Question Finder

A five-minute guide · you and the requester

What this is: a quick check you run on a request *before* you start building. You play back the concrete thing you would build — "if we did exactly what you asked, here's what we'd build…" — and let the requester react to it. The difference between what they asked for and what they actually need becomes visible, and you leave with the real question, restated and confirmed. It takes 10–15 minutes. This is a conversation, not a workshop.

The one line to hold: a request can be perfectly clear and still be the wrong question. The requester usually isn't wrong — they're answering the wrong question. Your job isn't to challenge them; it's to **describe what their words would produce**, and let them correct it.

1 · WHEN DO I USE THIS?

Use this when

GOOD FIT

Any request you're about to act on — especially a clear, confident, well-written one. **The strongest trigger: the requester has already proposed a solution ("build…", "create…", "add…") rather than describing the problem.** The effort is big enough that building the wrong thing would hurt.

WRONG TOOL

The brief is genuinely unclear or ambiguous › Brief Decoder. The requester doesn't yet know what they need at all › Requirement Discovery. You're looking for the cause of a problem that already happened › Root Cause Drill.

2 · WHAT AM I ACTUALLY DOING?

The one idea

People can't judge a request in the abstract — but they instantly recognise a wrong answer once they can see it. So you don't argue with the request, and you don't summarise it back. You describe, concretely, the thing their words would produce. The requester pictures it, and the mismatch becomes obvious *to them*. That's what makes the sensitive part easy: you never tell them they're wrong. You show them what they asked for, and they correct it themselves.

© Witmer

THE REAL-QUESTION TEST

1 · THE ASK
"build us a dashboard of weekly sales"

→

2 · THE PLAY-BACK
"so we'd build a page you read each Monday"
describe it, don't summarise

→

3 · THE GAP
"actually — I want to know when it drops"

The real question wasn't a dashboard. It was an alert.

A request can be perfectly clear and still be the wrong question.

You're not challenging their judgement — you're showing them what their words would produce.

3 · HOW DO I RUN IT?

The five moves

10–15 minutes, with the requester in the room (or on a call — this doesn't work by email). The bold part is what you say or do; the line under it is where the value is.

1. Take the request as given	2 min	*Read the request together, exactly as written.* Don't interpret or improve it yet — you need the stated ask on the table, in their words.
2. Play back what you'd build	4 min	***"If we did exactly what you asked, here's what we'd build…"*** Describe the concrete thing — screens, output, behaviour — specifically enough that they can picture themselves using it. **Describe, don't summarise.** Repeating their request back reveals nothing.
3. Watch for the gap	3 min	*Let them react. "Not quite…" "Actually…" "What I really want is…"* That's the real question surfacing. Then ask: **"What would you do with that?" Keep asking until they describe the outcome, not another solution** — "I need to know when sales fall," not "I need two dashboards."
4. Restate the real question	3 min	*Write it in one sentence.* The question that, if answered, would give them what they actually need — not what they originally asked for.
5. Confirm it	2 min	*Read the restated question back.* You're done when they say **"yes — that's what I actually needed."** If they hesitate, you're not there yet.

© Witmer

THE PLAY-BACK IS THE WHOLE TOOL — DESCRIBE, DON'T SUMMARISE

If you say "so you want a dashboard showing weekly sales by region?", they will say yes, and you will have learnt nothing — you've just repeated their words. If you say "so we'd build a page with a regional sales chart, updated every Monday; you'd open it each week and read the numbers", they can picture it — and *that* is when they say "actually, I don't want to read it weekly. I want to know when a region drops." **Concrete enough that they can imagine using it.** That's the whole trick. The real question wasn't a dashboard. It was an alert.

4 · WHAT IF IT GOES WRONG?

Common mistakes

What happened	The fix
You summarised instead of describing	Repeating the request back always gets a "yes." Describe the concrete thing it would produce — screens, output, behaviour.
It felt like a challenge	You're not telling them they're wrong. You're showing them what their words would produce, and letting them correct it. Stay descriptive.
They said "yes, that's right" — and meant it	**Don't try to force a different answer.** Sometimes the original question really is the right one. The value came from checking, not from changing it. Go and build it.
You started solving before confirming the real question	Stop. Confirm the restated question first — only then discuss solutions. This one happens constantly.
The real need turned out to be unknown	If they genuinely don't know what they need, that's a different tool — Requirement Discovery. Don't turn this into a discovery workshop.
You tried to do it by email	The play-back needs their live reaction. A written description gives them time to explain away the original ask.

© Witmer

Before you start building

- ✓ You described what you'd build — you didn't summarise their request
- ✓ They reacted to it, and you asked "what would you do with that?"
- ✓ The real question is written in one sentence
- ✓ They said "yes — that's what I actually needed"
- ✓ You haven't started designing the solution yet

5 · WHICH TOOL IS THIS?

Where this sits

Four tools work with unclear inputs, and they do genuinely different jobs. The one that matters most: **Brief Decoder clarifies an unclear brief. Question Finder challenges whether a clear brief is even the right question.** And the newest companion: **Requirement Discovery is used when the requester genuinely doesn't yet know what they need. Question Finder is used when they've asked a clear question — but it may be the wrong one.**

THREE TOOLS, THREE DIFFERENT JOBS

BRIEF DECODER	QUESTION FINDER	REQUIREMENT DISCOVERY
the brief is unclear	**the brief is clear — but wrong**	**they don't know what they need**
clarify it — turn a vague brief into an executable one	challenge it — is this even the right question?	discover it — surface needs not yet recognised

Next question: once you've confirmed you're solving the right problem, use **Estimate the Unknown** if you're unsure how reliable the available information is. (Question Finder, Estimate the Unknown and Root Cause Drill are the *reduce uncertainty before acting* family: are we solving the right problem? · how uncertain is the answer? · do we really understand why?)

The habit to take back: before starting any request, ask yourself — **"if I built exactly this, what would they do with it?"** If you can't answer, you don't understand the request yet. If the answer sounds wrong, you've found the real question. It costs two minutes and saves weeks of building the wrong thing.

© Witmer