



The Eight Ways Rollouts Die

Weekly-pulse reading. Each entry: the symptom, the mechanism, the countermove. Most dying rollouts exhibit #1, #3, or #6 — check those first.

1. The capture tax sneaks back in

Symptom: solver effort per capture creeping past a minute; agents "filling in the draft a bit" before flagging. **Mechanism:** someone well-meaning adds required fields, a pre-flag form, a quality note — each tiny, jointly fatal. The methodology died of this once already. **Countermove:** audit the capture path monthly against the 30-second cap; anything the AI or the curator could do instead, move to them. The charter's kill criterion exists for this.

2. The gate becomes a bottleneck

Symptom: queue age past a week; agents saying "I flagged that ages ago." **Mechanism:** capture volume outruns curator hours, usually because the trigger threshold is too eager, not because knowledge doubled. **Countermove:** in order — raise the capture threshold, add a curator in the flooding domain, batch dispositions daily. Never lower the quality bar to move the queue (that's death #3 wearing a fix's clothes).

3. The gate becomes a rubber stamp

Symptom: acceptance rate $\geq 95\%$ with edit-distance ~ 0 from week one; dispositions in seconds. **Mechanism:** overloaded or under-trained curators optimize for queue-zero. Wrong content enters the record with a human signature on it — worse than no gate, because now it's *trusted*. **Countermove:** re-run training candidate #4 (the fluent-but-wrong draft) unannounced; watch reject-reason telemetry — a gate that never rejects isn't judging. Fix load per #2.

4. Metrics theater

Symptom: dashboard full of activity numbers (articles created! candidates processed!) and no economics; nobody outside the team could say what the program saves. **Mechanism:** activity is easy to count and comfortable to report; cost per answer requires the channel inventory work. **Countermove:** the workbook's dashboard, sent monthly, six numbers, blended CPA first. If you can't fill the Channel Inventory, that — not more activity — is the week's priority.

5. The ungoverned bot bolt-on

Symptom: someone ships an AI answer surface "in parallel" reading the raw KB, because the governed loop "wasn't ready." **Mechanism:** demo pressure. The bot confidently cites the stale fifth of the corpus; the first visible wrong answer becomes the program's reputation. **Countermove:** the delivery rule in the content-

standards policy (validated-only, citations, refusal floor) applies to *every* surface, including experiments. Offer the impatient team the governed retrieval path — it's usually days, not months, at pilot scope.

6. The old corpus rots while the new loop shines

Symptom: new candidates flow beautifully; meanwhile the legacy 400 articles sit untyped, unowned, increasingly wrong — and retrieval still serves them. **Mechanism:** new-shiny bias; migration feels like a chore with no demo moment. **Countermove:** the metadata-extraction prompt makes migration a batch-review job, not a rewrite; retire the dead fifth early (it's the fastest trust win available); until migration completes, exclude unmigrated content from machine delivery.

7. The inversion panic

Symptom: month 4–6: "TTR is up, FTF is down — is this thing backfiring?" asked by someone with budget authority. **Mechanism:** shift-left worked; the easy volume left the assisted queue; nobody warned leadership. **Countermove:** prevention only — the Inversion Brief, delivered in month one, before the trend exists. If you're reading this mid-panic: send it now, with the known-vs-new mix chart showing *why* the queue got harder.

8. The hero dependency

Symptom: one person (usually the knowledge lead) is the gate, the workbook, the policies, and the training. Their vacation stalls the loop. **Mechanism:** competence concentrates; nobody notices because everything works. **Countermove:** two trained curators minimum from week 3 (the training doc exists so this is cheap); policies ratified and written down, not remembered; the monthly ritual documented in the workbook README. The system's whole premise is that knowledge shouldn't live in one head — including knowledge about the system.

Contextkeeping is a KCS-aligned operating model. KCS® is a service mark of the Consortium for Service Innovation; this material is not affiliated with or endorsed by the Consortium.