# "CLI coding agents: OpenAI Codex, Claude Code, Gemini CLI"

This title was summarized by AI from the post below.

**Matt Koppen**
10mo

Quick thoughts on CLI penAI Codex more than anything else...

**OpenAI Codex**
+
- Busts through toughe
- Much cleaner PRs: bu
- Seeming improbable
–
- Offers to do too muc ely), then says "how about I do X?" No! Focu
- Verbose and loves to
- Tends to want confirm
?
- Haven't tested it on t

**Claude Code**
+
- Plows through a _lot_
- Solid front-end work
- Strikes a good balanc an opus to read (see what I did there?)
–
- Can get very sloppy. PRs that don't build, syntax issues, piling in new code w/out any cleanup. Multiple rounds often needed to clean things up.
- Too often unfounded confidence. "I have done X and it will work this time." Um, it didn't work the last few times and you said the same.
- Blows through capacity pretty quickly at Pro level

(Note: Claude performance not apples-to-apples with Codex b/c I'm using Sonnet 4 due to Pro plan limitations. But that's a downside - Codex gives me GPT-5 on the same level plan!)

**Gemini CLI**
Well... what can I say about Gemini CLI. It's a relatively competent junior dev. In main projects, I feel ok giving it a very targeted feature to implement or bug to fix. I'd say it's 60/40 that it gets it when it's on 2.5-pro. If it's a goof project or experiment, I can let both 2.5-pro and 2.5-flash run wild and be ok if it's all a hash. And often it'll be a hash.

A plus? Finding humor in the all-too-often moments when it says (paraphrase) "Oh my, I really did screw that up. Let me go back and revert everything I just did."

Bottom line: Not my go to for anything critical.

**Cursor CLI and non-CLI coding agents**
I gave Cursor CLI a ver ugh Cursor I'd use GPT-5, Claude (Sonnet providers' CLI agents directly.

Same with non-CLI coc was ok, but mainly for prototyping, nothing s fine, but when I got started with Claude Co

Anyone had a different yone that leans on Gemini CLI (i.e. how?).

2 · 6 Comments

Like Share



**Shad Mickelbe** 9mo

I've been playir in helper in Cursor (with different r ive difference between the tw ect given a prompt of what you wa ific suggestions where you reall done even if it recognizes the

I built a working application with only writing very little code myself and it works great but was pretty annoying getting through some bugs and how certain features were implemented.

I've found the more narrow the focus on a change (perhaps just one file or even function/line) is much more reliable.

All that said it is still pretty exciting.

Like · Reply

To view or add a comment, **sign in**

## More Relevant Posts

**Omar Đečević**
9mo

I just published a short building to explore how AI can help with c
It covers how I approa at I learned along the way.

**https://lnkd.in/d6CDr**

**Socraites: AI-Powered**
drahil.com

25 · 1 Comment

**Like** **Share**



**Niaz Habib**
9mo · Edited

I've become a big fan of the OpenAI SDK — not because it's flashy, but because it's well engineered. In a world full of abstraction-heavy GenAI frameworks, this one feels refreshingly honest. Sharing a few thoughts on why it deserves more love 👇

The OpenAI SDK: Simplicity Done Right

If you haven't tried the OpenAI SDK, it's worth a look.
In the vast ocean of GenAI frameworks and agent builders, this one doesn't get nearly the love it deserves.
Everyone's talking about complex orchestration layers, agent graphs, and "frameworks for frameworks" — but sometimes, all you need is a clean, mature SDK that just works.
And that's exactly what this feels like.
You can tell it was built by engineers who actually write production code. It has that rare quality: it feels designed by experienced developers, for developers.

💡 A Breath of Fresh Air
I've tried a lot of GenAI frameworks lately. Many are impressive, but also... heavy.
They hide simple ideas behind layers of abstraction — "chains," "contexts," "graphs," "memory stores."
Great for demos, not s
The OpenAI SDK is diff o unnecessary magic.
No guessing.
Just solid engineering.

⚙ Why It Stands Out
Mature API design — c
Simplicity — minimal b
Easy to debug — what
Performant — no hidd
Fits anywhere — drop s native.
It's a rare SDK that scal a rewrite.

🧑‍💻 Built for Developer
There's confidence in it
It doesn't try to own yo our own architecture.
That's the hallmark of t

🧩 Growing — Without
Even as OpenAI adds n — the philosophy stays consistent.
Each layer adds power
You don't have to learn a new paradigm; it just fits naturally into what you already know.
That's rare — and it shows real design maturity.

❤ It Deserves More Love
Maybe because it's not flashy, the OpenAI SDK often flies under the radar.
But it's elegant. It's dependable. And it's built with respect for developers who care about clarity and control.
If you're building serious GenAI systems — not just demos — it's absolutely worth a closer look.

🔜 Coming Up Next
In my next post, I'll dive deeper into the Agent SDK and Agent Builder, and explore how OpenAI is

redefining what “agentic” can mean — without the bloat.
Sometimes the best tools aren’t the loudest — they’re the ones that just fit the way you think.

 Curious what others think — have you tried the OpenAI SDK yet? What’s your take compared to other GenAI frameworks?

26 · 1 Comment

Like Comment Share

To view or add a comment, **sign in**

**Saurabh Suri**
10mo

openai just did an ama

here's some the most i

> just shipped gpt-5-co h focused training on diverse coding tasks
>model performs signi due to specialized optimization
>team dogfoods heavi ith goal of zero manual coding next ye

team usage & product

> engineers can prototype large features with just ~5 turns of prompting, building 3 different feature versions in single day
> designers with no coding background now directly merge prs using codex, eliminating traditional developer bottleneck
> massive productivity gains reported: "no more writing crud endpoints or stream helpers" - focus shifts to higher-level architecture
> enables rapid experimentation where suboptimal prototypes can be scrapped without emotional attachment since they're "cheap to build"

performance trade-offs & complaints

> new gpt-5-codex significantly slower than previous models (15-20 minutes vs 3-4 minutes for same tasks)
> more intelligent but speed decrease breaks iterative development workflows for many users

technical capabilities & roadmap

> cli supports web search with --search flag, coming to ide extension soon with prompt caching fixes
> planning mode in development after internal debates about user control vs model autonomy
> exploring browser automation and voice integration (inspired by open source community demos)
> considering sub-agents and workflow automation but nothing actively in development

platform & integration challenges

> currently focused on ustrated with no timeline
> no free tier planned, windows experience p ershell command issues

token economics & rat

> teams working with l but we are just getting started" for ma pro users ($200/month ecifically
> confusing limit descr le session can run 25 minutes
> no usage tracking ui, ut they said its coming

future vision & philoso

> tools should become e background"
> abstraction level risir ng individual functions
> optimistic scenario: e eralist skills across design/product/engine

**Haoyi Yin**
10mo

□□💻 Here are 4 zero-cost methods for AI-powered coding in VSCode using Kilo Code ( **Kilo Code** ) a flexible alternative to GitHub Copilot that supports custom OpenAI-compatible APIs.

Method 1: Qwen-Code CLI + Free OAuth

Use Alibaba's Qwen-Co e Qwen3-Coder model. You get 2,000 f o token caps.

Setup for Qwen-Code:

1. Install via npm: npm
2. Run qwen and authe
3. In Kilo Code settings er or qwen3-coder-plus.

Method 2: ModelScope

ModelScope, Alibaba's s for models like Qwen3-Coder after lin ble.

Setup for ModelScope:

1. Register on **models** ken.
2. In Kilo Code:
• Provider: OpenAI Cor
• Base URL: **api.model**
• API Key: Your ModelS
• Model: qwen3-coder



Method 3: OpenRouter Free Models

OpenRouter aggregates models like DeepSeek V3 and Gemini 2.5 Flash. A one-time $10 top-up (your balance isn't used for free models) unlocks 1,000 free calls daily (20/min limit). Without it, you get 50.

Setup for OpenRouter:
1. Register on **OpenRouter** add $10 credit, and generate an API key.
2. In Kilo Code:
• Provider: OpenRouter

• API Key: Your OpenRouter Key
• Model: A free model like deepseek-v3:free or gemini-2.5-flash:free.

Method 4: Docker Self-hosted Gemini Balance

If you have multiple free Gemini API keys from Google AI Studio, you can self-host a load balancer to avoid rate limits. Gemini Balance deploys easily via Docker and is optimized for Gemini 2.5 Pro.

Setup for Gemini Balance:
1. Pull the Docker image: **https://lnkd.in/gWX9eP4d**
2. Configure a .env file with your Gemini keys and access tokens.
3. In Kilo Code:
• Provider: OpenAI Cor
• Base URL: http://<you
• API Key: Your access t

By load-balancing mult high-frequency AI coding assistance, tailo

These four methods gi side VSCode.

**#VSCode #FreeAI #Qv**

3

**Like** **Share**



**Patrick O'Sha**
9mo · Edited

I've been having relatively good success using agentic AI CLI tools like Claude Code (Anthropic) and Codex CLI (OpenAI) for coding tasks. The fact that they have access to the full power of the shell, my full project context, and integrate with my preferred and familiar Jetbrain IDEs (Rider, Webstorm,

etc), makes for a much better workflow than cut & paste code snippets. There is a continuity in conversation around the code base, and it does start to feel much more like a coding partner.

Of course, it is critical to use good software engineering practices with these new tools just as one would with any project - i.e. always review code diffs before committing to your git repo, do proper testing, etc. Also, their hallucinations and gaslighting over-exuberance about half-baked or even completely broken code can be a grating nuisance. But even with these limitations, I do find myself able to tackle more in less time. YMMV.

As I started spending more time in this mode, however, it became very jarring to abruptly hit up against API limits. My coding partner would suddenly take a multi-hours-long coffee break in the middle of a productive session!

So, I had an idea - wha me easily continue a coding session where I us to test the recent open source models, a l well locally on my machine (TL;DR - no, a un to explore)

The result is this open
**https://lnkd.in/g5jkQ**

Currently supported Cl
* Claude Code
* Codex CLI

Remote APIs:
* OpenRouter
* DeepSeek
* Google Gemini
* **Z.AI** GLM-4.5 (Claude

Local models:
Any local MLX or LM St

You can add connectio le. (Just ask Claude to add it for you)

For this project, I purposely was liberal with allowing the AI's to write nearly all of the code in the repo, including the git commit comments and README, to see how far it could get. I even used the tool itself while in development to switch between models to continue development on parts of it. Through the process, I learned a good deal about MLX, LiteLLM, LM Studio, the limitations in how LLM's access local tools, and the various coding LLM's that currently exist.

I hope it can serve as a useful resource for others that want to kickstart their own exploration of what is possible, and find the tools that work for them in the process. If you give it a try, I'd love to hear what you think!

(Btw, I wrote this post myself)

39 · 8 Comments

Like Comment Share

To view or add a comment, **sign in**

**Arnab Paul**
10mo



My Learning Journey w derstanding --->

Over the past few days ration) and applying it to a very practical use help of LLMs.

<-- What I Built -->

I implemented a pipeli , a Python file), and instead of feeding the sm to surface only the most relevant code sni that retrieved context.

**Here's a breakdown

-- Text Splitting -> Leve ode logically using custom separators like

-- Embedding & Retrieval -> Used sfr_embedding_code_400m_r_model from Salesforce (Open Model) for generating embeddings and retrieving top-k relevant chunks.

**Prompt Engineering**

- Created a custom PromptTemplate that ensures the model sticks to the given code context and avoids hallucination.

**Runnables & Chain**

- Built a RunnableLambda to dynamically pass retrieved code snippets into the prompt, then

pipelined it with an OpenAI LLM and a StrOutputParser for clean outputs.

**Final Execution**

- Queried the system with questions like “Merging of File and explain the code” and received contextually accurate answers extracted from the code.

<--- Key Learnings --->

**Smart Retrieval > Dumping Context **

- Feeding the entire co retrieval ensures only relevant pieces are use

**Custom Splitting is C

- Default text splitting specific separators (class, def) made retrie

**Hallucination Contro

- Explicitly instructing t ove reliability.

**Composable Chains

- Using RunnableLamb to extend.

<--- Why It Matters --->

Firstly I am thankfull to me to create something on Ai Doma

As codebases grow, un lenge. A RAG-powered approach can act like a th code references, explanations, and insig ntext.

This experiment is just the beginning, and I’m excited to explore more advanced retrieval strategies and integrations for real-world developer workflows.

Github - **https://lnkd.in/gZK9BCDH**

Curious to know how others are approaching RAG for code. Let’s discuss!

**#RAG #LangChain #LLM #CodeUnderstanding #AI #LearningInPublic**

Like Comment Share

To view or add a comment, **sign in**

**Omri Lahav**
10mo

Claude Bug Postmorte

A few weeks ago, durir y keyboard on the screen.

Claude is against me. T oded too much for one night. I'm tired too."

Did you also feel it?

We weren't hallucinatir

**Anthropic** just publish eparate bugs causing Claude to act weird. He

1) Requests were misr

Some prompts went to ague a question, but half the time the mess just not the one you needed.

→ Impact: outputs looked inconsistent, sometimes strangely off-topic, sometimes lower quality than usual.

2) Corrupted outputs

A config bug injected garbage characters into responses by mistakenly assigning higher probability to rare tokens, causing broken code, random tokens, nonsense in the middle of otherwise fine text.

→ Impact: developers got code that wouldn't even compile, or text that looked like Claude had a glitch mid-sentence.

3) Token-selection bug in the TPU compiler

An optimization bug sometimes caused Claude to literally skip over the right answer.

→ Impact: you'd get a confident but totally wrong response.

Each of these alone would be painful.
Overlapping together, they made it feel like Claude suddenly got stubborn and dumb (or both).

Now, what does it mean for your AI agents?

Your agents don't just rely on prompts and logic, they also rely on the LLM and its vendor.

These third-party LLMs
bug that crashes loudl
know it's wrong, you ju

After testing and evalu
they can start to behav
you're screwed.

When the LLM has this

- The output doesn't cr
- Guardrails don't save
"allowed but wrong" th
- Your careful pre-prod

So remember: agentic

17 · 1 Comment

Like

Share

To view or add a comment, sign in

**FireStrike**
47 followers
10mo

Software eats the worl nleashes their latest in coding intelligence, GP rites boilerplate, but can now refactor your out breaking a sweat. Novice? Doesn't matte watch Codex translate it into code.

But, there's a twist, the s Claude Code. OpenAI's move isn't jus pressive 74.5% on SWE-bench Verified, le om 33.9% to 51.3%. We're not looking at in t's possible for dev productivity.

But Codex isn't just for ne can pilot it, regardless of whether transformative for non-coders and seaso on builder.

Industry-wide, these a from monolithic prompt engines into s new normal. Competitors, brace yo IDEs will need to double down on AI int

However, if this is the r do we trust generated code in production? W unch an app? As the race between Claude a pipelines will require robust adaptation to keep pace.

The bottom line: AI code agents are no longer just helpers; they're teammates at the keyboard. Tech leaders should start prepping for upskilled teams tasked with validating and securing auto-generated software, while non-tech business units can begin leveraging AI to prototype faster.

Whether you're a veteran coder or a business leader eyeing digital acceleration, keep close tabs on this next wave; the boundaries of who creates, maintains, and secures software just changed.

More details and benchmarks in the original coverage:
**https://lnkd.in/g5DXfbCP** (OpenAI's new GPT-5 Codex model takes on Claude Code)

Don't just take our word for it, read the full story here: **https://lnkd.in/g5DXfbCP**

**#SECURITYOPERATIONS #BLUETEAM #CYBERSECURITY #SOC #DIRECTOROFAI**

Like Comment Share

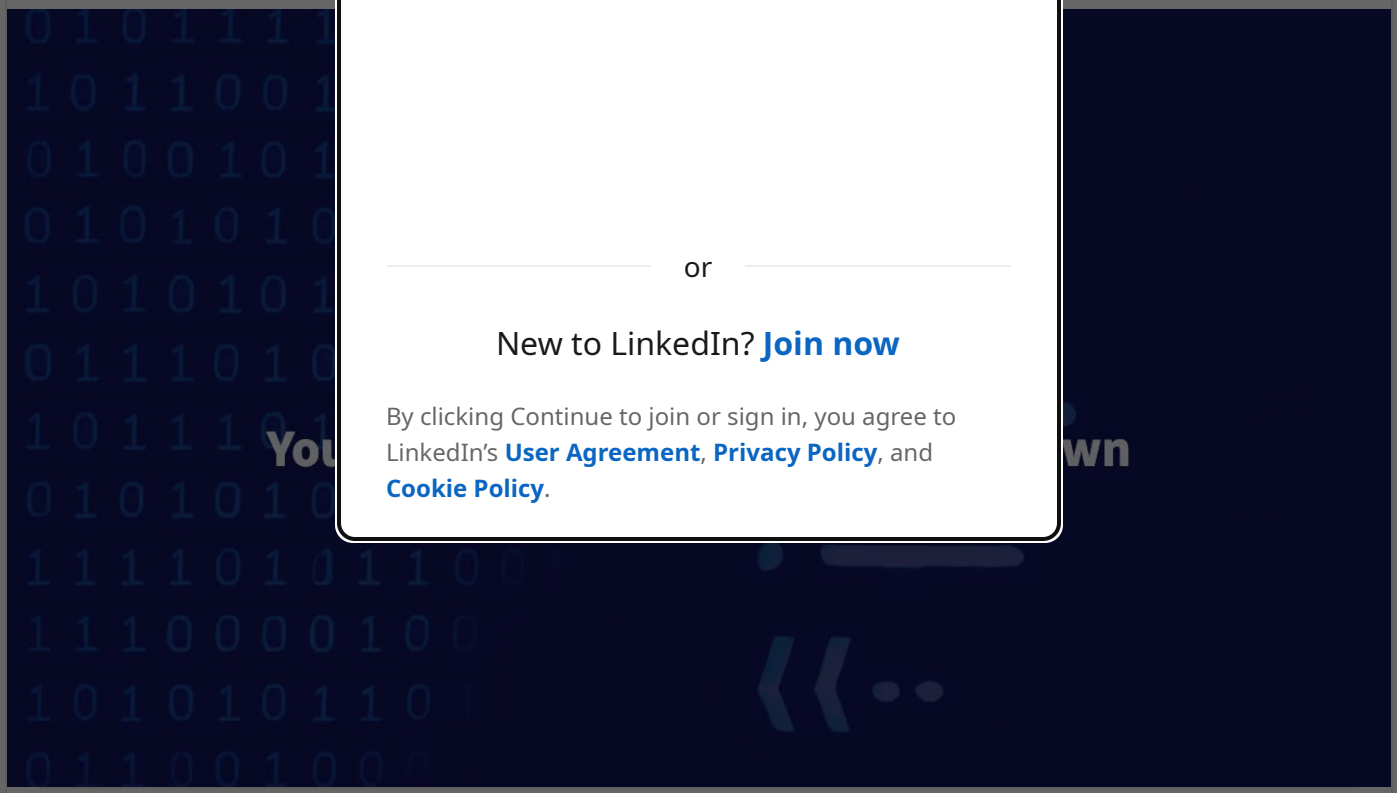
To view or add a comment, **sign in**

**Scott Farrell**
9mo

Your operating system
A traditional OS execut
execution.

wn.
edit a binary mid-

**...more**



Like Comment Share

To view or add a comment, **sign in**

1,391 followers

599 Posts

View Profile

Explore content

Career | Productiv | ...gence

Project Management

Show more