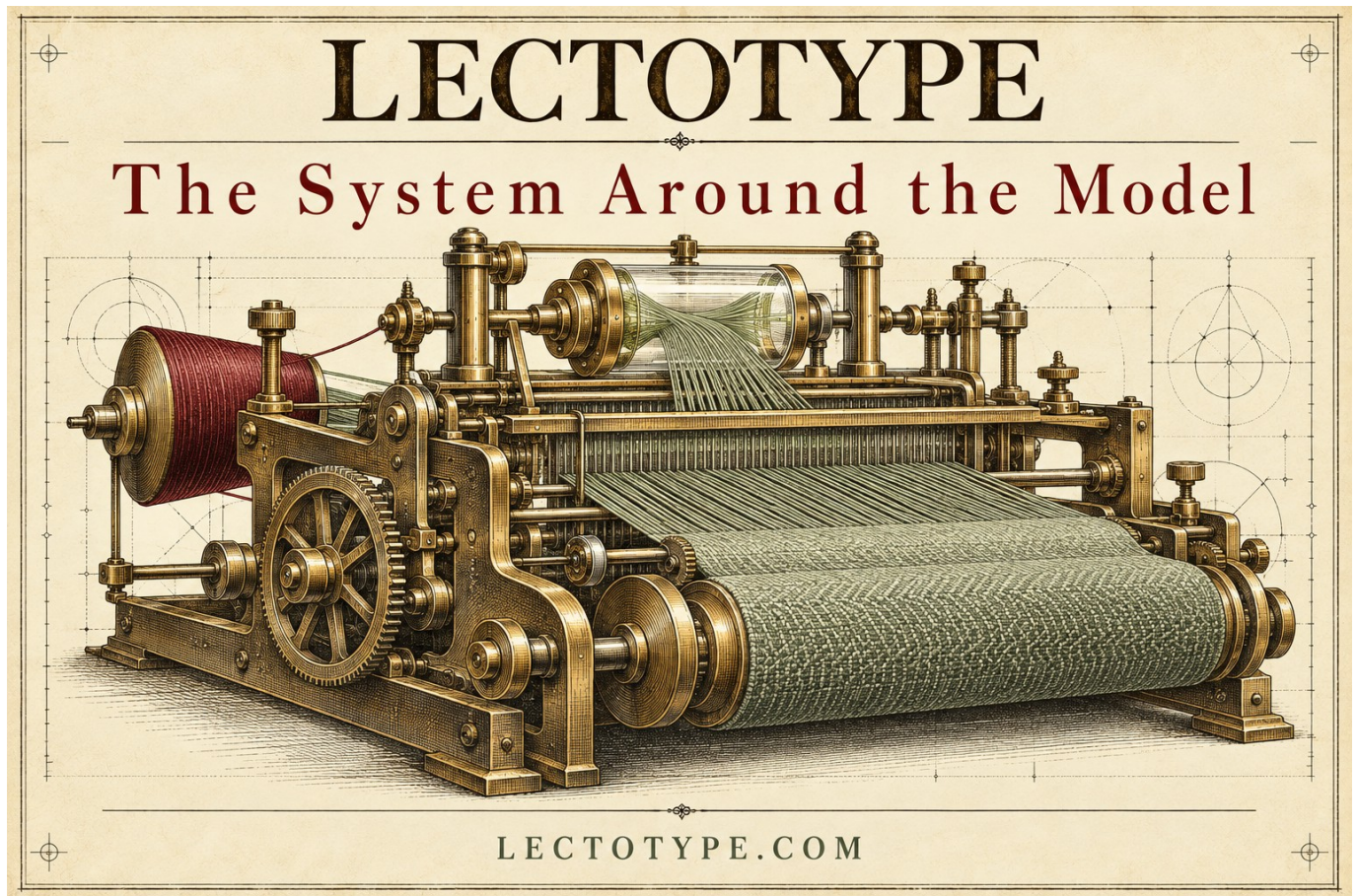




Agent builders create useful capability. Lectotype is developing the architecture around that work so clients can adopt it, measure it, and run it repeatedly.

David C. Robinson

CEO and Co-Founder, Human AI, Inc.

October 2026 · [Lectotype.com](https://lectotype.com)

The standard against which your AI is measured.

The System Around the Model

An agent builder can produce something impressive in a few days and spend the following months carrying it. The agent needs a source update, a client changes a rule, or an unexpected case requires a repair. Each deployment creates another set of operating responsibilities. A builder who wants a durable business has to solve the handoff as well as the initial build.

AI21 CEO Ori Goshen described the underlying problem in January 2024. Referring to LLMs, he wrote: *“They will never be as reliable as the enterprise demands without a system around them.”* His forecast emphasized evaluation and verification as enterprises moved into production. The people building models have been pointing to the surrounding system for years. [1]

Mistral CEO Arthur Mensch took the argument into business operations in his foreword to McKinsey’s agentic AI report. He called for agents integrated into core processes and wrote that *“openness, transparency, and control are central to your technology and implementation.”* His emphasis is useful for builders because a client buys a result within its own operating environment. The agent has to work with the company’s systems, decision rules, and people. [2]

That is the work we are organizing at Lectotype. We already have agents built with Claude Code and Codex running inside Lectotype workflows. We are developing the adaptation and operating process that surrounds them with approved evidence, defined steps, scoped credentials, checks, and human approval where required. The first client engagements will establish which integrations and operating features need to be completed for each production handoff.

A lead response workflow makes the opportunity concrete. An inquiry arrives, the system checks the customer record, prepares a relevant response, schedules the next step, and follows up on a proposal. Several of those operations follow a known procedure. A model can help interpret the request or draft the message, while ordinary software handles routing, dates, duplicate checks, and permitted changes. The builder’s domain knowledge determines how those pieces should work together.

Established engineering teams already combine model work with required software steps. Stripe’s published account of its Minions coding agents describes blueprints that include deterministic lint checks in the development loop. The check runs as part of the procedure before the agent’s branch is pushed. That placement is a useful design lesson for a business workflow: a required check should be part of execution rather than something the agent may decide to skip. [3]

The economics improve when the workflow can complete a useful sequence without repeatedly handing context back to a person. Research discussed by MIT Sloan examines how the arrangement of tasks and handoffs changes the value of AI automation. For a sales process, qualifying an inquiry, preparing a response, and scheduling a follow-up may form a useful chain. The business still retains approval over an exceptional price or an external commitment. [4]

Our commercial approach starts with a builder's existing client work. We will adapt one bounded agent workflow with the builder, define the client's acceptance criteria, and measure the operating result. The builder contributes the domain procedure and relationship with the client, while Lectotype contributes the governed workflow and adaptation work. Our partner program is being structured so the builder retains the client relationship and build fees, and participates in recurring subscription revenue.

The first engagement should produce evidence the builder can use in the next sale. We will compare representative cases with the client's current method and record acceptance, review time, interventions, and total operating cost. We will also plant cases that attempt an unauthorized action or rely on missing evidence. A useful workflow must complete legitimate work as well as stop cases that require permission, so unnecessary refusals belong in the results too.

Each adaptation will also show us where the product needs work. When engineers repeatedly build the same integration, repair the same transition, or prepare the same report, that work becomes a candidate for the next sprint. Reusable patterns will become templates as they are demonstrated across client cases. This gives development a commercial sequence: complete a defined client outcome, preserve what we learned, and reduce the effort required for the next deployment.

Builders who already understand a business process have a strong starting position. They know what the customer is trying to accomplish and which exceptions create the support burden. We want to work with those builders on one agent their clients already need. Bring the existing workflow and a client willing to define success, and we will establish the path from useful capability to recurring operation.

Sources

[1] Ori Goshen. [2024 Enterprise AI Forecast: From Pilot to Production](#). January 9, 2024.

[2] Arthur Mensch. [Foreword to Seizing the agentic AI advantage](#). June 13, 2025.

[3] Alistair Gray, Stripe Engineering. [Minions: Stripe's one-shot, end-to-end coding agents—Part 2](#). February 19, 2026.

[4] Kristin Burnham, MIT Sloan. [How AI is reshaping workflows and redefining jobs](#). April 22, 2026.