Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Coding Agents

Malte Luttermann – Institute for Humanities-Centered Artificial Intelligence (CHAI)

15.07.2026

Version from July 13, 2026

Agentic Tools Are Reshaping How We Code

THE SHIFT TO AGENTIC AI: EVIDENCE FROM CODEX

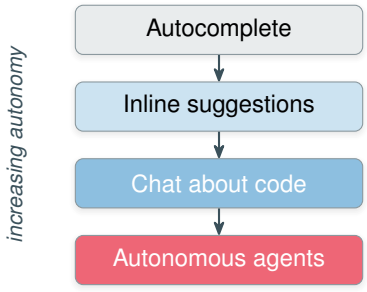
Drew Johnston[1,*] David Holtz[2,1] Alex Martin Richmond[1]
Christopher Ong[1] Prasanna Tambe[3,1] Aaron Chatterji[1,4]

[1]OpenAI [2]Columbia Business School [3]University of Pennsylvania, Wharton School
[4]Duke University Fuqua School of Business

We analyze usage data from OpenAI's Codex tool to present large-scale evidence of how agentic AI technology, which can take actions on a user's behalf, changes how people work. We use an automated, privacy-protecting pipeline to contrast usage across three populations: external personal-account users, external organizational-account users, and workers within OpenAI. We find that agentic AI usage is growing rapidly: the number of active users has grown more than fivefold in the first half of 2026, with the most rapid increase occurring outside the initial audience of software developers. Uptake is uneven across contexts: within OpenAI, Codex usage is nearly universal and has largely replaced business usage of ChatGPT. We document a similar shift to agentic tooling outside OpenAI, particularly within organizations, although external adoption remains lower and more uneven. In addition to headline usage figures, we observe measures of sophistication, and find that a growing number of users have used Codex to change their workflows substantially. We find that more than 10% of users manage three or more concurrent Codex agents at some point each week and that 26.6% use skills, which allow users to share instructions for complex workflows. Alongside these changes in usage practices, request complexity has increased: since the start of the year, the share of individual Codex users who submit at least one request for a task estimated to require more than eight hours for an experienced human to complete has increased nearly tenfold. Concurrently, output has grown rapidly—in June 2026, the median OpenAI employee in a legal role generated 13 times more monthly output tokens across Codex and ChatGPT than they did in November 2025, while the median researcher generated more than 50 times as many. We conclude by discussing the implications of these patterns for productivity, job reorganization, and workforce restructuring.

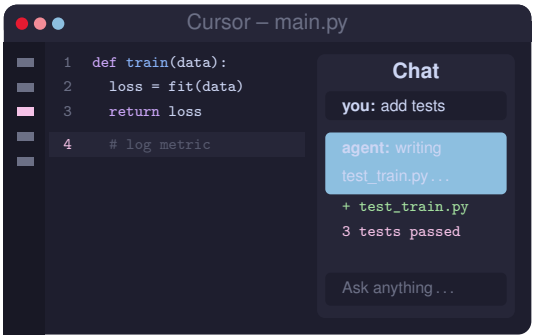
Agentic Tools Are Reshaping How We Code

- Modern assistants go beyond autocomplete
  - Suggest, refactor, explain
  - Plan and execute multi-step tasks
- Two prominent tools today:
  - **Cursor** – An agentic code editor
  - **Codex** – OpenAI's coding agent



Cursor – »AI-First Code Editor«

- Built as a fork of VS Code
  - Familiar UI, extensions, keybindings
- Agentic features deeply integrated into the editor
- Common models available (e.g., from Anthropic, OpenAI)
- Website: https://cursor.com



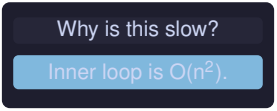
Cursor – Three Modes of Interaction

Tab

Predicts your next edit—not just the next token, but the next *change* across lines.

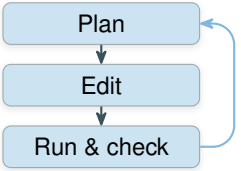
Chat

Ask questions about your code, get explanations, or request small edits with context.



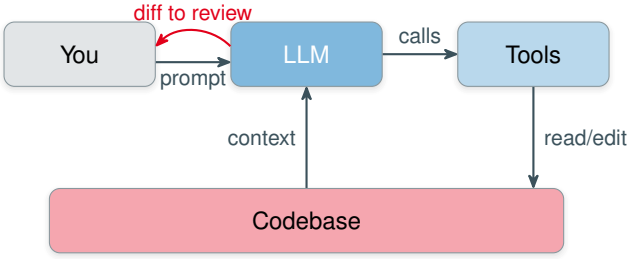
Agent

Plans and executes multi-step tasks: edits files, runs commands, iterates until done.



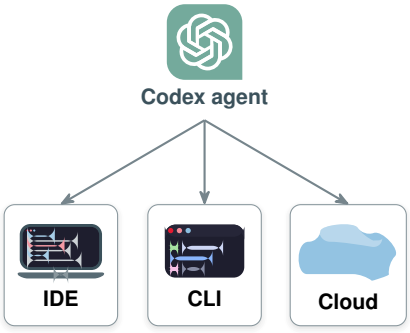
Cursor – How the Agent Works

- You describe a goal in natural language
- The agent
  - reads relevant files
  - proposes edits
  - can run commands
  - asks for confirmation on risky steps
- You review and accept the diff



Codex – OpenAI's Coding Agent

- Agent designed for coding tasks
- Runs in three places:
  - As an **IDE extension** (VS Code, Cursor)
  - As a **CLI tool** in your terminal
  - In the **cloud** as a background worker
- Powered by OpenAI's foundation models
- Website: developers.openai.com/codex/



Codex – IDE Extension

- Sidebar panel inside VS Code or Cursor
- Typical workflow:
  - Open your project
  - Describe the task in the panel
  - Inspect proposed diff and approve
- Sessions can be handed off between IDE, CLI, and cloud

Codex – Local vs. Cloud

Local (IDE / CLI)

- Direct access to your files
- Fast feedback loop
- Best for interactive work

Cloud (background)

- Runs on OpenAI's machines
- Tackles long tasks in parallel
- Returns a pull request for review

Cursor vs. Codex – Comparison

	Cursor	Codex
Type	Standalone editor (VS Code fork)	Agent + IDE extension + CLI
Models	Choice of foundation models	OpenAI models
Strengths	Tab edits, local file access	Long-running tasks, cloud handoff
Runs in	Desktop app	Your existing IDE, terminal, or cloud

Both tools share the same idea: An *agent* that reads, writes, and runs code with you.

Using These Tools as a Student

Helpful for...

- Prototyping a new idea quickly
- Explaining unfamiliar code
- Writing tests and docs
- Helping to debug and refactor

Be careful with...

- Code you do not understand
- Hallucinated APIs and citations
- Sensitive data and credentials
- Skipping the learning step

Rule of thumb: **Never** commit code you cannot explain.

Summary

- **Cursor**: AI-first editor with tab, chat, and agent
- **Codex**: OpenAI's coding agent in IDE, CLI, and cloud
- Both turn the editor into a collaborator – not a replacement
- Use them to learn faster, not to think less