

Kilo Code

The Agentic Engineering Platform for Enterprise ROI



**The AI coding tool you buy
today will be obsolete in 18
months**

unless it's model-agnostic.

The only durable advantage is the ability to switch.

The AI Landscape is Shifting

Why the Status Quo is Becoming a Liability



Rapid Obsolescence

Best practices from 6 months ago are already obsolete



Model Churn

The best model today won't be the best tomorrow



Paying Twice

You have cloud AI credits you can't use with current tools



Vendor Lock-in

Copilot/Cursor/Claude Code lock you to one provider's roadmap



Point Solution

Sprawl

Stitching together IDE, CLI, cloud, and review tools



Bottleneck Shifting

Fast code doesn't help if specs, tests, and docs can't keep up

Your current AI coding tool locks you to yesterday's best model.

The Cost of Waiting

What happens if you stay on your current path?

Next 6 Months

2-3 new frontier models will release. Your locked-in tool may not support them. Competitors who can switch will ship faster.

Next 12 Months

Your unused cloud credits expire or get scrutinized by finance. "Why are we paying for AI twice?" becomes a board-level question.

Next 18 Months

The productivity gap compounds. Teams using flexible AI tooling will have shipped 3-4x more features. Talent will prefer companies with modern AI stacks.

Every quarter you wait is a quarter your competitors don't.

What Best-in-Class Looks Like

The Enterprise AI Coding Stack You Actually Need



Credit Utilization

Your \$500K Anthropic commitment actually gets used. No separate billing. No double-paying.



Strategic Autonomy

Open Source, Open Models, and Open Pricing means you choose what you need and can walk away at any time.



Verifiable Costs

No obscured token usage. Visibility into Kilo Gateway spend or BYOK means you verify spend directly. Never any hidden markup.



Model Risk Mitigation

When a new model comes out or Opus is expensive, you -- not us -- choose whether a more expensive model delivers enough ROI.

This isn't a fantasy. It's what Kilo Code customers experience today.

The Market Today

Three categories, three compromises

Developer Autocomplete

GitHub Copilot, Tabnine

Fast suggestions, but locked to one model. Developers only. No SDLC coverage.

AI-Native IDEs

Cursor, Windsurf

Powerful in-editor agents, but you abandon your tooling. Still single-provider.

CLI Agents

Claude Code, Aider

Terminal-first power users only. No governance. No team visibility.

Writing more code faster just moves the bottleneck.

The only way to eliminate it is a tool that addresses the full SDLC.

Introducing Kilo Code

The Agentic Engineering Platform

Dimension	Copilot	Windsurf	Claude Code	Kilo Code
Always have the best model	Vendor-controlled	Vendor-controlled	Anthropic only	✓ 500+ models or BYOK
Use your cloud commitments	x	x Can remove access	Anthropic only	✓ Any provider, direct billing
Eliminate bottlenecks across SDLC	Dev only	Dev only	Dev only	✓ All-in-one platform
Eliminate vendor lock-in	x	x	x	✓ Open source + source available

The only platform that combines model freedom, full SDLC coverage, and enterprise control.



What Makes Kilo Different

The Platform Built for Enterprise Reality

Three Pillars of Kilo Code

Model Freedom

500+ models – including Opus 4.5, GPT-4.5, Gemini 2

BYOK – use your Anthropic/OpenAI/AWS credits

No lock-in – Apache-2.0, audit every line

Full SDLC Coverage

IDE-first – VS Code + JetBrains where 80% of devs work

CLI + Cloud – power users and automation when ready

Code Reviews – automated PR analysis

Non-devs too – PMs, QA, writers use Architect mode

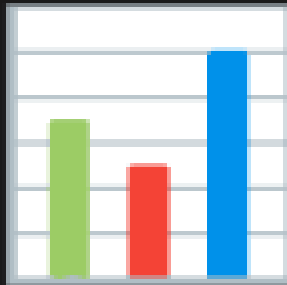
Enterprise Control

Full visibility – usage data you own, for optimization and provider negotiations

Policy controls – model allowlists, cost caps

SOC2 + RBAC – no data retention on paid plans

One platform. Any model. Total control.



Proof

Results at Scale

Proven at Scale

#1

on OpenRouter

1.5M+

Developers

25T+

Tokens Processed

Market validation and production scale

Customer Results

Real outcomes from real teams

FINTECH / E-COMMERCE

Plug&Pay

Challenge: Fragmented AI stack (Cursor, CodeRabbit, Claude Code, BugBot) with opaque pricing

Result: 80% developer adoption, unified billing, Code Reviews on all repos

"We pay for what we use, without hidden costs."

– Daniël Langezaal, Software Engineer

EVENT MANAGEMENT

Admin Collective

Challenge: Trapped in Replit after \$15K+ invested, couldn't export code

Result: Rebuilt entire app in 2 weeks with full code ownership

"I rebuilt the entire app in Kilo Code in two weeks."

– Dona Dowlan

TASK MANAGEMENT

22 Tasks

Challenge: Solo founder hitting usage caps on Copilot and Claude

Result: 20% → 60% complete in 1 week, meaningful progress in 1 hr/day

"For the first time, it feels like meaningful work with a reasonable partner."

– Andrey Guenov

Read full stories at kilo.ai/customers

Getting Started is Easy

Week 1-2

Pilot Team

- 5-10 people (mix of devs & non-devs)
- Install extension
- Connect BYOK to cloud provider
- Start with real tasks, not demos

Week 3-4

Measure & Expand

- Track velocity & cloud credit utilization
- PMs start using Architect mode
- Compare ROI to current tools

Month 2+

Scale with Confidence

- Roll out across organization
- Custom modes for your domain
- Full management plane deployment

Low-friction adoption, immediate value, measurable ROI from day one

The Kilo Speed Run

No-risk proof of value



Embedded Engineer

A Kilo engineer joins your team



1-2 Day Sprint

Intensive focus on a real problem




Real Results

Solve an actual project challenge

See Kilo Speed firsthand—on your codebase, with your team.

 sales@kilocode.ai

 kilocode.ai

 kilocode.ai/docs

Appendix

Additional materials for Q&A

Questions We Hear

"Is open source enterprise-ready?"

Yes. Apache-2.0 means you can audit every line. We add enterprise management plane, SSO, RBAC, and support on top. Kubernetes is open source too.

"How hard is migration from Copilot/Cursor/Windsurf?"

Most teams run both in parallel during pilot. No data migration needed—Kilo works with your existing repos. Typical full cutover: 2-4 weeks.

"What if models get worse or pricing changes?"

That's exactly why model-agnostic matters. When OpenAI raised prices 2x last year, Kilo customers switched to Claude or Gemini in one click. You're never trapped.

More Questions

"Our security team will have concerns."

BYOK means your code goes directly to your cloud provider. We never see it. SOC2 Type II certified. No data retention on paid plans.

"What if our vendor removes a model we depend on?"

When Windsurf removed Opus 4.5 access, customers had no visibility and no recourse. With Kilo + BYOK, you see usage, you own the provider relationship, and you switch models in one click. Your capability is never held hostage.