



A C M O ' S F I E L D N O T E S

My First 30 Days *as an AI-Native CMO.*

What I actually did. Not a framework. Not a thought piece.

Context: CMO, \$50M ARR vertical fintech ISV

A P L A Y B O O K I N T H E O P E N / B U I L T D U R I N G , N O T A F T E R

CONTEXT

The 30/60/90 was built for a world that doesn't exist anymore.

THE OLD ONBOARDING

- Listening tour.
- Stakeholder interviews.
- Slide deck of observations.
- Strategy memo by day 60.
- First real output by day 90.

WHAT I'M DOING INSTEAD

- Wire into raw signal on day 1.
- Build systems while doing the work.
- Ship visible output in week 2.
- Every decision compounds context.
- The team sees it working before hearing about it.

THE SITUATION

Where I landed.

\$50M

ARR

Growing ~35% YoY. Past founder-led GTM, pre-scale.

ISV

FINTECH VERTICAL

Embedded payments + compliance tooling for regional banks.

13

PEOPLE IN GTM

Marketing, sales, CS. Everyone wears two hats.

0

AI STANDARDS

Individuals experimenting. No shared context, no standards.

The market moves. The team is small. Founder-led motion has topped out. I have maybe 90 days of patience before I'm expected to ship an outcome, not a plan.

Go all the way. *No half measures.*

01

Signal before strategy.

I don't need alignment on day 3. I need raw customer truth. Transcripts, pipeline, win/loss. The strategy emerges from the signal — not the other way around.

02

Build while doing.

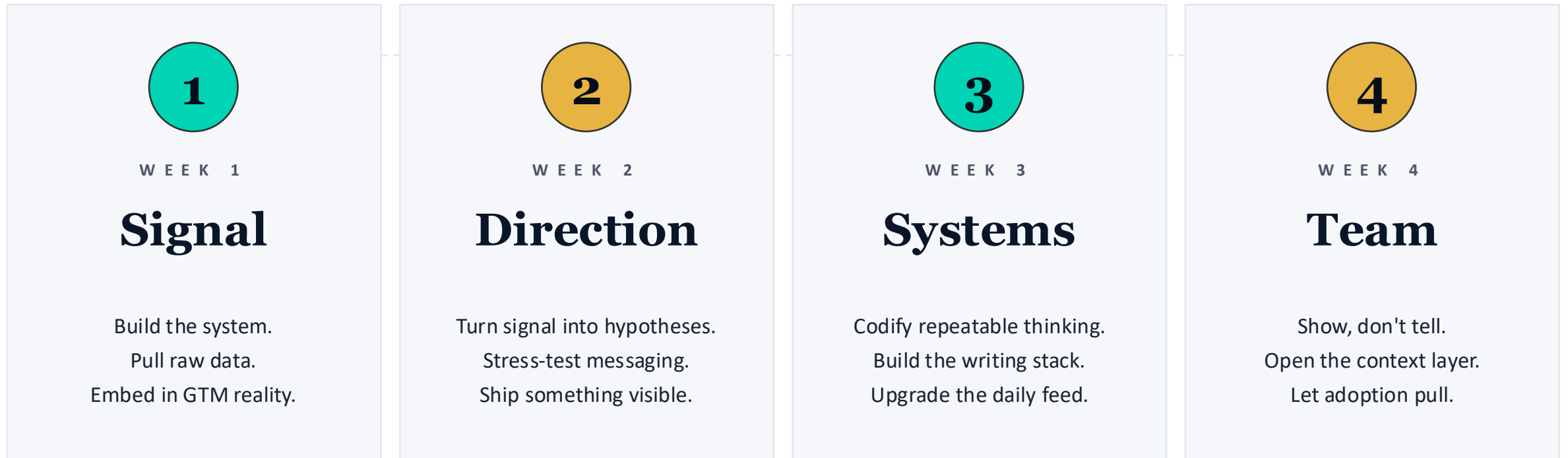
I won't sit down and design a perfect AI system. I'll do real work, notice what repeats, then systemize. If it works twice, it becomes a skill.

03

Compound context.

Every call, doc, decision, and insight goes back into a structured context base. My system gets smarter every day. This is the actual moat.

Four weeks. Four moves.



Each week has an output. If I miss the output, something is wrong with the system — not the week.

01

WEEK 1

Signal.

Wire into reality before you try to change it.

Goal: raw truth, not opinions. Ugly-but-reusable workflows. A context base that starts compounding.

Set up the Personal AI OS — properly.

Don't just create folders — define behavior. The folder structure is the least interesting part. The operating rule is everything.

`~/cmo-os/`

`/context/` *calls, pipeline, market, internal*

`/calls/` *transcripts + notes*

`/pipeline/` *CRM exports, deal notes*

`/market/` *competitors, category*

`/internal/` *meetings, team insights*

`/work/` *active projects, drafts, analyses*

`/system/` *skills, prompts, voice files*

THE OPERATING RULE

After every meeting, Slack thread, or customer call — log three things:

- 1 *What did I learn?*
- 2 *What surprised me?*
- 3 *What repeated?*

Skip this step and the system stops compounding. The logging IS the leverage.

Pull raw data. Don't wait for the dashboard.

On day 2 I asked our RevOps lead for exports — not reports. Raw CSVs beat polished dashboards because I can run my own pattern analysis against them.



What I asked for

- Last 20–30 closed-won deals
- Last 20–30 closed-lost deals
- Current pipeline snapshot
- Last 10–15 call transcripts



What I ran on it

- Win/loss pattern analysis
- Messaging extraction from calls
- ICP reality check vs. marketing assumptions
- Deal velocity by segment



What I wrote down

- "We win when..."
- "We lose when..."
- "Customers consistently say..."
- "Our marketing site says X — customers say Y."

First layer of truth. By Friday I had a one-pager of patterns from the data. Zero stakeholder interviews required.

Insert yourself into GTM reality.

I shadowed 9 sales calls in 3 days — discovery, demos, pricing conversations. Not to learn "the business." To build a pattern library.

WHAT I TRACKED LIVE



Where deals slow down

Compliance review? Pricing? Procurement?



Where reps struggle

Which objections don't have a clean answer?



What gets repeated

Same question 3+ calls = a content gap.



Exact customer phrasing

Words we're not using that they are.

WHAT WENT INTO /context/calls/

- Objection patterns → tagged & counted
- Verbatim phrasing → pulled for messaging
- Moments of confusion → flagged as friction
- Competitor mentions → tracked by stage

"Not learning the business. Building a pattern library."

Build the first workflows. Keep them ugly.

By day 5, friction shows up. The same analysis. The same search. The same synthesis. That's the signal to codify.

WORKFLOW 01

/analyze-pipeline

Instead of manually reviewing deals, a repeatable analysis surfaces:

- Stuck deals (no activity >14 days)
- Risk signals (sentiment drop, stakeholder silence)
- Common traits of deals sliding in-quarter
- Where marketing-sourced vs. sales-sourced diverges

WORKFLOW 02

/calls-to-messaging

Input: call transcripts. Output:

- Top pains (ranked by frequency + emotional weight)
- Real phrases — verbatim customer language
- Messaging angles scored against current homepage
- Objections categorized & ready for rep enablement

The rule: if it works twice, it's a system. Don't wait for polish.

End of Week 1: What I actually have.

Not clean. Not comprehensive. But real — and mine.



Raw customer truth

Win/loss patterns from 40+ deals. Verbatim language from 10+ calls. No opinions attached.



Early pipeline patterns

Where deals actually stall. Which segments close. Which marketing claims do vs. don't hold up.



Two usable workflows

/analyze-pipeline and /calls-to-messaging. Ugly. Re-runnable. Already saving hours.



A growing context base

Every call, meeting, and surprise is logged. The system is already smarter than it was on Monday.

What I don't have yet: a strategy, a team plan, or a polished deck. Those come next — grounded in what's above, not in my opinions.

02

WEEK 2

Direction.

Turn signal into a working thesis. Then test it.

Goal: rough hypotheses, messaging stress-test, a shared GTM context layer, one visible ship.

Write the first working thesis.

Not a strategy. Not a deck. A set of sentences rough enough to be wrong — and specific enough to test.

ICP

Our real ICP is probably regional banks with \$2B–\$20B in assets running legacy core systems — not the mid-market fintechs our marketing assumes.

USE CASE

Our strongest use case is compliance automation for ACH + wire, not the broader "embedded payments" story we've been telling.

MESSAGING GAP

Our messaging is weakest at the procurement stage — we lose deals to "too risky, unfamiliar" far more than to competitor feature wins.

PIPELINE

Pipeline is breaking between discovery and technical eval. Reps aren't equipped to run the security conversation without a solutions engineer.

These four sentences drive everything for weeks 2–4. Each one becomes a test.

Stress-test messaging against real language.

Our messaging said one thing. Customer calls said another. I put them side-by-side and the rewrites wrote themselves.

WHAT WE SAY



"Next-gen embedded payments infrastructure."



"AI-powered compliance automation."



"Enterprise-grade scalability and resilience."

WHAT CUSTOMERS SAY



"We need something that won't get us written up in an audit."



"Every week we spend 6 hours on exception review. Can you cut that?"



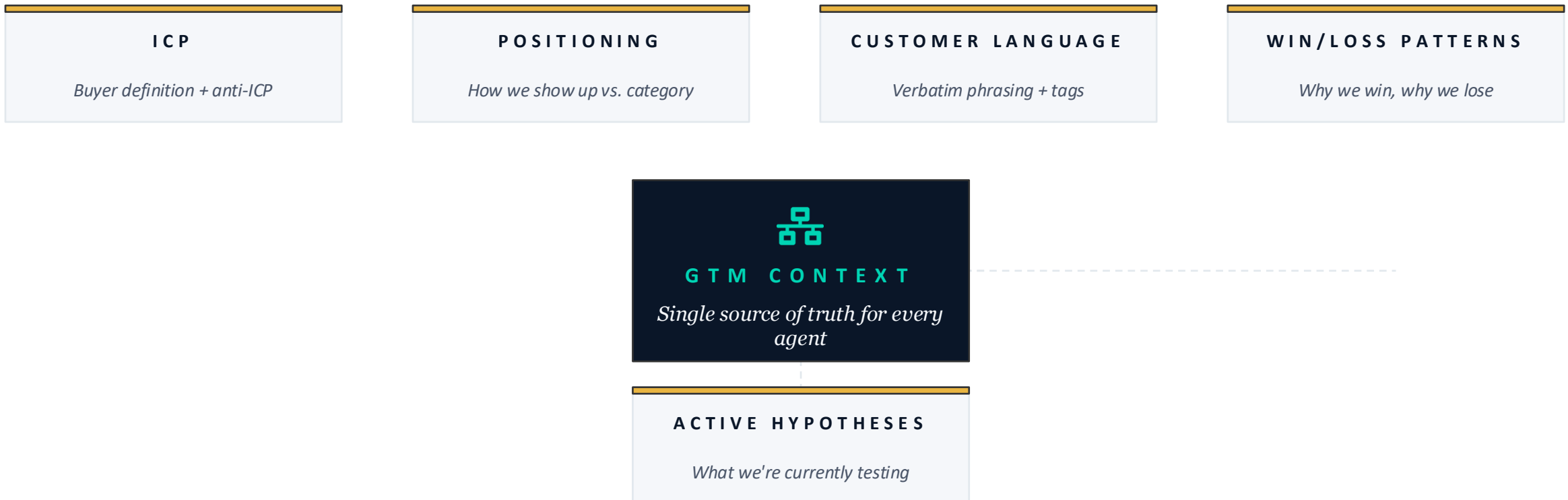
"Our last vendor went down during month-end. That can't happen again."

The rewrite rule: homepage headline, core value prop, 1-2 campaign angles — all based on real language, not brand opinion.

Build the GTM Context layer — v1.

This is the single highest-leverage thing I build in 30 days. Not a deck. A living system any agent on the team can draw from.

Not over-formatted. Populated with raw insights, quotes, and patterns — updated continuously.



Ship something externally visible.

By day 14 something real is live — visible to customers, prospects, the team. This creates credibility, feedback loops, and momentum. All three matter.

01

Homepage rewrite

New headline + hero subhead driven by verbatim customer language. Replaces category jargon.

02

Campaign relaunch

Kill the running campaign if the angle doesn't match real pains. Relaunch with one sharp message.

03

Messaging update

Sales-facing one-pager reflecting the new positioning. Rep enablement for the procurement objection.

Constraint: pick ONE. Must reflect new insights. Must ship fast. Perfection is the enemy of momentum.

End of Week 2: What shifted.

WHAT I NOW KNOW

The real ICP is 3x tighter than our marketing suggests. That changes everything downstream — ads, content, site, sales plays.

The procurement stage — not the demo — is where we lose. "Risk" is the blocker, not "features."

Our category language repels the buyer. They use the words "audit," "exception," "downtime." We use "embedded."

WHAT'S LIVE



New homepage hero

Headline reflects real buyer language. Measured against baseline CTR.



Sales one-pager v1

Procurement-objection kit. Already used on two calls this week.



GTM Context v1

Five sections populated. CSM and AE have started contributing.



Four hypotheses

Written, dated, testable. Everything downstream tests against them.

03

WEEK 3

Systems.

Stop being the bottleneck.

Goal: codify repeatable thinking, build the writing stack, upgrade the daily control panel.

Find the patterns. Turn them into skills.

By day 15 I look back at two weeks of work and ask three questions. The answers become my first real skill library.

THREE QUESTIONS

- 01** *What decisions did I make repeatedly?*
- 02** *What analysis did I re-do from scratch?*
- 03** *What thinking pattern keeps showing up?*

SKILLS I CODIFIED THIS WEEK

/icp-check

Scores a deal, prospect, or account against our ICP criteria.

/objection-lookup

Given an objection, returns the best in-context rebuttal + supporting proof.

/win-loss

Synthesizes last N closed deals into patterns + messaging implications.

/morning

Personalized briefing: pipeline shifts, call insights, Slack signals, priorities.

Build the writing system — properly.

AI doesn't write for me. But with a good voice file and two focused skills, it's the sharpest editor I've ever worked with.

STEP 1

Voice file

```
/system/voice.md
```

How I write: cadence, vocabulary, tone, what I never say. Plus 5–10 examples of my best writing. Always pulled into context.

STEP 2

Two skills

```
/draft /iterate
```

/draft takes messy input → structured output.
/iterate sharpens clarity and flow — line-edit mode, not rewrite mode.

STEP 3

Real work

```
ship it
```

Test the stack on internal docs, messaging frameworks, and this week's content. If it holds up on real work, it's valid.

Upgrade /morning from update-feed to control panel.

Every CMO gets drowned in updates. My /morning skill gives me decisions, priorities, and actions — pulled from my actual context, not a generic template.

✘ BEFORE — update feed

- 12 new Slack threads
- 4 calendar invites
- 8 pipeline updates
- 3 LinkedIn posts from competitors
- Email digest: 47 items

"I read it all and then forget it all."

✔ AFTER — control panel

3 DECISIONS NEEDED FROM YOU TODAY

Approve campaign budget / sign off hiring plan / green-light rewrite

2 SIGNALS FROM PIPELINE

Two \$100K+ deals stalled — both on procurement risk

1 THEMATIC PATTERN

"SOC 2 timeline" came up in 4 of last 7 calls — content gap

End of Week 3: Leverage starts to compound.

For the first time, my morning takes 15 minutes instead of 90. That's not convenience — that's time I now redirect into real thinking.

7

NAMED SKILLS

Each one earned — built from repeated real work.

~5h

SAVED / DAY

Pipeline analysis, briefing, messaging, editing.

1

VOICE FILE

Pulled into every writing task automatically.

∞

COMPOUNDING

Every new skill builds on the last. The flywheel is real.



The quiet risk

Everything above only works for me. At the end of Week 3 I'm still a single-player system. Week 4 is where this gets hard.

04

WEEK 4

Team.

Single-player → multi-player. This is where things break.

Goal: show the team what works, open the context layer, let adoption pull — don't push.

Show. Don't tell.

I blocked 60 minutes with the full GTM team (13 people). No slides. No "AI strategy." Just: this is how I'm working. Screen share. Trade notes.

DO

- **Screen share the actual system**
Folders, files, skills — the real thing.
- **Walk through a real workflow end-to-end**
Pick one deal or one call. Show the output.
- **Invite questions; take notes live**
Their objections are your roadmap.

DON'T

- **Present a polished "AI strategy"**
People tune out. They've heard it.
- **Mandate adoption timelines**
Pressure kills curiosity. Curiosity drives adoption.
- **Pretend it's all figured out**
Honesty about the mess is your credibility.

Open the GTM Context. Let the team in.

The context layer I built in Week 2 was for me. Now I turn it into a shared brain — access for all, contributions encouraged, no gatekeeping.

FROM

Single-player context

- ✗ My local folders, my eyes only
- ✗ One person logging insights
- ✗ Knowledge lives in my head
- ✗ Team reinvents the wheel constantly



TO

Shared GTM brain

- ✓ Read + write access for all of GTM
- ✓ CSMs contribute customer quotes
- ✓ AEs log objection patterns in-flow
- ✓ Any agent on the team pulls the same truth

Scale two workflows. Not twelve.

I could hand the team a dozen skills. I won't. Two workflows — the ones with the clearest ROI — are the wedge. Everything else is noise until these land.

/analyze-pipeline

FOR: AEs + sales leadership

Gives reps a 5-minute version of their own QBR prep. Surfaces stalled deals, flags risk, ranks follow-ups.

→ *Weekly pipeline review time: 4h → 45 min*

/calls-to-messaging

FOR: Product marketing + content

Turns every 10 calls into real customer language, ranked pains, and tested messaging angles — without manual listening.

→ *Campaign concept time: 2 weeks → 3 days*

The rule: don't force adoption. Let the people who see it working pull it in. Pressure kills curiosity.

Observe where it breaks.

Final stretch: I watch. I don't intervene unless asked. Breakage is the point — it tells me what the real system needs to be.

Inconsistent usage

Two AEs are running /analyze-pipeline daily. Three haven't touched it.

WHAT I DO

Pair the power users with the non-users. Not in training. In real deal reviews.

Confusion at the edges

CSMs try to use /calls-to-messaging but their call types are different.

WHAT I DO

Their workflow branches from ours. Build the CSM version — don't force-fit.

Duplicated effort

Content marketer is running the same analysis I'm running — separately.

WHAT I DO

That's a shared context failure. Their work belongs in GTM Context, visible to me.

Reframe: this isn't failure. This is the real system emerging.

What I actually built in 30 days.

Not a plan. Not a deck. Five systems that compound from here.

01	A context engine	Every call, meeting, deal, and surprise logged into structured buckets. Compounding knowledge.
02	A decision system	/morning skill: decisions, signals, patterns — not updates. 15-minute daily control panel.
03	Early GTM intelligence	Pipeline and call analysis that runs repeatably. Patterns surface without manual effort.
04	A shared layer of truth	GTM Context open to the team. Five sections, growing weekly. Every agent pulls from the same source.
05	The beginnings of team leverage	Two workflows in production with the team. Breakage patterns identified. Real system starting to emerge.

Single-player is easy. Multi-player is where it breaks.

WHAT GOT EASY FAST

- **Personal workflows**
Weeks, not months, to full flow.
- **Context compounding for me**
Every week I'm sharper than the last.
- **Speed of iteration**
Idea → draft → ship in hours, not days.

WHAT STAYED HARD

- **Team deployment**
Everyone works differently. No standard patterns yet.
- **Security + data governance**
Customer data in prompts needs real rules.
- **Shared context discipline**
Getting 13 people to log consistently is its own problem.

The system emerges from the work. Not the other way around.



My first instinct was to design a clean hierarchy, a master prompt library, a perfect framework. It was wrong. Every time. The best systems came from doing real work, noticing what repeated, and wrapping structure around the pattern after the fact.



Over-architecting

Designing the perfect folder structure before doing work.



Skill proliferation

Building 20 skills when 3 would have been used.



Premature team rollout

Pushing to the team before it's real for you.

Context engineering is the most undervalued work in GTM today.

01

Agents are only as good as their context

The same model with different context produces very different outputs. The work isn't in the prompt — it's in the structured knowledge the prompt can draw from.

02

The boring work is the leverage

Logging a call. Tagging an objection. Copy-pasting a customer quote into the right folder. Unsexy. But this is what makes the next quarter's outputs 10x better.

03

It compounds in ways that surprise you

Once our GTM Context had three weeks of data, CSMs started using it to spot expansion signals. That wasn't the plan. Good context creates its own uses.

If I had to do it again.

A few things I'd change on day one — knowing what I know at day 30.



Pull the CS call library sooner

I waited 10 days to get CS call transcripts. They had the best expansion signal in the whole company. Should have been day 2.



Set data-handling rules earlier

I started pasting customer data into prompts without documented rules. Retrofitting governance is harder than setting it up upfront.



Resist the homepage redesign urge

Shipped the hero rewrite in week 2. Good. But I also burned energy on design polish too early. Message > design for the first 30 days.



Involve one trusted GTM partner in Week 1

I stayed solo for too long. A single AE or CSM with skin in the game would have surfaced blind spots two weeks earlier.

What's next: Day 31–90.

Three moves to turn the Week 4 foundation into real GTM leverage.

DAY 31–50

Full-team multiplayer

Turn the two workflows into five. Formalize the context-logging rituals. Hire or redeploy someone as the first "context engineer" role in GTM — curation is real work.

DAY 51–75

Re-architect one motion

Pick the single GTM motion where AI leverage is highest — for us, it's the procurement-stage security conversation. Redesign it end-to-end with agents in the loop.

DAY 76–90

Commit to the thesis publicly

By day 90, positioning, pricing, and ICP are no longer hypotheses — they're commitments. Board update, team offsite, customer messaging: one coherent story.

The AI-native CMO doesn't have a smarter *strategy*.

They have a smarter *system* for getting to strategy.

The work is the same. The loop is tighter. The context compounds. The team sees it working before they hear it explained. And by day 30, what looks like a "30/60/90" is something stranger and better — a living operating system for the go-to-market function.



TRADE NOTES

If you're a GTM leader leaning on AI —
what's pulling you in, and what's holding you back?



This playbook is a work-in-progress. I'll share more as I go — what works, what breaks, what surprises me. I'm not offering a copy-paste template. I'm offering the field notes.

I read every reply.