AI Rollout Playbook

By Marcus Turner · November 2026

How to take any business from "we should use AI" to a tested, adopted tool, in 8 repeatable steps. The worked example throughout is Coastal Care Family Medicine, a fictional clinic with made-up data, where all 8 steps were run end to end.

The 8 steps at a glance

Steps 1–4 are discovery: understand the business before you touch any AI. Steps 5–8 build, test and roll out one use case, then loop back to step 3 for the next. Times are for one use case at a small business.

Step	What you do	What you produce	Coastal Care example	Typical time
1. Understand the business	Learn what it does, its scope, size and systems	A one-page business profile	Clinic profile	1–2 hours
2. Map the roles	Who does what, and who hands off to whom	A roles and contacts table	Team and escalation table	1 hour
3. Find the pain points	Interview, log questions, map the as-is journey, measure the baseline	Audit notes, a question log, an as-is map and baseline numbers	Workflow audit and as-is map	1–2 weeks
4. Pick one use case	Gate out what's not allowed, then score the rest	A scorecard with a clear first pick	Use-case scorecard	1–2 hours
5. Write the rules	Turn knowledge in people's heads into SOPs, with owners and a never list	SOPs for the use case's scope	SOPs 1–5	1–2 days

Step	What you do	What you produce	Coastal Care example	Typical time
6. Build and test	Set up the AI tool, test it with real questions, fix the source or the method, retest	A working tool and a test log	Claude Project, Skill and a 23-question test log	1–2 days
7. Roll out and train	Quick-start guide, training session, a named owner	Training materials and a rollout plan	Quick-start guide, rollout plan and feedback log	1–2 weeks
8. Measure and improve	Compare against the baseline; decide the next use case	An adoption tracker and before-and-after results	Adoption tracker	Ongoing

Checklists for each step

Each step ends when its boxes are ticked. Copy this section into a new document at the start of each project.

1. Understand the business

- [ ] What it sells or does, for whom, and how big it is (staff, customers, volume)
- [ ] Hours, locations and the systems it uses
- [ ] Who the decision-maker is, and who approves new tools

2. Map the roles

- [ ] Every role in scope, with what it owns
- [ ] The handoffs: who passes work to whom
- [ ] The escalation path: who answers when someone is stuck

3. Find the pain points

- [ ] Interview the person who feels the problem, the one who absorbs it, and the one who pays for it
- [ ] Log real questions or errors for 1–2 weeks, grouped by topic
- [ ] Draw the as-is journey, with the pain points marked where they hit
- [ ] Write down the baseline numbers you'll measure against

- [ ] Record the constraints: what the tool must never do, and who must approve it

4. Pick one use case

- [ ] List 4–6 candidate uses
- [ ] Gate first: anything that breaks a constraint is out
- [ ] Score the rest on weighted criteria the decision-maker cares about
- [ ] Pick the top one, and note the runners-up as next use cases

5. Write the rules

- [ ] Split the work into SOPs: one trigger to one outcome, cut at handoffs
- [ ] Give every SOP an owner, numbered steps, tables for rules, and a named person for each escalation
- [ ] Write the never list
- [ ] Have the owner settle every decision point

6. Build and test

- [ ] Knowledge files, instructions (client-specific) and a Skill (reusable method)
- [ ] A smoke test: one normal lookup plus one test per guardrail
- [ ] A full test log of 15–20 questions: lookups, thresholds, ambiguity, cross-document questions, not-covered questions and pushback
- [ ] For each miss, decide whether the fix belongs in the source or the method; fix it, update the files, and run the regression test
- [ ] Record the first-try pass rate, the final pass rate and any known gaps

7. Roll out and train

- [ ] A one-page quick-start guide: what it's for and not for, how to ask (the situation, not the person), how to read an answer, the rules, and what to do when it's wrong
- [ ] An owner (champion) and a backup, written into the roles so the duty survives turnover
- [ ] A short training session, mostly hands-on with real situations, including a guardrail drill
- [ ] A phased launch: a small pilot, then everyone, then standard for new hires
- [ ] A low-friction feedback loop: the tool writes a fixed-format entry, staff add it to a shared log, and the owner reviews it weekly
- [ ] A decision on where the freed time goes, made with the manager

8. Measure and improve

- [ ] Measure the same baseline numbers weekly after launch
- [ ] Track adoption (weekly users), the feedback logged and fixed, and the problems prevented
- [ ] Show before and after honestly: label simulated or partial data, note small samples, and expect results to level off rather than reach zero
- [ ] Report the value through the leader's lens (people, customers or money)
- [ ] Choose the next use case from the scorecard's runner-up list

Reusable templates

Each template is a shape to copy, not content. Start from the Coastal Care version and swap in the new business's details.

Template	Its shape
Interview guide	5 open questions for each of 3 roles. The person who feels it: their first weeks, most-asked questions, what they did when stuck, a mistake, what they wish they'd known. The one who absorbs it: how they train, repeat questions, volume and peak times, costliest mistakes, what's out of date. The one who pays: the cost, what "ready" means, success in 6 months, what the tool must never do, who approves, and "if this person got that time back, what would you want them doing?"
Question log	Topic, count, share, an example question, and which document answers it
SOP	Owner and approver; purpose and the 5 key rules; tables for rules; numbered steps; special cases; an escalation table with named people; a never list (in the shared SOP)
Scorecard	A pass/fail gate column, then weighted 1–5 criteria (impact, frequency, safety, ease) and formulas for score and rank

Template	Its shape
Project instructions	Who it helps; safety rules first; answer only from the files, otherwise escalate; which Skill to use, with a fallback; how to write a feedback entry and where it goes; tone
Skill	The reusable answer method: checks in order (safety, private data, source, ambiguity), a fixed answer format, and a never list. It names no client, so it carries over to every project
Test log	Question, expected key points, result (Pass, Partial, Fail), what went wrong, fix made, area and what it tests, plus first-try and final pass rates

Rules of thumb

These are the lessons from Coastal Care that apply anywhere. Each one came from a real decision or a real test result.

1. **Audit before you write rules.** The audit tells you which rules are worth writing.
2. **The problem is usually access, not knowledge.** 86% of new-hire questions were already answered somewhere. The win was making answers findable.
3. **One SOP runs from one trigger to one outcome.** Split where the work changes hands, where it runs past about 10 steps, or where parts change at different speeds.
4. **Every rule names a number or a person.** "Ask the front desk lead" works. "It depends" leaves the AI guessing.
5. **Gates before weights.** Rule out what isn't allowed first, then score what's left.
6. **Privacy by design.** Build the tool so it never needs sensitive data: staff describe the situation, never the person.
7. **Fix the source before the AI.** Most misses (4 of 6 fixes) were gaps in the SOPs, not assistant errors.
8. **Re-test after every fix.** That's regression testing: a fix shouldn't break what already worked.
9. **Tell a one-off from a pattern** before you change anything.
10. **Test the guardrails hardest:** emergencies, private data, questions the documents don't cover, and pushback.

11. **Testing the AI also audits your documents.** The assistant found a conflict between SOPs and a missing procedure.
12. **Write down known gaps.** An honest open item beats a fake-perfect result.
13. **Test the whole workflow, not just the answers.** A perfect answer still fails if staff can't find where to log feedback.

Turning time saved into value

Time saved isn't value until it's spent on something the business cares about. Decide where the freed time goes during discovery, then measure it.

- **Ask early:** "If this person got 2 hours a week back, what would you want them doing?"
- **Make prevention visible:** prevented problems are invisible, so count them. At Coastal Care that meant insurance flags fixed before the visit and booking errors caught the day before.
- **Speak the leader's language:** frame the same result through the lens the decision-maker cares about most. That might be people (stress, turnover), customers (waits, safety) or money (denied claims, overtime, extra shifts).
- **Go a layer deeper:** put a dollar value on each prevented problem and each week of faster ramp-up, using the business's own figures. Never use guessed numbers.

How real projects differ

The Coastal Care example ran cleanly because every input was controlled. Real projects take longer and get messier, mostly in discovery and adoption, not in the build.

In the worked example	On a real project	What to do
Interviews and data were simulated	People are busy, answers conflict, and the data surprises you	Book interviews early. Write down disagreements instead of smoothing them over. Let the data change your pick
SOPs were written fresh	SOPs are missing, outdated or contradict how people actually work	Start from how the work really happens, and have each owner sign off
Policy decisions were made on the spot	Decisions wait on managers, compliance or legal	Name the decision-maker in step 1, and list open decisions in a tracker

In the worked example	On a real project	What to do
Approval was assumed	Privacy, IT and security reviews take weeks	Ask "who approves?" in the first interview, and start reviews early
Everyone wanted the tool	Some staff resist, or quietly ignore it	Name an owner, train in the team's own workflow, and show quick wins
Scope held to one use case	People ask for more mid-project	Point new ideas to the scorecard's runner-up list for the next round
Discovery took hours	Discovery takes 1–2 weeks	Budget for it. It's the part that decides whether the rest works

When you're on your own: follow the checklists in order, use the templates, and keep a test log from day one. When something doesn't fit the playbook, write down what you did and why, then update the playbook.