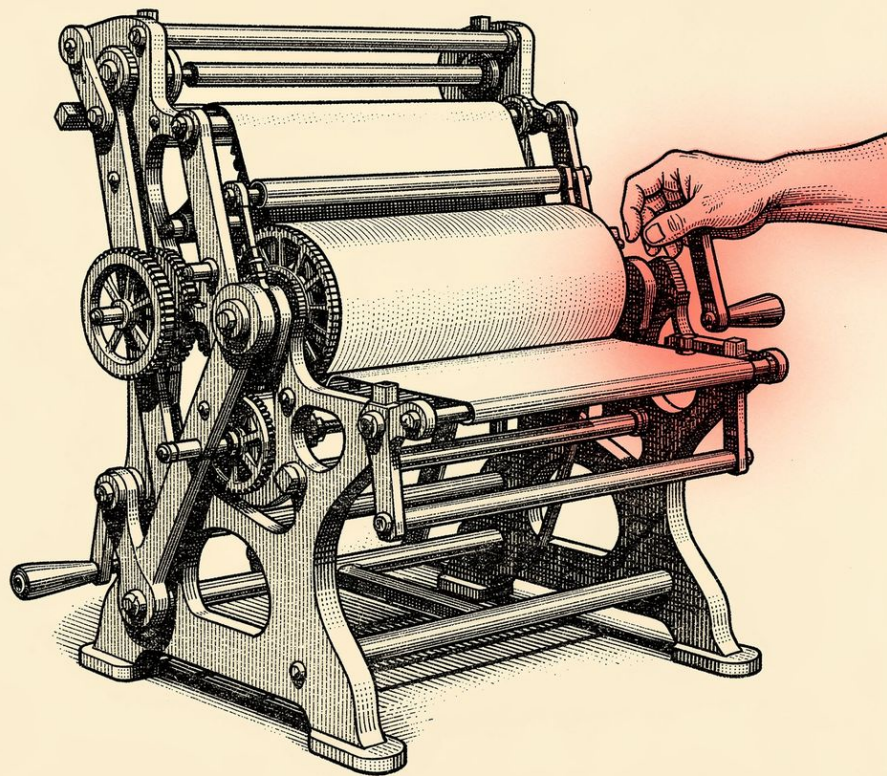




FRAMEWORK

AI-Enabled Creative Operations

How a lean team punches above its weight

The interesting question is what a team **stops doing** once AI can write.

And what has to stay human on purpose.

HOW ADOPTION USUALLY GOES

Everyone develops their own prompting habits in private.

Six months later the team is faster and less consistent, which is the worst of both. The knowledge stayed in people's heads.

A prompt that works lives in one person's chat history **and** **dies there.**

The same corrections get made every week by different people who don't
know the others made them.

Part 01

Skills, not prompts

A skill is team infrastructure. A prompt is personal technique.

- 01 A trigger · when it should fire, written broadly enough that it does
- 02 A resolution step · state assumptions so a person can correct a setting
- 03 Two modes · write, and check someone else's draft
- 04 Reference files · voice, proof library, caps, inclusive language
- 05 Hard gates · rules that block output regardless of score
- 06 A feedback loop · the part everyone skips

**Half of all voice problems are a
piece written at the **wrong
setting**, not written badly.**

Part 02

The split

WHERE THE LINE SITS

Automation on the front end. Human gates on what matters.

Automate

Finding what needs attention. Assembling the first draft. Enforcing documented rules. Producing variants to a spec. Getting design to MVP stage.

Keep human

Deciding what we work on. Judging whether it's good. Deciding what the rules should be. Choosing the spec. Overriding a rule with a reason.

Every item on the right is a judgment call **the model can't feel.**

An accelerating pipeline can't make them for you. A team that hands them over stops having a point of view.

Part 03

The feedback loop

A draft that passes every rule **and still gets rejected** is the signal that matters most.

It means a rule is missing, and it's the only way you find the rules nobody
knew they had.

TWO RULES ABOUT CHANGE

A skill anyone can silently edit is a skill nobody can trust.

- 01 Never edit reference files from a live session. Changes go through a scheduled review.

- 02 Version it, and document the known gaps. A skill that says what it can't do yet is more useful than one pretending to be complete.

Part 04

Where to start

FOUR WORKFLOWS, IN ORDER

Each one buys the time for the next.

- 01 Content decay pipeline · flag, draft, route to an editor
- 02 Design workflows · arrive at 80%, refine what needs judgment
- 03 Skills library · voice, research, editorial direction
- 04 Discoverability · structured so a claim survives with its attribution

WHAT IT DOESN'T DO

The category overclaims. This doesn't.

- 01 It doesn't produce a point of view. It can enforce a position; it can't have one.
- 02 It doesn't fix a missing strategy. Faster production produces more of the wrong thing, sooner.
- 03 It doesn't reduce headcount on its own.
- 04 It requires real investment. Tech-company output on the lowest tier of tooling is a choice.

KNOWING IT WORKS

Five signals.

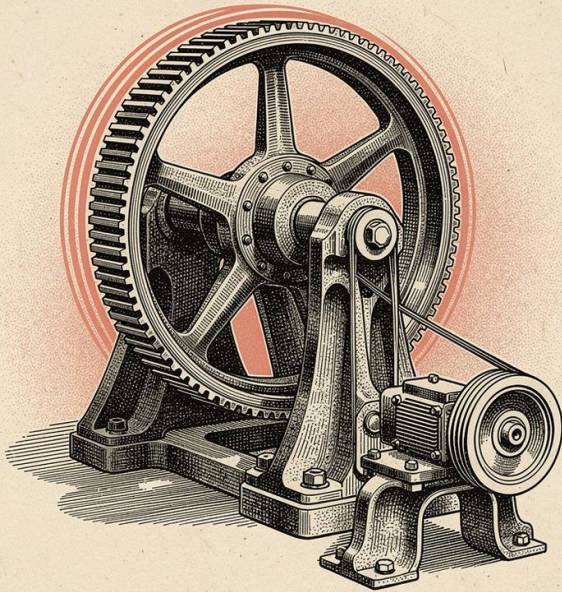
- 01 Someone who didn't build the skill uses it unprompted

- 02 Review comments cite the standard, not preference

- 03 New hires produce on-voice work in week one

- 04 The correction log has entries from other people

- 05 The floor rises. The ceiling still comes from people.



FROM THE SHELF

Technology is an accelerator of momentum, not a creator of it.

JIM COLLINS

GOOD TO GREAT

None of the companies Collins studied ignited their transformation with technology. They applied it once they already knew what they were best at.

**Automate the front end. Keep
the judgment.**