



• ARCHITECTURE NOTES · RESOURCE

Adding AI without trashing voice

Voice degradation is solvable. The fix is structural. Prompt engineering will not get you there.

Companion to: [How to add AI to your agency without trashing your brand voice.](#)



Why this matters

A handful of agencies are running production AI on client content. Most are still stuck because the first attempt collapsed brand voice and nobody trusts the second attempt. The pattern repeats: a strong start, a quiet flattening, a rescue call. This deck is for agency leads about to add AI to client production work, and for teams whose first attempt drifted and want to know what the second attempt has to look like.

- The two failure modes that account for most rollouts gone wrong
- The three artifacts that prevent both
- What to build first if you only build one
- The investment shape: weeks, not months



The two failure modes

01 Silent register drift

Output reads competent for a few weeks, then someone reads back two months of posts and notices the brand has flattened into a generic agency register. Nobody can point at a specific bad post; the cumulative effect is brand erosion. Reviewers do not catch this consistently

02 Per-batch contamination

Drafts start on-voice for the first ten outputs, then a model update or a switched provider quietly nudges the register sideways. By draft fifty the team is rewriting every output, which costs more operator time than writing from scratch would have.

03 What both share

Both are predictable. Both are caused by the same gap: there is no mechanical voice contract enforced in the dispatch path. Reviewers cannot hold a register line draft after draft; software can. The fix for both failure modes is the same.

04 Why prompt engineering will not fix this

Prompts drift across model versions and across batches. A prompt-only fix lasts until the next provider update. Voice has to be enforced as a written specification audited mechanically, not as a paragraph prepended to every call.



The three artifacts that fix it

01 Written voice specification

A document the team agrees on. What the brand sounds like, what sentence shapes it uses, what vocabulary is forbidden, what the rhythm looks like when it is working. The spec becomes the input to every drafter and the audit standard for every review. Cheapest of the three to

02 Automated scanner in the dispatch path

Runs on every draft before any human sees it. Enforces the voice spec mechanically. Drafts that fail get rewritten automatically; drafts that fail repeated rewrites park for manual edit. One engineer-day to write. Pays back the first time it catches a bad register that a tired reviewer would

03 Brief schema sharper than a junior copywriter brief

AI output quality is downstream of input quality. Vague briefs produce shallow drafts at any model size, regardless of vendor. The model is rarely the bottleneck. The schema settles inside a month of weekly iteration.

04 Why three, not one

A spec without a scanner gets ignored by week three. A scanner without a spec has nothing to enforce. A scanner and a spec without a brief schema regress toward the model default register. The three artifacts together are what hold the line.



Most agency rollout vs. structural rollout

Most agency rollout

- Prompt engineering as the primary tool
- Voice held by reviewer judgment, not software
- Brief is a paragraph or a Notion doc
- Failure shows up in week eight, not week one
- Rescue call when the model gets blamed
- Headcount climbs back as cadence rises

Structural rollout

- Voice spec written in week one
- Scanner deployed in week two
- Brief schema iterated weekly through month one
- Failure caught at the dispatch gate, not at the calendar level
- Marginal cost drops to scanner plus operator review
- Headcount stays flat as cadence climbs



The investment shape

01 Voice specification

One focused week if the team has never articulated voice before. Spec writing is co-written with the team, not handed in. Output is a document the drafter reads and the reviewer audits against.

02 Scanner

One engineer-day to write. Runs voice gates, sentiment scans, dead-link checks at the per-artifact level. Pays back the first time it catches a register break a tired reviewer would have approved.

03 Brief schema

Settles inside a month of weekly iteration. Six fields in our practice. The schema is the boring part; iterate on the bullets, not the form.

04 After install

Marginal cost of high-quality on-voice copy drops to roughly the cost of running the scanner plus operator review. Headcount stays flat as cadence climbs. That is the structural shift that makes outcome pricing possible inside an agency that used to bill by the hour.



Build order if you can only do one at a time

- **Build the scanner first**

If you only build one of the three, build the scanner. A spec without enforcement gets ignored by week three. Software does the audit job software does well.

- **Brief schema second**

After the scanner. The schema is what keeps your output from regressing toward the model default register. Vague briefs sink the rest of the stack.

- **Voice spec third (but never skip it)**

Cheapest to write, most expensive to skip. Every agency that has called Arthea in for a rescue had skipped this step and tried to fix the output by upgrading the model.

- **Co-write the spec, do not hand it in**

A spec written without the team is a spec the team will not enforce. The week-one co-write is what makes it real.

- **Deploy the scanner before the team scales output**

Per-batch contamination shows up around draft fifty. The scanner has to be in the path before you cross that volume.

- **Lock the brief schema after a month**

Iterate weekly through month one, then stop iterating on the form. Iterate on the bullets after that.



Anti-patterns that look like fixes

01 Upgrading the model to fix the output

A bigger model on a vague brief still produces shallow drafts. The model is rarely the bottleneck. Fix input quality first.

02 Reviewer-only enforcement

Tired reviewers approve drifted register. Three months later the brand has flattened. Software holds the register line; humans cannot do this consistently across batches.

03 Prompt-only voice control

Prompts drift across model versions. A prompt-only fix lasts until the next provider update. Voice is held in a spec, not in a system prompt.

04 Skipping the spec because the team "knows the voice"

Tacit knowledge does not survive a new hire, a switched provider, or a tired week. The spec is what makes the voice transferable. Skipping it is the most common mode of rescue.



Investment shape (verbatim from the article)

SPEC

1 week

Focused, if the team has never articulated voice before.

SCANNER

1 engineer-day/month

To write. Runs in the dispatch path on every draft.

BRIEF SCHEMA

Settles inside a month of weekly iteration.

STEADY-STATE COST

scanner + review

Marginal cost of on-voice copy drops to scanner plus operator review.



- NEXT STEP

Voice is structural, not a prompt.

Spec, scanner, brief schema. Three artifacts, weeks not months, and the cost line decouples from cadence. If you only build one, build the scanner.

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