



• ARCHITECTURE NOTES · RESOURCE

Rebuild vs. iterate

How to read the maintenance log and decide whether your automation needs a patch or a fresh brief.

Companion to: *When to throw out an automation and rebuild it from the brief.*

Why this decision is the most expensive one in operations

An automation that fit the company at fifteen people quietly stops fitting at fifty. The flow still runs, the dashboards still light up, and the operator still ships. What changes is the cost of every additional change. Iterating a workflow that should have been rebuilt is the most expensive mistake in operations because the maintenance debt compounds every month and never appears as a single line item. This deck gives you the signals, the read of the maintenance log, the rebuild brief, and the trade-offs so you can call rebuild or iterate with confidence and defend the call to the rest of the team.

- How to tell rebuild from iterate using the maintenance log alone
- How to write the rebuild brief from current company size, data, and owner
- The trade-offs and the four anti-patterns that mask a needed rebuild
- Five signals to watch every quarter so the call gets easier over time



The four lenses on every running automation

01 Company size at the time of the brief

Every automation encodes assumptions about headcount, span of control, and review capacity. A flow built for a fifteen-person team usually assumes one reviewer, one queue, one owner. At fifty people those assumptions collapse silently and the patches start.

02 Data shape at the time of the brief

The original brief locked in a schema: which fields existed, which were optional, which downstream system owned the truth. When the data shape changes around the flow, patches accumulate as branches and special cases that the brief never anticipated.

03 Current owner and the review surface

The original owner often left, got promoted, or now owns three other systems. A flow only stays healthy when one person can hold the whole brief in their head. When ownership has drifted, the next change tends to be a patch on a patch.

04 The maintenance log itself

The log is the highest-signal artefact you have. Five edits in a quarter is loud. Four of them clustered around the same module is louder. The log tells you whether the architecture is wrong or the patches are wrong, and the answer is almost always the architecture.



Rebuild signal vs. iterate signal

Rebuild signal

- Five or more edits in the last quarter
- Four of those edits clustered on the same module
- Original owner is no longer the day-to-day operator
- Company size has roughly tripled since the brief
- New patches contradict assumptions in older patches
- No single person can describe the full flow end to end

Iterate signal

- Workflow runs reliably across the last quarter
- Operator has a clear, specific improvement in mind
- The change is local to one branch of the flow
- Original brief still describes what the flow should do
- Owner is the same person who shipped it last time
- No outstanding patches contradict each other



Illustrative read of a maintenance log

EDITS LAST QUARTER

5

Threshold cited in the article. Above this, treat the flow as a rebuild candidate by default.

EDITS ON ONE MODULE

4 of 5

Cluster signal. The architecture is wrong, not the patches. Stop patching and re-open the brief.

HEADCOUNT AT BRIEF

15

Illustrative anchor from the article. Re-read every assumption the brief made about review and ownership

HEADCOUNT TODAY

50

Illustrative anchor. Span of control, queue volume, and reviewer capacity all break around this



Rebuild runbook: from log to new brief

— **1. Pull the last four quarters of edits on the workflow**

Count them, cluster them by module, and tag each as patch, feature, or bug fix. The pattern is what you are reading for, not the totals.

— **2. Name the original brief out loud**

What was this workflow supposed to do, at what company size, with what data, and for which owner. If you cannot reconstruct it, that itself is a rebuild signal.

— **3. Write the new brief from current reality**

Current company size, current data shape, current owner. Treat the old brief as historical context, not a starting point.

— **4. List every assumption that no longer holds**

The article is explicit: half the rebuild work is realizing the original brief encoded assumptions that no longer apply. Surface them before any building starts.

— **5. Decide what the new flow does NOT do**

Most rebuilds are smaller than the patched version because the patched version absorbed scope that no longer belongs in this workflow.

— **6. Identify the single owner of the rebuild**

One person holds the brief end to end. If you cannot name them, the rebuild will produce another patch-prone flow.

— **7. Ship the new flow alongside the old one**

Run both in parallel briefly so you can compare outputs and catch the assumptions you missed. Then retire the old flow cleanly.



Four anti-patterns that look like iterate but are rebuild

01 The patch on a patch

Every recent edit reaches into a fix from the previous edit. The flow has stopped describing the work and started describing its own history. Iterating here adds another layer; rebuilding from the brief erases the layers.

02 The brief no one can recite

When you ask three people what this workflow is for and you get three different answers, the brief has decayed. You can iterate on shared understanding; you cannot iterate on its absence.

03 The orphaned owner

The flow technically has an owner on a wiki page, but that owner has not touched it in two quarters and three other people have. Ownership has actually moved. A rebuild forces the new owner to write the new brief.

04 The size-shift miss

The flow assumes a reviewer can eyeball every output. At three times the volume that assumption is silently false. Output quality drops without anyone noticing because the patches keep firing.



Five signals to track quarterly

EDITS PER QUARTER

meaningful majority

Trend matters more than absolute count. Flat or rising for two quarters is a

MODULE CONCENTRATION

single

Share of edits hitting the same module. When a majority cluster, treat as

OWNER CONTINUITY

single

One named operator who shipped the last three changes. Anything else is

BRIEF FIDELITY

recitable industry+

Three teammates give the same one-sentence answer for what the flow does. If

SIZE-SHIFT GAP

Headcount or volume change since the brief. Large multiples invalidate



- NEXT STEP

Rebuild from the brief, not from the flow.

When the maintenance log shows five edits clustered on one module, stop patching and re-open the brief at the current company size, current data shape, and current

[Read the full architecture note ->](#)