

Human-Centric Software Development Augmented by AI

By Terry Yin and Song Tao

In early 2025, we worked with a client in Singapore to explore how AI could augment—not replace—software development. The project was built on a 20-year-old legacy system with complex cross-team dependencies. Rather than chasing productivity for its own sake, our goal was to discover how humans and AI could work together meaningfully.

From Human Needs to Human-AI Collaboration

Our approach begins and ends with humans. Requirements come from real people with real needs—not from AI hallucinations. AI can help fill in details, but it must never define the problem itself. True requirements demand empathy—the human ability to feel another person’s experience.

Similarly, analysis must remain a human-centered, collaborative process. It determines direction and meaning. Giving it entirely to AI would be like outsourcing thinking itself.

Development and delivery, however, are where AI truly shines—provided humans stay in the loop. Our teams use AI for code generation, testing, and automation, but we remain deliberate about when and how to rely on it.

The Risk of Cognitive Debt

When AI solves problems for us, we gain convenience but lose depth. “Cognitive debt” arises when we skip the struggle that produces real understanding. Engineers often use AI to solve problems they don’t fully grasp. Once the code works, dopamine rewards them, but the learning never consolidates.

Over time, the system’s product reality drifts away from the team’s mental model. Eventually, production issues appear that no one—human or AI—can truly diagnose. The antidote is to ensure shared understanding through collaboration, not delegation.

Flipping Automation: Lessons from Jidoka

We drew inspiration from Jidoka (自動化)—the Toyota principle of “automation with a human touch.” Instead of blindly automating, Toyota built machines that automatically stopped when something went wrong. Counterintuitively, this made production more efficient.

We applied the same principle to AI: automation must pause itself when uncertain. AI should not produce endless defects faster—it should produce feedback faster. The goal is not full automation, but meaningful human-AI interaction.

Feedback Over Progress

We learned to focus on feedback, not progress.

- Unit tests give narrow feedback—useful but shallow.
- End-to-end tests simulate user interaction, providing rich behavioral feedback.

We continuously run end-to-end tests locally with deterministic environments. Every developer can replicate the full stack on their laptop—frontend, backend, Kafka, Redis, and databases. This high-fidelity feedback loop gives both humans and AI immediate insight into what's broken and why.

Human-in-the-Loop: Planning, Acting, and Stopping

“Human-in-the-loop” doesn't just mean supervision. Humans design strategies, decompose problems, and decide when to stop. AI executes, analyzes, and occasionally corrects itself. The developer's job is to separate planning (human thinking) from acting (AI doing).

Stopping conditions matter. We don't stop only when a test fails—we stop when the AI's path becomes overly complex, costly, or off-course. The human's role is judgment, not micromanagement.

Evaluating Rather Than Creating

Craig Larman once said humans will increasingly evaluate AI outputs rather than create from scratch. That may be true for senior developers who can already judge quality. But how do junior developers grow if AI does all the work? Evaluation is not a given—it's a learned ability that requires making mistakes, reflecting, and understanding deeply. If AI takes away the struggle, it also takes away the learning.

There Is No AI Moat

AI tools are commodities. Whatever you use, your competitors can too. The real advantage lies not in owning the tools, but in preserving and sharing human knowledge—keeping your codebase understandable, your architecture transparent, and your domain

knowledge living.

Instead of racing to build your own “AI agents,” invest in making your systems—and your teams—learnable. When the next breakthrough model arrives, buy it. But make sure your people and your products are ready to learn from it.

Closing Thoughts

Our work is not about automating developers away. It’s about augmenting human judgment.

We emphasize three principles:

1. Human-centric development – empathy drives the need.
2. Jidoka mindset – automation must know when to stop.
3. Feedback-driven work – seek meaningful feedback, not superficial progress.

AI can accelerate how far we go, but humans still decide where we’re going.