
# Supply Chain Planning Never Made the Switch

*Autonomy Platform · Public Blog*

PERSPECTIVE

Strategy August 2026

# Supply Chain Planning Never Made the Switch

Markets moved from discretionary to systematic decades ago: every position sized against a calibrated edge, every outcome attributed. Supply chain planning never made that transition, because its unit of work is a cell in a plan and a cell has nowhere to put a probability.

A piece has been circulating about why quantitative funds keep winning when the math they use is public. Bernoulli published in 1713. Bayes in 1763. Shannon in 1948. Markowitz, Kelly, Sharpe, Black-Scholes: all in textbooks, all free. Renaissance Technologies' Medallion fund returned roughly 39% annualised after fees from 1988 to 2018, on Gregory Zuckerman's reconstruction in *The Man Who Solved the Market*, and it was not built on a formula nobody else had.

The argument is that the edge is the wiring. Probability tells you how uncertain you are. Signal processing asks whether the pattern is real. Sizing decides how much loss you can survive. Portfolio theory asks how the bets interact. Optimisation turns constraints into action. Each layer covers the weakness of the one before it, and the composition is the thing that is hard to copy, not any single equation in it.

I think the argument is right, and I think it has almost nothing to do with hedge funds.

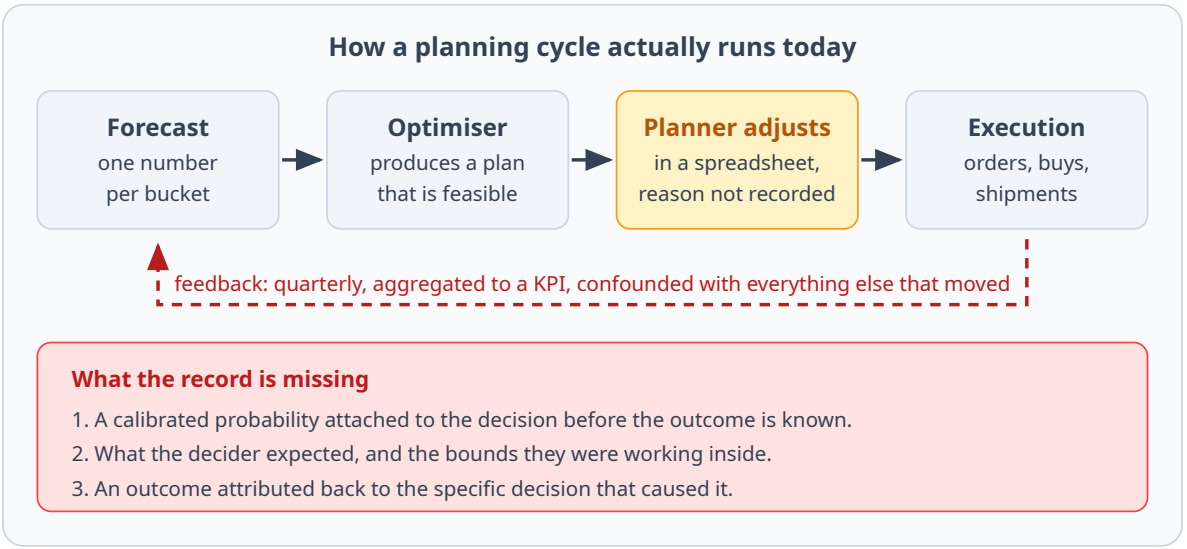
What actually happened in markets is a regime change that most industries have not made. Trading went from **discretionary** to **systematic**. Not from human to computer, which is the version people usually reach for, but from a practice where the decision function lives in someone's judgement to one where it is written down, measured per decision, and improved against evidence. Discretionary trading did not disappear because it was stupid. It lost because you cannot compound something you cannot measure.

Supply chain planning has not made that switch. It is one of the last large decision domains still run discretionarily, and it is the one where the switch is worth the most.

## What discretionary looks like when it is wearing an optimiser

Here is the objection I get first, and it is a fair one: planning is full of math. MRP, DRP, MEIO, linear programming, safety stock formulas with a service-level z-score in them. Some of it is fifty years old and very good. How is that discretionary?

Because the math is in the wrong place. It sits inside a step that produces a plan, and the plan is a grid of numbers. Look at what happens around it.




*A cycle full of mathematics that still cannot tell you whether any single decision in it was good.*

Notice what is absent. The forecast is a point, so nothing downstream knows how uncertain it was. The optimiser returns a feasible plan, not a scored one. The planner changes it, often correctly, and the reason evaporates. Then the quarter closes, service was 94.1%, and no one can say which of the ten thousand decisions inside that number helped.

That is the definition of discretionary. Not “a human did it”. **A decision was made, and the system retained no basis on which it could ever be judged.**

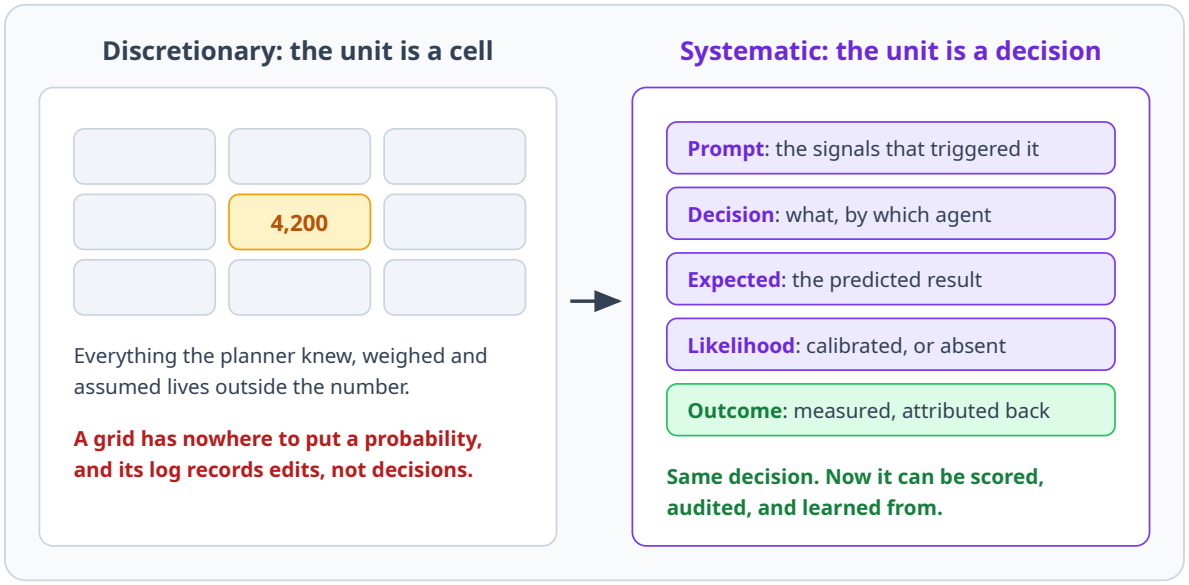
# The unit of work is the problem

The reason this persists is architectural, and it is worth being precise about, because it is not a feature gap that a vendor can patch.

An advanced planning system is organised around a plan: a grid of quantities, indexed by product, location and period. That grid is the primary object. Everything the system stores, versions and publishes is a set of cells. A cell can hold a quantity.

Systems in this class do record who changed it and when. There is a change log, and there are scenario versions, and both are usually good. What the log is keyed to is the **edit**, not the **decision**. So it will tell you that someone set 4,200 on Tuesday morning, and it cannot tell you what they expected that to achieve, how likely they thought it was, or what actually happened. You can bolt a confidence score onto a screen, but you cannot make a grid into a ledger, because a grid does not have rows that correspond to decisions.

Systematic trading did not win by adding confidence scores to a portfolio view. It won because the primitive changed. The unit of work stopped being a position and became a **trade with an attached thesis, size, and outcome**. Everything else follows from that: you can compute a hit rate, size against an edge, attribute P&L, and improve.




*The switch is not better math. It is a change in what the system stores as its primary object.*

This is the whole architectural bet behind Autonomy. The primary object is not a plan, it is a decision, and the plan is what falls out of a lot of them. Every decision row has a slot for each of those four fields, is hash chained since July so the sequence is tamper evident, and is queryable per decision rather than per period.

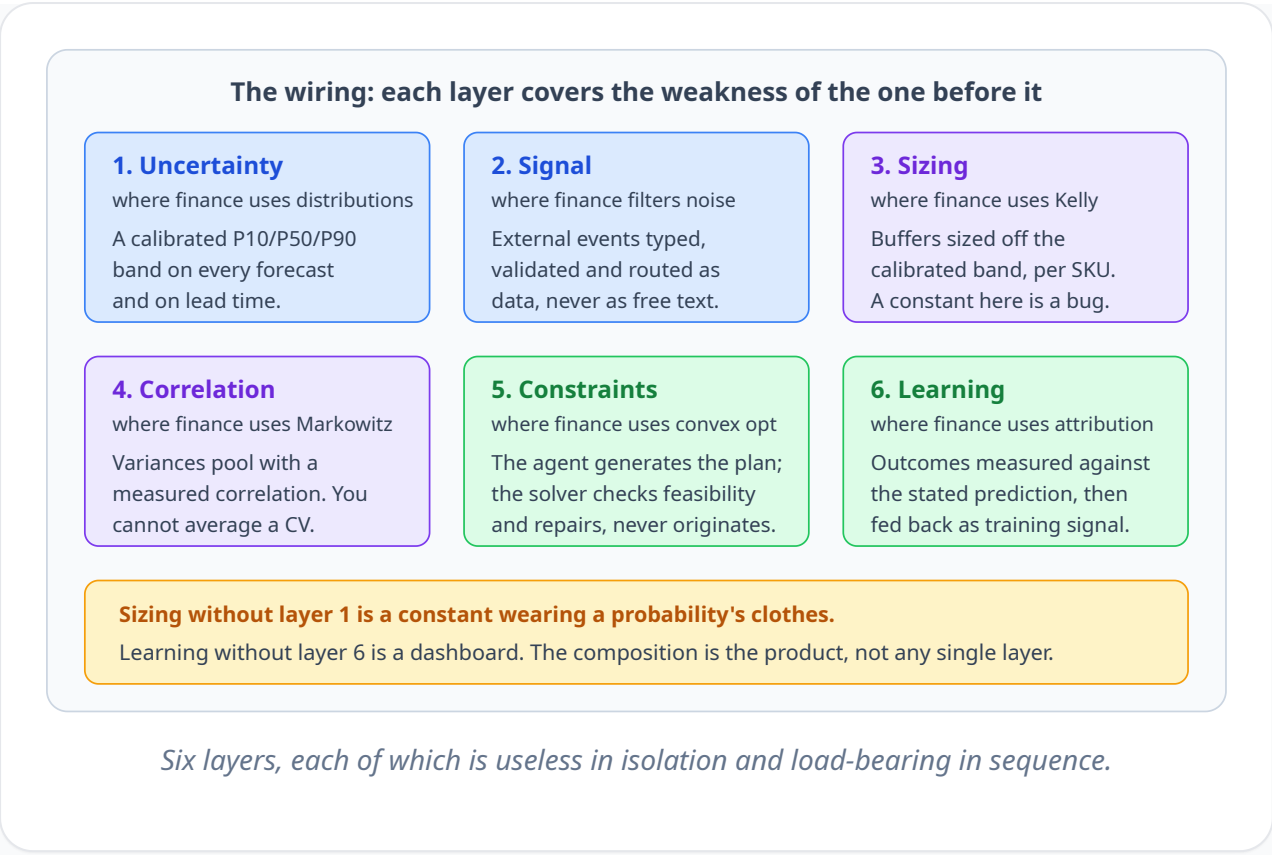
The fourth field is where we hold ourselves honest, and it is worth being exact about what it means. The likelihood is the conformal-calibrated number or it is nothing. A model's raw confidence output is not that number and never stands in for it: on several of our own agents that output is a head the training loss never touched, and it reads above 0.99 on most states, which is a thermometer that always says twenty degrees. So when a decision class has not been calibrated yet, the row carries no likelihood, and by contract an absent likelihood routes the decision to a human instead of being scored as a zero or filled with something plausible.

Today that is most of them. I would rather ship a system that refuses to show you a confidence it has not earned than one that shows you a number nobody computed.

We are also binding each row to an authenticated agent principal, so the record becomes non-repudiable rather than merely tamper evident. That part is genuinely in progress, and I would rather say so than imply it is finished.

## The layers, and why none of them work alone

Once the unit of work is a decision, the rest of the stack has somewhere to attach. This is where the quant-stack argument earns its keep, because it is a warning as much as a blueprint: Kelly sizing on top of an edge you have not verified is not conservative, it is a faster way to lose. Every layer is load-bearing for the next one.




*Six layers, each of which is useless in isolation and load-bearing in sequence.*

Layer 3 is where I can give you a real number, and it is one of ours going wrong. On one of our reference networks a lane capacity had been assembled from a genuine physical ratio multiplied by a level nobody had ever measured. Dimensionally impeccable, physically meaningless. It capped achievable customer fill at 0.78, at any inventory level, permanently; once we sized the lane off the demand it actually carries, the same network reached 0.98. Worse, while it was in place the scorecard had been reading its own unreachable service band as a reason to stop buffering. A month of agent conclusions had been measured through it.

The same shape turned up separately in our lane defaults as `` `19958.0 * 1.0` ``: a real 53-foot dry van payload, multiplied by a truck count nobody had counted. We find these because every planning parameter in the system has to declare where it came from: extracted from the source system, derived from the tenant's own history, calibrated as an explicit ratio times an explicit level, or flagged as unknown. A parameter that

cannot name its rung cannot be defended to a customer. Nobody grepping for hardcoded values finds `19958.0 * 1.0`, which is exactly why the rule names the two halves separately, so a missing half is visible.

I said above that a constant in the buffer path is a bug. One is open right now: the buffer still sizes against a single network-wide demand variability, while the forecast bands that should feed it vary by more than four times across products. The band is calibrated, it is on every forecast row, and it does not yet reach the buffer. That is the gap between having the layers and having them wired, and it is the honest state of layer 3 today.

That is what systematic actually feels like day to day. Not clever models. A system that is built to catch you.

## The three things that do not transfer

The analogy is useful, and it fails in three specific places. I would rather name them than let a prospect find them.

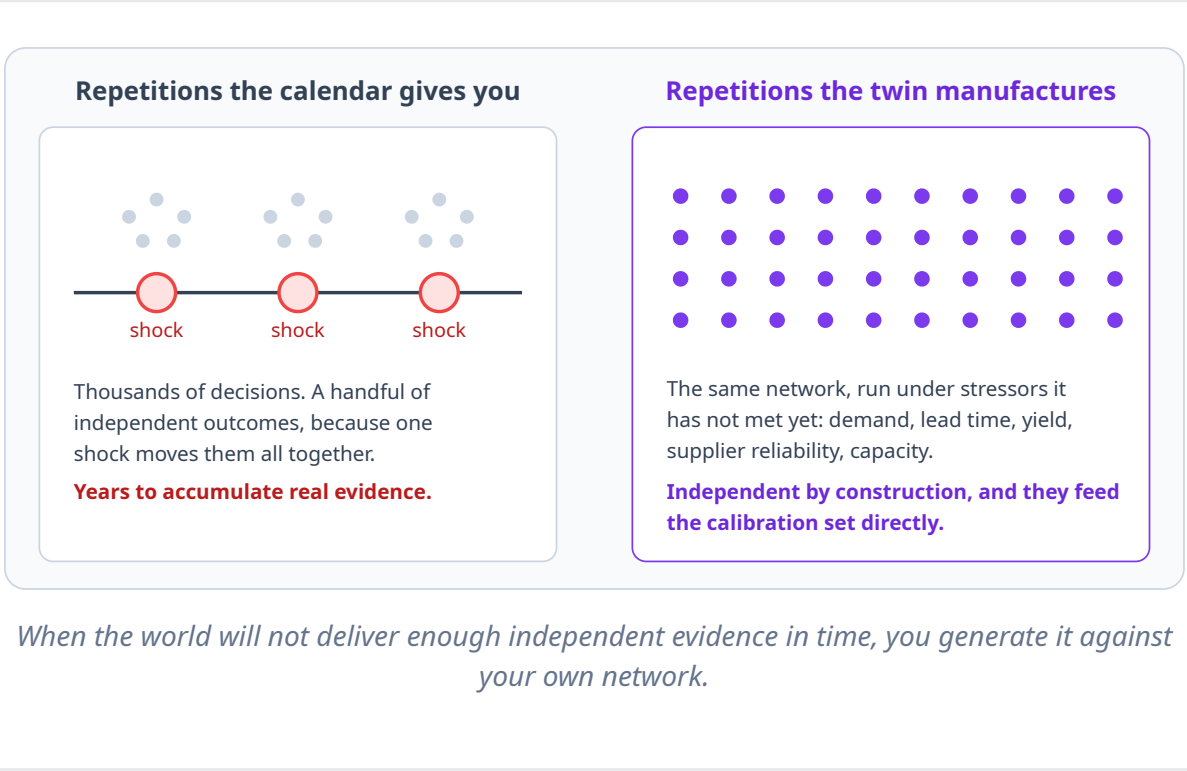
**We have no speed moat and no data moat.** A large part of the quant edge is latency, colocation and exclusive data. Supply chain planning has no meaningful latency race, and we deliberately give up the pooled-data flywheel: no model, prior, calibration or policy parameter is ever trained on, pooled from, or transferred between the real data of two customers. That is a rule in our codebase, not an aspiration. It costs us the thing most AI vendors lean on.

**The feedback signal is slow and confounded.** A trader gets marked to market daily against an unambiguous number. A planner finds out months later, and cannot separate “the buffer policy worked” from “demand was

mild.” This is why causal attribution is the layer I would not ship without, and it is also the one where our own wiring is least finished: measuring every impact edge interventionally is specified and building, not done. Without it the loop in the first diagram does not close, it just runs faster.

**Raw decision count overstates the evidence.** The tempting version of this argument is that a planner makes thousands of small decisions a day, so the law of large numbers should do its work. It does not, because those outcomes are heavily correlated: one demand shock moves every product at once. Thousands of decisions can contain a handful of independent tests. This is the same mathematics as layer 4 in the diagram above, applied to your own evidence base, and it is the objection a quantitative CFO will raise first.

So where do the repetitions come from?




*When the world will not deliver enough independent evidence in time, you generate it against your own network.*

That is what the digital twin is for, and it is the reason it sits in our architecture as a peer of the agent layer rather than as a scenario toy. The agents are pre-trained generically, against synthetic networks under stress, with no customer's data anywhere in that step. They are then tuned on the individual customer's own network topology, and the resulting bands are calibrated on data the calendar has not produced yet. It is the same move Monte Carlo plays in the quant stack, pointed at a different problem: not pricing an option, but manufacturing the repetitions that make a small edge legible.

That split is deliberate, and it is what makes the previous paragraph's privacy claim survivable. Generic weights trained on no customer's data are the one thing that moves between deployments. The tune never leaves the tenant.

One deliberate difference: we do not Monte Carlo the plan itself. Uncertainty enters the plan through the calibrated band, once, in a form you can audit. Sampling the plan gives you a distribution of plans and no way to explain any of them to a planner at seven in the morning.

## The part everyone skips: discipline

The quant argument ends somewhere unfashionable. The math is old. The reason most people cannot use it is that they will not follow it under pressure. They quit in week three because a real edge feels bad early, or they double their size after a good run.

Supply chain has been running its own version of this experiment for thirty years, and it has a name: the planner overrides the optimiser, the plan loses credibility, and the six-figure APS becomes a spreadsheet feeder.

Every vendor in this category knows the pattern. The usual responses are both wrong. Locking the planner out fails because planners frequently know something the model does not, and being right while ignored is how you lose a team. Leaving the override unmanaged fails because you learn nothing from it.

Our answer is to make the override a first-class, measured act. Agents decide and act inside declared bounds. They inform when a decision crosses a policy boundary or when calibrated confidence is low and urgency is high. A person can inspect any decision and get the four fields back: what prompted it, what was decided, what result was expected, and how likely that result is. And a person can override, which is not an undo. It is a new decision, made by someone with information the agent did not have, and it is recorded as such, scored against what actually happened, and fed back into training.

That is the discipline layer. It does not ask a human to trust a system blindly, which is a thing no operator should ever do. It asks the system to earn it, one measured decision at a time, and it keeps score in both directions.

## What this is worth

If you accept the framing, the prize is not a better forecast. It is a change in what compounds.

In a discretionary regime, improvement comes from hiring better planners and running better meetings, and it leaves when they do. In a systematic regime, every decision the network makes is a small, recorded experiment with a stated prior, and the operating policy improves whether or not

anyone remembers why. That is the actual reason Medallion is the example everyone reaches for. Not the returns. The fact that the returns came from a process that got better at a rate a human organisation cannot match, because the process could see itself.

I will hold myself to the standard this argument implies. Claiming a systematic regime obliges you to produce measurements, and the honest position today is that our substrate carries the calibrated band, the decision ledger, the twin, the causal attribution layer and the governed override, while several of the compositions between them are further along in some planes than others. I have named three of those gaps in this post rather than leave them for a prospect to find, and I would be sceptical of anyone claiming a uniformly wired stack, including us. What I will say is that the primitive is right, and that is the part you cannot retrofit.

The equations were never the moat. In markets that was true in 1988. In supply chain it is true now, and the switch has not happened yet.