

PoV

Generative AI and Coding - Rise of the Centaur Coder

Authors

Adish Apte



Table of Contents

01	Introduction	03
02	Is effective software coding possible with generative AI?	04
03	Advantages of generative AI for programmers	05
04	What are the Pros and Cons of Generative AI for software coding?	07
05	Will Generative AI play Dalek to exterminate the Coder tribe?	08
06	Rise of the Centaur Coder	09
07	So, what management interventions are required to make this new technology tick?	11
08	In Summary	12
09	References	12

Introduction

Generative AI – GPT and BERT models based on the transformer model, ever since the seminal paper "Attention is all that matters" by Ashish Vaswani et al. in 2017, have revolutionized NLP. ChatGPT, in particular, has made its presence felt in the public consciousness. Transformer models allow parallelization of text input and, more importantly, enable neural network models to focus on the right words in a sentence. Large Language Models (LLMs) are created through autoregressive and self-supervised pre-training of trillions of tokens sourced from the World Wide Web, digital bookstores, and software repositories. The models are continually growing.

Most of these models are jack of all trades, pre-trained on all the text there is to graze in the verdant token meadows of the World Wide Web. Some models are fed on specialized fodder, for example, GitHub Copilot and AlphaCode from DeepMind are trained on GitHub repositories.

This PoV discusses the impact of generative AI on software coding, its advantages, and associated pros and cons. It also deep dives into the rise of the Centaur Coder and the management interventions required to make this new technology tick.



Is effective software coding possible with generative AI?

GitHub Copilot, released in June 2021 (OpenAI Codex with 12B parameters), helps convert code between programming languages and plain English language requirements. It lacks the problem-solving skills needed for complex real-world challenges.

This is where DeepMind steps in with AlphaCodeⁱ, which has been developed and designed to write programs at competitive levels. In February 2022, DeepMind released AlphaCode, another LLM-based platform designed to write software programs at a competitive level. It achieved the position in the top 54%ⁱⁱ of participants in the coding challenges hosted by Codeforces – a first in AI. AlphaCode is trained for critical thinking, logic, algorithms, coding, and NLP and used GitHub repositories with 41 billion parameters for training. It supports C++, C#, Go, Java, JavaScript, Lua, PHP, TypeScript, Ruby, Scala, Rust, and Python.

This is a rapidly evolving space, and newer tools to watch outⁱⁱⁱ for are Tabnine, CodeT5, Replit, and Google's secret project, Pitchfork.

Advantages of generative AI for programmers

Our architects experimented with ChatGPT and Copilot, and the results on the coding front from ChatGPT are truly remarkable.

Making coders faster



ChatGPT from a coding perspective^{iv}

Developers frequently look up code on GitHub and Google for samples of code. Like a seasoned developer, ChatGPT provides a consolidated output to the query. While it might take a developer 15 minutes to find and comprehend the code online, ChatGPT can accomplish the same task in just a few seconds.

ChatGPT can generate boilerplate code for developers, like modern IDE (integrated development environment) like Eclipse and VSCode.

Code refactoring is possible. E.g., when asked to regenerate pom.xml (build file for java), which was generated for Java11 to support Java 17, it converted pom.xml to support Java 17 and could build projects using generated pom.xml; the quality of generation is also good.



Copilot from a coding perspective^v

Enables faster development, giving many coding suggestions like mini-Google searches.

Autocomplete for chunks of code, repetitive sections of code, and entire methods and/or functions.

Convert code comments to a runnable code.

Application building

ChatGPT cannot generate a whole application. However, developers can steer this process by breaking the application into smaller chunks and generating these building blocks.

Faster training

Getting new developers trained in new languages is easy. Seasoned coders exploring new programming languages with less tedious documentation can learn faster.

Build better SDLC artifacts

Generative AI allows for writing better requirements specifications/user stories, test cases, defects, and other artifacts. This high-quality SDLC data creates additional possibilities.

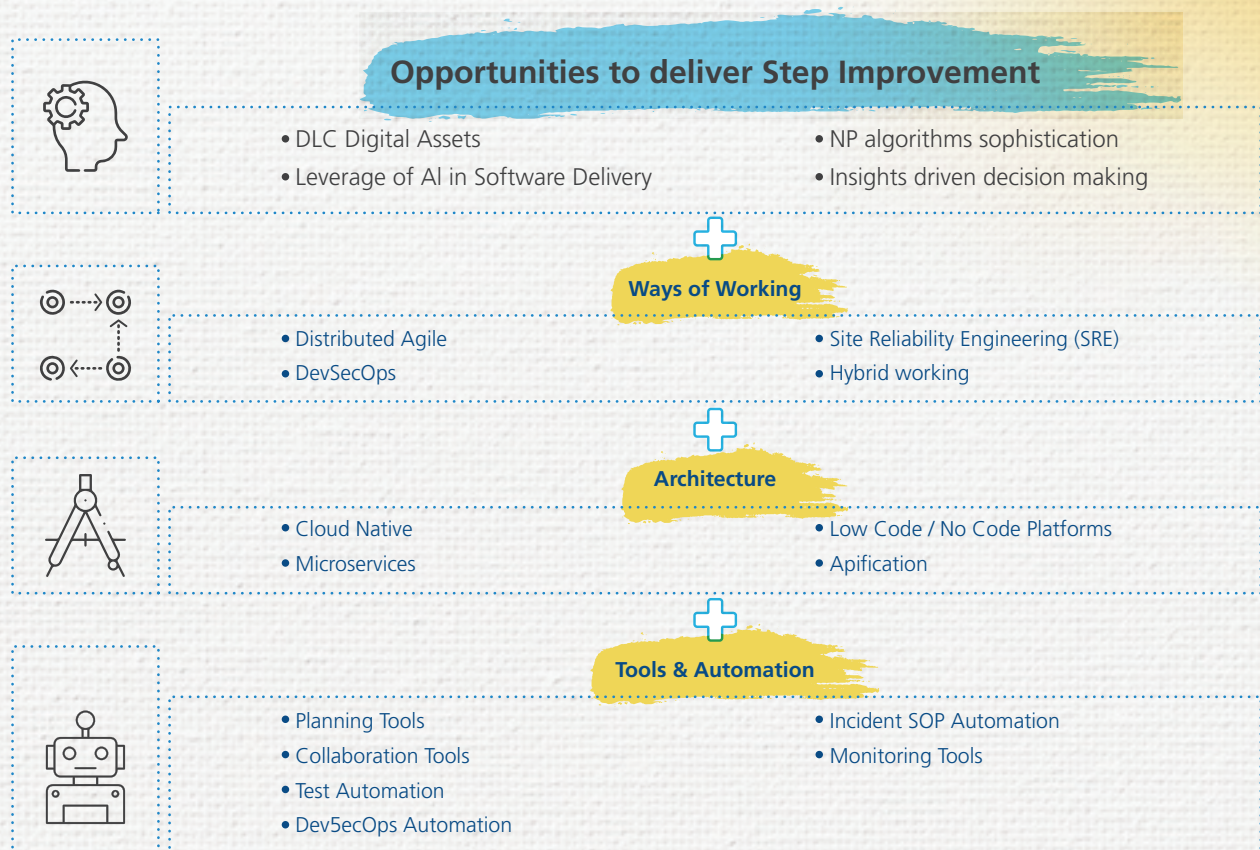


Figure 1: Latent need for insight-led software delivery acceleration

What are the pros and cons of generative AI for software coding?



Autoregressive models

A statistical model can be created that predicts based on past data. GPT-2 of 2019 cannot respond to COVID-19 queries, but may work for automated software coding.



Open Source

Hyperscalers see new LLMs as their significant IP investments; recent models (post-2022) have yet to be open sourced by Google or OpenAI. We may need to access Cloud APIs, which can be additional costs to the client.



IP leakage with Cloud APIs

Not being able to host your model means relying on centralized entities for training and inference. This also entails sharing our sensitive data, which will need client approval. The hosting company's policy changes could also lead to significant operational impact.



Licensing concerns

While models are trained on GitHub repositories, in the case of GitHub Copilot, a small portion of the code presented was copied verbatim. This leads to apprehensions and risks of infringing the original owner's copyright.

Will generative AI play Dalek to exterminate the coder tribe?

Intense debate is outside the purview of this note. However, two clear camps seem to emerge:



Pragmatic Optimist

Through the short history of software development, programmers have faced similar somber projections.

01

WordPress was supposed to eliminate web developers.

02

RPG was supposed to make COBOL programmers redundant.

Software development is very complex for the problems it solves. It needs critical thinking, analysis, and problem-solving skills.



Doomsday Pessimists

Much like Moore's law for CPUs, LLMs will continually become more extensive and effective, large enough and potent enough to displace the software coder soon.

Rise of the Centaur coder

The coder community should embrace AI tools, and management needs to adapt their operating models accordingly.

Centaur chess is a freestyle chess tournament. In 2005, a few amateur chess players and three ordinary computers, called centaurs, crushed all competition, including Hydra, the world's best chess supercomputer in 2005.

We are now entering the Centaur coder phase (including developers, testers, architects, and product owners) in the evolution of software development. Generative AI use cases will support a Centaur coder's work to increase productivity.



Centaur coder – Software developer

01

Generative AI as a co-programmer aids with building code more productively and faster. GitHub Copilot is already integrated as a plug-in Visual Studio IDE.

02

It also helps prepare better documentation of user stories, test cases, defects, and other SDLC artifacts, enabling the right solution the first time for deriving insights for future change requests.

03

AI-assisted tools for software delivery can be seen forming an ecosystem and growing, utilizing tools like LTIMindtree's Canvas to enhance STM and quality further.

04

GenAI will help coders learn new technologies faster, with easy search and easily explainable outputs from these LLMs.



Centaur coder – Tester

01

Gen AI will help correlate test cases and code files better for better test coverage and identification of automation opportunities.

02

Continuous testing—unit and regression—helps provide quick feedback to coders to fix their code.

03

AI can suggest alternative paths for explorative testing and tease out insidious bugs.

04

Testers can leverage AI-led insights to evaluate and retest to close fixed defects faster.



Centaur coder – Architect

01

Design and architecture patterns:

Architects can ask for various **architectural patterns** for specific computing needs. LLMs will list different patterns, provide more information on architecture patterns, and compare them. Architects can also ask for implementation details till the code level.

02

Product evaluation:

Architects can ask for a comparison or a point of view on a specific product, for example, CMS platforms like Drupal, Adobe AEM, and Liferay. Gen AI provides a good comparison, so that we can check whether a specific feature is available in that product.

03

Architecture decisions:

It helps decide on better fitment for designing specific architecture components.

04

Architecture diagrams:

A text-based AI model cannot generate PPTs or diagrams. But it cleverly generates diagrams using pure text and special characters. It can generate code that can be used in LucidChart or Draw.io.

So, what management interventions are required to make this new technology tick?

Setting up a Center of Excellence (CoE)

An NLP CoE must keep a tab on this mini-Cambrian explosion of LLM use cases and increase best practices within the delivery organization.

Data data data

Fine-tuning pre-trained models will be necessary to deliver differentiated outcomes. This needs good quality and a high volume of custom data. Data governance for collating cross-organization corpus will need sharper focus.

Change management

The entire software delivery community needs to appreciate and embrace the new technology and initiate training programs to operationalize it fully.

Upgrading the operating models

Tweak the operating model to incorporate the above.

New skills

As the volume of code output increases and business demands speed to market, the emphasis will shift to ensuring CI/CD pipelines. DevOps, Cloud, and NLP engineers will need integration to provide the boilerplates for software coders to focus on rapidly developing and releasing business functionality to production.

Right metrics

Code/developer is a flawed productivity metric. It is highly desirable to have DORA or SPACE frameworks and focusing on value stream management.

Policy and framework for data privacy and licensing

Sharing logs and errors to find issues will expose confidential data to the internet or ChatGPT servers. Copilot is currently untested on legal grounds. In November 2022, a class-action lawsuit was filed, challenging Copilot's legality. Policies and organizational setups to address these concerns will need to be shaped.

In summary

Generative AI LLMs seem to be here for the long haul. They will grow in power and efficacy in helping IT teams swiftly convert business requirements in plain English to production code.

An ecosystem of AI-assisted tools is emerging to enhance software delivery productivity further (Scopemaster, Diffblue, Rollbar, Canvas, etc.).

Businesses will continue to demand ever-increasing speed to market and quality from IT teams. Organizations must tweak their IT operating models and shift to AI-led and insights-driven software delivery.

References

- i DeepMind's AlphaCode Explained: Everything You Need to Know, Alberto Romero, Mar 9, 2022:
<https://towardsdatascience.com/deepminds-alphacode-explained-everything-you-need-to-know-5a86a15e1ab4>
- ii Competition-level code generation with AlphaCode, SCIENCE, Dec 8, 2022:
<https://www.science.org/stoken/author-tokens/ST-905/full>
- iii 5 AI Tools That Can Generate Code To Help Programmers, Janakiram MSV, Mar 14, 2022:
<https://www.forbes.com/sites/janakirammsv/2022/03/14/5-ai-tools-that-can-generate-code-to-help-programmers/?sh=ed17cc45ee0c>
- iv Using AI To Code Better? ChatGPT and Copilot change everything, Theo, Dec 3, 2022:
<https://www.youtube.com/watch?v=yZqIMNvqjg>
- v Your AI pair programmer, GitHub:
<https://github.com/features/copilot>
- vi "Centaur" Are Often More Powerful than Computers, Mike Beneschan, May 24, 2021:
<https://medium.com/geekculture/centaurs-are-often-more-powerful-than-computers-549f827130d9>
- vii The lawsuit that could rewrite the rules of AI copyright, James Vincent, Nov 8, 2022:
<https://www.theverge.com/2022/11/8/23446821/microsoft-openai-github-copilot-class-action-lawsuit-ai-copyright-violation-training-data>

Author's profile



Adish leads the Canvas Platform in LTIMindtree. With over 25 years of experience, Adish brings strong expertise across product management, relationship management, and sales. Besides leading the Canvas Platform development, he is focused on innovation with AI/ML in NLP and Knowledge Graphs & Ontology to deliver new insights for accelerating software delivery.

Adish Apte

Principal Director -
Product Engineering

LTIMindtree is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 700 clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by 82,000+ talented and entrepreneurial professionals across more than 30 countries, LTIMindtree — a Larsen & Toubro Group company — combines the industry-acclaimed strengths of erstwhile Larsen and Toubro Infotech and Mindtree in solving the most complex business challenges and delivering transformation at scale. For more information, please visit <https://www.ltimindtree.com/>