

ICP Self-Validating Scrape

Instead of accepting whatever the scraper returns, make the agent grade its own list against your ideal customer profile and rescrape until it passes.

01 When to use this

Use this any time you build a lead list. It is the most novel prompt in the course and the pattern generalizes far past scraping. Reach for it whenever your manual quality check has a number in it, because that number can be written into the skill.

02 The prompt

I want you to build me a lead list, and I want you to check your own work before you hand it to me.

My ICP is:

- Roles: <founder, CEO, co-founder, owner, co-owner, partner>
- Industry: <INDUSTRY>
- Company size: <HEADCOUNT_RANGE>
- Location: <GEOGRAPHY>
- Must have: <DELIVERABLE_EMAIL, PHONE, OR OTHER HARD FILTER>
- Disqualifiers: <WHO IS NOT A FIT AND WHY>

The loop:

1. Scrape 100 leads using <SCRAPER_OR_ACTOR> with filters you think match the ICP above.
2. Randomly sample 20 of the 100.
3. Judge each of the 20 against my ICP one by one. For each, say pass or fail and give one line of reasoning.
4. If fewer than 15 of the 20 pass, adjust the filters based on what specifically failed, and scrape again from step 1.
5. Repeat until a sample scores 15 or better out of 20. Hard cap at <MAX_ROUNDS> rounds, then stop and tell me what is blocking you.

When the sample passes:

- Report the score from every round so I can see how the filters moved.
- Trim the output to these columns only: first name, last name, email, company name, company size. Drop everything else.
- Save it to <OUTPUT_PATH>.

Do not ask me to approve each round. Run the loop yourself.

03 What it is actually doing

Every experienced operator does the same thing after a scrape. They open the file, eyeball twenty rows, and decide whether the list is any good. If it is junk they change the filters and run it again.

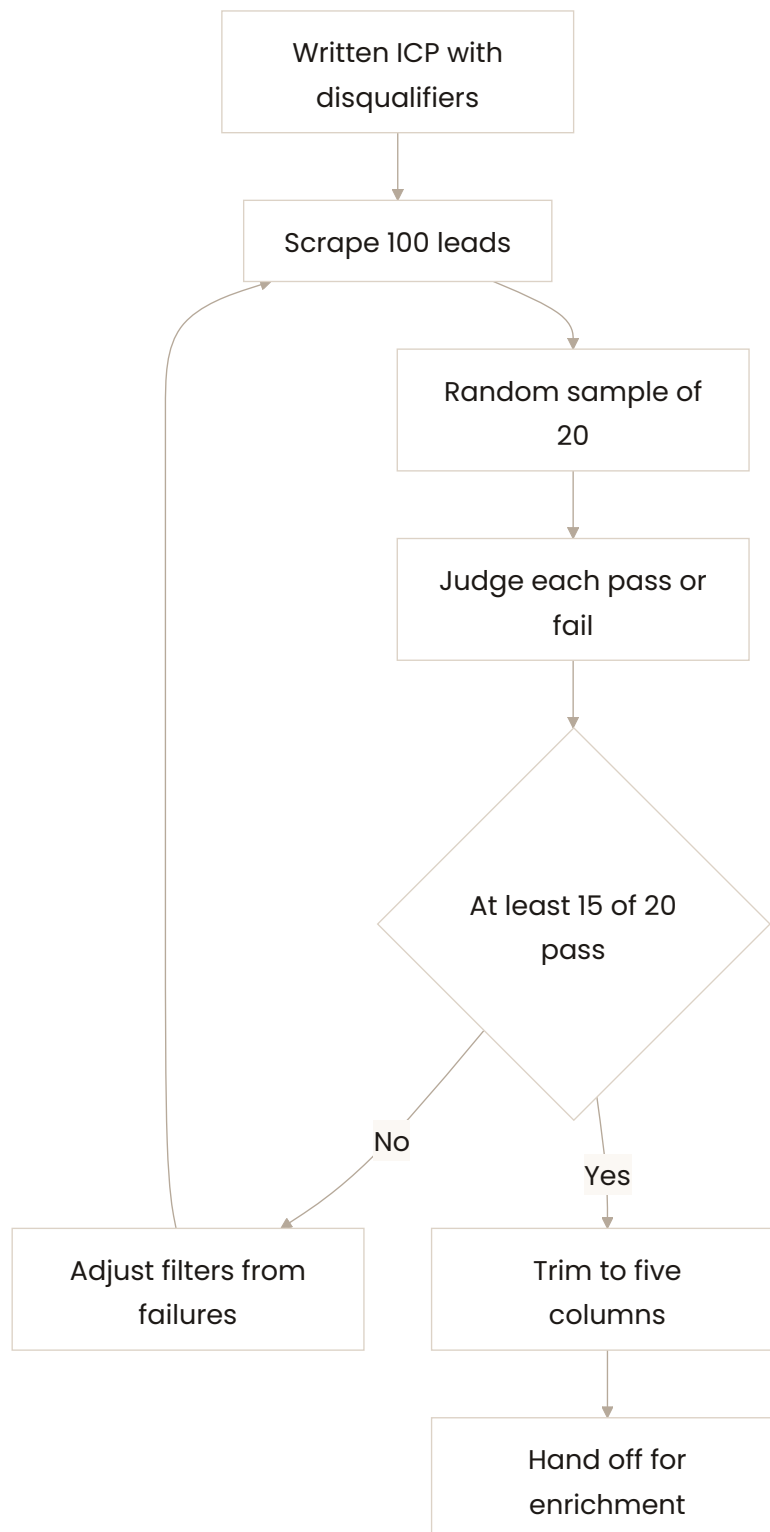
That habit is a loop. It has a check and a threshold and a retry. Which means you can write it down and hand the whole thing to the agent, including the judging.

That is what this prompt does. It tells the agent how many to scrape, how many to sample, what counts as a pass, and what number of passes is good enough. Then it tells the agent to fix the filters itself and go again. You invoke it once and walk away.

Nick's live run went 9 out of 20, then 9 out of 20 again, then 15, then 19. Four autonomous rounds from one invocation, ending in a list that actually matched the ICP. He never touched it between rounds.

The general lesson is bigger than scraping. Your existing standard operating procedure almost always contains a checkable threshold somewhere. Find the number, write the number into the skill, and let the agent iterate against it. A goal like until it looks good is not checkable. A goal like 15 of 20 is.

04 How it flows



05 Step by step

01 Write your ICP down properly

Roles, industry, size, location, hard filters, and disqualifiers. Nick targeted founder, CEO, co-founder, owner, co-owner, and partner. Vague ICP means the judge step has nothing to judge against.

02 Pick the sample size and threshold

Scrape 100, sample 20, require 15 to pass. Those numbers are arbitrary but they are specific, and specific is what makes the loop terminate.

03 Make it show its reasoning per lead

Ask for pass or fail plus one line of why on each sampled lead. That reasoning is what it uses to adjust the filters on the next round.

04 Set a hard round cap

Always cap iterations. Without a cap, a bad ICP or a thin data source turns into an expensive loop that never converges.

05 Wire the scraper through a connector

Nick used an Apify actor steered through MCP. Each actor exposes an MCP configurator URL you paste in as a custom connector, so Claude drives the scraper with no API plumbing.

06 Trim the columns in the same run

A raw scrape carries 30 or more fields. Cut to what the send needs. If you skip this you feel the friction on every single run.

07 Read the round-by-round scores

The score history tells you whether your ICP was the problem or the filters were. A list that never gets past 10 of 20 is an ICP problem.

06 Words explained

ICP

Ideal Customer Profile. A written description of exactly who you want to sell to, specific enough that someone else could sort a list with it.

Self-validating loop

A process where the agent checks its own output against a stated standard and redoes the work until it passes, with no human in between.

Actor

A prebuilt scraper on a platform like Apify. You give it filters and it returns rows.

Custom connector

A link you paste into Claude that lets it drive an outside tool directly. Apify gives each actor its own configurator URL for this.

Sample

A small random slice of the full list, used to judge the whole thing without reading every row.

Done-criteria

The checkable condition that ends the loop. It has to be a number you can measure, not a feeling like good enough.

07 Watch out for

- A loop with no round cap can keep scraping against an impossible ICP. Always set a maximum.
 - If the ICP is vague, the judging step is theater and the score means nothing.
-

- Raw scrapes carry 30 or more columns. Skip the trim instruction and you fight the file on every run.
- Scraping is not free. Nick's reference is about 1.50 to 1.80 US dollars per 1000 leads, and one live run of 100 leads with emails cost 18 cents.
- A score that never improves across rounds means your source or your ICP is wrong, not your filters.

08 How it connects

Cold Email Personalization

This produces the validated list that the personalization prompt enriches. The two run back to back as one pipeline.

Self-Healing and Error Logging Clauses

Both notes hand the agent a rule for what to do when its own work fails, instead of leaving it to stall or hand you garbage.

Cold Outreach Enrichment Pipeline

The framework note recording the four-round live result and the column-trim step.

Automation Ladder – Prompt Skill Loop Routine

A self-validating scrape is what makes the skill safe to run unattended on the loop rung, because it grades itself.

Credit. Framework and approach by Nick Saraev, from the self-validating scrape segment of the cold outreach build in his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.