



Achievability, and what fragility actually measures

Autonomy Platform · Public Blog

PERSPECTIVE

© 2026 Azirella Ltd. All rights reserved worldwide.

[Read more at azirella.com/blog](https://azirella.com/blog)

TL;DR

In September I wrote that the likelihood of realization was my measure of fragility, and that was the wrong word for it: a probability says how often a plan breaks, while fragility in Taleb's sense is about how steeply it degrades as the error grows. So the probability gets its own name, Achievability, and fragility becomes the slope that sits beside it. Two plans can share an Achievability of 0.92 and be forty-seven points apart once the error grows by half.

KEY POINTS

- "Likelihood of realization" is renamed Achievability: scored before you commit, the share of recorded error patterns under which a plan would meet every commitment it makes. One word for the probability, and it is not the same word as the one for how fast it falls.
- Fragility becomes the slope of Achievability under stress, which is the question a single probability cannot answer. Two plans at 0.92 can be forty-seven points apart at 1.5 times the recorded error, and a scorecard reading only the first number ties them.
- No plan can be antifragile. The slope is non-positive by construction, so Gartner's third state belongs to the loop that replaces plans rather than to any plan inside it, and that gives antifragility a form you can actually measure.

- Both numbers are a rank over stored error patterns, which is a lookup rather than a simulation. Decision Achievability is live; the per-plan measure and the slope are filed as Autonomy-Core §3.669 and not built.

In September I wrote that [optimization is also too confident](#), and that a plan leaves two questions unanswered. I said I wanted to be precise about the words. I then used one word for two different things, which is the opposite of that.

Here is the sentence I would withdraw: “This is my measure of **fragility**.” I was describing the likelihood of realization, the probability that a plan’s expected result actually comes true. Later in the same piece I used fragility again, this time in Taleb’s sense, where a thing is fragile because the damage grows faster than the error. Those are two different properties of a plan. They can disagree. A plan can be very likely to hold and catastrophic in the rare case it does not, and the word I was using cannot tell you which of those I meant.

So this post does two things. It gives the probability its own name, and it gives fragility back to the thing Taleb used it for.

Achievability is the probability

Achievability scores a plan before you commit to it: the share of recorded error patterns under which it would meet every commitment it makes. It runs from zero to one. Everywhere the site has said “likelihood of realization”, read Achievability from here on.

The rename is not cosmetic. “Likelihood of realization” is four words that a planner has to decode before they can use it, and in practice it got shortened in conversation to “the likelihood”, which is a word that already means six other things. Worse, naming the probability *fragility* inverts its sign: a high likelihood of realization is good news, and a high fragility should be bad news, so the term was fighting the number it described. Achievability moves in the direction people expect. Higher is better. A planner can read it off a screen without being taught.

The name now covers two scopes, and keeping them apart matters more than it sounds. **Decision Achievability** is what every agent decision already carries: the calibrated probability that this decision delivers the outcome it predicted, sitting on the

audit row beside the prompt, the decision and the expected outcome. **Plan**
Achievability is the one this post is about: the probability that a whole plan would hold every commitment it makes at once.

The second is not an average of the first, and it must not be built as one. A plan that holds its fill rate and blows its capacity has not delivered, so the question is joint, and a joint answer is lower than any single commitment's own probability. Both scopes are set out on [the Achievability page](#).

A plan is answerable for the commitments it made

That phrase is doing work, so let me be exact about it.

A plan commits to things. A fill rate it will hold, a capacity it will not exceed, a date a customer was given, a cash position finance is relying on. Each of those is a threshold the plan either respects or breaches. Achievability asks one question across all of them at once: in what share of the error patterns we have on record does the plan respect every single one?

Every one matters. A plan that holds its fill rate and blows its capacity has not delivered. This is why the number is not an average of per-commitment probabilities: it is a joint question, and the joint answer is lower than any individual one, often much lower.

The number comes from the record, not from a simulation

Take the errors the business has actually made. Not a distribution fitted to them, not a distribution someone chose: the realised misses, one row per period, with every quantity's error in that row sitting together exactly as it occurred.

That togetherness is the whole mechanism. When demand ran hot in the same week a supplier ran late, that row holds both. Applying the row to a candidate plan reproduces the correlation without anyone having to model it, estimate it, or assume it away. Supply chain errors move together, and a method that takes them apart and reassembles them has already lost the thing that makes them dangerous.

Apply each stored row to the plan, ask whether every commitment survived it, and count. The share that survived is the Achievability. It is a rank in a sorted list: at decision time, a lookup, not a solve. That is the same move [conformal prediction](#) makes on a single quantity, carried up to a whole plan.

This is not Monte Carlo, and the distinction is not a quibble. Nothing is sampled, no future is generated, and no random number is drawn. Every row is used exactly once, so the answer is the same every time it is computed and reproducible from the stored record months later. We do not Monte Carlo a plan, and nothing here changes that.

Two plans at 0.92

Now the reason one number is not enough.

Picture two plans for the same network. Both score 0.92: in nine-and-a-bit of every ten recorded error patterns, both hold every commitment they made. A scorecard reading only Achievability ties them, and a planner would reasonably take either.

Now ask what happens if the business makes somewhat larger errors than it has been making.

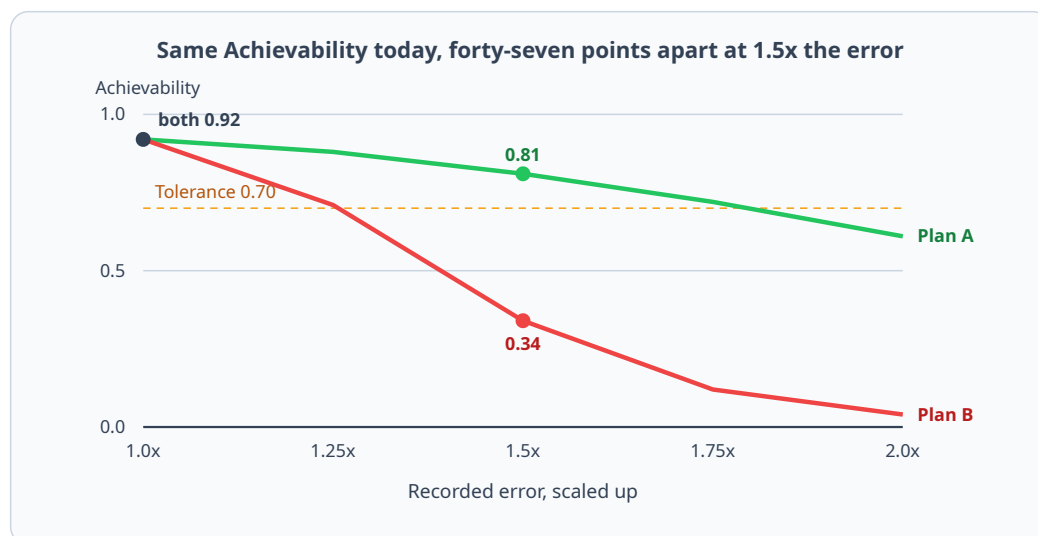


Illustration. The figures are invented to show the shape. Plan A gives up eleven points by the time the error is half again as large; Plan B gives up fifty-eight.

Plan A loses eleven points. Plan B loses fifty-eight and is below any tolerance a business would set. They were indistinguishable on the number the scorecard read, and they are not remotely the same plan. Everything that separates them lives in how fast the line falls, and no probability, however well calibrated, carries a rate.

Fragility is the rate

Fragility is how fast Achievability falls as the recorded error is scaled up. Plan A is robust. Plan B is fragile. The word now means what it sounds like it means.

There is a second reading of the same curve that planners find easier to argue with, and I prefer it on a screen: the multiplier at which the plan drops below tolerance. Plan A survives errors about eighty per cent larger than the business has been making. Plan B survives about thirty per cent larger. That is a sentence a planner can take to a supplier review, because the thing it is denominated in is a thing they have opinions about.

This is also, directly, the stress test Gartner's supply chain practice has been recommending for two years. Tim Payne's argument for antifragility leans on probabilistic planning and stress testing as the capabilities that separate the firms that gain from uncertainty from the ones that merely survive it. A stress test that returns a pass or a fail against one named scenario has thrown away most of its own result. The slope is the same exercise, reported as a rate.

Where I have to be careful: scaling is not a forecast

Scaling the recorded errors by a half manufactures error magnitudes the business has not actually seen. I should say that plainly rather than have it said for me, because it sits close to something this site has ruled out.

The rule we hold to is that you do not attach a likelihood to a disruption scenario, and **I still think that is right**. Nothing here attaches one. The multiplier is a dial, not a prediction: the curve says *if errors were half again as large, this is what would happen to the plan*, and it is silent on how likely that is. Read it as a sensitivity of the plan, never as a probability of a shock. The moment someone writes "there is a twelve per cent chance of a 1.5x error environment" on a slide, the instrument has been misused.

There is a stricter version available, and I am not yet sure it is not the better one: instead of scaling, recompute Achievability on the worst recorded windows. The roughest quarter the business has actually had, rather than an invented multiple of its average. That uses only realised data and gives up nothing to the objection above. It is lumpier, it gives you a handful of points rather than a smooth curve, and it cannot reach past the worst thing that has happened to you. We have not decided between them.

Taleb's fragility is about curvature, not slope

Having taken the word back, I have to admit the slope is not quite what Taleb meant either.

In *Antifragile* (2012) a thing is fragile because the response accelerates: a mistake twice as large costs far more than twice as much. That is a claim about the bend in the curve, not its tilt. The slope is the first thing you notice; the acceleration is the Talebian part.

It matters because two plans can share a slope and behave differently. One gives up its Achievability steadily across the whole range. The other holds almost flat and then falls off a cliff just past where the business normally sits, which is the shape that ends careers, and it is the shape the missing slack produces: while the error is inside the buffer it costs nothing, and past the buffer every unit is paid at the worst price available.

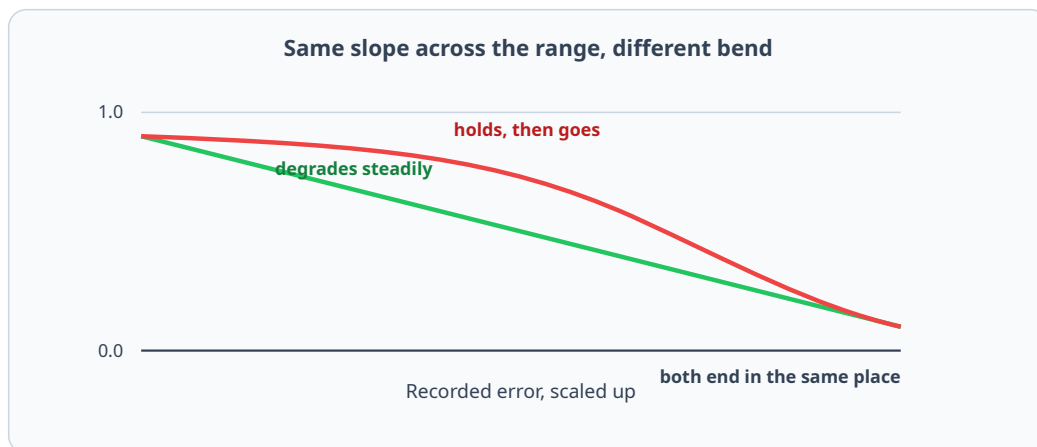


Illustration. Both plans lose the same Achievability over the full range, so they share an average slope. Only the red one is fragile in Taleb's sense.

So the honest position is that the slope is the right default to report, because it is what separates plans a scorecard would otherwise tie, and the bend is the refinement. Which of the two reaches a screen is not settled.

Gartner's three states describe an organisation, not a plan

Gartner sorts supply chains into three states: fragile, which loses value when exposed to uncertainty; resilient, which holds its value; and antifragile, which gains from it. The term comes from Taleb. It is a useful frame and a lot of readers now arrive carrying it, so it is worth being clear about where these two numbers sit inside it.

They sit at a different level. Gartner's states are properties of an organisation. Achievability and Fragility are properties of one plan. A resilient organisation can hold a fragile plan, and most do, routinely, without any way of knowing. That gap is most of the reason a per-plan measure is worth building at all: the organisational diagnosis does not reach down to the thing anyone actually executes.

No plan can be antifragile

This follows from the arithmetic, and I think it is the most useful thing in this post.

Antifragility is gain from disorder: a positive response to stress. For a fixed plan against a fixed set of commitments, the slope cannot be positive. Scaling the errors up cannot increase the share of error patterns under which every commitment holds, because every pattern that breached a threshold at a smaller scale still breaches it at a larger one. The best a plan can do is a slope of zero, which means nothing it committed to was ever close to binding.

An antifragile plan is not difficult. It is unavailable.

What can be antifragile is the loop that replaces plans. If the system learns from each cycle, then the plan it produces next month, measured at the same stress level, should be more achievable than the one it produced this month. That is a real claim, it is falsifiable, and it is measurable with the instrument already described: hold the multiplier fixed, plot Achievability across successive planning cycles, and see whether the line goes up. A system whose plans get better under stress it has already experienced is gaining from disorder in a sense Taleb would accept and a CSCO can audit.

I have not seen anyone put a number on antifragility. This is the number.

The objection I would raise against myself

Achievability is a single scalar, and this site has argued against single scalars, in [the post about one number and two kinds of not knowing](#). A 0.5 can mean the lane really is a coin toss, or it can mean we have six weeks of history on a site we have never seen. One number cannot carry that, and I still think that is right.

Fragility is part of the answer but not all of it. A plan scored against a thin record will tend to show a steep slope, because a handful of stored patterns cannot populate the middle of the curve, so the two numbers do partially separate the cases. They do not separate them cleanly. A short record and a genuinely volatile network can produce the same pair.

The rest of the answer has to be the resolution of the measure itself. With two hundred stored patterns, Achievability cannot be finer than about half a percentage point, and anything beyond that is the interface inventing precision the record cannot support. A screen showing 0.996 from a two-hundred-row record is lying, politely. The number has to arrive carrying how many patterns it was computed from, and the interface has to refuse to render digits it has not earned.

There is a second problem I have no clean answer to. Picking the best plan from a set biases the winner's Achievability upward: some of what made it score highest is that it happened to suit the stored patterns. The fix is to re-score the chosen plan against a second, held-out slice of the record before committing to it, which costs nothing to run and is easy to skip. I expect it to be skipped unless it is a gate rather than a report.

What is built

Decision Achievability is real. Every agent decision carries a calibrated score and a provenance beside it, and the provenance is declared by the producer rather than inferred, so a score that came from a heuristic rather than a trained head is labelled as one instead of passing quietly as calibrated. That is the half that already works.

Plan Achievability does not exist, and neither does Fragility. I checked rather than remembering: there is no per-plan scoring in the codebase at all. Everything in this post from the joint count onwards is a design. It is filed as the top of our near-term work order, and the rename alone is 2,265 occurrences across 228 files and three database columns, which is the honest reason it has not happened yet rather than an excuse for it.

So the vocabulary is ahead of the code, deliberately, and I would rather say which half is which than let a reader assume the better answer. Nothing here has been run against a real quarter of a real company's data. Until it has, it is architecture.

Achievability says whether the plan holds. Fragility says how hard you would have to push it. Neither is the other, and I should not have used one word for both.

See Autonomy in action

Walk through how Autonomy models, executes, monitors, and governs supply chain decisions with autonomous AI agents.

[See It Live](#)