FLIGHTY.AI / R&D FIELD GUIDE 001

BEFORE YOU BUILD IT, TRY TO KILL IT.

A field guide for using AI to find evidence before you spend your time building the wrong thing.

A QUIET BANDS FIELD GUIDE

v2 / RESEARCH SYSTEM

AI MADE BUILDING CHEAP.

IT DIDN'T MAKE BEING WRONG CHEAP.

The dangerous question is not "Can AI build this?" It usually can.

The useful question is: what evidence would have to exist before this deserves a build?

THE THREE THINGS YOU NEED TO PROVE

01 THE PROBLEM IS REAL.

02 PEOPLE ALREADY REACT TO IT.

03 THE CONSEQUENCE MATTERS.

DON'T LET AI UPGRADE YOUR EVIDENCE.

ONE SOURCE IS ONE SOURCE.

Do not turn a single company's process into an industry pattern.

WORK EXISTS != BUSINESS EXISTS.

Manual work can be real without creating a software opportunity.

MISSING FEATURE != DEMAND.

A competitor not doing something is not proof buyers want it.

BURDEN != WILLINGNESS TO PAY.

Time, errors and frustration matter. They still do not prove a budget.

KNOW WHEN TO STOP SEARCHING.

Sometimes the next answer belongs to a practitioner, not another browser tab.

MAKE EVERY CLAIM AUDITABLE.

Use this structure whenever AI gives you a research conclusion.

CLAIM	**What exactly are we saying?**
EVIDENCE + SOURCE	**What supports it, and where did it come from?**
SCOPE	**One person? One company? Several sources? Industry-wide? Unknown?**
WHAT THIS PROVES	**The narrowest conclusion the evidence supports.**
WHAT THIS DOES NOT PROVE	**Demand? Frequency? Economics? Willingness to pay? Market size?**
CONFLICTS / GAPS	**Missing sources, stale evidence, contradictory facts, weak proxies.**
NEXT EVIDENCE NEEDED	**Search, artifact, practitioner interview, or manual test?**

PROBLEM SCOUT

Find evidence that the problem or recurring work actually exists - without proposing a solution.

PROMPT

I am investigating this claim: [CLAIM]. Do not brainstorm products or help me build anything. Find direct evidence that the problem or recurring work exists. Prioritize primary sources, practitioner language, job duties, SOPs, forms, tickets, reports, public complaints and other observable artifacts. Separate direct evidence from inference. Do not generalize beyond the scope of the source.

REQUIRED OUTPUT

- **CLAIM / EVIDENCE / SOURCE / SCOPE**
- **WHAT THIS PROVES / WHAT THIS DOES NOT PROVE**
- **CONFLICTS OR GAPS / NEXT EVIDENCE NEEDED**

STOP CONDITION: If you cannot find the work without describing your proposed solution, the signal is weak.

WORKAROUND HUNTER

Find what people already do when the problem appears. Behavior is stronger than polite agreement.

PROMPT

For this problem: [PROBLEM], investigate what people do today instead of my proposed solution. Look for spreadsheets, email chains, manual reviews, duplicate entry, extra staffing, consultants, existing software, checklists, work queues, exceptions, delays, or choosing to do nothing. Show evidence for each workaround. Do not assume a manual process is inefficient or worth automating unless the evidence supports that claim.

REQUIRED OUTPUT

- **CURRENT WORKAROUND + WHO DOES IT**
- **ARTIFACTS OR SYSTEMS USED**
- **FREQUENCY / CONSEQUENCE - only if evidenced**
- **SOURCE + SCOPE + UNCERTAINTY**

THE MONEY QUESTION COMES LATER. First prove the behavior exists.

ASSUMPTION KILLER

Give AI one job: try to prove you wrong.

PROMPT

I want to build: [IDEA]. Act as a skeptical research partner. List the assumptions that must be true for this to deserve a build, then actively search for evidence that weakens each assumption. Look for existing alternatives, low-frequency use, low consequence, solved workflows, failed attempts, switching friction, and reasons people may tolerate the current state. Do not rescue the idea. Do not recommend features.

REQUIRED OUTPUT

- **ASSUMPTION**
- **EVIDENCE FOR / EVIDENCE AGAINST**
- **SOURCE + SCOPE**
- **CURRENT VERDICT: supported / weak / unresolved**
- **FASTEST WAY TO REDUCE UNCERTAINTY**

THE GOAL IS NOT TO KILL EVERY IDEA. The goal is to make the idea earn survival.

EVIDENCE AUDITOR

Run this after research. Its job is to downgrade claims that got stronger than their sources.

PROMPT

Audit the research below as if you are trying to prevent an expensive false conclusion. For every material claim, identify the exact supporting source and whether the wording overstates it. Flag single-source generalizations, proxy evidence, stale evidence, conflicting dates, unsupported numbers, burden estimates presented as staff time, and any leap from 'work exists' to 'buyers want software.' Rewrite each overstated claim to the narrowest defensible version. Research: [PASTE RESEARCH]

REQUIRED OUTPUT

- ORIGINAL CLAIM
- SUPPORTING SOURCE
- PROBLEM: none / scope / conflict / inference / stale
- DEFENSIBLE REWRITE
- WHAT REMAINS UNKNOWN

If the source does not support the claim, say so. Do not silently repair it with general knowledge.

HUMAN HANDOFF

Recognize when the internet has done its job and the next answer belongs to a person doing the work.

PROMPT

Based only on the evidence below, identify the 3-5 questions that cannot be answered reliably through desk research and require a practitioner. Prioritize questions about frequency, human touch time, rework, consequences, systems used, decision ownership, budget, and what happens in the real workflow. Do not ask whether they 'like my idea.' Turn each evidence gap into a neutral interview question. Evidence: [PASTE EVIDENCE]

REQUIRED OUTPUT

- UNRESOLVED QUESTION
- WHY DESK RESEARCH CANNOT SETTLE IT
- WHO CAN ANSWER IT
- NEUTRAL INTERVIEW QUESTION
- WHAT ARTIFACT WOULD HELP VERIFY THE ANSWER

WHEN THE NEXT ANSWER IS HUMAN, STOP POLISHING THE RESEARCH REPORT.

BUILD GATE

Decide what the evidence supports next. No score. No fake validation percentage.

PROMPT

Review the evidence below conservatively. Do not judge whether the idea is exciting. Decide only what the evidence currently justifies. Choose one next state: RESEARCH MORE, TALK TO HUMANS, TEST MANUALLY, EXPLORE A BUILD, or PARK IT. Explain what we can see, what is still assumed, and the smallest next experiment that would reduce uncertainty. Evidence that work exists must not be treated as evidence that a software opportunity exists. Evidence: [PASTE EVIDENCE]

REQUIRED OUTPUT

- **STATE**
- **WHAT WE CAN SEE**
- **WHAT IS STILL ASSUMED**
- **WHY THIS STATE FITS**
- **SMALLEST NEXT EXPERIMENT**

EXPLORE A BUILD means a technical conversation may be justified. It does not mean product-market fit.

DON'T PITCH. RECONSTRUCT THE WORK.

01 **What happened the last time this problem showed up?**

02 **Walk me through what you did from start to finish.**

03 **Where did a person have to touch it?**

04 **What gets sent back, corrected, or done twice?**

05 **What systems, files, forms or spreadsheets are involved?**

06 **How often does this happen?**

07 **What happens if nobody fixes it?**

08 **Who cares enough about this to change the process?**

SHOW ME THE FUCKING SPREADSHEET.

Not literally every time. The point is to find the artifact created by the work.

Spreadsheet
Ticket queue
Email chain
Form
SOP
Report
Calendar
Exception list
Invoice
Screenshot

Artifacts turn 'people probably struggle with this' into something you can inspect.

WHAT HAPPENS NEXT?

RESEARCH MORE	Core facts are still assumptions.
TALK TO HUMANS	You know who holds the missing answer.
TEST MANUALLY	The outcome matters; software may not be necessary yet.
EXPLORE A BUILD	Multiple evidence types justify a technical conversation.
PARK IT	The evidence does not justify more investment right now.

NOW PUT YOUR EVIDENCE UNDER PRESSURE.

The guide teaches the method. The Quiet Bands Pressure Test turns what you found into a conservative next step.

PRESSURE-TEST YOUR IDEA

quietbands.com/rd/pressure-test/

No score. No fake certainty. No AI telling you your idea is amazing.

BUILD LESS. LEARN EARLIER.

FLIGHTY.AI x QUIET BANDS