AI Enablement for Serious Engineering Organizations

Build the AI-enabled engineering system your organization actually needs.

Writing code is getting cheap. The expensive parts of engineering now are judgment and context: knowing what to build, keeping agents pointed at the right problems, and answering for what ships. Most organizations respond by giving each engineer a coding assistant. That produces a few impressive power users and leaves the organization unchanged.

> *We help you design and prototype the code agents and AI infrastructure your team keeps wishing for, built against your real systems, budgets, policies, and data boundaries.*

OUR POINT OF VIEW

The future of engineering is multiplayer.

Giving one engineer a great assistant makes one engineer faster. The change that matters happens when the whole team shares context, practices, and infrastructure.

AI should expand capacity and ambition.

The systems we design create **cost-optimized** engineering capacity at scale. We call it tokenoptimization, not tokenmaxxing. Doing today's work with fewer people is the least interesting option. Spend the new capacity on work that never made the cut: the migration nobody could staff, the tool everyone wanted.

The why must remain human.

Agents can carry the mechanics; people orchestrate them. Agents can't be accountable, and they don't care why the work matters. Purpose and judgment stay with people, and the tooling should leave people with more agency.

Your constraints are design inputs.

Security review, privacy rules, audit trails, procurement, where your code and context travel: requirements to design for, not obstacles to route around. Trusted intermediaries in front of the labs; proof that nothing phones home. The interface between the models and your institution should belong to you — including, when it earns its cost, a model tuned to your work.

Workflows matter more than tools.

Any tool you adopt is temporary. The way your team works with agents is what compounds: how work gets split, where context lives, how agent output gets reviewed.

The mechanics will change. The values should not.

Build something bounded, use it, let evidence drive the next decision. Your standards for trust and judgment should outlast the tooling.

WHAT WE BUILD TOGETHER

- Coding-agent practices that hold up across teams, not just for one power user
- Agent interfaces wired into your repositories, permissions, and workflows
- Systems that capture organizational context and keep it where agents can find it
- Patterns for review, testing, and oversight when agents write most of the code
- Cost controls that keep agent runs token- and budget-aware
- Experiments that settle what to build, buy, govern, or own

HOW WE WORK TOGETHER

There's no fixed on-ramp. Start with a day at our Chicago lab, a working session with your technical leads, hands-on time with one of your teams, a bounded prototype, or a research question. Every format has the same goal: your own evidence, your own capability, and the ability to keep going without us. If you're shopping for staff augmentation, model licenses, or a big-firm transformation program, we're the wrong shop.

WHY 2389

We hit these questions in our own work first. Our lab builds agent infrastructure, coding systems, orchestration tools, and context-management software. We learn by shipping, we publish what we find, and we open-source what we can.

Let's compare notes.

Bring us the problem that makes AI adoption hard where you are: the mission, the backlog, the policy, the systems mess. Build more. Keep the why human.

work@2389.ai
2389.ai · Chicago, IL