Tacit-Knowledge-Is-Your-Moat

Autonomy Platform · Internal Architecture Document

CONFIDENTIAL

Tacit Knowledge Is Your Moat. Here's the Math.

Why Berkeley's "next competitive moat" isn't a metaphor — and what it takes to actually fill it.

Trevor Miles | Azirella Ltd | May 2026

A short prior post first

Two months ago, *Your ERP Knows the Structure. It Doesn't Know the Planner.* made the case that supply chain planners have spent thirty years quietly being the system's missing semantic layer — what Knut Alicke calls the experiential ontology and what Pieter van Schalkwyk's three-layer stack calls the socio-technical dynamics layer. The diagnosis: every override is a senior planner encoding tacit knowledge into a number, and every system in the industry throws that knowledge away the moment the new number is saved.

This post takes the next step. **If overrides are the visible tip of tacit knowledge — the part that surfaces when a planner disagrees with the system — what about the much larger part that never produces an override?** And how do we know, formally, that capturing this stuff actually compounds into a durable advantage rather than evaporating into another knowledge-management binder?

The answer comes from forty-year-old mathematics, by way of a Berkeley business journal article published in March.

The moat thesis, made formal

Berkeley's California Management Review published *Tacit Knowledge Is Your Next Competitive Moat* this March. The argument is sharp: in a world where every public model has read the same internet, the durable difference between two operations teams is the experience that lives in the people who run them — and, increasingly, in the systems that have learned alongside those people.

This is not a metaphor. There is forty-year-old reinforcement-learning mathematics that says exactly when tacit knowledge constitutes a moat and when it is just folklore. The operative concept is the **value function**.

Imagine two manufacturers running the same ERP, the same APS, the same planning cadence. Tomorrow, both fire up the same off-the-shelf AI planning agent.

Company A treats every planner override as friction — a number on a dashboard to drive down. Tacit knowledge walks out the door at retirement.

Company B treats every override, every interview, every counterfactual walkthrough as training data for an operating substrate. Tacit knowledge gets externalized into the system that acts on it.

Eighteen months later, Company A's agent is exactly as smart as the day it was deployed. Company B's agent has internalized thirty years of *"I had a feeling about this lane in August."* The software is identical. The moat is the difference between the two.

This is what CMR means. The moat is not the AI. The moat is *the experience the AI has been allowed to inherit*.

Why the moat has historically leaked

Management theory has had a blueprint for tacit-to-explicit knowledge conversion for thirty years. Ikujirō Nonaka's SECI model — built on Michael Polanyi's foundational insight that *we know more than we can tell* — describes four modes of conversion:

- **Socialization** (tacit → tacit): two people work side by side; one absorbs the other's judgment without it ever being written down.
- **Externalization** (tacit → explicit): expertise is articulated into something a system can hold — words, schemas, rules, weights.
- **Combination** (explicit → explicit): articulated knowledge is recombined into higher-order structures.
- **Internalization** (explicit → tacit): explicit knowledge is acted on enough times that it becomes intuitive again.

The bottleneck has never been the theory. The bottleneck has been that explicit knowledge had nowhere useful to live. A binder. A wiki. A SharePoint site. Things nobody opens.

What is new in 2026 is that the substrate finally exists. Externalized knowledge can now live inside a system that acts on it directly, in production, in milliseconds. Which brings us to the math.

What's actually inside Autonomy

Strip away the marketing layer of any modern operations AI and you find the same formal object: a **Markov Decision Process**. A state space (the world), an action space (what the system can do), a transition rule (what happens next), and a reward (how good the outcome was). The system's job is to learn a *value function* — for each possible state, the expected long-run reward of being there if you keep acting well. From the value function, the right action falls out almost for free. This is what *value iteration* does. The canonical treatment is Chapter 4 of Sutton & Barto's *Reinforcement Learning: An Introduction*; Berkeley's CS188 §4.3 covers the same material in gentler form, with interactive visualizations.

Now consider what a senior planner has accumulated over thirty-one years. She has been observing states (your inventory positions, demand signals, supplier behaviour), taking actions (overrides, escalations, expedites), and getting feedback (what happened next). Each event has been a gradient update on a value function in her head. She does not think of it that way. But that is what it is. Her judgment *is* a value function. That is why it is hard to articulate. That is why a checklist of her practices does not reproduce her competence.

The Autonomy platform is built, at substrate level, as the same kind of object — operating over the same supply chain she has been operating over:

- **State** is the canonical state of the supply chain — sites, products, lanes, inventory, capacity, commitments — refreshed continuously from the ERP.
- **Actions** are the decisions the platform emits — buffer levels, lane assignments, urgency adjustments.
- **Transitions** are simulated by a digital twin, with honest uncertainty bands wrapped around every forecast instead of false single-point precision.
- **Reward** is the Balanced Scorecard — service, cost, working capital, sustainability, resilience — weighted to each tenant's strategy.

What emerges across millions of episodes — synthetic and real — is a value function we call **Experiential Knowledge**: the platform's accumulated, structured record of having lived through the supply chain it is asked to run.

This is the deeper claim CMR is making, restated formally: *experience is not a soft asset.* It is a mathematical object. Once you treat it that way, you can store it, version it, retrain on it — and, critically, you can extract it from a human and write it into the same store.

Four channels into the substrate

If the platform's competence is a value function built from trajectories, then the senior planner's competence is a value function built from her trajectories. Transferring her knowledge is not a matter of writing a procedure manual she will never finish. It is a matter of getting her trajectories — and her judgment about them — into the substrate the agents already learn from.

Map the four SECI conversion modes to four concrete channels:

1. Override-with-reasoning *(externalization).* Covered in detail in the prior post. Every override carries a free-text reason; an override classifier routes the highest-signal corrections back into the retraining loop.

2. Trajectory shadowing *(socialization, mechanized).* Recording the planner's decisions during the months before retirement — including the times she watches the agent act and chooses *not* to intervene. That non-intervention is signal too. The agent learns to act like she did, in the situations she handled — the same way a junior planner learns by watching a senior one, just with perfect recall and no shifts off.

3. Structured EK interviews *(externalization, deliberate).* A long-context language model running locally, on the customer's tenant, sits with a retiring planner for a few hours. *Here is a scenario. What would you do? Why? What would change your mind? What are you watching for that the system isn't?* Transcripts are paraphrased, structured, and converted into priors on Balanced Scorecard weights and guardrail directives. The planner does not write — she talks, the way she would to a junior who is shadowing her.

4. Counterfactual scenario walkthroughs *(externalization + internalization in one loop).* The digital twin replays a historical disruption, or invents a plausible new one. The planner makes the calls. Her decisions become labelled training examples — particularly for the kinds of compound disruptions where her judgment is most valuable and a fresh agent is most lost.

Note what each channel does in Nonaka's terms. Channel 1 captures the tacit at the moment it surfaces against an agent's recommendation. Channel 2 mechanizes the apprentice-shadowing-master pattern that has been the dominant tacit-knowledge transfer mode in industry for centuries. Channel 3 is structured externalization, a job a long-context language model is unusually well-suited to. Channel 4 closes the loop: the planner externalizes by deciding; the agent internalizes by training on what she decided.

The combination is the operational version of SECI's spiral. Each rotation deposits more knowledge into the substrate. The substrate compounds. The moat grows.

The window

CMR's framing is not just rhetoric. The Bureau of Labor Statistics projects 26,400 logistics openings annually through 2034, and the experienced cohort is retiring faster than organisations can backfill. Every quarter without a systematic capture mechanism is institutional knowledge lost permanently — not because there is not a market for it, but because the people who hold it are no longer in the building.

The optimistic version of the future is not that AI replaces those planners. It is that the math of value iteration finally gives us a way to *keep* them — not as a Slack channel of war stories, but as an operating prior, weighted into every decision the platform makes, indistinguishable in effect from their continued presence on the team.

That is what Experiential Knowledge means inside Autonomy. It is the formal claim that experience is something you can hold onto.

If your best planner is retiring this year, the question is not whether you can afford to capture her judgment. It is whether you can afford the eighteen months after, when the system she trained — by overriding it, by complaining about it, by occasionally being right about Dallas in August — is the one running without her.

Trevor Miles is the founder of Azirella Ltd and creator of Autonomy, a Decision Intelligence platform for supply chain planning. He previously held leadership roles at Kinaxis and i2 Technologies.

References: - California Management Review (Berkeley), *Tacit Knowledge Is Your Next Competitive Moat* (March 2026), cmr.berkeley.edu - Sutton, R. & Barto, A., *Reinforcement Learning: An Introduction* (2nd ed.), Chapter 4 — Dynamic Programming, incompleteideas.net/book/the-book-2nd.html - UC Berkeley CS188, §4.3 *Value Iteration*, inst.eecs.berkeley.edu/~cs188/textbook/mdp/value-iteration.html - Nonaka & Takeuchi SECI model — *Managing Knowledge in Organizations: A Nonaka's SECI Model Operationalization*, Frontiers in Psychology (2019), frontiersin.org - Bureau of Labor Statistics, *Occupational Outlook: Logisticians 2024-2034* - Prior post: *Your ERP Knows the Structure. It Doesn't Know the Planner.* (March 2026)