Root Cause Drill

Facilitator guide · a team drill that traces a recurring problem, with evidence, down to the cause that's really driving it

Your job: run one recurring problem down to its root — and make the team **prove every step** on the way. You hold one rule above all others: no cause earns the next one beneath it until it has evidence. When someone can't back a cause, that's not a failure of the activity — it's the moment the room sees it's been guessing. Keep the drill moving down one proven link at a time, until removing the final cause would stop the recurrence.

The one line: the first cause is often where curiosity stops — not where the problem starts.

· WHEN DO I USE THIS?

Use this when

GOOD FIT

- A problem has **come back after being fixed** at least once
- The team is about to apply the obvious fix again
- A "solution" only ever holds for a few weeks
- Different people blame different things, none of it evidenced

USE ANOTHER TOOL

- Not sure you're solving the right problem at all → **Question Finder** (challenges the request upstream)
- Working back from a future goal, not a present symptom → **Future Back** (backward from success)
- A one-off problem with no history of recurring → a normal fix; you don't need a drill

The test question: "Has this exact problem been 'fixed' before and come back?" If no, this isn't the tool — Root Cause Drill earns its keep on *recurrence*.

PART 1 RUN IT

everything you need to start

1 · WHAT IS THIS REALLY DOING?

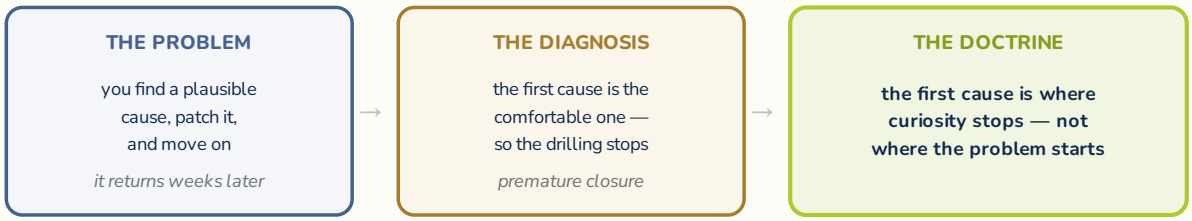
The one idea

THE ONE IDEA

Recurring problems survive because we stop at the first comfortable cause. This drill makes the team keep going — and **prove every link** — so a shallow stop becomes visible instead of passing as an answer.

© Witmer

WHY THE PROBLEM KEEPS COMING BACK



A cause you can patch isn't always the one driving it.

2 · WHAT DO I NEED?

What you'll need

Item	What it is, and when it's used
The capture sheet (core)	The hero of the activity: linked cause boxes running down the page — symptom at the top, a cause and its evidence at each step, ending in the named root. An unfinished chain is **meant to look unfinished**. Leave room to add links if the chain grows.
One recurring problem (core)	Brought by the team — a real symptom you can point to, already fixed once and returned. Chosen before the session, not invented in it.
Evidence to hand (core)	Logs, numbers, timelines, screenshots, examples. Part of the drill is noticing where the evidence *doesn't* exist — and if it doesn't exist yet, discovering that is a valid outcome of the drill.
Facilitator	Runs the flow, **holds the evidence gate**, writes the chain, keeps the drill going down not sideways. Does not supply causes.
Participants	Propose causes, bring the evidence, decide when the root is reached.

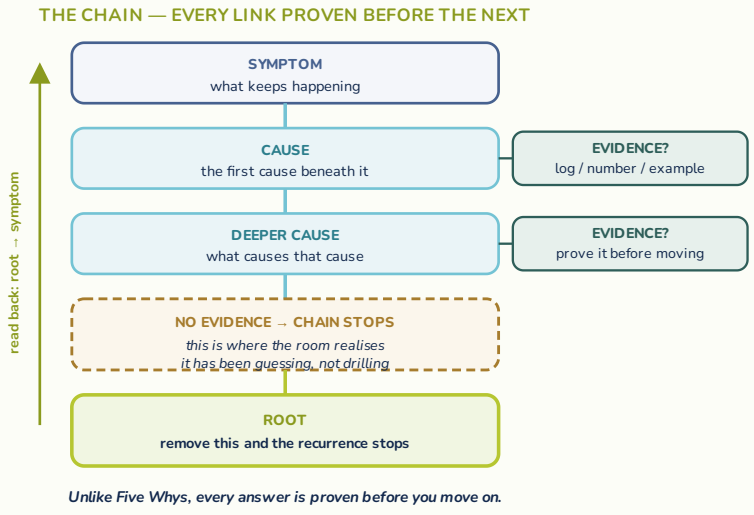
3 · HOW DO I RUN IT?

The drill, step by step

1 · Set up & brief	**2 min**	Appoint yourself facilitator and explain the drill in your own words in a minute: *we trace one recurring problem down, and we can't move to the next cause until this one is proven.* Write the recurring symptom in the top box.
2 · Name the symptom	**3–5 min**	Get the symptom stated precisely — what happens, how often, since when. *Describe the symptom before anyone suggests a cause* — hold the room there; it's tempting to jump.

© Witmer

3 · Lay the first cause	**3 min**	Ask for *the first cause you'd normally fix* and write it directly beneath the symptom. This is the comfortable one — the one you'd usually stop at.
4 · Prove it	**3–5 min**	Ask for the evidence — a log, a number, an example. **No evidence, no next link.** Say it plainly: *"We need evidence before we go deeper. What log, number or example shows this is true? If we don't have it, we stop here — or go and get it and come back."* If they can back it, write the evidence beside the cause; if not, hold at this link (see the tip below).
5 · Drill again	**repeat**	Ask what causes the proven cause; write it as the next link down; prove that one too. *Follow one chain only — down, never sideways.* If the room starts branching into separate causes, pull it back to a single line. Set aside any alternative causes until this chain is complete.
6 · Name the root	**3 min**	Stop when removing the bottom cause would *end the recurrence, not just reduce it.* Circle it as the root. Test: "if we removed only this, would the problem stop coming back?"
7 · Read the chain back	**2 min**	Read the chain aloud from the root up to the symptom. *If every link still makes sense read from root to symptom, you've found it.* Write the one change you'll make and how you'll check it worked.



FACILITATOR — THE TURNING POINT

The real moment isn't the root — it's the first time you say **"we can't go further, this cause has no evidence."** That's when the room feels it's been guessing. Don't rush past it. And if the chain stops because the evidence simply *doesn't exist yet*, that's a useful result, not a dead end: the team's next action is to **go and gather that evidence** before drilling further

© Witmer

— never invent the next cause just to finish the chain. Sometimes the honest root is "nobody actually knows — yet."

FACILITATOR — IF MORE THAN ONE ROOT APPEARS

If the team lands on more than one possible root, don't merge them into one chain. Finish the strongest evidence-backed chain first, then start a second drill if it's still needed. Two clean chains beat one tangled one — this comes up often in real retrospectives.

PART 2 MAKE IT LAND & MAKE IT STICK

why it works, and what to watch

4 · WHY DOES THIS WORK?

Why it works

Recurring problems survive because people stop at the first plausible cause — it's comfortable, and patching it feels productive. The information to go deeper is usually right there; the drilling just stops. Forcing an evidence check at every link removes the comfort: you can't advance on a guess, so a shallow stop is *seen* instead of passing as an answer. That single rule — prove it before you move — is the whole difference between this and asking "why?" five times.

5 · WHAT IF IT GOES WRONG?

Common mistakes

What you'll see	The fix
It turns into Five Whys — a string of unproven "why?"s	Reinstate the gate. After every cause, ask for the evidence *before* the next "why". The proof is the point; the question alone isn't.
A cause has no evidence, but the room wants to push on	Hold the line. Mark the link unproven and stop, or send someone to fetch the evidence. Guessing past the gap defeats the tool.
The team branches into several causes at once	One chain only. Set aside the other candidates; follow the single most-evidenced line down. You can drill a second chain afterwards if needed.
They stop at the first comfortable cause	Ask the recurrence test: "if we removed only this, would it stop coming back?" If not, it's a symptom, not the root — keep drilling.
The "root" is really a person to blame	Redirect to the system: what allowed that to happen? A root you can remove is a process or condition, not a name.

© Witmer

What people say — and your move

What you'll hear	Your move
"We already know why this happens."	Then it'll be quick — show me the evidence, and we'll drill one level below it. If you're right, the chain confirms it fast.
"We don't have time to gather evidence."	We've fixed this twice already without it. The evidence is what stops us fixing it a third time. Mark the link unproven for now and note what to go and get.
"Isn't this just Five Whys?"	Same shape, one hard rule added: every "why" has to be proven before the next. That rule is the whole difference.
"The real cause is [a person / another team]."	Maybe — but a name isn't something we can remove. What condition let that happen? Drill to the process, not the person.

Before you close, check:

- ✓ Every link on the chain has evidence beside it
- ✓ You followed one chain down, not a spread of causes
- ✓ Removing the root would stop the recurrence, not just reduce it
- ✓ The chain reads cleanly from root back up to the symptom
- ✓ You've written one change and how you'll check it worked

6 · HOW DO I MAKE THIS STICK?

Take it back to work

THE HABIT

Before you fix a recurring problem, ask one question: "what's the evidence this is the cause — and what's beneath it?" If you can't answer the first half, you haven't found the cause yet. If you haven't asked the second, you've probably stopped too early.

For a returning team: keep the last drill's change and its check. Did the recurrence actually stop? If it came back, the root was a symptom — drill again, one link deeper.

© Witmer

Appendix

a worked run · what good looks like · where this sits

7 · A WORKED RUN

Watch the chain reach the root

The problem (recurring): a team's releases keep slipping — three of the last four went out late. It's been "fixed" twice by adding buffer time. It keeps happening.

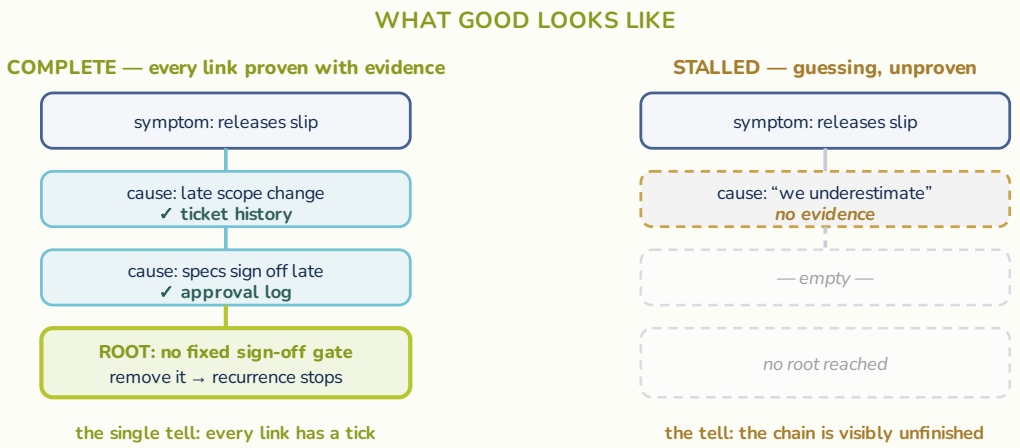
Stage	What happened in the room
Symptom	"Releases slip" → sharpened to "3 of the last 4 releases shipped 1–2 weeks late."
First cause	"We underestimate the work." The comfortable one. **Evidence?** The team looked — estimates were actually close on two of the three. *Unproven — the chain nearly stopped here.*
Re-drilled	Reframed with evidence: "scope changed late in the sprint." **Evidence:** ticket history showed scope added after day 6 on all three slips. ✓ Proven — drill down.
Deeper cause	"Why does scope change late?" → "specs are signed off after the sprint starts." **Evidence:** approval log showed sign-off dates trailing sprint start. ✓ Proven.
Root	"There's no fixed sign-off gate before a sprint opens." Recurrence test: fix the gate and late scope can't enter → slips stop. **Root confirmed.**
Read back & commit	Root → symptom read cleanly. Change: no sprint opens without signed-off specs. Check: track scope-added-after-day-6 next two sprints.

The lesson, not stated but felt: the first cause ("we underestimate") was the one they'd patched twice — and the evidence didn't support it. The real root was two links below it.

© Witmer

8 · WHAT DOES A GOOD DRILL LOOK LIKE?

What good looks like



A shallow stop doesn't hide — it leaves the chain standing incomplete.

Left: a drill that reached the root. Right: a drill that stopped at the first comfortable guess.

9 · WHERE DOES THIS FIT?

Where this sits in the Gym

Root Cause Drill is one of the **"reduce uncertainty before acting"** family. Each changes the direction of thinking:

Tool	Its direction — and when to reach for it
Question Finder	**Upstream.** Challenges the request itself — "are we solving the right problem?" Use before you drill, if the problem itself is in doubt.
Future Back	**Backward from success.** Works back from a future outcome — "which problem must we solve to reach the goal?"
Root Cause Drill	**Down beneath a recurrence.** Drills from a present, repeating symptom to the cause driving it — "why does this keep happening?"

After the drill: once you've named a root and a change, take the change into delivery like any other commitment — and if it's a decision the team keeps reopening, lock it so it sticks.

Unlike Five Whys, Root Cause Drill requires evidence before moving to every new cause — that evidence gate is what makes it a drill, not a round of questions.

© Witmer