AI POWER HOUR · COMMUNITECH × MOMENTUM · JUNE 15, 2026

Make AI Push Back Before You Decide.

Momentum Product Co. · momentumproductco.com

The skill isn't a better prompt — it's a system that argues with you. Ask any tool "should I do X or Y?" and it picks one and lists ten reasons it's obviously right. For a decision you have to defend, that confidence is a trap. The five moves below force the model to expose its assumptions, critique itself, and show you the alternatives. Underneath is every question the room asked — answered live.

THE FIVE-MOVE FRAMEWORK — RUN THEM IN ORDER ON ANY DECISION

01 Restate

Make it prove it understood you — and name the assumptions you never stated.

02 Critique

Highest-leverage move. Make it find the holes a skeptical CTO and head of sales would.

03 Constrain

Tie it to real money, time, and headcount. Make it say what has to give, and why.

04 Compare

Top three paths side by side — no straw men. The one fact that would flip the pick.

05 Challenge

The move everyone skips. Argue the strongest case against your #1 — and for doing nothing.

QUESTIONS ANSWERED LIVE IN THE ROOM

Q. Is it worth running the same question through different AI tools to compare?

Yes — but only if they share the same context. Different results across Claude, ChatGPT, and Gemini usually mean you loaded different data into each. Keep one centralized folder so context is identical. Pick the tool by the job: Claude leads on executing work beyond the chat; Gemini is better at image generation; OpenAI has strong voice tools.

Q. If I just run the "restate / assumptions" prompt, does the recommendation change?

No — you haven't asked it to do anything differently. That move only surfaces what it factored in. The answer changes when you take a surfaced assumption, tell it it's wrong, and re-run. You can't correct an assumption you can't see.

Q. Can I fix a bad assumption — like "I don't actually have the budget" — by adding it to the prompt?

Yes, and you should. Paste the listed assumption back with the correction ("our budget is X," "our sales process isn't standard") and watch the decision shift. One corrected assumption can flip the whole answer.

Q. How good are today's tools at critiquing their own answer vs. a different model critiquing it?

Much better than before. Older models were "addicted to their answers"; modern ones (4.8 / current Opus) self-critique usefully. We do tune the approach by model — newer ones respond better to what to do than what not to do. A second model as critic still works, but it's no longer the only way.

Q. If I drop down a model (4.8 → 4.6), is it less attached to its answer?

Depends on the stakes. Low-stakes decisions stayed consistent; high-stakes ones hold their position more — which you want. Most variance shows going from Opus down to Sonnet and below, not between adjacent versions.

Q. How do I feed in different stakeholder perspectives — like a skeptical sales manager?

Build that context into your system, not your prompt. Encode each stakeholder lens as a skill you refine over time and let the framework pull on it. Context isn't dumping words into the chat window — it's reusable structure.

Q. Can I call a skill any time? Does it have to live in a project?

Call it from any chat — no project required. Heather opened a clean chat, asked "YouTube vs. a Maven course?", said "apply the adversarial skill," and it ran the whole five-move framework. Skills can also auto-trigger on keywords.

Q. Are custom skills the same as changing the chatbot's personality? Where's that stored?

Different things. Personality and tone ("don't greet me by name") live in Settings memory and shape how it talks generally. Skills are reusable assets you invoke deliberately — "write in my voice," "apply your conservative financial lens." One adjusts manner; the other applies a method.

Q. Should I be updating my skills with the instructions I use?

Yes — and you don't write them by hand. Tell Claude "package this into a skill called Adversarial Review" and it builds it from the conversation. Also borrow open-source expert skills (pricing, segmentation, cold outreach) — they're just files you drag and drop in.

Q. On pushback, should I trust what the model already "knows"?

No — for anything time-sensitive, tell it to research fresh instead of relying on training data. Bake it into the skill: "don't reference what you think you know — research it, cite sources, discard anything older than six months." Critical for competitive intel, which goes stale in days.

Q. If it doesn't search the web, it's answering from a year-old snapshot?

Essentially yes. Ask "MacBook Air or Pro?" and it recommends whatever was current at training. It'll also walk you through product screens that have since been deprecated. That's why you force citations and recency filters for anything that changes.

Q. If I identify myself, can AI figure out who I am from my socials — and is it reliable?

You can try, but expect a flattering bias — it tends to tell you you're an expert. More useful: point it at honest sources. A Reddit scan for a client surfaced verbatim negative quotes that never showed up in their support tickets — people complain online, not in tickets.

CONFIDENCE TRAPS TO WATCH FOR

Sycophancy. It agrees because you sound confident, not because you're right. Ask it to argue the other side.

Fake confidence. Certainty is tone, not proof. Make it rate its confidence and flag the weak links.

Instant flip-flop. If it drops a good answer the moment you push, that's a tell.

Rigged comparison. Alternatives built to lose aren't a comparison. If you'd never pick option B, it's a straw man.

Skipping the challenge. Most stop at Move 4. Protect Move 5 — don't let it soften the case against itself.

Stale knowledge. Year-old "latest" picks and deprecated UIs. Force recency and sources.

A confident answer is the start of a decision — not the decision itself.