

Hunting Season for the Rest of Us

Varun Chوريا · 2026-07-24 · <https://www.varunchoria.com/hunting-season-for-the-rest-of-us/>

About a week into prepping for one of these companies, I could drop the category's favourite piece of jargon into a sentence without flinching. Felt good about it too.

Then I tried explaining to myself why anyone would buy it, and I got about 11 seconds in before I ran out of road.

That's the thing nobody warns you about with interview prep. You can accumulate a fairly convincing vocabulary in a weekend, and vocabulary is worse than knowing nothing, because knowing nothing makes you ask questions and vocabulary makes you confident. Confidence invites the follow-up. The follow-up is where you die.

I'm between roles right now, interviewing for GTM and product marketing seats, so I have been finding this out repeatedly and in real time. The market is not helping. Every posting gets a thousand applicants, half of them AI-blasted. AI now does the junior version of most marketing jobs, so "I can execute" stopped being a pitch somewhere around last year. Meanwhile companies expect one senior person plus a few agents to cover what a pod of 3 covered in 2022. Floor went up, ceiling went up, and most of us are standing in the gap between them wondering which way to jump.

I [spent a decade deliberately engineering luck](#) and even I'll admit this season is harsh. You can't do anything about the market. You can do quite a lot about how you show up to the 4 or 5 shots you get.

So after that little incident I stopped prepping and spent 3 days building a thing instead. This is what it is, how it's put together, and what it cost, which was nothing.

What I actually built

It's a static site. 40 HTML pages sitting in a folder on my desktop, no hosting, no build step, no backend. I open `index.html` and the whole war room is there.

Screenshots below are anonymized prototypes. I'm mid-process with these companies and publicly dissecting your potential employer is an excellent way to stop being mid-process. The blue callouts are notes on why each piece exists.

Interview War Room

Three live processes, eight rival dossiers, one learning system. All claims hyperlink to sources.

Companies

Company A · Sr PMM

no round scheduled

profitable

Security scaleup. Kill-shot POV: several products, one story.

Company B · Director Mktg

R1 done, log outcome

analyst Leader 2026

Enterprise software giant. Primary mandate is agentic marketing. The live agent demo is the closer.

Company C · Group Mgr PMM

no round scheduled

IT services titan. BU-altitude discipline: the unit grows fast, the parent barely moves.

WHY THIS MATTERS

One glance answers: where does every hunt stand, and what do I build next

WHY THIS MATTERS

The weekend list is proof-of-work artifacts, each built to answer an objection before it's raised

This weekend

1. **AI-visibility audit + 3 readouts** (~4h)
2. **Open-source the research skill + launch post** (~4h)
3. **Live account-research agent demo** (~6-8h)
4. **Two battle cards** (~7h)
5. **Seniority-consistency audit** (~1h)

Here's the folder, roughly:

```
interview-dossiers/
  <company>/raw/<date>/      scraped pages, archived by fetch date
  research/                  sitemaps, rival homepages, brand data
  research/rivals/<rival>/    one folder per competitor, shared
  site/                      the 40 pages
    index.html               hub
    <company>/               10 pages, one per company
    rivals/                  8 competitor teardowns, shared
    kb/                      link out to the learning base
    assets/style.css         one stylesheet for everything
```

2 things in there took me 2 bad versions to arrive at, and they're the only structural ideas in the whole system I'd defend.

The rivals folder is shared. 3 of the companies I'm talking to compete with overlapping sets of vendors. In version 1 I researched each competitor separately inside each company's folder, which

meant writing the same teardown of the same vendor twice with slightly different adjectives. Now a rival gets researched once, lives in one place, and gets linked from every company that needs it. This is the single reason a new company now takes a few hours instead of a day.

Learning is separate from research. There's a second folder, gtm-kb, that has nothing to do with any specific company. 4 disciplines in it: ABM, content marketing, SEO plus AI search, and interview craft. Each one is an issue tree, then branch docs, then a self-test. The last question on every self-test is an applied artifact that has to land in an actual company dossier, so studying and prepping feed each other instead of competing for the same evening.

That split came from noticing that company research goes stale in weeks and industry understanding doesn't. Mixing them meant throwing away good learning every time a process died.

The version history, since the failures are more useful than the final state:

v1 was markdown files. Which is what the research skill still outputs, and it's correct for the machine. It is unreadable for a human at volume. I had maybe 60 files and no way to answer "what do I know about this company" without opening 9 of them.

v2 was one giant HTML page per company. Everything on one scroll. Solved the file-hunting problem and created a worse one, which is that a 4,000-word page is a wall and you don't read walls the night before a round. You skim them and feel prepared, which is the exact failure mode I was trying to fix.

v3 asked a different question. Not "how do I organize this information" but "what job does this page do for me, and when." That produced 10 pages per company, each named for its job: overview, 101, 201, strategy and people, market, buyers, competition, fit and objections, drills, sources. I open drills the morning of. I open fit when I'm nervous. I never open all 10 at once, which is the point.

Same lesson as [designing without AI slop](#). The model will generate all of this happily. Deciding what belongs on which page is still your job and it is most of the work.

Company A · Senior PMM

THE ONE INSIGHT TO WALK IN WITH

The company is really **three disconnected value props**: test before ship, monitor after ship, fix inside the AI dev loop. A buyer doesn't buy three things. **The PMM's first job is the connective narrative.**

WHY THIS MATTERS

The TL;DR is my opening point of view, the one insight I'd bring on day zero. Not a summary

Five numbers

4.6/5

peer-review platform, a few hundred ratings

~40k

units processed last year (their figure)

108%

net revenue retention

1.6x

more defects in AI-written work

WHY THIS MATTERS

Numbers I can defend under cross-examination. Sourced, dated, hyperlinked

WHY THIS MATTERS

The landmine detector: facts that changed mid-process, verified from primary sources

DO NOT WALK IN WITHOUT KNOWING

A co-founder stepped out of the CEO seat a few months ago (their own post). The about page still shows stale titles. Never assert them.

3 things on the overview page do the real work, and you can build all 3 in a Google Doc in an hour. The rest of the site is convenience.

The one insight. Not a summary of the company. The single thing you'd fix or build in your first 90 days, stated like you actually believe it. Mine for one company: they sell several products as separate things and the buyer doesn't want several things, so the first job is the connective narrative. That's an opinion, and opinions are what these conversations are secretly testing, because [judgment is what we're all paid for now](#).

5 numbers you can defend. Sourced, dated, hyperlinked. When you drop a number and someone leans in, "that's from your own exec's interview last December" ends it.

The landmine box. Facts that changed recently and could detonate mid-conversation. In one process I found the CEO had quietly changed a few months earlier. The company's own about page still had the old titles on it. Picture confidently referencing the wrong CEO in round 1. Now picture being the only candidate in the pile who asks a good question about the transition.

The primers, which are the highest-leverage part

2 per industry, and this is the direct fix for the 11-seconds problem up top.

The **101** is the industry from zero, written as a story. Why does this thing exist, who pays for it, what forces them to pay, how does the money actually move. Every term shows up when the story needs it and arrives attached to a reason, which is the only way I've found to make terms stick.

The **201** is what makes you sound like you've been in the room before: the debates practitioners genuinely argue about, the unit economics, 20 years of history compressed into 4 acts, and a day in the life of the person who signs the cheque.

War Room / Company B / Learn: 101

Industry 101 • the category, from zero

A story about how big companies actually run this function, why it is a mess, and why the mess is worth billions to fix. ~15 minutes.

Start here: the second-biggest number in the company

Take a 50,000-person company. Its biggest cost is people. Its *second* biggest cost is **everything it buys**: raw materials, laptops, cloud contracts, consultants, office coffee.

If the company runs a 10% margin, a dollar of new profit through *sales* means selling ten dollars of product. A dollar saved on *the cost side* is a dollar of profit, directly. **A dollar saved here is worth ten dollars of revenue.**

IF YOU REMEMBER ONE THING

This is the profit lever hiding in plain sight, and it is under-automated. That is the entire investment thesis of the category.

WHY THIS MATTERS

The 101 and 201 primers teach the what AND the why, so I never say a word I can't defend one follow-up deep

WHY THIS MATTERS

Every section ends with one memory hook. Recall beats vocabulary

Here's the difference in practice. Vocabulary-level prep gets you: "they've shipped an autonomous agent for the part of the workflow humans always did by hand." The 201 version gets you: that was never a capability problem, it was a unit-economics problem. The buyer always had the upper hand in those transactions. Actually using it cost more in analyst hours than it saved, so for about 30 years nobody bothered. Agents collapse the cost of paying attention, which is what finally makes the leverage worth exercising.

I can hold a 10 minute conversation on the second one. The first one is a sentence I could have gotten from the homepage.

To build them: ask whichever model you use to teach you the industry as a narrative, from first principles, with analogies, and a "if you remember one thing" line per section. Then a second pass for the debates, the economics, the history, the buyer's day. Then read both out loud and mark every sentence you couldn't defend if someone pushed. Those marks are your study list. Mine was embarrassingly long the first time.

The tools

Claude Code does the research, through a skill I wrote called [interview-recon](#). It chains into [Master Shifu](#) when something case-shaped comes up and needs a framework. Firecrawl handles bulk scraping on the free tier. Context.dev covers what Firecrawl can't reach, which is LinkedIn public pages, bot-protected review sites, sitemaps, brand data. That one has its own story and it's at the bottom.

Plain HTML for output because it opens instantly, prints as a leave-behind, and will still work in 10 years when whatever note-taking app I'd otherwise have used has been acquired and shut down.

One rule about models: use the strongest reasoning model you have for the synthesis step and make it cite a URL for every claim. Not out of distrust exactly. It's that you'll be repeating these facts out loud in a room where being wrong once costs you the room.

The skill is open source, same as Master Shifu, because job hunting is hard enough without paywalled prep.

 github.com/vcxcvii/interview-recon

```
git clone https://github.com/vcxcvii/interview-recon.git
~/.claude/skills/interview-recon
```

Then say “prep me for my interview at [company]”. 3 design decisions in it worth stealing even if you never install it. It announces its backend and its credit budget up front and asks before blowing past it, because most research skills prescribe paid tools and then quietly degrade when you don't have them. It shows you a research plan before it fetches anything, so you stay the editor. And it appends rather than overwrites, so round 2 loads what round 1 learned and refreshes only what went stale.

The 4 moves that did the damage

Everything in the war room carries a hyperlink and a date. 4 research moves produced most of what's actually in there.

Read the earnings calls of the public companies in the space. Private companies don't publish numbers. Their public neighbours do, and those neighbours set the narrative your interviewer is swimming in whether they know it or not. Transcripts are free on Motley Fool and on company IR pages. One hour with one transcript taught me that the whole industry's AI revenue reporting is a definitional arms race, everyone counting slightly different things and daring each other to object. That came up in a real conversation within the week.

Audit their content like you already own it. Every site has a public sitemap at `sitename.com/sitemap.xml`. Pull it, classify the posts by buying intent, chart the result. One company I looked at publishes several times more content than its loudest rival and roughly 70% of it is top-of-funnel fluff. That became a chart and a first-90-days recommendation. Nobody expects a candidate to walk in with the content audit already done, which is exactly why it works.

Content audit · classified from sitemaps

Company B · ~1,400 content URLs



WHY THIS MATTERS

I scraped every player's public sitemap and classified the content by buying intent. This chart is the artifact I hand over in the interview

Insurgent rival · ~450 content URLs



THE CHART'S STORY

"They out-publish the insurgent 3:1 in volume, but roughly 70% is top-of-funnel. My first move: a comparison-tier sprint while category attention is hot."

WHY THIS MATTERS

Walk in with a first-90-days move backed by their own numbers

Mine both kinds of reviews. G2 and Gartner Peer Insights tell you what customers think versus what the homepage claims. Employee review sites tell you what you're walking into. One company in my pipeline sits under 3 out of 5 across several hundred employee reviews, with the same complaint about pay after promotion showing up over and over. That didn't kill my interest in the role. It set my posture for the conversation about money, which is research doing a job that has nothing to do with answering their questions.

Read everything the interviewer has published. One hiring manager had posted their strategic thesis months before I ever spoke to them. My prep borrowed their language and pushed it one step further than they had taken it. That isn't flattery, it's arriving already aligned instead of spending the first 20 minutes establishing that you might be.

The part I'd keep if I could only keep one

Reading prep is not interview prep. Interviews are spoken, adversarial and on a clock, and none of those things are true of a document.

My written answers were sharp. My spoken answers were about 40% filler until the third rep. There is no way to discover that by reading.

So before every round I run live mocks with ChatGPT in voice mode. Copy this and adjust:

You are playing my interviewer for round 2 at [company].
Here is their public writing so you can capture how they think and what they care about: [paste posts/talks].

Rules:

- Ask one question at a time, like a real interview.
- Push back on weak answers. Don't let me ramble past 90 seconds.
- Interrupt me like a busy executive would.
- After 20 minutes, stop and grade me on: clarity of my opening point of view, structure under pushback, domain fluency, evidence discipline (did I use numbers), and the quality of my questions to you.
- Score each 1-10 with specific feedback, and compare against my last attempt: [paste last rubric].

2 honest caveats. This simulates the archetype, not the person. You're practising against what someone with that job and those published opinions tends to value and push on, and a revenue-minded CSO does push differently than a brand-minded CMO. And you have to do it out loud. In voice mode. The entire finding is the gap between how good your answer looks written down and how it sounds coming out of your face.

Voice mock · Round 2 simulation · Company B

Mock interview rubric · attempt 3

DIMENSION	SCORE	FEEDBACK
Opening POV clarity	8/10	Landed the thesis in 40 seconds. Cut the second example.
Structure under pushback	6/10	Paused well once, filled silence twice. Agree with what's true first.
Domain fluency	9/10	The unit-economics answer was senior-level.
Evidence discipline	7/10	Two claims lacked numbers. You have the numbers. Use them.
Reverse questions	9/10	The role-durability question is one nobody else will ask.

DELTA VS ATTEMPT 2

Structure +2, evidence +1, filler words down from 14 to 6. Next rep: the salary deflection, cold.

WHY THIS MATTERS

ChatGPT voice plays the interviewer's archetype from their public writing, then grades me on a rubric I can't argue with

WHY THIS MATTERS

Deltas across attempts turn repetition into training, not rehearsed mistakes

The rubric is what turns this from a novelty into training, because reps without feedback are just rehearsing your mistakes with extra steps. Across 3 attempts my filler words went from 14 to 6. More useful than that, the pushback score taught me something no [communication listicle](#) ever managed to make stick: agree with whatever is true in the objection before you start defending. The rubric caught me skipping that. Twice.

Every question I fumble goes into an objection bank with a written answer, so the same punch doesn't land twice.

Objection bank · scripts, not vibes

▼ "You have zero domain background."

SCRIPT

"Your buyer is an enterprise exec I've sold to for years. Buyer psychology is my home turf; vocabulary is a sprint, not a marathon. Here's my 30-day domain plan, and here's work I've already done." *Then put artifacts on the table.*

WHY THIS MATTERS

Every objection pre-scripted: what's really being assessed, the answer, and the proof artifact that backs it

▼ "Short stint on your resume. Flight risk?"

SCRIPT

Pull, not push: "I am optimizing for a long build and choosing carefully. I've researched you to the level of your org changes and analyst citations. That's not shopping-around behavior."

► "Everyone claims AI results. Prove yours."

EMPLOYEE REVIEWS: UNDER 3/5 ACROSS SEVERAL HUNDRED

Negotiate hard upfront. Raises reportedly do not come later. Probe role durability in the reverse interview.

WHY THIS MATTERS

Culture and comp research sets my negotiation posture, not just my answers

Build yours with one prompt: "here's my resume and the JD, list the 10 hardest objections an interviewer would have about me, what each one is really assessing, and draft my answer." Then rewrite every answer in your own words, because a script you didn't write is audible from space.

If you only have one evening, do these 3: one earnings transcript from the biggest public company in your target's space, one written point of view about the first thing you'd fix, and one voice mock with the rubric above. Let it humble you tonight so the interviewer doesn't get to on Thursday.

The bill for all of it, since I know what job-search finances feel like: Firecrawl free tier, about 500 credits. Context.dev, about 350 credits. Claude Code and ChatGPT were subscriptions I already had, and if you have neither, the free tier of any frontier model covers the primers, the objection bank and text-mode mocks. 3 focused days for the system and the first 3 companies. Incremental cash spent, zero.

The world is kind

Now the part I didn't plan for and can't take any credit for.

5 days ago Yahia Bakour, who founded Context.dev, sent me one of those routine founder check-in emails. Saw you signed up, how's the API treating you. I told him the truth, which was that I loved the product, I was building an interview research agent on top of it, and I couldn't afford a paid plan right now because I'm between roles.

He replied the next morning. He'd dropped 10,000 credits into my account, on the house, and bumped my rate limit. No strings, no call, no "let's find a way to work together." When I offered to do some free marketing work in return, his entire ask was that if I ever meet someone who needs the API, send them his way.

My whole [thesis about luck](#) is that it compounds on honest effort made visible, and I still believe that. But sometimes it's much simpler than the thesis. Sometimes a stranger is just kind, at the exact moment when kindness lands hardest. Half the research depth in this post runs on credits I was given by someone with no reason to give them.

So, holding up my end. If you're building anything that needs clean web data, scraping, extraction, brand intelligence, sitemaps, go and try [context.dev](#). It's accurate in the places where other tools fail silently, and it's built by someone who treats a broke job-seeker like a future customer rather than a freeloader. That's a founder worth betting on.

When this season ends I'll publish the recap: real screenshots, and an honest account of what actually worked once I was in the rooms. The rest of what I build in public lives on [side quests](#), and if you'd rather hire this thinking than copy it, [here's how I work](#).

The market is harsh and nobody is coming to make it less harsh. But the same tools that raised the bar will get you over it, mostly for free, if you prep like an operator instead of a candidate.

Happy hunting. If any of this helps you land something, tell me. That's the whole reason for writing it down.