

Mit KI von der Idee zur App

Vibe Coding
20. Oktober 2025



itcc.li



Andrej Karpathy ✓

@karpathy



The hottest new programming language is English

[Post übersetzen](#)

9:14 nachm. · 24. Jan. 2023 · **8,2 Mio.** Mal angezeigt



Post



Andrej Karpathy

@karpathy



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.

...

I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it.

...

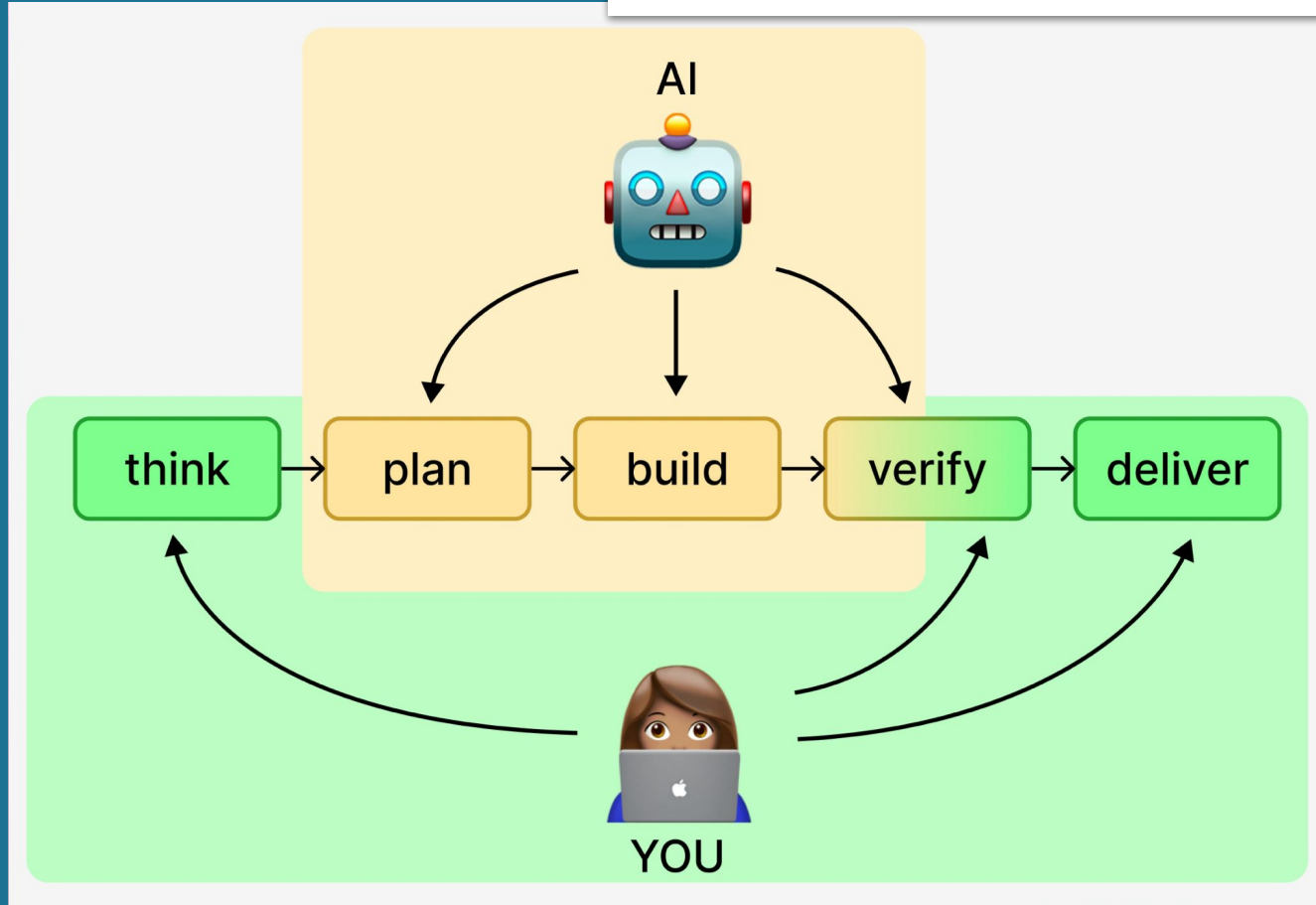
Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. ... but it's not really coding ... it mostly works.

12:17 vorm. · 3. Feb. 2025 · **5,1 Mio.** Mal angezeigt



Andrea Bizzotto 🇺🇦 @biz84 · 13. Okt.

This is kinda how I build with AI these days.



Vibecoding, iterativer Entwicklungsprozess

- 1 Plan erstellen
- 2 Plan prüfen
- 3 Schritt umsetzen
- 4 Fehler analysieren („Warum?“)
- 5 Plan anpassen & nächsten Schritt

Beispiel **StudyFlow**:

Ein minimalistisches, fokussiertes Lerntool, das Timer, Flow-Tracker und Fortschrittsanzeige vereint.

Plan erstellen

Prompt

Erstelle einen Entwicklungsplan für StudyFlow, eine minimalistische Lernfokus-App mit Timer, Flow-Tracker und Fortschrittsanzeige.

Gliedere den Plan in 5–7 Schritte von Idee bis MVP.

Plan prüfen und umsetzen

Prompt

Prüfe den Plan kritisch und verbessere ihn, falls nötig.

Prompt

Setze Schritt 1 des Plans um und erkläre kurz das Ergebnis.

Fehler analysieren

Prompt

Analysiere folgendes Problem:
[Fehlermeldung oder Beschreibung einfügen]

Warum ist der Fehler aufgetreten?



Plan prüfen & nächsten Schritt

Prompt

Prüfe und verbessere falls notwendig den Plan.
Dann setze den nächsten Schritt um.

Qualitätsprüfung für Produktions-Release

I've vibe coded an app, and now want to make sure it's of high enough quality to ship to production. I'm not a professional developer. Suggest 3-5 comprehensive and understandable LLM prompts I should now offer my agentic CLI to check for software quality attributes like usability, reliability, performance, observability, security, repeatability, scalability, availability, flexibility, and maintainability.

Tools

- ChatGPT
 - chat.openai.com
 - openai.com/codex
- Claude (claude.ai, [Claude Code](#))
- Gemini (gemini.google.com, [Gemini CLI](#))
- AI Studio (aistudio.google.com)
- Stitch stitch.withgoogle.com
UI-Designs + Frontend-Code
- Jules - KI-Coding-Agent
 - jules.google.com
- [How to use Stitch and Jules to create functioning web apps](#)

Tools

- Lovable (lovable.dev)
 - KI-App-Builder, vollständige Web-Apps aus Prompts (Frontend + Backend + Datenbank).
- V0 (v0.app)
 - Erzeugt React/Next.js-Code mit shadcn/ui und Tailwind.
- Bolt (bolt.new)
 - KI-Code-Editor von Stackblitz.
- Replit (replit.com)
 - Online-IDE mit integriertem KI-Assistenten („Replit Agent“) und Cloud-Runtime.

<https://code.visualstudio.com/blogs/2024/12/18/free-github-copilot>

The screenshot displays the Visual Studio Code editor interface for a project named "polaris-vibe-coding". The Explorer sidebar on the left shows the file structure, including a "src" directory with a "components" subdirectory containing "hello-world.ts". The main editor area shows the "index.html" file with the following content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Lit Hello World</title>
  </head>
  <body>
    <hello-world name="Simon"></hello-world>
    <script type="module" src="/src/main.ts"></script>
  </body>
</html>
```

The Chat sidebar on the right shows a conversation with GitHub Copilot. The first message, highlighted with a green border, is: "create a lit.dev scaffold project with a hello world web app". The response, highlighted with a yellow border, states: "I'll set up a minimal Lit + Vite + TypeScript project scaffold with a Hello World web component by creating the necessary files and configs in your workspace." It lists the files created: "package.json", "tsconfig.json", "tsconfig.app.json", and "vite.config.ts". The chat also shows a "Todos (3/3)" section with the following items:

- Enable GPT-5 for all clients
- Enable access to the latest GPT-5 model from OpenAI.
- Learn more about how GitHub Copilot serves GPT-5.

The terminal at the bottom shows the command "npm run dev" being executed, resulting in the Vite development server starting at "http://localhost:5173/". The status bar at the bottom indicates the current file is "index.html" at line 9, column 29, with 2 spaces, UTF-8 encoding, and LF line endings.