ARCHITECTURE NOTES · RESOURCE

Why most AI rollouts collapse at month three

A diagnostic companion. Voice drift, model regressions, brief decay, reviewer fatigue. The four mechanical failure modes the article names, and the contract that fixes all four.

Companion to: Why most AI rollouts collapse at month three.

AI rollouts succeed in week one and die quietly between week eight

The pattern is predictable, and so is the fix. The article names four failure modes that compound until the team rewrites every output and quietly stops shipping. This deck walks each one with the diagnostic signals to watch, and lands on the single fix that addresses all four. Use it as the structured companion to the article, with the framework expanded for an operating team.

- For content leads and AI program owners past the first sixty days of a rollout.
- Each failure mode has a specific signal, a specific compounding curve, and a specific reason scanners miss it.
- The fix is the same for all four: written contracts that automate the discipline reviewers cannot sustain.
- No invented metrics. The article is short. This deck stays inside it.

The four failure modes, in the order the article ranks them

01

1. Voice drift

The most expensive of the four, per the article. Output reads competent for a few weeks. Then someone scrolls back and notices the brand has flattened. Nobody can point at a single bad post. The cumulative effect is erosion.

02

2. Model regressions

The provider releases a new model. The prompt that worked last month produces noticeably worse output. Scanners catch nothing because they were calibrated against the old behavior. The team blames the brief.

03

3. Brief decay

The schema your team built in week one slowly fills with shortcuts. Fields go blank or get padded. The quality of the brief drops below the quality of the draft, and the draft follows.

04

4. Reviewer fatigue

The deadliest of the four, per the article. The reviewers who caught problems in week one stop catching them in week ten. Each individual review still happens. The discipline behind it does not.

How week one looks versus how week ten looks

Week one - when the rollout is alive

- Voice in the output is recognisable as the brand.
- Prompts produce stable output across runs.
- Briefs are filled out with specific hooks and key points.
- Reviewers catch failures and the team fixes the contract.
- The team ships on cadence and feels good about it.

Week ten - when the rollout is collapsing

- Output reads competent and forgettable. Brand has flattened.
- A new model release shifted behaviour and nobody adjusted.
- Brief schema has shortcuts in three of seven fields.
- Reviewers approve drafts that would have been rejected at week three.
- The team rewrites every output and quietly stops shipping.

The single fix the article names: written contracts

01 **A voice contract that scans every output**

The voice drift failure mode is invisible to humans because nobody can point at one bad post. The contract is the artifact that flags the cumulative drift before week ten arrives.

02 **A model regression alert that does not depend on calibrated scanners**

When the provider ships a new model, the contract has to flag the behaviour shift. Scanners calibrated on the old behaviour will not.

03 **A brief schema that refuses incomplete inputs**

Brief decay starts with shortcuts in fields. The schema refuses the run if the brief has not been completed against the contract.

04 **A reviewer process that does not depend on willpower**

Reviewer fatigue is the deadliest failure mode because it compounds silently. The contract automates the discipline so the reviewer is not the load-bearing layer.

A month-three rescue pass, in order

- **Pull the last thirty drafts and read them in one sitting.**
  Voice drift is invisible per-draft. It is visible across thirty drafts in a row.
- **Diff the current model behaviour against the original calibration set.**
  A new model release that nobody noticed is the most common cause of a sudden quality cliff.
- **Audit the briefs that fed those thirty drafts.**
  Score completeness against the schema. Find the fields that have started filling with shortcuts.
- **Re-read the reviewer notes from week three and from last week.**
  Reviewer fatigue shows up as shorter notes and fewer rejections. The pattern is unmistakable in the diff.
- **Write the four contracts before shipping again.**
  Voice scan. Model regression alert. Brief schema gate. Reviewer process. The article is explicit: the fix for all four is the same.
- **Re-baseline the calibration set for the new model.**
  Scanners only catch what they were calibrated against. Recalibrate after every model release.

Anti-patterns we see at month three

What teams reach for first

- Hiring a prompt engineer to fix what looks like prompt rot.
- Swapping models in the hope the new one is sharper.
- Adding a sixth reviewer to catch what the first five are missing.
- Rewriting individual outputs by hand and shipping them anyway.
- Quietly extending the cadence until it stops slipping.

What the article says works instead

- Writing the voice contract before touching the prompt.
- Re-baselining scanners against the new model behaviour.
- Automating the discipline reviewers cannot sustain.
- Refusing to ship until the brief schema is complete.
- Treating the four failure modes as one problem with one fix.

Signals to watch from day one of the rollout

VOICE DRIFT

Cumulative

Read thirty drafts in a row on a recurring cadence. Per-draft inspection misses it.

MODEL REGRESSION CADENCE

Every release

Re-baseline scanners after every provider model release.

BRIEF COMPLETENESS

Schema-gated

A run does not start until the brief passes the schema check.

REVIEWER ENGAGEMENT

Tended

Note length, rejection rate, and time-per-review tracked across weeks.

NEXT STEP

The fix for all four failure modes is the same

Written contracts that automate the discipline reviewers cannot sustain. Voice, model, brief, reviewer. Build the contracts before week eight. Hold the cadence past

Read the full architecture note ->