

The AI Bootcamp

22

Judge AI Output Like an Editor

In this chapter you'll learn

A bar written before you read, a revision loop that converges, and 4 reasons to stop

The AI Bootcamp

THE PROBLEM

Most AI education is a pile of disconnected tips. A LinkedIn carousel here, a YouTube tutorial there, a \$2,000 cohort course that goes out of date before the refund window closes. None of it is built to be read in order, so nothing compounds. You finish twelve videos and can recite twelve facts, not perform one new skill.

THE FIX

The AI Bootcamp is built the other way. 90 skills, packed 2 to 3 per chapter, 36 chapters, 3 tracks. Every chapter states what it needs from you and what it hands to the chapter after it, so skill 47 only works because skill 12 already landed. Read it in order and by the end you are not collecting facts about AI, you are running work through it.

LETTER FROM THE EDITOR

We built this because we kept meeting smart people who felt behind on AI for no reason other than nobody had organized the material for them.

This book does not assume you are technical, and it does not water anything down either. Every chapter earns its place: a real skill, a real exercise, a one-page companion you can flip back to in month three when you have forgotten the details but still need the answer.

If a chapter loses you, or a chapter changes how you work, I want to hear about it either way.

Kris

Editor, AI Central

Write to me at editor@thecentral.ai

THE 3 TRACKS

AI 101

Get running

Pick a tool, set it up, write a clear instruction, use it safely at work.

AI 202

Prompt properly

The framework behind every good instruction, and catching AI when it is wrong.

AI 303

Build without watching it

Connect AI to your tools, and automations that run while you are not looking.

IN THIS LESSON YOU'LL LEARN

By the end of this chapter you can state what good looks like for a piece of work before you generate it, run a revision loop that gets closer instead of wandering, and recognise the moment when editing it yourself or starting again beats asking for one more version.

You have spent 9 chapters getting better answers out of it. Nobody has yet told you how to decide whether an answer is any good, so you have been deciding by feel, and by feel you accept whatever arrives looking finished. The revision loop is the other half of the problem. It is where most people quietly give back the time they saved: 6 rounds in, holding a version worse than round 2, with no way back to it.

CONTENTS	22.1	True, sound and good are three different questions
	22.2	Write down what good looks like before you read a word of it
	22.3	Change one thing, keep the version you liked, and go back rather than forward
	22.4	Stop when describing the change costs more than making it

TOPICS REQUIRED

Chapter 21, and the piece of your own thinking you put through its critique. Bring one more thing with you: a draft the assistant produced that you were not happy with and could not say precisely why. That draft is the working material for this chapter.

AFTER THIS CHAPTER

Chapter 23 assumes you already have a standard. Its catalogue of failures then reads as a list of specific ways a known bar gets missed, rather than a list of things to be generally wary of.

22.1 True, sound and good are three different questions

You now have three ways to interrogate a piece of work and they arrived in three different chapters. Running the wrong one is how a draft passes every check you know and is still unusable.

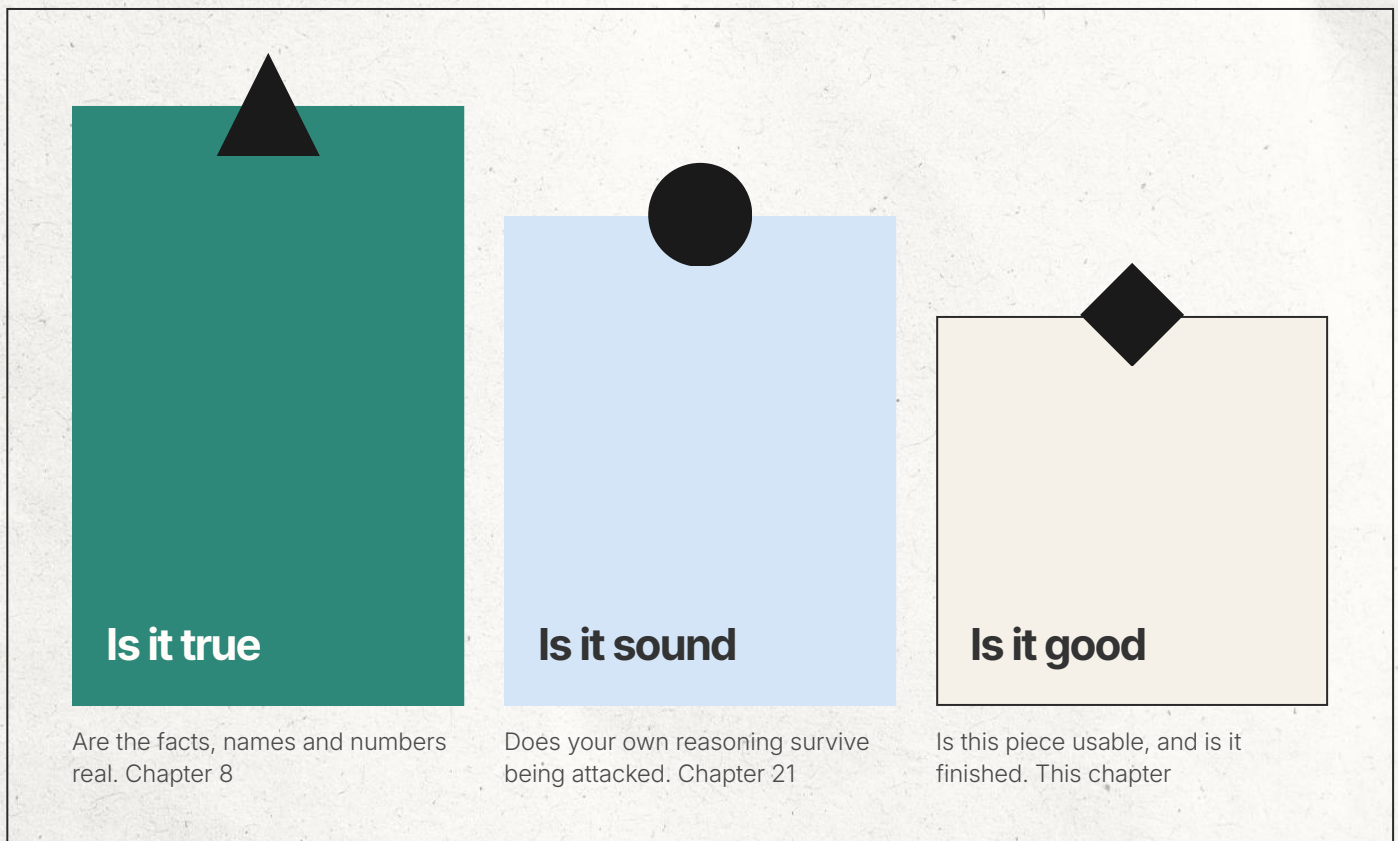


Fig 22.1 Three checks, three chapters, three verdicts that do not substitute for each other. A draft can pass any 2 and fail the third, which is the normal way something goes out broken. The bar heights are visual rhythm, not importance.

22.1

Chapter 8. Verification. A claim is true or it is not, and the answer lives outside the draft.

Chapter 21. Critique. Your reasoning, argued against, before you commit to it.

This chapter. Editing. The piece in front of you, measured against what you needed it to be.

Here is the problem this chapter exists to fix. You have got good at asking, and you have got good at checking. What you still do not have is a way of deciding, on purpose rather than by mood, whether the thing that came back is any good.

Three different questions are involved and they are easy to blur now that you have met all three. Is it true is a question about the world, and the answer is not in the draft: that is Chapter 8, and it is settled by going and looking. Is my thinking sound is a question about you, settled by making it argue the other side: that is Chapter 21. Is this good is a question about the artefact, and it is settled by comparing what arrived against what you needed.

Watch how a piece fails one while passing the others. A summary where every figure checks out, built on an argument that survived a hostile reading, can still be 900 words when the person receiving it has 90 seconds and will read the first line only. Nothing in it is false. Nothing in it is muddled. It is unusable, and neither of the earlier checks would have caught that.

It runs the other way too. A crisp, well-shaped, exactly-right-length brief can be built on an invented statistic. Reading it as an editor will not save you, because it looks like what you asked for. That is Chapter 8's job and this chapter does not repeat it.

An editor does not ask whether they liked something. They ask whether it does the job it was commissioned to do, for the person who will receive it, in the space it has. That is a colder question than it sounds and it is answerable, which is the whole point: good here is not a feeling, it is a small set of conditions you can write down in advance and then check against.

Before you read on, go back to the last draft you accepted from an assistant and used. Ask which of the 3 questions you actually ran on it. Most people find they ran one, quickly, and called it judgement.

22.2 Write down what good looks like before you read a word of it

Read the draft first and you will end up grading it against itself. The standard has to exist before the output does, or the output becomes the standard.

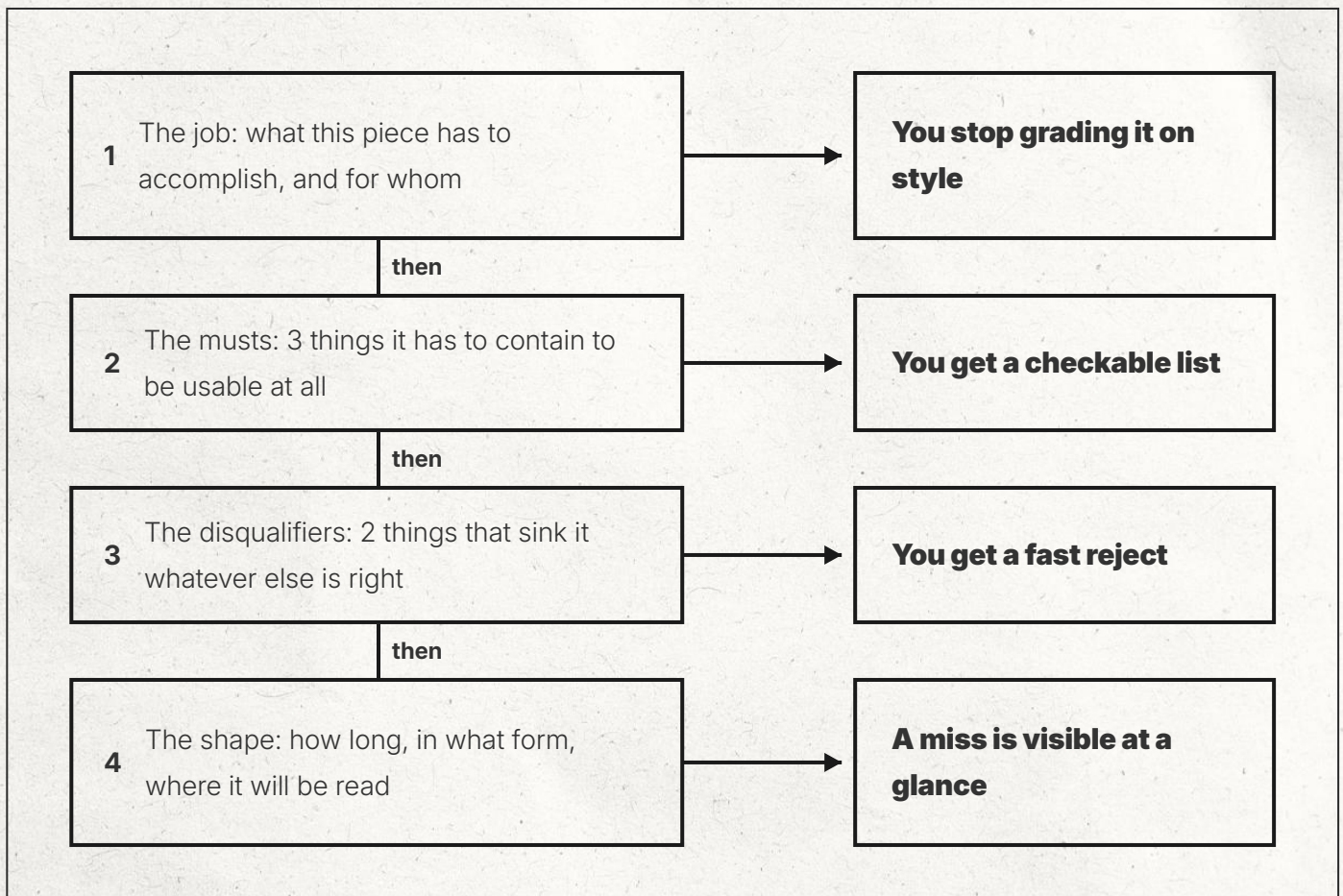


Fig 22.2 Four lines, written before you generate anything. Notice that 3 of them could be checked by a stranger who has never met you: that is what separates a bar from an opinion about a draft.

22.2

Write it, do not think it. A bar you only keep in your head bends to fit whatever turns up. Four lines on paper does not bend.

Name the disqualifiers. Say what makes it unusable, not only what would make it better. Rejects should be cheap and fast.

It doubles as the ask. Most of the bar belongs in the instruction anyway. Chapter 13's framework and this are the same 4 lines used twice.

Almost everyone works in the same order: generate, read, react. The trouble is that reacting is comparative, and once you have read the draft the only thing available to compare it against is the draft. So you grade it against itself, notice the two things you would change, change them, and call it done. The standard you applied was set by what you were handed.

There is a measurement of how strong that pull is, and it is worth knowing even though it was taken on machines rather than people. An August 2026 study fed a prior score to 8 assistants as nothing more than context metadata and asked them to judge independently. Their ratings moved toward the number they had been shown, with an effect size up to 0.71. On one dataset the anchor blocked 48% of the corrections the model would otherwise have made and flipped just over 10% of its correct judgements to the wrong label. Telling it in the prompt to disregard the metadata reduced the effect and did not remove it. If a stray number does that to a system that was explicitly instructed to ignore it, be honest about what a finished-looking draft does to you.

The fix is 4 lines, written before anything exists to be influenced by. What the piece has to accomplish and who receives it. Three things it must contain to be usable at all. Two things that disqualify it however good the rest is. And the shape: length, form, and where it lands. For a customer update that might read: it has to stop them calling support; it must name the cause, the fix, and the date it is fixed by; it is dead if it blames the customer or if it uses the word unfortunately; 120 words, plain sentences, readable on a phone. You now have something to check rather than something to react to.

The obvious next thought is to make the assistant apply the bar for you, and that is worth doing for exactly one kind of question. Given an explicit, checkable list, it is a decent inspector: does every claim carry a number, is this under 120 words, is there a named next step. Given the open question of whether the piece is any good, it is a poor arbiter, and the reasons are specific. It rates longer answers higher: controlling for output length raised one leading automatic evaluator's agreement with human rankings from 0.94 to 0.98, which tells you how much length was carrying. It rates its own writing higher than humans do, and a June 2026 result found that simply labelling who wrote something moves the score in both directions with no change in quality. And it rewards fluency: on 3,563 rated peer reviews examined in July 2026, length and venue explained most of the quality gap, and the verdict was not detectably aligned with the rubric's own measures of analytical quality. Confident, well-turned prose scores well for being confident and well-turned.

One more number, because the optimistic figure gets quoted a lot. Strong assistants do reach over 80% agreement with human preference, about the level humans reach with each other. But an evaluation of 21 judge models from 9 providers, published in June 2026, found that raw agreement overstates real discrimination by 33 to 41 percentage points once you correct for the agreement you would get by chance. The same study found 2 production judges repeating themselves at above 0.95 while carrying severe position bias. Consistent is not the same as correct, and a machine that gives you the same wrong verdict twice will feel more trustworthy than one that hesitates.

So use it as an inspector, not a critic. Hand it the bar as a numbered list and ask for a verdict on each item with the sentence that proves it quoted back, rather than a score out of 10. Work published in September 2026 found that judging from extracted evidence rather than the raw text beat the next best approach by 23%, which is the same discipline stated formally. Judging output that no human ever reads is a different problem again, and it belongs to Chapter 34. Write your 4 lines for the next real thing you need before you open the assistant at all.

22.3

Change one thing, keep the version you liked, and go back rather than forward

The revision loop is where the time you saved goes. You ask for a fix, get a version that fixes it and breaks something else, ask again, and 6 rounds later you are holding something worse than round 2 with no way back to it.

Layering

Every instruction is typed on top of the last version, in one thread, and the thread holds the only copy.

TRAITS

- Each round starts from the round before it
- Several changes per message, so causes blur together
- The version you liked exists only in the scroll

PROS

- + Feels fast, and genuinely is for the first 2 rounds
- + Nothing to copy out, nothing to keep track of

CONS

- A fix in one part quietly rewrites another
- When round 6 is worse than round 2, round 2 is gone

Branching

The best version so far lives outside the thread, and every change starts from that version, one at a time.

TRAITS

- One change per round, named out loud
- The best version sits somewhere you control
- What must not change is restated every round

PROS

- + You can always return to the last version you liked
- + You can tell which instruction caused which change

CONS

- Costs you a copy and a paste each round
- Slower than layering until about round 3

Fig 22.3 The same 6 rounds spent two ways. The difference is not how well the instructions are worded: it is whether a version you liked survives the round that comes after it.

22.3

One change. Three requests in one message gets you a rewrite, and no way to tell which request caused what.

Name the untouchables. "Keep the opening line and every number exactly as they are." Otherwise a fix in paragraph 4 edits paragraph 1.

Do not ask it to undo. Paste the version you liked back in as the new starting text. Undo is not an operation it can perform.

Two things are happening at once in a wandering revision loop, and it helps to separate them. The first is that a long thread is a worse place to work than a short one. In a study of over 200,000 simulated conversations across 6 generation tasks, published in May 2025, performance was on average 39% lower when the same work was spread over several turns than when it was asked in one go. The researchers broke that down and found only a small loss of raw capability. Almost all of it was a rise in unreliability. Their summary is the sentence to remember: when it takes a wrong turn in a conversation, it gets lost and does not recover. It fixes on an early reading of what you want and keeps leaning on it long after you have said something that contradicts it.

The second thing is that it will not reliably notice its own regression. There is a well-known 2023 result where a model critiquing and rewriting its own answers improved across 7 tasks by around 20%, and it gets quoted as though repeated self-revision is free progress. What followed does not support that. A 2024 paper from Google DeepMind found that without feedback from outside itself, a model struggles to self-correct and sometimes gets worse. Another found that on a task with a checkable answer, whether the critique was any good barely affected whether the loop improved anything. An August 2026 paper puts it plainly: unconstrained self-correction may preserve, amplify, or introduce errors. Revision pays when something outside the loop says what is wrong. In your work, that something is you.

Which gives you 4 rules, and the first is one change per round. If your message contains and also, you have asked for two things, and what comes back will be a rewrite in which both requests are partly served and you cannot attribute anything. Ask for the single most important change. Read what happened. Then ask for the next one.

Second, keep the version you liked, somewhere that is not the thread. A document, an email to yourself, a note. This takes 4 seconds and it is the rule people skip, because in the moment round 3 always looks like it is going to be an improvement on round 2. Sometimes it is not, and by then round 2 is 40 messages up a scroll and mixed in with 3 versions that look almost the same.

Third, go back rather than forward. When round 4 is worse than round 2, the instinct is to ask it to put back what it removed, and that is a fifth layer on 4 layers of drift. Paste round 2 in as the starting text, say what you want changed about it, and let round 5 be a branch off the good version rather than a patch on the bad one. Fourth, name what must not change, every time. Keep the opening exactly as it is and change only the third paragraph is a sentence that costs you 12 words and saves an entire round.

The cheapest of the 4 is the copy-out, so start there. Whatever draft you have open now, find the best version of it in the thread and paste it somewhere you own before you ask for anything else.

22.4

Stop when describing the change costs more than making it

The loop has no natural end, so you have to supply one. There are 4 reasons to stop and only 1 of them means the piece is finished.

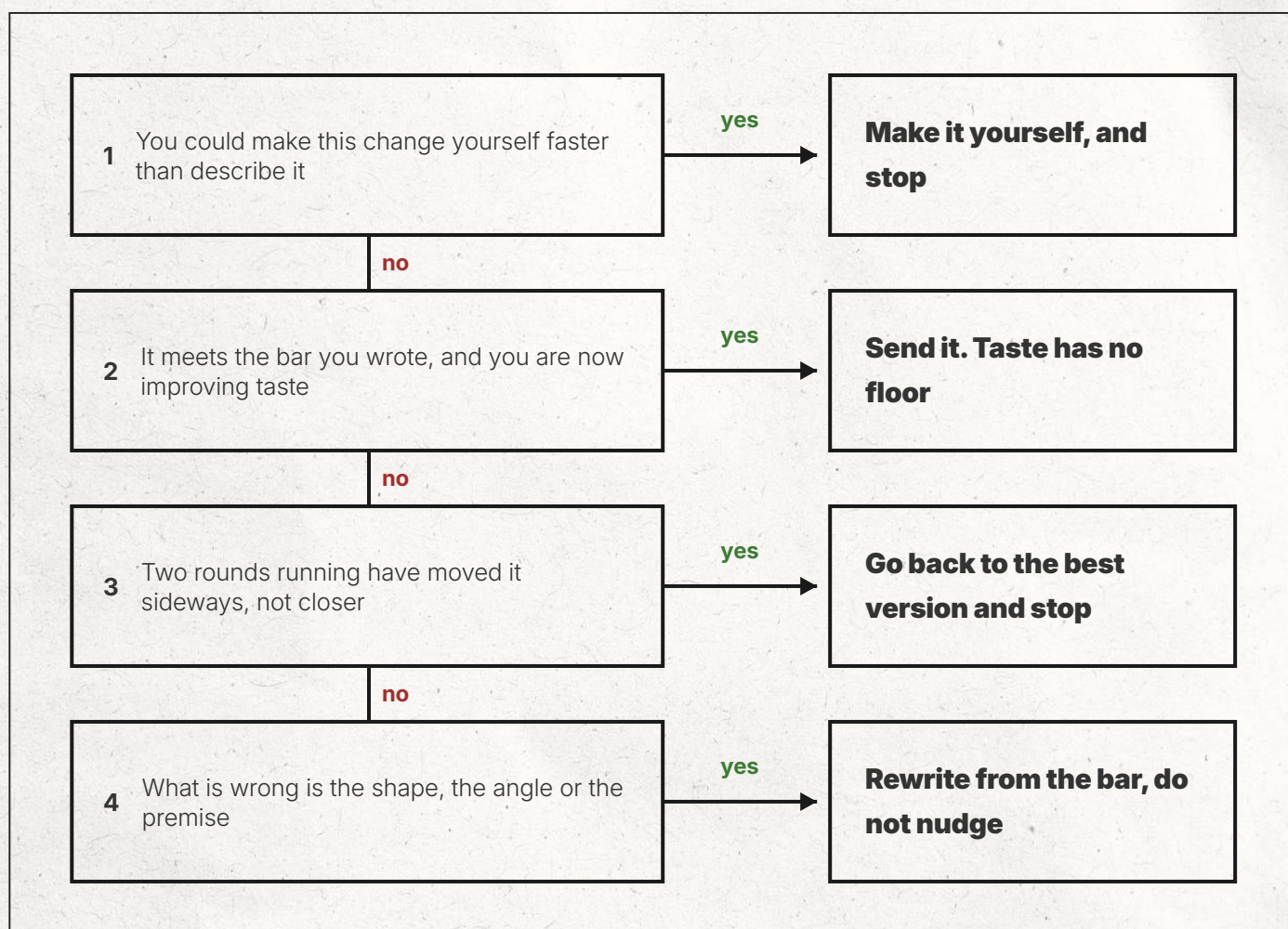


Fig 22.4 Read down and stop at the first line that is true. Only the second means the work is done: the other 3 mean the loop has stopped paying, which is the more common and less obvious reason to get out of it.

22.4

The crossover. Small, local, taste-level changes are faster to type than to specify. Systematic ones are the opposite.

80% is not close. If the wrong 20% is the structure, the work remaining is not 20% of the job.

A dead draft still paid. It almost always shows you which line of your bar was missing. Add the line and ask again.

Nobody tells you when to stop, so people stop when they get bored or when they run out of patience, and both of those arrive later than the point at which the loop stopped being worth running. There is formal work on this: an August 2026 paper models refinement as a stopping problem and computes when the expected gain from another round no longer covers its cost. It assumes a verifier scoring each round. You are the verifier, so you need a test you can actually run in your head.

Start with the crossover, because it is the one senior readers get wrong most consistently. Describing a change precisely enough for someone else to execute it costs a sentence or two of careful writing. Making it yourself costs a keystroke. For a wrong word, a sentence that should go, two paragraphs in the wrong order, you lose by asking: by the time you have written change the third sentence so it leads with the number rather than the caveat, you could have changed it 4 times. Handing it back is worth it when the change is systematic rather than local. Every section needs a named owner and a date runs 9 times off one instruction, and that is a trade you win.

The second stop is the reason you wrote a bar in the first place. It meets the 4 lines. Anything further is taste, and taste has no floor: there is always a better verb. If you kept going past this point on the last 3 things you made, that is not diligence, it is the absence of a stated finish line, and you now have one.

The third is the sideways test, and it is the practical version of everything in 22.3. After each round, ask one question: is this closer to the bar, or just different? Different-not-closer twice running means the loop has stopped converging and more rounds will not restart it. Go back to the best version you kept and stop there. It will feel like giving up on an improvement that was one message away. It almost never was.

The fourth is the one worth most to you. A piece that is 80% right sounds nearly finished, and that is only true when the missing 20% is sentences. When what is wrong is the structure, the angle, or the premise it was built on, nudging produces exactly what you would expect: the wrong piece, with better sentences, after 5 rounds. The test is whether you can name the change as an edit or only as a description of a different piece. If it is the second, open a new thread, take your 4 lines, add the one you now know was missing, and ask again from nothing. The failed draft was the research.

You are allowed to throw a draft away. Doing it at round 3 instead of round 9 is the whole skill, and it gets easier once the bar is written down, because you can point at the line it will never meet. Go back to the draft you brought into this chapter and decide which of the 4 lines applies to it. Then take that decision, and the draft, into Chapter 23.

READING ENDS HERE

Everything before was reading. Everything after is doing.

3 exercises, About 20 minutes and one real decision that opens Chapter 23.

Do these before Chapter 23

3 exercises. About 20 minutes

Pick one real piece of work you have not asked for yet. Write the bar before you open the assistant. Nothing exists to be influenced by, which is the entire point of doing it in this order.

THE JOB, AND WHO
RECEIVES IT

3 THINGS IT MUST CONTAIN

2 THINGS THAT DISQUALIFY
IT

Now generate it, and score it against your bar without editing a word. Write down the line it failed, and only that line. If it passed all 3 boxes, your bar was too low: write a harder one and try again.

Run exactly one revision round on the line it failed, and nothing else. Then write one sentence: closer to the bar, sideways, or worse. If it is worse, paste the previous version back in and stop there.

Chapter 22 on one page

THE THREE QUESTIONS

Is it true, Chapter 8 · is my thinking sound, Chapter 21 · is this piece good and finished, here. Passing 2 of 3 is the normal way something goes out broken

THE BAR, IN 4 LINES

The job and its reader · 3 things it must contain · 2 things that disqualify it · the shape, length and where it lands. Written before you generate

THE 4 LOOP RULES

One change per round · name what must not change · keep the best version outside the thread · branch off it rather than layering another fix on the bad one

WHAT A SELF-SCORE IS WORTH

Good as an inspector against a checkable list you wrote. Poor as a critic: it rates longer, more confident and more familiar-sounding writing higher, its own included

THE 4 REASONS TO STOP

Faster to make the change than describe it · it meets the bar · 2 rounds moved sideways · what is wrong is the shape, so rewrite from the bar instead of nudging

BRING TO CHAPTER 23

Your written bar and the draft you judged against it. Chapter 23 takes that same draft and names what actually went wrong inside it, which is a question you can only ask once you know what it was supposed to be.

BOTTOM LINE

You can define good before you look, run a loop that converges instead of wandering, and tell a piece that needs one more round from a piece that needs starting again.

NEXT CHAPTER

How to Fix a Bad Answer

Chapter 23 turns the standard you now apply into a catalogue. The specific ways it goes wrong: refusals, drift across a long thread, padding, invented sources and flattery, and the tell that gives each one away before you have read to the end.

BRING WITH YOU

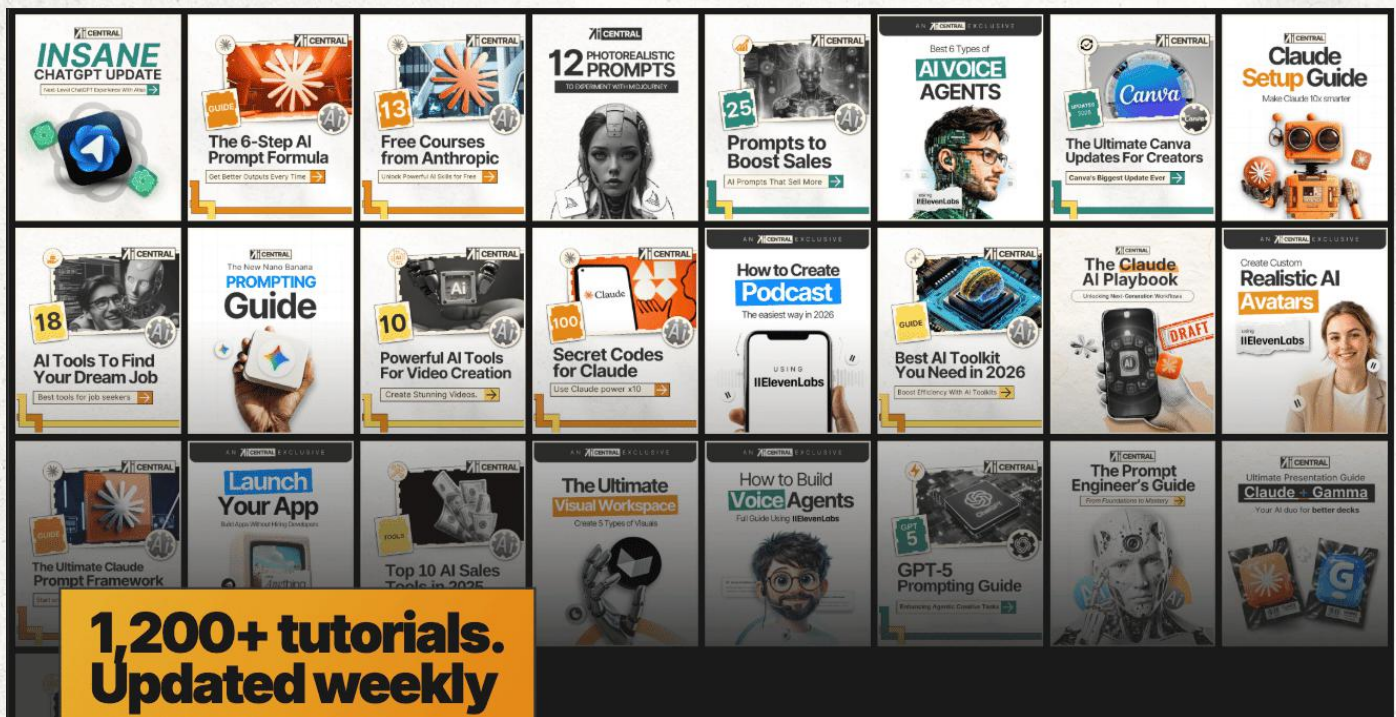
- ☐ The 4-line bar you wrote in this chapter
- ☐ The draft you rejected, and the line of the bar it failed

GO FURTHER

The Ultimate AI Library

Every guide, prompt pack and playbook AI Central has published, in one place. The reference shelf this course keeps pointing you to.

explore the library



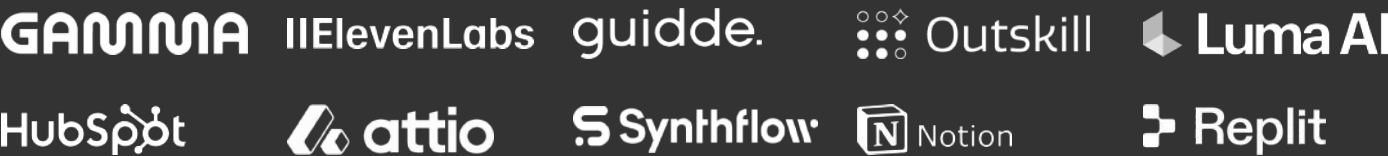
WHO MAKES THIS



AI Central Media is a London-based editorial team. Our flagship publication covers practical AI for senior professionals, and 100+ AI companies, SaaS platforms and growth teams have advertised with us since 2023.

100+ AI companies, SaaS platforms, education brands and growth teams have advertised with us since 2023	613K Accounts reached every month, across the 7 channels we operate	151 Countries our readers are in, plus all 50 US states
---	---	---

PROUD PARTNER OF



We turn attention into pipeline.

visit thecentral.ai

➤