SAM ✦ RENDERS

A CREATIVE FOUNDER'S FIELD GUIDE

3 Questions Before You Ship Any AI Feature

The three questions I learned the hard way at Healify — the ones that separate AI features users trust from the ones they quietly stop using.

Sam Renders
Multi-platinum artist (Sam Feldt, 10B+ streams) · AI entrepreneur · GP, Hello World Investments
Building Healify on agentic infrastructure.

WHY THESE THREE

Most AI features fail for the same three reasons.

Not technical reasons. Judgment reasons.

I've spent 18 months shipping Healify, an AI health platform built on LangGraph agents. In that time I've killed more features than I've shipped — not because the engineering was broken, but because the judgment was wrong.

The three questions in this guide are the ones I learned the hard way. I run every feature through them now before it touches the build queue. They've saved me roughly 40% of the engineering time we used to waste.

Steal them.

1

What would the user do if this feature didn't exist?

If the honest answer is **"nothing"** or **"something better than what we're building"** — kill it. Most AI features are solutions chasing problems. The agent layer makes it cheap to build, so we build, and mistake **"this works"** for **"this matters."**

How to test it

Write the user's day. One paragraph. Now imagine your feature isn't in it. Does the day get worse? Or do they just do the slightly-clunkier thing they were already doing? If the day doesn't get worse, the feature isn't pulling weight.

Healify example

We built auto-suggested meal swaps based on daily calorie deficit. Worked great. Killed it — when we wrote the user's day without it, nothing got worse. They were already swapping meals their way. Our version added a third place to look at lunch.

Ship instead

Whatever the user is **already** doing that you can make 10× faster or clearer. AI products that win attach to existing behavior. They don't manufacture new ones.

2

What does this feature look like when it's wrong?

Not **"if"** wrong. **When.** Every AI feature is wrong some percent of the time. The question isn't whether you can prevent it — you can't. It's whether the failure mode is recoverable.

How to test it

Write the worst-case user moment, specifically: who the user is (age, context, emotional state), what they asked, what the AI said (the wrong version), what they did next, and whether they came back tomorrow. If the answer is **no**, the failure mode is too expensive. Redesign so the failure is recoverable.

Healify example

Our first bloodwork-explainer gave clinically correct, emotionally flat answers. Wrong scenario: a 34-year-old, kid in the next room, just got a cholesterol number she didn't expect. Cold answer. Didn't come back. The model wasn't wrong — the judgment was. We rebuilt to lead with emotional context, then logistics. Trust scores doubled.

Ship instead

Features whose failure mode is **"didn't help much this time"** — not **"made the moment worse."** The bar is whether the user comes back the next day.

3

Can the model deliver this without saying anything cringe?

This is the one engineers underweight and creatives can't unsee. A clinically correct answer in the wrong tone is still a wrong answer. Users don't separate logic and tone. They experience them as one thing.

How to test it

Read the agent's output out loud, in the voice you'd use with a friend. If it sounds like a corporate disclaimer, a wellness-influencer caption, or a clinical-trial brochure — kill it. The model is hedging because the prompt told it to. The prompt is wrong.

Healify example

We rewrote one system prompt four times in a week. Not because the logic was wrong — because the agent kept sounding like a therapist with no skin in the game. We stripped every reflex hedge, every "consult your doctor," every safety platitude that wasn't actually doing safety work. Trust scores went up. The clinical content didn't change. The voice did.

Ship instead

Features whose voice you'd be proud to read aloud. If you wouldn't, the user already isn't reading it the way you hoped.

USE IT LIKE A CHECKLIST

Before any AI feature gets engineering time

- ☐ Q1 — A real user's day measurably worsens without it
- ☐ Q2 — Worst-case failure mode is recoverable, not catastrophic
- ☐ Q3 — Voice is one you'd be proud to read aloud to a friend

The throughline: none of these are technical questions. They're judgment questions. Taste questions. The model gives you infinite drafts. Your job is to pick the right one — the skill creative careers train for a decade before you ever touch a model.
