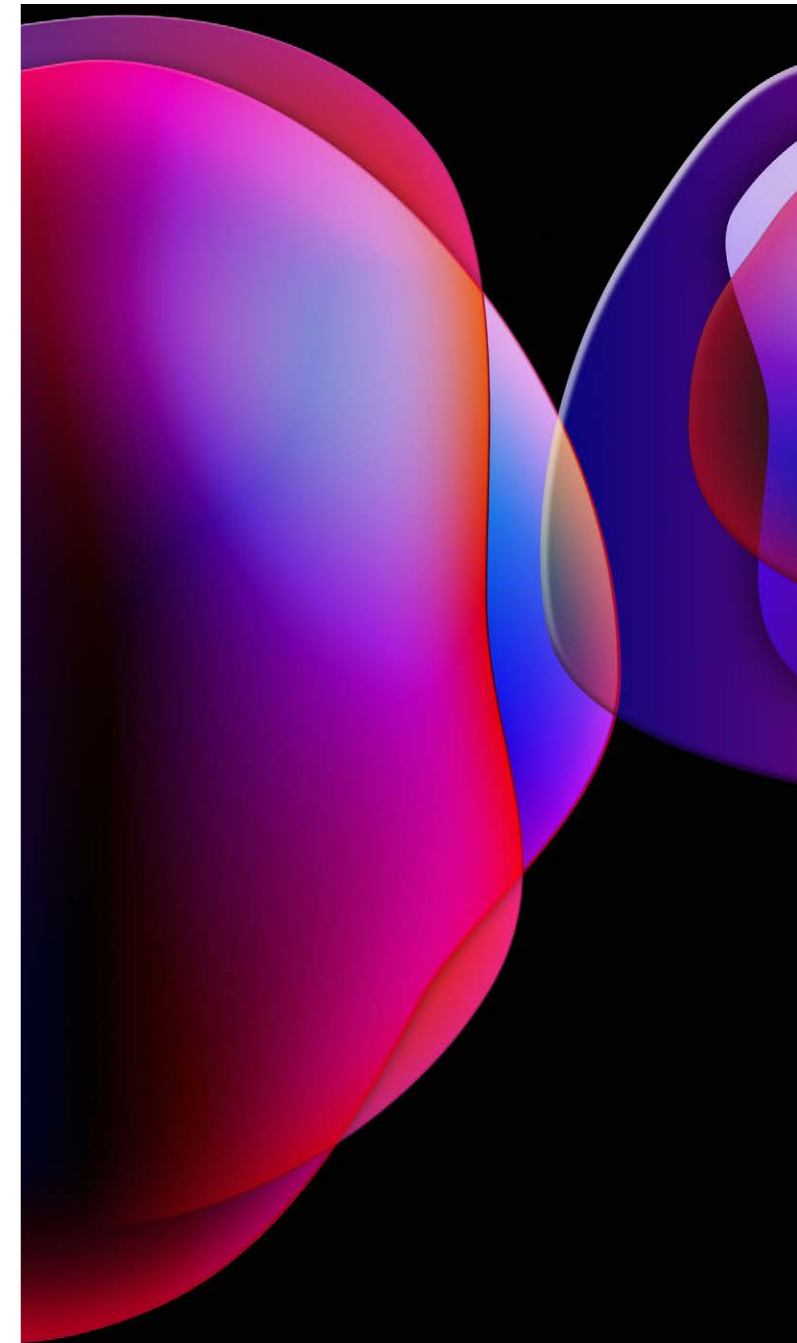
Within this use case, the application of Generative AI for **refactoring tasks is already widespread**, where its usefulness is being demonstrated in improving the readability, performance and maintenance of the code, without the need to change its observable behavior to users since it allows automating certain actions such as:

- **Rename variables and functions** to make its purpose more evident.
- **Break down large functions** into smaller, more specific functions.
- **Remove duplicate code**.
- **Simplify** conditional structures.
- Improve code **organization**.

This process gives teams greater ability to **quickly analyze large volumes of code** and indicators to **detect subtle inefficiencies** and patterns of poor performance that could be taken into consideration during a later manual review process.



*Metabob is a Generative AI **tool specialized in refactoring** that uses LLMs and its own Graph Neural Networks (GNNs) to detect problems, explain them and solve them. Metabob uses a mechanism of attention to understand semantic and relational markers, resulting in a more complete representation of each input. Your proprietary LLMs They also use a mechanism of attention and use billions of parameters to predict the token most likely to follow a given input.*



Maintenance

Algorithm optimization

One of the most revolutionary functions is the use of Generative AI to **optimize the creation of the algorithms** that will support the developed program or application. Generative AI can use advanced machine learning models to **generate highly efficient and personalized algorithms**. The generation of algorithms through AI opens a window to obtain significant benefits in performance, being able to greatly surpass traditional approaches by being able to analyze large volumes of data and learn from complex patterns. Generative AI models learn to balance different objectives, such as efficiency, precision and resource utilization, generating algorithms that offer optimal solutions in a greater variety of situations, ensuring that the software remains optimized even in dynamic environments and allowing multi-objective optimization, simultaneously addressing multiple goals or constraints. It also allows the exploration a greater variety of solutions and learn underlying patterns, generating more adaptable algorithms.



*AlphaTensor is a new software created by Deepmind (owned by Google), an evolution of AlphaZero (a software capable of programming mathematical algorithms that caused great excitement a few years ago by being unbeatable by humans in games like Go); which has proven to be capable of **creating faster and more efficient algorithmic formulas** than those that mathematicians had been using after decades of study. Deep learning techniques have been combined so that the machine can discover the rules of matrix multiplication on its own without human programmers having previously taught it any mathematical notions. Experiments like this are a representative sample of how far Generative AI can go and how, in the future, development teams could use this type of machines to generate algorithms instantly, through text commands written in natural language, achieving more efficient and precise results than those that can be created through the human brain.*

Improved security

Generative AI can play a key role in **proactively addressing potential security vulnerabilities and improving incident detection and response techniques**. The opportunity for improving software security is in the identification of vulnerabilities and the detection of attacks. AI models are capable of ingesting and analyzing large volumes of source code, audit logs, behavioral patterns, and security data to identify potential weaknesses and anomalies in the system.

Within these analyses, the models are capable of detecting vulnerabilities that can later lead to code injections, cross-site scripting (XSS) or denial of service (DoS) attacks, as well as discovering anomalous patterns that could indicate intrusion attempts.

Finally, Generative AI models are also capable of analyzing existing encryption algorithms and identifying possible weaknesses or suggesting improvements to build more robust algorithms.



*Veracode Fix is a GPT-based Machine Learning model, trained on proprietary data, specialized in reviewing **and correcting insecure code**, serving to detect vulnerabilities that could go unnoticed by human developers. Veracode Fix is capable of automatically generating secure code patches that developers can review and apply to fix security flaws, without the need to manually code a code.*

Conclusions

In the study, we have observed the **transformative potential of generative AI in the software industry**, as well as the implications and consequences that its adoption entails in innovation and competitiveness.

The use cases and examples presented demonstrate the emergence of a whole new ecosystem of solutions and tools designed to **help developers with their daily tasks, to be more agile**, efficient and precise. We have explored the potential of Generative AI to revolutionize tasks related to automatic code generation, interface design, error identification and complex problem solving. This technology can accelerate the speed of development, improve software quality and allow the creation of more personalized solutions adapted to user needs. Likewise, generative AI provides the opportunity to explore new creative possibilities and to find innovative solutions to technical challenges.

However, we are still in an **early experimentation phase** and it is too early to know how far this technological disruption can take us in terms of business. What is clear is that it requires companies to adopt this trend with a strategic approach and it is essential that organizations, regardless of their sector or activity, begin to study the benefits and implications of integrating this technology into their business models.

When adopting Generative AI, it will be essential for **companies to invest in the education and training of their teams in charge of developing software**. Professionals will need to acquire skills aimed at understanding and applying generative AI, so much so that courses and training programs are already emerging in a new knowledge space known as “prompting” and which is nothing more than the development of the human capacity to communicate with artificial intelligence in the most effective way possible.

Therefore, future learning involves understanding how a better stage of collaboration between humans and intelligent systems, so organizations must establish a culture of experimentation and continuous learning, promoting innovation and interdisciplinary collaboration.

Successful adoption of Generative AI also entails changes in organizational structure and business models. **Companies must be willing to adapt to new ways of working and redefine team responsibilities**. This involves integrating AI specialists, data scientists and designers into software development processes, as well as creating environments that encourage creativity and experimentation.

In terms of innovation and competitiveness, the adoption of Generative AI can make the difference between leading and lagging companies in the market. Those organizations able to take advantage of the potential of this technology will be able to develop more advanced, personalized and efficient products and services. Generative AI can boost a company's ability to quickly adapt

to changing market demands and anticipate customer needs, resulting in a **very significant competitive advantage**.

As Generative AI continues to transform the field of software development, new opportunities and challenges will arise for organizations, and those that manage to harness its full potential will be able to boost their competitiveness in a business environment that is experiencing an unprecedented historical revolution.

softtek.com

