

Kilo Speed

How We're Already Living in 2028



Scott Breitenother

CEO & Co-Founder

Kilo

The Transformation

How engineering is evolving

MARCH 2025

1 feature

every 2-3 weeks

TODAY

1-2 features

per week

Same team size. Same deadlines.

**We stopped working like it's 2023
and started working like it's 2028**

The Old Paradigm

What's Dying

Developer as code monkey

- You write every line
- Manual boilerplate

Collaboration as default

- Endless meetings
- PR review bottlenecks

Velocity killers

- Waiting on others
- Context switching

The New Paradigm

What's Emerging

Developer as orchestrator

- You manage AI agents
- You guide vision

Collaboration as exception

- Only when it adds value
- End-to-end ownership

Velocity multipliers

- No waiting
- Deep focus

Part 2: How Kilo Engineering Works Today

The Agent Team Model in Action

The Agent Team Model

Each developer manages 3-5 specialized agents simultaneously

Agent Types

Orchestrator Agent

Breaks down complex projects into manageable subtasks, assigns them to other agent modes, and coordinates between them

Architect Agent

Research your codebase to create design systems and written implementation plans

Code Agent

The default mode that transforms your requests and written plans into production-ready implementations

Ask Agent

A knowledgeable technical assistant focused on answering questions without changing your codebase

Debug Agent

An expert problem solver specializing in systematic troubleshooting and diagnostics

How We Ship This Fast

Individual Ownership

- One engineer owns Deploy end-to-end
- Uses Orchestrator to break down the feature
- Ships working code in 5 days, not 5 weeks

No Coordination Tax

- No waiting for signoff before getting to work
- Agents handle implementation, developer handles vision

Continuous Validation

- Debug agents catch errors automatically
- Ship with confidence, iterate on feedback

What This Actually Looks Like



Developer

- Sets direction
- High-level requirements
- Creative decisions
- Architecture choices
- Quality gates



Agents Handle

- Boilerplate
- Implementation details
- Error recovery
- Documentation lookup
- Syntax debugging



Result

10x

Productivity increase on
implementation work

The Empowerment Model

Reduced Dependencies

No waiting for backend team, frontend team, etc.

Full-Stack Autonomy

Agents help you work across the entire stack

Faster Feedback Loops

Test, iterate, ship without coordination overhead

Part 3: Eating Our Own Dog Food

The Meta-Advantage

The Meta-Advantage

You might be thinking: Of course Kilo can ship fast, you built the tool.

But that's exactly the point.

The Product Development Flywheel

1

Use Kilo Code

to build Kilo Code

2

Feel Pain Points

first, as users

3

Ship Features

to fix them

4

Ship Faster

using those features

We feel the problems. We fix the problems. We ship faster. Repeat.

Recent Example: Parallel Agents

Before Parallel Agents

Working on Deploy meant context-switching between CLI work, config detection, and documentation

After Parallel Agents

One developer orchestrates three agents working simultaneously

Result

Deploy shipped in 1 week instead of 3

The Difference

Other Tools

- ❌ Built by teams that don't use them daily
- ❌ Features based on theory
- ❌ Slow to fix pain points

Kilo Code

- ✅ Built by developers who live or die by it
- ✅ Features based on real pain
- ✅ Fast iteration on what matters

That's why we have:

- ✅ MCP marketplace
- ✅ Cloud Agents

Every feature we ship makes us faster
at shipping the next feature.

And you can enter that same cycle.

Part 4: Why This Isn't Happening Everywhere (Yet)

The Barriers

The Tool Problem

Most AI coding tools are black boxes

✗ Don't know what model you're using

✗ Locked into one provider's rate limits

✗ Can't see the prompts or context

✗ Can't customize or extend

✗ Tied to only one lab or set of models

✗ Can't choose the best model for the task

✗ Vendor can remove models mid-contract

The Trust Problem

Hallucinations

AI that makes up APIs and best practices

Hidden Failures

Errors that don't get caught until production

Context Gaps

AI that doesn't understand your codebase

The Control Problem

This isn't just a developer problem—it's a procurement risk



Can't run agents in parallel

Stuck with sequential execution



Can't specialize agents for different tasks

One-size-fits-all approach



Can't switch models based on task complexity

Paying premium prices for simple tasks

Part 5: How Kilo Code Solves This

The All-in-One Platform Built for the Future

Open by Design



Apache-2.0 Licensed

Inspect every prompt, every decision



500+ Models

Use the right model for each task



No Hidden Compression

You see exactly what the AI sees



BYOK or Kilo Gateway

Total pricing transparency, pay exact list price



Enterprise Usage Analytics

See consumption by team, model, and task

Built for Agent Orchestration



5 Specialized Modes

Orchestrator, Architect,
Code, Ask, Debug



Parallel Execution

Run multiple agents
simultaneously



MCP Marketplace

Add tools like Context7 for
hallucination-free code

Works Where You Work



CLI

For power users



JetBrains

All JetBrains editors



VS Code

All VS Code-based IDEs



Tab Autocomplete

Fast, knowledgeable suggestions

Proven at Scale

#1

on OpenRouter

750k+

Developers

6.1T

Tokens/Month

Customer Spotlight: Plug&Pay

€1B+ payment processor · Netherlands & Belgium

80%

Developer adoption

```
<div class="bg-blue-900 bg-opacity-30 p-4">  
  <div class="text-3xl font-bold" style="color: #000000;">  
    <div class="text-sm opacity-75">Tools c  
  </div>
```

```
<div class="bg-purple-900 bg-opacity-30 p-4">  
  <div class="text-sm opacity-75">Replaced: Cursor, CodeRabbit, Claude Code, BugBot,  
</div>
```

"We watched other tools shift from unlimited requests to abstract credit models overnight with random markups. It felt like a bait-and-switch. We needed a partner, not just a vendor."

– Daniël Langezaal, Software Engineer, Plug&Pay

And We're Just Getting Started

Here's what we've just shipped

Just Shipped: November/December 2025

Cloud Agents

Run AI agents in the cloud without consuming local resources. Execute long-running tasks, complex workflows, and resource-intensive operations while your machine stays free for other work.

Shipped

Code Reviews

Automated code review agents that analyze pull requests, suggest improvements, catch bugs, and ensure code quality standards. Get instant feedback before merging.

Shipped

App Builder

Visual app builder that generates production-ready code. Design interfaces, define logic, and let Kilo Code create the implementation following best practices.

Shipped

Just Shipped: November/December 2025 (cont.)

Managed Indexing

Deep semantic understanding of your entire codebase. Kilo Code can navigate, understand, and modify large projects with full context awareness with no setup needed.

Shipped

Agent Manager

Run multiple AI agents simultaneously directly in your IDE. Coordinate complex tasks across different parts of your codebase without switching to the CLI.

Shipped

Notice the pattern?

**Each feature builds on the last.
Each makes developers more autonomous.**


And we're shipping them at pace because
we use Kilo Code to build Kilo Code.


Part 6: What This Means for Your Team

From Roadmap to Reality

Look at Our Roadmap Again

 Cloud Agents

 Code Reviews

 App Builder

 Managed Code Indexing

 Parallel Agents in IDE

A year ago, that's a...

12-18 month roadmap

for a 20-person team

Now?

6 weeks

for < 15 engineers

That's not because we work harder.

**One engineer with Kilo Code can do
what three engineers could do before.**

And we're proving it in production,
every single week.

ROI Reality Check

↗ **3 hours saved / week / engineer** ≈
\$1,500/month

\$ Typical inference spend: < \$100/month



ROI payback in **first 2 weeks** of
deployment



Full visibility = leverage in vendor
negotiations

The 2-3 Year Timeline

2025

Early Adopters

Using agent orchestration (like Kilo)

2026

Standard Practice

At high-velocity teams

2027

Expected Baseline

For competitive engineering orgs

The Skills Shift

Less

- ✗ Writing boilerplate
- ✗ Debugging syntax
- ✗ Waiting for help
- ✗ Context switching

More

- ✓ System design
- ✓ Agent orchestration
- ✓ Quality judgment
- ✓ Creative problem solving

New role: Developer as conductor, not musician

The Cultural Shift

From

"Let's schedule a meeting to align"

To

"I'll ship a prototype and we can iterate"

Empowerment over coordination

Trust over gatekeeping

Shipping over discussing

The Proof Is in Production



Coders LLC

Owner

Dashboard

Organization

Usage

Integrations

Review Agent

Cloud Agent

Deploy

Model Access

Custom Modes

Audit Logs

Subscription

Invoices

Admin

ES

Emilie Schario
emilie@kilocode.ai

Organization Details

Organization Information

Basic organization details

Organization ID

ff7a9a5e-bda9-46d7-9fd2-be561fd28014

Name

Coders LLC

Balance **\$99.46**

Buy More Credits

View Billing

Usage

Organization usage spend (last 30 days)

Spent **43.83** Total **7,307**

Total Input Tokens

75.17M

Total Output Tokens

961.90K

View Detailed Usage

Providers and models

48 providers and **704 models** selected

Set default model

Members (7)

Team members and pending invitations

+ Add Member

+ Invite Member

Alex Gold

alex.gold@kilocode.ai

Joined: Oct 8, 2025

Owner



Brendan O'Leary

brendan@kilocode.ai

Joined: Sep 16, 2025

Member



Brian Carlson

brian@kilocode.ai

Joined: Oct 1, 2025

Owner



David Fisher

david@kilocode.ai

Joined: Nov 3, 2025

Member



Emilie Schario

emilie@kilocode.ai

Joined: Sep 26, 2025

Owner



George Sumenkovic

george@kilocode.ai

Joined: Nov 7, 2025

Owner



Job Rietbergen

job@kilocode.ai

Joined: Nov 14, 2025

Owner



PRODUCTION

"Doesn't this reduce code quality?"

✓ No, it shifts quality work from syntax to architecture

You spend time on what matters: design, not typos

✓ Debug agents catch errors automatically

Continuous validation throughout development

✓ You review agent output, agents don't review yours

More time for design, less time for nitpicks

"What about junior developers?"



They learn faster by managing agents than writing CRUD by hand

Focus on concepts, not syntax memorization



They see best practices in action, not just in docs

Real-world examples generated instantly



They ship real features on day one, not after months of onboarding

Immediate contribution and confidence building



They learn system design earlier in their career

Elevated to architecture sooner

Why Kilo Code?



Open Source

Apache-2.0 licensed
See every prompt



Full Control

500+ models
BYOK or Kilo Gateway



Agent Orchestration

Parallel execution
Specialized modes



Transparent Pricing

No commission
Pay exact list price
Usage data you own



Auto Recovery

Automatic failure detection
Self-healing workflows



Extensible

MCP marketplace
Custom integrations

Thank You

Book a Speedrun – ship a real feature in 60
minutes

Questions? Let's talk.
scott@kilocode.ai