

Fionn Tobin

The PSA framework: build your first real AI automation

The three-step framework I use with clients: Prompt, Skill, Automate.
Dial in one prompt, turn it into a skill, schedule it in Claude Cowork.
Full example inside.

who it's for

complete beginner

time needed

1 week

cost

free

a free guide

Everyone wants AI to automate the boring, repetitive parts of their business. Almost no one shows you the easy way to actually do it.

Most people start with a workflow tool, wire seven boxes together, and end up with an automation that produces something nobody trusts. So they check its work every time, which means they saved nothing.

Here's the framework I use with clients instead. I call it **PSA: Prompt → Skill → Automate**. Three steps, in that order, no skipping.

The framework in one line

Get one prompt to 100%, freeze it as a skill, then put the skill on a schedule. The order is the whole trick: you only automate something after it already works every single time.

We'll build a real one together: a morning brief that reads your calendar and email and tells you what today actually looks like. By the end you'll have it landing on its own, every weekday, before you've opened your laptop.

Step 1: prompt

Don't start with a workflow tool. Start with a single prompt in Claude.

Write the first version, run it, and grade it honestly. It'll be about 60% good. That's fine, that's the starting point. Change one thing, run it again tomorrow. 70%. Then 80. Then 90. Don't move on until it reliably does exactly what you want, every time, with no editing from you.

The mistake is automating at 80%. An automation multiplies whatever you feed it, so a prompt that's slightly off once a day is slightly off 250 times a year.

Here's the daily brief prompt to start from. Fill in the brackets, and make sure your email and calendar are connected first ([the setup guide](#) covers connectors):

```
You are my chief of staff. Every time I run this, produce my morning brief for today.
```

```
My job: [your role and company, one line]. My priorities this quarter: [the 2 or 3 things you're actually trying to move].
```

```
Pull from my connected calendar and email. If either isn't connected or returns nothing, say what's missing instead of guessing.
```

```
Build the brief in this exact order:
```

```
1. TOP 3. The three things that matter most today, one line each. Pick them from my calendar, my
```

unanswered emails, and my priorities above. If more than three things feel urgent, the rest go in section 5, not here.

2. CALENDAR. Today's meetings in time order. For each one: time, who, what it's about, and one line of prep I'd be glad I read ("their last email asked about pricing, have an answer ready"). Flag any clashes or back-to-backs with no break.

3. NEEDS A REPLY. Emails from the last 3 working days that are waiting on me, newest first. For each: who, what they want, and one line on the angle you'd take in the reply. Skip newsletters, receipts, notifications, and anything I'm only cc'd on.

4. WAITING ON. Threads where I'm the one waiting and it's gone quiet for 3 or more days. Who owes me what, how long it's been quiet, and a one-line nudge I could send.

5. WORTH KNOWING. Anything else it would embarrass me not to know today: a deadline buried in an email, a meeting that moved, [anything specific you always want watched].

Rules:

- Each section fits on one phone screen. No padding, no pep talk.
- Never invent a meeting, email, or name. If you're not sure something is real, leave it out or mark it "check this".
- All times in [your timezone].
- Bold the names and times so I can scan it.
- End with one line: the single thing that, if it's all I do today, makes today a win.

Now run it every morning for a week. Each time something's off, don't fix the output, fix the prompt. Section 3 keeps catching newsletters? Add the sender to the skip list. Prep notes too vague? Add an example of a good one. That's the climb from 60 to 100, and it usually takes about a week of real mornings.

Step 2: skill

Once the prompt is dialed in, turn it into a **skill**: a repeatable version of that prompt Claude can call on demand, with access to whatever tools and connectors it needs.

This is the step almost everyone skips, and it's why their "automations" break constantly. A prompt lives in a chat somewhere; you retype it, you paraphrase it, it drifts. A skill freezes the version that works.

You don't need to write any code. Claude has a skill creator built in, and you build the skill by talking to it. Open a new chat and paste this:

Use the skill creator to turn the prompt below into a skill.

Name: daily-brief

When to use it: whenever I ask for my morning brief, my daily brief, or what today looks like.

It needs access to my email and calendar connectors.

Keep the prompt's structure and rules exactly as written. That includes the section order, the skip rules, and the "never invent anything" rule. Those took a week to get right.

Ask me anything that's unclear before you build it.

Here is the prompt, exactly as I run it every morning:

[paste your dialed-in prompt from step 1]

Claude will ask a few clarifying questions, then build the skill and hand you the file. Save it, then turn it on: in Claude it's **Settings → Capabilities → Skills**, in Cowork it's **Customize → Skills** in the left sidebar.

Test it before moving on. Open a fresh chat and type "run my daily brief". You'll see "Using daily-brief" appear while Claude works. If the output matches what your week of tuning produced, the skill is good. If not, ask Claude to adjust the skill and test again.

Step 3: automate

Now the payoff. Open **Claude Cowork**, run the skill once by asking for your brief, and when you're happy, type **/schedule** in the chat.

Tell it what you want in plain English: "Run my daily-brief skill every weekday at 8am and leave the finished brief for me." It'll confirm the details with a couple of quick questions, you hit Schedule, and that's it. There's also a manual route if you prefer forms: **Scheduled** in the left sidebar → **New task** → **Set up manually**.

Two things worth knowing:

- **It runs in the cloud.** Laptop closed, asleep, in a bag: the brief still lands on schedule. Everything shows up on the Scheduled page, where you can pause it, edit it, or run it on demand.
- **Scheduled runs use your connectors and the files in your Claude account**, not folders on your computer. Our daily brief reads email and calendar through connectors, so it's built for exactly this.

You'll need a paid Claude plan, and scheduling is still rolling out, so if you don't see it yet, check back in a week or two.

Getting the frequency right

The whole game from here is matching the schedule to the decision it feeds. Some anchors:

- **Daily** (or weekdays): anything you'd otherwise check every morning. Your brief, yesterday's sales, overnight support messages.
- **Weekly**: anything that feeds a weekly rhythm. Pipeline recap before Monday's meeting, a Friday summary of what shipped.
- **On demand**: anything triggered by an event rather than a date. Keep it as a skill you fire when the moment comes: "prep me for this call", "summarise this thread".

If you're reading an output and thinking "nothing's changed since last time", the cadence is too fast. If you're surprised by what's in it, too slow. Adjust on the Scheduled page and move on.

Once you've nailed that, you've built your first real AI automation. Then the framework repeats: next boring task, new prompt, week of tuning, skill, schedule. Two or three cycles in, your mornings look very different.

The one rule

Only automate at 100%, and start with work that flows **to you**, not from you. Briefs, summaries, recaps, research: schedule freely, the only person reading is you. Anything that leaves the building, an email to a client, an invoice, a post, stays human-sent. The automation drafts, you press send.

This guide lives at <https://fionntobin.com/guides/psa-framework> and gets updated when the tools change. More free guides: <https://fionntobin.com/guides>