NEWS · JUNE 28, 2026

GitKraken Is Now the Code Flow Company — and the Name Actually Means Something

By Dr. Holger Flick



As a GitKraken Ambassador I've watched the company evolve from "the nice Git GUI with the green-and-purple commit graph" into something with a much bigger thesis. This month they made it official: GitKraken is repositioning as **the Code Flow company**. Normally a corporate rebrand makes me reach for the eye-roll, but this one names a problem I'm living every single day — so let me explain what it actually means for developers like us.

The shift: generating code isn't the same as shipping it

GitKraken's whole argument fits in one sentence from their announcement: **"Generating code is not the same thing as delivering software."**

That lands because it's true. AI can now produce in hours what used to take weeks. But the bottleneck didn't disappear — it *moved*. The hard part is no longer typing the code. It's coordinating it: keeping context across

multiple agents, reviewing output you didn't write, integrating much larger volumes of changes, and getting all of it from idea to production without losing the plot.

GitKraken calls that movement **Code Flow** — how work travels between developers, coding agents, repositories, planning systems, reviews, branches, and production. Or, in CEO Matt Johnston's words: *"Moving that code from plan to main, that is Code Flow."*

The data behind the rebrand

This isn't pure marketing vibes. In May 2026 GitKraken surveyed **more than 550 software developers**, and the numbers explain why they're betting the brand on this:

- **96%** reported some level of AI adoption on their team.
- **65%** said more than half their team has adopted AI tools.
- **30%** are actively running **multiple AI agents in parallel**, with another **33%** experimenting with it.
- **28.5%** already let AI operate **autonomously** — on its own or alongside other agents.

If you'd told me a couple of years ago that nearly a third of developers would be orchestrating *several* agents at once, I'd have been skeptical. But that matches my own workflow now, and it reframes the job. As Johnston put it: *"What's changing isn't just how developers write code — it's how they work."*

The real bottlenecks, named

What I appreciate is that GitKraken's research names the friction points instead of waving at "AI is hard." The challenges developers reported are the exact ones I hit:

- Which model actually suits which kind of work.
- The right number of simultaneous agents before it becomes chaos.
- Whether a repository is even *ready* for agent-driven development.
- Holding standards steady across parallel workstreams.
- Safely integrating much larger volumes of code than a human wrote by hand.

None of those are code-*generation* problems. They're all coordination, visibility, and governance problems — which is precisely the gap "Code Flow" is meant to fill.

What it means for the products

The rebrand is the story, but it's backed by the existing toolset reframed around this idea — from plan to main:

- **GitKraken Desktop** — the Git GUI, now with an agent mode.
- **GitLens** — the IDE extension with 40M+ installs across VS Code, Cursor, Windsurf, Trae, and Kiro, gaining agentic capabilities.
- **Kepler** — a delivery engine / agent development environment for agent-driven work.
- **GitKraken CLI** — multi-repo command-line workflows.

- **GitKraken Insights** — engineering intelligence aimed at answering the question every leader is now asking: *what's the ROI of our AI investment?*
- **Git Integration for Jira** — connecting planning to the rest of the flow.

My take

I'm an Ambassador, so take the enthusiasm with the appropriate grain of salt — but the reason I find this interesting is that it matches reality on my own machine. The moment you go from "AI helps me write a function" to "I'm supervising three agents across two repos," your problem stops being *authoring* and becomes *flow*.

Naming that, and pointing a whole product line at it, is a smarter bet than chasing yet another code-generation feature.

Whether "the Code Flow company" sticks as a slogan, I don't know. But the underlying observation — that velocity has outrun coordination — is the most accurate description of agentic development I've read from a tooling vendor this year.

Sources: GitKraken: The Code Flow Company · GitKraken Introduces Code Flow (PR Newswire) · GitKraken Unveils Code Flow (SD Times)

INDEPENDENT · AD-FREE · READER-SUPPORTED

Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

Ways to support

KO-FI

Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

Support on Ko-fi —