

The whole method, four steps

STEP 1 Pick one problem Small, specific, and it slows you down. Not "fix my shop." The one that gets you every Monday.	STEP 2 Teach it your business Your data, your rules, your words. What you would tell a new employee.	STEP 3 Build small, check it It will not know your shop yet. You are the one who can tell it what it is missing.	STEP 4 Fix it and repeat Corrections make it smarter. Then pick the next problem.
--	--	--	---

Three rules that made it work

Set your standards up front. It holds them from then on. What that looked like: We said on day one that every number has to show where it came from. That rule is now a button called Show the Math: click any figure and see the arithmetic behind it.	See it work on the real screen before you trust it. What that looked like: Our stale-data warning tested perfectly and would almost never have appeared in real life, because it read delivery dates and wedding orders are dated weeks ahead. Only the real screen showed that.	Write everything down. What that looked like: Every time Cameron overruled a recommendation, what he changed and why got written down. His reasoning now sits next to next week's order instead of in his head.
--	---	--

Your first 30 minutes, tonight

- 1. **Pick the one problem** that slows you down every single week. Small and weekly beats big and someday.
- 2. **Write down what you would tell a new employee** about that problem: how you decide, your rules of thumb, what goes wrong. That page is your shop's knowledge, on paper for the first time.
- 3. **Read your page to an AI assistant** and ask what it would need to help. It will ask you questions. Answer them like you are training a promising new hire.

Words that work: starter prompts

- "Here is how my shop handles ____ today. Ask me questions, one at a time, until you understand how I decide."
- "What would you need from me to help with this every week?"
- "Show me your math. Where did that number come from?"
- "That does not match what I see in my shop. Here is what I actually have."
- The interview question that started Norton's: "What is in my head that my shop manager would need to know to run the shop without me for two weeks?"

What you say on day one

Set these once. They hold.

- "You recommend. I decide."
- "Every number has to trace back to a file I can open."
- "If the data cannot answer, say so and tell me what would fix it."
- "Do not make my team type anything they are not already typing."

Data you already have

Nothing new to create. You keep these already.

- Point of sale exports (sales history)
- Supplier and wholesaler invoices
- Price sheet, recipes, order history
- The waste bucket. Start writing down what you toss.

We used Claude. The method works with any good AI assistant. It took real effort up front, and it pays it back every week. You know how you eat an elephant: one bite at a time. This sheet is the first bite.

What the conversation actually looks like

No code, no forms, no special software. Three real moments from this build, with dates, taken from the project record and shortened to fit.

JUNE 10 · THE FIRST CONVERSATION WAS NOT WITH THE AI

Joe to Cameron and Dana, before any data, any software, or any decision about tools.

Joe What is in your head that your shop manager would need to know to run the shop successfully without you? If you got to take a real two-week vacation, what would you have to write down first?

Joe Then start writing the job description for your AI, just like you would for a new hire. What you write becomes the actual instructions we load into it, word for word, so the detail you put in is the employee you get out.

Cameron We will be buying based on need and not just on a feeling. We will be operating the business on what we need and not just what we think.

The move: *Two weeks of this came before anything was built. Cameron and Dana never had to learn a technical word. They had to answer questions about their own shop, which they already knew.*

JULY 3 · THE RULES, SET BEFORE ANYTHING WAS BUILT

Joe to the AI. These four rules were written once, in plain English, and they have held ever since.

Joe Every number has to trace back to a file I can open. Never invent a number to please me, Cameron, or Dana. If the data cannot answer the question, say so plainly and tell me what would fix it. You recommend, we decide.

AI Understood. I will show the source and the math behind any figure, and say when the data is too thin to answer instead of estimating.

Joe This cannot be a black box. Cameron only hands over his ordering if he can check it.

The move: *Say the standard out loud before you ask for anything. That rule is now a button in the app called Show the Math: click any figure and see the arithmetic behind it.*

JULY 28 · CAMERON WAS RIGHT AND THE TOOL WAS WRONG

The dashboard reported that about 41 percent of the week's projected units had no recipe link.

Cameron Everything in my recipe builder is tagged.

Joe He was right, and it was on our side, not his. His register sells the parent product, Loose Flowers, while his builder tags the versions, Loose Flowers Stock. The matching was never connecting the two. Untagged went from about 41 percent to about 2 percent, and what was left was a genuinely useful list of things not in the builder yet.

The move: *He did not need to know anything technical. He knew his own products, and that was enough to find it. When the tool disagrees with you, say so out loud. You are usually the one who knows.*

Notice what is missing: no technical words, and no right answers anybody needed in advance. Every one of these is somebody knowing their own shop and saying so plainly. That is the job only you can do.

THE ONE PROBLEM I AM GOING TO START WITH
