NEWS · JUNE 28, 2026

# GitKraken ist jetzt die Code-Flow-Company — und der Name bedeutet tatsächlich etwas

By Dr. Holger Flick




Als GitKraken Ambassador habe ich miterlebt, wie sich das Unternehmen von "der netten Git-GUI mit dem grün-violetten Commit-Graphen" zu etwas mit einer deutlich größeren These entwickelt hat. Diesen Monat wurde es offiziell: GitKraken positioniert sich neu als **die Code-Flow-Company**. Normalerweise löst ein

Corporate Rebranding bei mir reflexartiges Augenrollen aus, aber dieses benennt ein Problem, das ich jeden einzelnen Tag erlebe — lassen Sie mich also erklären, was es für Entwickler wie uns tatsächlich bedeutet.

## Der Wandel: Code generieren ist nicht dasselbe wie Software ausliefern

GitKrakens gesamtes Argument passt in einen Satz aus der Ankündigung: **"Generating code is not the same thing as delivering software."** — Code zu generieren ist nicht dasselbe, wie Software auszuliefern.

Das trifft, weil es stimmt. KI kann heute in Stunden produzieren, wofür früher Wochen nötig waren. Aber der Engpass ist nicht verschwunden — er hat sich *verschoben*. Der schwierige Teil ist nicht mehr das Tippen des Codes. Es ist seine Koordination: den Kontext über mehrere Agents hinweg behalten, Output reviewen, den man nicht selbst geschrieben hat, deutlich größere Mengen an Änderungen integrieren — und all das von der Idee bis in die Produktion bringen, ohne den Überblick zu verlieren.

GitKraken nennt diese Bewegung **Code Flow** — wie Arbeit zwischen Entwicklern, Coding Agents, Repositories, Planungssystemen, Reviews, Branches und Produktion wandert. Oder, in den Worten von CEO Matt Johnston: *"Moving that code from plan to main, that is Code Flow."* — Diesen Code vom Plan bis nach main zu bringen, das ist Code Flow.

## Die Daten hinter dem Rebranding

Das ist kein reines Marketing-Gefühl. Im Mai 2026 hat GitKraken **mehr als 550 Softwareentwickler** befragt, und die Zahlen erklären, warum das Unternehmen die Marke darauf setzt:

- **96 %** berichteten von einem gewissen Grad an KI-Einsatz in ihrem Team.
- **65 %** gaben an, dass mehr als die Hälfte ihres Teams KI-Tools nutzt.
- **30 %** betreiben aktiv **mehrere KI-Agents parallel**, weitere **33 %** experimentieren damit.
- **28,5 %** lassen KI bereits **autonom** arbeiten — allein oder gemeinsam mit anderen Agents.

Hätte man mir vor ein paar Jahren gesagt, dass fast ein Drittel der Entwickler *mehrere* Agents gleichzeitig orchestrieren würde, wäre ich skeptisch gewesen. Aber das entspricht inzwischen meinem eigenen Workflow, und es definiert den Job neu. Wie Johnston es formulierte: *"What's changing isn't just how developers write code — it's how they work."* — Es ändert sich nicht nur, wie Entwickler Code schreiben, sondern wie sie arbeiten.

## Die echten Engpässe, beim Namen genannt

Was ich schätze: GitKrakens Untersuchung benennt die Reibungspunkte konkret, statt vage auf "KI ist schwierig" zu verweisen. Die Herausforderungen, die Entwickler nannten, sind genau die, auf die auch ich stoße:

- Welches Modell eignet sich tatsächlich für welche Art von Arbeit.
- Die richtige Anzahl gleichzeitiger Agents, bevor es im Chaos endet.
- Ob ein Repository überhaupt *bereit* für agentengetriebene Entwicklung ist.
- Standards über parallele Workstreams hinweg konstant halten.
- Deutlich größere Mengen Code sicher integrieren, als ein Mensch je von Hand geschrieben hätte.

Nichts davon sind Code-*Generierungs*-Probleme. Es sind durchweg Probleme von Koordination, Sichtbarkeit und Governance — und genau diese Lücke soll "Code Flow" schließen.

## Was das für die Produkte bedeutet

Das Rebranding ist die Geschichte, aber sie wird vom bestehenden Toolset getragen, das um diese Idee herum neu gerahmt wird — vom Plan bis nach main:

-

**GitKraken Desktop** — die Git-GUI, jetzt mit einem Agent-Modus.

- **GitLens** — die IDE-Erweiterung mit über 40 Millionen Installationen in VS Code, Cursor, Windsurf, Trae und Kiro, die agentische Fähigkeiten erhält.
- **Kepler** — eine Delivery Engine bzw. Agent Development Environment für agentengetriebene Arbeit.
- **GitKraken CLI** — Multi-Repo-Workflows auf der Kommandozeile.
- **GitKraken Insights** — Engineering Intelligence, die die Frage beantworten soll, die sich inzwischen jede Führungskraft stellt: *Was ist der ROI unserer KI-Investition?*
- **Git Integration for Jira** — verbindet die Planung mit dem Rest des Flows.

## Meine Einschätzung

Ich bin Ambassador, nehmen Sie die Begeisterung also mit der angemessenen Prise Salz — aber der Grund, warum ich das interessant finde, ist, dass es der Realität auf meinem eigenen Rechner entspricht. In dem Moment, in dem Sie von "KI hilft mir, eine Funktion zu schreiben" zu "Ich beaufsichtige drei Agents in zwei Repos" übergehen, ist Ihr Problem nicht mehr das *Schreiben*, sondern der *Flow*. Das zu benennen und eine ganze Produktlinie darauf auszurichten, ist eine klügere Wette, als dem nächsten Code-Generierungs-Feature hinterherzujagen.

Ob "the Code Flow company" als Slogan hängen bleibt, weiß ich nicht. Aber die zugrunde liegende Beobachtung — dass die Geschwindigkeit der Koordination davongelaufen ist — ist die treffendste Beschreibung agentischer Entwicklung, die ich dieses Jahr von einem Tooling-Anbieter gelesen habe.

---

*Quellen:* 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 —**