MARKETING YOUR STARTUP USING CLAUDE

A Founder's Field Guide to Running the Marketing Skills Framework

An Independent Field Guide

Built on the open-source Marketing Skills framework by Corey Haines

*“You don't need a marketing department.
You need a marketing brain your agent can borrow.”*

A NOTE ON THIS BOOK

Before You Start

This book is for the founder who is, at this exact moment, both the CEO and the entire marketing department. You have a product. You may even have a few customers who love it. What you don't have is a growth marketer, a copywriter, an SEO specialist, and a paid-media buyer sitting down the hall — and you don't have the budget to hire them yet. What you do have is Claude, and, as of this book, a library of forty-plus marketing skills built specifically to turn a general-purpose AI agent into something closer to a fractional CMO: a collaborator who knows conversion optimization, copywriting, SEO, paid media, lifecycle email, and growth engineering, and who applies the right playbook the moment you describe your problem.

The skills this book teaches come from an open-source project called Marketing Skills for AI Agents, created by Corey Haines and released under the MIT license. It is, at the time of writing, one of the most-used marketing skill collections built for coding agents like Claude Code — organized into seven categories that cover almost everything a startup marketing function actually does: SEO and content, conversion optimization, content and copy, paid and measurement, growth and retention, sales and go-to-market, and strategy. This book does not reproduce that project's files. Instead, it teaches you how the framework thinks, chapter by chapter, category by category, so that when you sit down with Claude and your own product, you already know which skill to reach for, what questions it will ask you, and what a good session with it looks like.

You'll follow one recurring founder through the book: Dana Whitfield, who is building Northstar Ops, scheduling and dispatch software for small home-service businesses — plumbers, electricians, cleaners, the trades that run on trucks and phone calls. Dana is not a marketer. Dana is a former field-service dispatcher who got tired of watching good crews get run by bad software, and built something better. Every chapter follows Dana through one real marketing problem, working with Claude the way you will. In the later chapters, when the questions turn to scale — paid media budgets, brand campaigns, revenue operations — you'll also meet Marcus Odei, a Series A founder further down the same road, so you can see the framework flex from zero to a real marketing org.

Two things before you begin. First, this book assumes you can already talk to Claude — in Claude Code, the desktop app, or claude.ai — and that you're comfortable installing a skills folder into a project or plugin marketplace; Chapter 1 walks through exactly how. Second, this is a field guide, not a reference manual: read it start to finish once, doing the exercises as you go, and you'll come out the other side with a real, working product-marketing context file, a real 90-day plan, and real drafts of your homepage, your SEO foundation, your first ad campaigns, and your launch plan — built for your actual company, not a hypothetical one.

CONTENTS

Contents

PART I

Foundations

Set up the one file every other chapter depends on.

CHAPTER ONE

Why an Agent Needs a Marketing Brain

Dana had opened Claude a dozen times in the two weeks since Northstar Ops got its first paying customer, and a dozen times had typed some version of the same message: *“write me a homepage,”* or *“write me a cold email,”* or *“what should our pricing page say.”* **Every single time, the output was competent, generic, and a little bit wrong.**

It wasn't that Claude wrote badly. It was that Claude knew nothing about Northstar Ops beyond the two sentences Dana had typed that morning, and had no framework for what a good homepage for a scheduling tool aimed at electricians should even accomplish — what to lead with, what objection to kill first, what a home-service business owner actually cares about at ten at night when they're finally looking at software instead of running calls. The copy was fluent. It was also the kind of copy that could have belonged to any of four hundred other SaaS products. Dana rewrote most of it by hand and started to suspect that “just ask the AI” was a much smaller unlock than the tweets made it sound.

The gap Dana ran into is not a Claude problem. It is a context and expertise problem, and it is exactly the gap the marketing skills framework exists to close. A skill, in the sense this book uses the word, is a folder of instructions — a SKILL.md file, plus reference material — that tells an agent like Claude how a specific kind of expert actually works. Not “write good copy” in the abstract, but the specific questions a conversion copywriter asks before touching a headline, the specific structure a technical SEO audit follows, the specific way a fractional CMO sizes a marketing plan to a company's stage and budget. When you install a skill, you are not teaching Claude new facts about the world; you are handing it a professional's checklist and a professional's judgment about when to use which move.

One Framework, Seven Categories

The specific library this book teaches — Marketing Skills for AI Agents, built by Corey Haines and released as an open-source, MIT-licensed project — currently ships more than forty of these skill files, organized into seven categories that map cleanly onto how a real marketing function is organized: SEO and content, conversion rate optimization, content and copy, paid media and measurement, growth and retention, sales and go-to-market, and strategy and monetization. Ask Claude to “optimize this landing page for conversions” and, with the skill installed, it reaches for the cro skill — a conversion-rate-optimization framework that knows to check your value proposition placement, your form friction, and your social proof before it touches a single word of copy. Ask it to “set up a five-email welcome sequence” and it reaches for emails. Ask it to plan a whole quarter and it reaches for marketing-plan, which behaves less like a document generator and more like a fractional CMO running a structured engagement.

Figure 1. *The marketing skills map — one shared context file at the center, seven categories of specialized skills fanned around it.*

What makes this more than a very long prompt library is the thing at the center of that map: a single shared file, product-marketing.md, that every other skill checks before it does anything. You build it once, in Chapter 2, and from then on you never again explain what Northstar Ops does, who it's for, or how it talks. Every skill you invoke after that — whether it's writing ad copy eight chapters from now or drafting a churn-prevention flow twelve chapters from now — reads that file first and produces something that already sounds like your company, not like a stranger's best guess.

A skill isn't a better prompt. It's a professional's checklist Claude runs before it opens its mouth.

WATCH FOR

- Treating skills as templates. A skill doesn't hand you finished copy — it asks the questions a real specialist would ask first, then produces work grounded in your answers. Skipping the questions is how you get generic output again.

- Installing everything and using nothing. Forty-plus skills is a menu, not a checklist to complete. You'll use a handful heavily and a few others once a quarter. That's the intended shape.
- Forgetting the context file exists. The single most common way people get mediocre results from this framework is skipping Chapter 2 and jumping straight to a skill. Every skill is only as good as the context it's given.

KEY TAKEAWAYS

- A “skill” is a folder of instructions — a checklist, a set of questions, and a body of expertise — that tells Claude how a specific kind of marketing specialist actually works.
- The framework this book teaches organizes 40+ skills into seven categories: SEO & content, CRO, content & copy, paid & measurement, growth & retention, sales & GTM, and strategy.
- Every skill checks one shared file — product-marketing.md — before doing anything, which is what keeps a solo founder from re-explaining their company in every single conversation.
- You don't need to memorize the skill list. You need to recognize the situation (“this page isn't converting,” “I need a launch plan”) and trust that naming the problem in plain language will surface the right skill.
- Skills are open, editable markdown files — you can read exactly what expertise Claude is applying, and adjust it if something doesn't fit your business.

EXERCISE

Exercise

Self-Reflection

Open your last five conversations where you asked Claude for marketing help — a headline, an email, a page. For each one, write one sentence naming what context you wish Claude had already known before you asked.

Collaborate

Ask Claude directly: “If I gave you a written summary of my product, audience, and positioning, what would you have done differently on [pick one of the five]?” Notice how specific its answer gets once you name a real gap.

Reflection

– What's the one piece of context that, if Claude had it permanently, would improve almost everything you ask for?
– Where has “just ask the AI” quietly cost you time you didn't notice you were spending on rewrites?

REFLECTION

- *Where in your own workflow have you mistaken fluent output for correct output?*
- *What would change if every marketing request you made started from the same shared understanding of your company?*

WHAT'S NEXT

Chapter 2 builds that shared understanding: the product-marketing.md file every other chapter, and every other skill, will lean on.

CHAPTER TWO

Your Product Marketing Context

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

- ✓ Install the marketing skills framework into a project Claude can use, via Claude Code, the plugin marketplace, or a manual copy.
- ✓ Run the product-marketing skill to build your own product-marketing.md context file.
- ✓ Explain why customer language beats polished description, and capture your own in the file.
- ✓ Know when and how to update the file as your positioning changes.

Dana's second attempt started differently. Instead of asking Claude for a homepage, Dana asked it to help build something that had never occurred to her to write down: a single document describing, in plain language, what Northstar Ops actually was.

This is the bookend artifact this book is built around. You'll build a first version of it in this chapter, in about twenty minutes, and by the closing chapter you'll have expanded it into a living operating document that your whole future marketing hire — if you ever hire one — could pick up and understand your company from cold. For now, it starts small: one markdown file, stored at .agents/product-marketing.md (or .claude/product-marketing.md if you're working purely inside Claude Code), that every other skill in the framework reads before it lifts a finger.

Getting the Framework Installed

Before you can run any skill, Claude needs access to the files. If you're working in Claude Code, the fastest path is the command-line installer: run npx skills add coreyhaines31/marketingskills from your project directory, and it detects Claude Code automatically and installs into .claude/skills/. If you'd rather install everything through Claude Code's own plugin system, add the marketplace with /plugin marketplace add coreyhaines31/marketingskills and then /plugin install marketing-skills. If you're working somewhere that doesn't support either — the desktop app without Claude Code, or a browser session — you can clone the repository directly and treat the skill files as reference material you paste in in relevant part, though you'll get the smoothest experience with Claude Code, since that's what the framework is built against. Whichever path you take, do it once, before you continue reading, so the rest of this chapter is something you're doing, not something you're reading about.

THE METHOD

The First-Session Method

1. Install the framework (npx skills add coreyhaines31/marketingskills, or the plugin marketplace equivalent) into your project.
2. Tell Claude: “Let's set up my product marketing context using the product-marketing skill.” Let it run the interview — don't pre-write your answers.

> 3. Answer in your own words, not marketing-speak. When a question feels too broad ("who is your audience?"), ask Claude to narrow it, or answer with a specific, real customer.
> 4. When Claude summarizes a section back to you, actually read it and correct anything that sounds like it came from a pitch deck instead of from you.
> 5. Save the file, then open it and read the whole thing once, end to end, exactly as if you were a stranger meeting your company for the first time.

Why the Interview Matters More Than the File

The product-marketing skill's job is to capture foundational positioning and messaging information so you never have to repeat yourself to another skill again — but the file itself is not really the point. The interview is. A good version of this skill won't ask "what problem do you solve," because founders answer that question in vague, defensible generalities: "we help small businesses run more efficiently." It will ask something sharper: what's the number one frustration that actually brings a customer to you. Dana's honest answer had nothing to do with efficiency. It was: "my dispatcher was scheduling three trucks to the same job because we were still doing it on a whiteboard and a group text, and one Tuesday we lost a customer over it." That sentence is usable. "We help small businesses run more efficiently" is not.

This is the pattern behind every well-run skill interview in this framework, not just this one: ask for the specific, frustrating, concrete moment, not the polished summary. When Claude asks you a question and your instinct is to answer in the language of a slide, that's the signal to stop and ask yourself what actually happened the last time a customer described their problem to you, in their own words. Capture that. A conversion copywriter will thank you for it in Chapter 7; so will whoever writes your ads in Chapter 9.

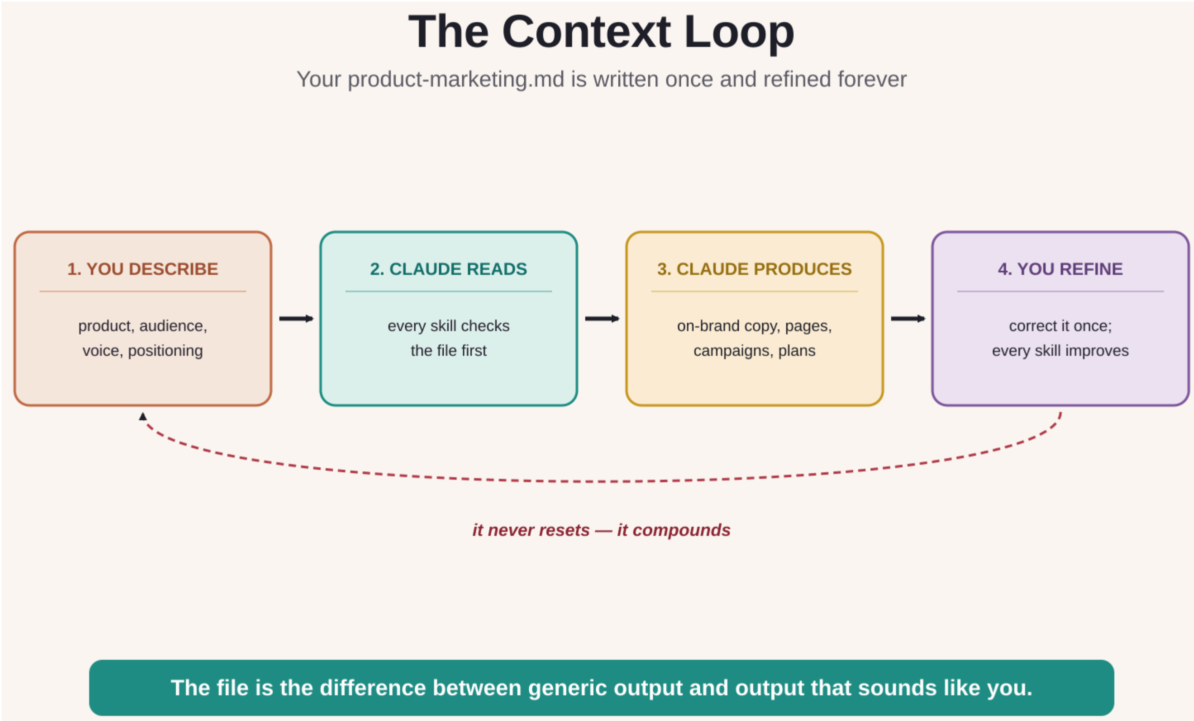

Figure 2. *The context loop — you describe once, every skill reads it, and each correction compounds instead of resetting.*

Notice what the loop in Figure 2 implies: this file is never finished. The first version you write this chapter will be thin — a page, maybe two, built from a twenty-minute conversation. That's correct. The mistake would be treating it as a one-time chore instead of a document that gets a little sharper every time a skill's output makes you wince and you go fix the paragraph that caused it. Six months from now, your product-marketing.md should read like it was written by someone who has watched two hundred customers use your product, because by then, in a sense, it has been.

A Second Example: The Second Founder

Marcus Odei ran the same exercise eighteen months earlier, for Ledgerline, a compliance and reconciliation tool for mid-market fintechs. His first draft of the file was almost useless — heavy on category language ("the leading platform for automated reconciliation"), light on anything a copywriter could use. What changed it wasn't a better skill; it was Marcus going back after his first two customer calls post-launch and adding the actual sentence one customer used: "I stopped trusting our own numbers before month-end close." That line ended up, almost unedited, as the opening line of Ledgerline's homepage two chapters from now. The lesson generalizes regardless of company size: the file gets better when you feed it what customers actually say, not what you assume they'd say.

> **THE MOVE THAT MATTERS MOST**
>
> Every time a real customer describes their problem in their own words — on a call, in a support ticket, in a review — copy that sentence into your product-marketing.md verbatim. It will outperform anything you or Claude invent from scratch.

KEY TAKEAWAYS

- Install the framework once, at the start of a project, so every skill you run afterward can find and read your context file automatically.
- The product-marketing skill's interview matters more than the resulting document — answer with specifics and customer language, not category language.
- The file lives at .agents/product-marketing.md (or .claude/ as a fallback) and every other skill checks it first, before asking you anything.
- Your first draft should be thin and honest, not polished and vague. It gets sharper over time, not right the first time.
- Treat customer language as raw material: when a real customer describes their problem in their own words, that sentence belongs in your context file.

EXERCISE

Exercise

Self-Reflection

Before running the skill, write down — in one sentence, in your own words, no jargon — the specific frustrating moment that led your most recent customer to you.

Collaborate

Run the product-marketing skill with Claude now. Let the interview run its full course before you save anything. Where Claude's summary sounds more polished than true, correct it on the spot.

Reflection

- Which section of the file was hardest to answer honestly, and what does that difficulty tell you?
- What's one sentence a real customer has said to you that belongs in this file verbatim?

Stretch Goal

Pull up three recent support tickets, reviews, or sales call notes and mine them for customer language you can fold into the file this week.

REFLECTION

- *What would a total stranger understand about your company after reading only this file?*
- *Where are you still describing your product the way you wish it were, rather than the way customers actually experience it?*

WHAT'S NEXT

With the context file in place, Part II turns to strategy: how the marketing-plan skill takes that file and turns it into an actual quarter of prioritized work.

PART II

Strategy & Positioning

Turn the context file into a real plan for a real quarter.

CHAPTER THREE

The 90-Day Marketing Plan

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Run the marketing-plan skill as a structured, fractional-CMO-style engagement rather than a one-shot document request.

✓ Size a plan's budget and channel mix to your actual stage — pre-seed, seed, or growth.

✓ Distinguish a growth pattern from a channel, and use that distinction to avoid copying a competitor's playbook wholesale.

✓ Produce a first working 90-day plan for your own company.

Dana's first instinct, three weeks after building the context file, was to ask Claude for “a marketing plan” the same way she'd have asked a contractor for “a quote.” What came back from the marketing-plan skill was not a document. It was a conversation.

The skill opened by reading the product-marketing.md file Dana had built in Chapter 2, then asked a question no template would have asked: what's actually true about where Northstar Ops is right now — how much runway, how many people, what's already been tried, what worked, what flopped. Dana explained: two people, herself and a part-time contractor doing support, eleven paying customers, a homepage that had never been touched since launch, and a marketing budget best described as “whatever's left after payroll,” which this month was about six hundred dollars. The skill didn't flinch or pad the plan with paid-media line items that would have been fantasy at that budget. It asked a second question: what's the single outcome that would make the next ninety days feel like a win. Dana said: twenty-five paying customers, without hiring anyone.

Why a Plan Should Behave Like an Engagement, Not a Document

Most requests for “a marketing plan” produce a plan that could describe almost any company: a content calendar, a social schedule, a vague nod toward SEO, a paid budget sized to nothing in particular. The marketing-plan skill in this framework is deliberately built to behave differently — more like a fractional CMO running a real engagement than a document generator. It draws on published frameworks for how founding-stage marketing actually gets built (the budget-planning and growth-pattern models in this framework come from Corey Haines's Founding Marketing), and it resists the urge to hand you a plan before it understands your stage, because a plan sized wrong in either direction wastes exactly the resource you have the least of. Too ambitious, and you burn a shoestring budget chasing tactics built for a Series B company. Too conservative, and you leave growth on the table you could actually afford.

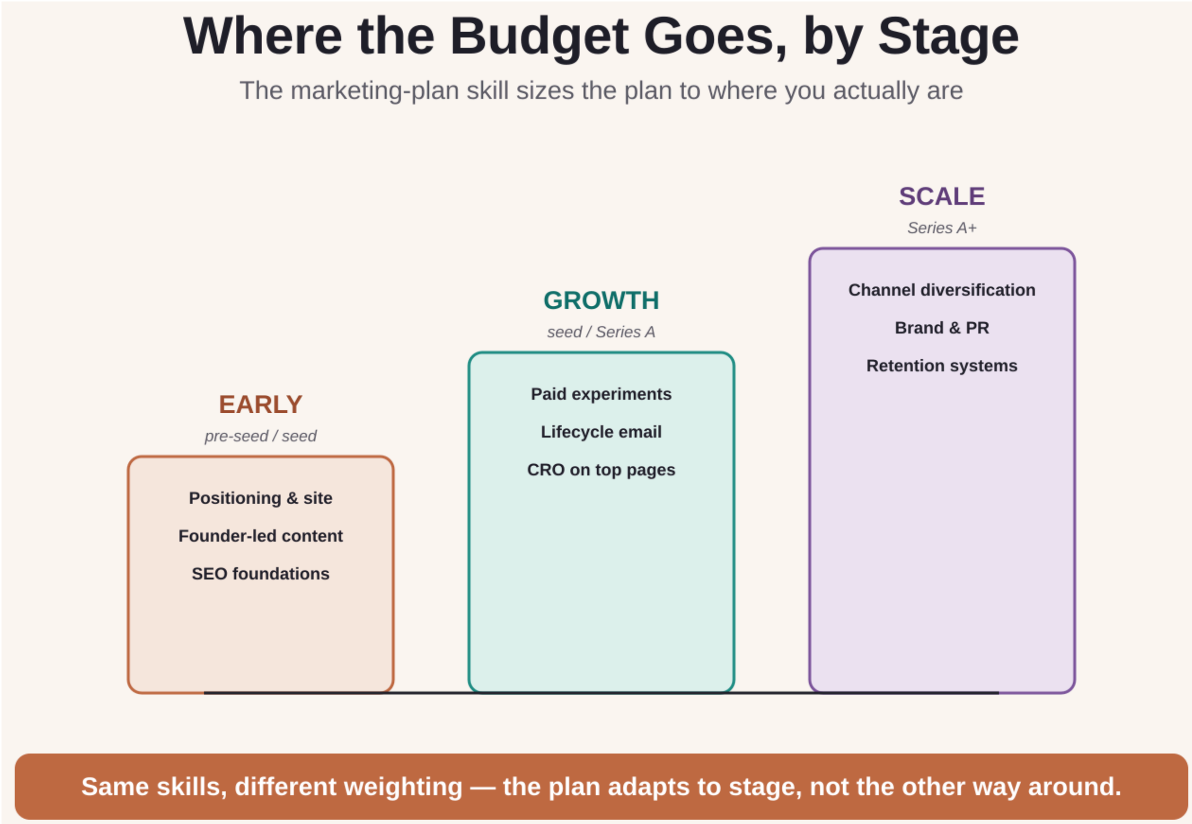

Figure 3. *Budget and focus shift by stage — the same skill, weighted differently for early, growth, and scale companies.*

Figure 3 shows the shape of that shift. At Dana's stage — pre-seed, two people, a few hundred dollars a month — the plan the skill produced weighted almost everything toward positioning, a rebuilt homepage, and founder-led content: Dana answering the exact questions home-service owners were already typing into Google and Reddit, in her own voice, because that's inventory that costs time, not money, and because at eleven customers, every one of them still remembers exactly who Dana is. It's a different plan than the one Marcus ran at Ledgerline post-Series-A, where the same skill, run against Ledgerline's context file, weighted toward paid experiments and lifecycle email, because Marcus had a small team who could execute in parallel and a real budget to test channels at a pace Dana couldn't afford.

Growth Patterns, Not Just Channels

The second thing the marketing-plan skill pushed back on was Dana's instinct to list channels — “we'll do some SEO, some social, maybe try a few ads.” A channel is where you show up. A growth pattern is the underlying mechanism that actually compounds, and the skill's job is to find the one or two patterns that fit a company's specific economics before it ever discusses channels. For Northstar Ops, with a low-consideration product used by busy tradespeople who trust word of mouth from other tradespeople more than any ad, the pattern that fit was referral-and-community: get the eleven existing customers talking to the other business owners in their trade associations and local contractor Facebook groups, and build the tooling to make that easy, rather than trying to out-bid category leaders on paid search for "scheduling software." For Ledgerline, selling into finance teams who research extensively before ever taking a call, the fitting pattern was content-and-authority: become the source finance leaders trust on reconciliation, and let that trust convert slowly through a longer, more deliberate sales cycle.

THE METHOD

The Quarter-Sizing Method

1. State your real constraints first — team size, runway, and actual available budget — before asking for any plan. Resist rounding up.
2. Name one outcome that would make the quarter a win, specific and countable (customers, revenue, activation rate), not a vague direction.
3. Let the skill propose a growth pattern before it proposes channels. If it jumps straight to channels, ask it to justify the underlying mechanism first.
4. Push back on anything sized for a company you're not. If a line item assumes a team or budget you don't have, say so and ask for the version that fits.
5. Leave the engagement with a plan you could execute starting Monday — not a strategy deck, a punch list.

A channel is where you show up. A growth pattern is why it compounds.

WATCH FOR

- Asking for "a marketing plan" in one line and accepting whatever comes back. The value of this skill is almost entirely in the back-and-forth about your real constraints — skipping it produces the same generic output you'd get from any template.
- Copying a competitor's channel mix because it's visible. You can see a competitor's ads; you can't see their unit economics, their team size, or how long that channel actually took to pay off.
- Re-running the plan every time a metric dips. A 90-day plan needs roughly 90 days to show whether the underlying pattern is working — give it the runway before declaring it wrong.

KEY TAKEAWAYS

- The marketing-plan skill behaves like a fractional CMO engagement: it asks about your real stage, team, and budget before proposing anything.
- Plans should be sized to your actual stage — early-stage plans weight toward positioning and founder-led content; growth-stage plans weight toward paid experiments and lifecycle systems.
- A growth pattern is the underlying mechanism (referral, content-and-authority, product-led virality) that makes growth compound; channels are just where that pattern shows up.

- Naming one specific, countable 90-day outcome up front keeps the whole plan honest and prevents scope creep into a vague strategy document.
- The right plan for your competitor is very often the wrong plan for you — match the pattern to your economics, not their visible tactics.

EXERCISE

Exercise

Self-Reflection

Write down your actual team size, runway, and monthly marketing budget — the real numbers, not the ones you'd be comfortable saying out loud to an investor.

Collaborate

Run the marketing-plan skill with Claude, feeding it your product-marketing.md and your real constraints from above. Push back at least once on anything that feels sized for a different company.

Reflection

- What growth pattern did the plan land on, and does the underlying mechanism actually make sense for how your customers decide to buy?
- What's the one countable outcome you're now accountable for in the next 90 days?

REFLECTION

- *Where have you been choosing channels because they're visible, rather than because the underlying pattern fits your business?*
- *What would you actually do differently this quarter if you fully believed your stated budget was fixed?*

WHAT'S NEXT

A plan needs raw material to execute against. Chapter 4 covers where that raw material comes from: marketing ideas sized to your situation, the psychology behind why they work, and the offer underneath everything you sell.

CHAPTER FOUR

Ideas, Psychology & Offers

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Use the marketing-ideas skill to generate tactics filtered by your actual stage and resources, not a generic “100 growth hacks” list.

✓ Apply at least three psychological principles from the marketing-psychology skill to a real page or campaign.

✓ Distinguish an offer from the copy that describes it, and use the offers skill to strengthen the thing underneath your pricing page.

✓ Connect pricing decisions to the offer and positioning work already captured in your context file.

“What marketing should I even be doing?” is the question that had been sitting, unanswered, in the back of Dana's head since before the 90-day plan gave it a shape.

The plan from Chapter 3 said, in effect: lean into referral and community. It did not say exactly what to do on a Tuesday afternoon. That's the gap the marketing-ideas skill is built to close — not as a substitute for strategy, but as a working library of proven tactics, filtered against the plan you already have, so that “what should I actually build this week” has a concrete answer instead of a Twitter thread's worth of scattered inspiration.

Ideas Filtered by Reality, Not Volume

Asked for ideas cold, most tools — and most people — will hand you a long, undifferentiated list: run a giveaway, start a podcast, try TikTok, build an affiliate program. Somewhere in that list is probably something that would work for Northstar Ops. Finding it by trial and error is expensive when your monthly budget is six hundred dollars. The marketing-ideas skill instead checks the product-marketing context first, then filters its library of proven SaaS marketing tactics against your specific stage, resources, and the growth pattern already agreed in your plan, before it ever proposes a specific idea. For Dana, that meant the giveaway and the TikTok strategy never came up; what came up was a referral incentive structured around the trade associations Northstar's existing customers already belonged to, and a short list of the exact questions home-service owners were typing into Google that Northstar Ops was positioned to answer better than the category leader.

The skill's real value is in what it declines to suggest as much as what it does. A tactic that would work brilliantly for a product-led, self-serve tool with a five-minute signup is often close to useless for a business that sells through relationships and word of mouth, and vice versa. Naming your growth pattern in Chapter 3 is what lets this filtering actually happen instead of producing the same generic hundred-and-forty-item list everyone gets.

The Psychology Underneath the Tactic

A referral incentive is a tactic. Reciprocity — the reason people feel obligated to return a favor — is the psychology underneath it, and understanding the underneath is what lets you adapt a tactic when the first version doesn't quite land. The marketing-psychology skill exists to apply mental models and behavioral science deliberately, rather than by accident, to the pages, emails, and offers you're already building. Dana's first pass at a referral program offered a flat discount for both the referrer and the new customer — reasonable, but flat. Working through the psychology skill surfaced a sharper version: frame the reward as recognition inside the trade community Northstar customers already respected (a feature on Northstar's page for top local shops, alongside the discount), which layered social proof and status on top of reciprocity instead of relying on the discount alone. The response rate roughly doubled.

Marcus ran the same skill against a very different problem: Ledgerline's trial-to-paid conversion, which was stalling around the point finance teams had to get sign-off from a director who'd never used the product. The relevant principle wasn't reciprocity — it was loss aversion, and the specific tactic it suggested was showing, inside the trial itself, exactly what reconciliation work would revert to manual spreadsheets the day the trial ended, rather than only advertising what the paid tier added. Naming the loss outperformed describing the gain, which is a pattern that shows up again and again once you know to look for it, and is precisely the kind of judgment a skill like this exists to bring to a page you'd otherwise write from instinct alone.

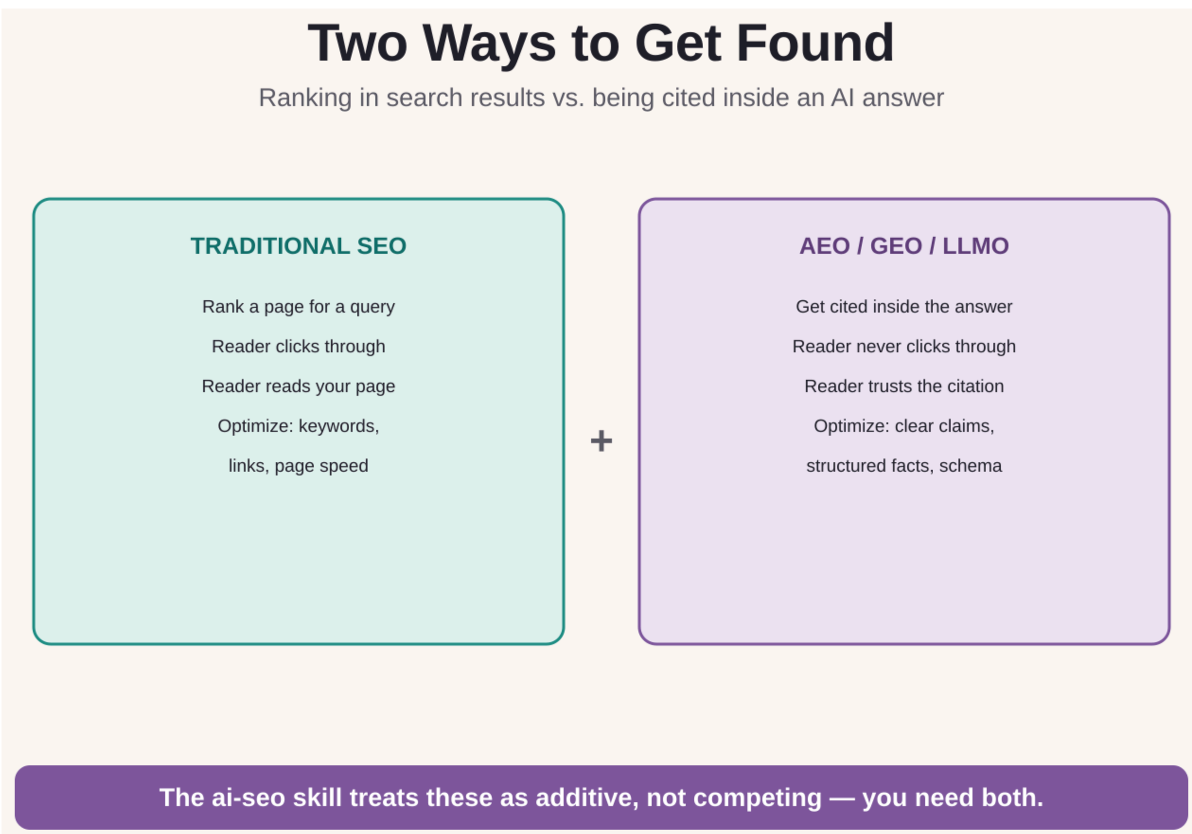

Figure 4. *SEO and AI-search optimization are additive, not competing — a preview of the choice Chapter 6 covers in depth.*

A USEFUL HABIT

Before publishing any page or email, ask Claude which specific psychological principle the copy is leaning on, and whether that principle actually fits how your customer makes this particular decision. If it can't name one, the copy is probably running on habit, not intent.

The Offer Underneath the Page

Most “our copy isn't working” requests are, once you dig in, actually “our offer isn't strong enough” requests wearing a copywriting costume. The offers skill exists for exactly this distinction: it handles the thing you actually sell — value framing, bonuses, guarantees, how risk is allocated between you and the customer — separately from the page that describes it, because no amount of copywriting fixes an offer that doesn't give a hesitant buyer a reason to stop hesitating. For Northstar Ops, the sharper offer that came out of this work wasn't a price change; it was a thirty-day, white-glove onboarding guarantee — Dana personally helping migrate a shop's existing schedule by hand if their team couldn't do it themselves — which addressed the real fear (“what if switching tools breaks a chaotic week”) more directly than any homepage headline could.

Pricing sits right next to the offer, and the pricing skill is where packaging and monetization strategy get decided — tiers, what's gated, what's usage-based versus flat — informed by the same context file everything else in this book draws from. Dana's pricing conversation with Claude wasn't about picking numbers; it was about recognizing that Northstar's single flat-rate tier was punishing the smallest shops (a two-truck operation paying the same as an eight-truck one) and leaving obvious upside on the table with the larger ones. The fix was a simple per-truck tier structure, which the pricing skill helped model against Dana's actual customer mix before it ever went live.

KEY TAKEAWAYS

- The marketing-ideas skill filters a large library of tactics against your actual stage and growth pattern — its value is in what it rules out, not just what it suggests.
- Every effective tactic has a psychological principle underneath it (reciprocity, loss aversion, social proof); naming that principle is what lets you adapt the tactic instead of copying it blindly.
- “The copy isn't working” is very often “the offer isn't strong enough” in disguise — the offers skill strengthens the thing you're selling, separately from the page that describes it.
- Pricing and packaging decisions belong to the same context as everything else — the pricing skill models tiers against your real customer mix, not abstract best practice.
- A referral discount, a guarantee, and a pricing tier are all expressions of the same underlying offer — keep them consistent with each other.

EXERCISE

Exercise

Self-Reflection

Pick one page or email currently underperforming. Write one sentence naming what you believe the actual problem is — the copy, or the offer underneath it.

Collaborate

Ask Claude to run the marketing-ideas skill for three tactics that fit your stage, then the marketing-psychology skill to name the principle underneath the strongest one, then the offers skill to check whether your current offer supports that tactic.

Reflection

- Which tactic did the skill rule out that you would have tried anyway, and why does it not fit your situation?
- Is your current pricing structure actually matched to how your different customer segments get value, or is it one-size-fits-most out of convenience?

REFLECTION

- *Where have you been treating a weak offer as a copywriting problem?*
- *What's one psychological principle you're currently leaning on by instinct that you could now name and apply deliberately?*

WHAT'S NEXT

With a plan, a filtered set of tactics, and a sharpened offer in place, Part III turns to getting found — starting with SEO, and the newer discipline of getting cited inside AI-generated answers.

PART III

Getting Found

Earn attention in search results and inside AI answers alike.

CHAPTER FIVE

SEO for the Agent Era

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

- ✓ Run a structured technical and on-page SEO audit with the seo-audit skill instead of a scattershot checklist.
- ✓ Use site-architecture to fix navigation, URL structure, and internal linking before writing a single new page.
- ✓ Know when programmatic-seo is the right tool — and the real risk of using it too early.
- ✓ Add basic schema markup so both search engines and AI systems can parse what your pages actually say.

Northstar Ops had a homepage, a pricing page, and nothing else — no blog, no help center, nothing indexed beyond those two URLs. Dana's assumption was that SEO meant writing blog posts, and that blog posts were a luxury for later. The seo-audit skill's first move corrected both assumptions at once.

Rather than starting with content, the skill started with what already existed: it crawled the two live pages and came back with a short, unglamorous list of technical problems that were actively costing Northstar visibility — missing meta descriptions, a title tag that just said "Home," no structured heading hierarchy, images with no alt text, and a site that loaded slowly enough on mobile to plausibly be losing visitors before they ever read a word. None of this required writing anything new. All of it was costing Dana rankings for the two pages Northstar already had, for free, before a single blog post existed. This is the order the skill enforces on purpose: fix what's broken before you build what's new, because new content built on a broken foundation just means more broken pages.

Structure Before Content

Once the technical issues were logged, the seo-audit skill handed off to site-architecture, which is where page hierarchy, navigation, and URL structure get planned deliberately instead of accreting by accident. For a two-page site this sounds like overkill, and that's exactly the point at which it matters most: the architecture you choose now becomes the scaffolding every future page hangs on. Working with Claude through this skill, Dana mapped out a structure built around the actual questions home-service owners search for — /scheduling-software-for-plumbers, /scheduling-software-for-electricians, /scheduling-software-for-cleaning-services — each a real page addressing a real trade's specific scheduling headaches, linked from a new "Who It's For" section on the homepage, rather than one generic "Features" page trying to speak to all three trades equally and specifically to none of them.

This is also where the programmatic-seo skill enters, and where it needs the most caution. Programmatic SEO builds pages at scale from templates and data — exactly the pattern behind those /scheduling-software-for-[trade] pages Dana was planning, and a legitimate, powerful tactic when the underlying data genuinely differs page to page and each page earns its existence. The risk, and the reason this skill comes with real guardrails, is generating hundreds of thin, near-identical pages that differ only in a swapped-out keyword, which search engines increasingly recognize and penalize. Dana's three trade pages worked because each one addressed genuinely different pain points — an electrician's scheduling problem (permit inspections, multi-day jobs) is not a cleaner's scheduling problem (recurring weekly visits, key management). Ten trade pages, each with real substance, beats two hundred thin ones every time.

Fix what's broken before you build what's new. New content on a broken foundation is just more broken pages.

A Second Example: Schema at Scale

Marcus's SEO problem at Ledgerline looked nothing like Dana's. Ledgerline already had sixty blog posts, a real content team producing four a week, and rankings for a wide spread of reconciliation-related terms — but almost none of that content was showing up as a source when someone asked ChatGPT or Perplexity a related question, even for topics Ledgerline covered better than anyone ranking above it. That gap sent Marcus to the schema skill: adding structured data markup so that both traditional search engines and AI systems could parse exactly what each article claimed, who wrote it, and how it related to other pieces on the site, rather than inferring it imperfectly from unstructured prose. Within a few weeks of adding FAQ schema, author schema, and article schema across the existing library — no new content required — Ledgerline began showing up as a cited source in AI-generated answers to reconciliation questions it had been quietly winning on Google for a year without anyone noticing.

THE METHOD

The Foundation-First Method

1. Run seo-audit on whatever pages already exist, even if it's only one or two. Fix every technical issue it surfaces before writing anything new.
2. Use site-architecture to plan the page hierarchy and URL structure you want in six months, not just the next page you're about to publish.
3. Before using programmatic-seo, confirm each planned page would genuinely earn a visit on its own — if the only difference between pages is a swapped keyword, don't build it.
4. Add schema markup to your most important existing pages before writing new content — it's often the highest-leverage, lowest-effort fix available.
5. Revisit the audit quarterly. SEO foundations decay as a site grows; don't treat the first pass as permanent.

WATCH FOR

- Writing new content before fixing technical debt. A great article on a page search engines can't properly crawl or parse is invisible no matter how good it is.
- Programmatic SEO used as a shortcut instead of a genuine data advantage. If you couldn't defend each page's usefulness individually, don't publish it.
- Treating schema as a one-time task. As pages change, schema needs to change with them, or it starts actively misrepresenting your content.

KEY TAKEAWAYS

- seo-audit fixes what already exists before anything new gets built — technical debt on existing pages is usually the highest-leverage, lowest-effort fix available.
- site-architecture plans page hierarchy and URL structure deliberately, so new pages hang off a scaffolding instead of accreting randomly.
- programmatic-seo is powerful when each generated page genuinely differs and earns its existence, and actively harmful when it produces thin, near-identical pages at scale.
- schema markup helps both traditional search engines and AI systems parse what a page actually claims — it's often the fastest way to make existing content newly visible.
- SEO foundations decay as a site grows; treat the audit as a quarterly habit, not a one-time project.

EXERCISE

Exercise

Self-Reflection

List every page currently live on your site. For each, write one honest sentence: does it have a real title tag, meta description, and clear heading structure, or are you not sure?

Collaborate

Run the seo-audit skill against your live site with Claude. Fix the top three technical issues it surfaces before doing anything else this week.

Reflection

- What's one page you've been planning to write that, under closer inspection, would actually be a thin, near-duplicate of a page you already have?
- Which of your existing pages would benefit most from schema markup, and what would that unlock?

REFLECTION

- *Where have you been building new content to compensate for foundational problems you haven't actually fixed?*
- *What does your site's URL structure currently communicate about how your product is organized — and is that the structure you'd choose on purpose?*

WHAT'S NEXT

> *Fixing the foundation is necessary, but it answers only half the getting-found question. Chapter 6 covers the other half: how content gets found, cited, and trusted inside AI-generated answers, not just search results.*

CHAPTER SIX

Winning AI Search

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Explain the difference between ranking for a query and being cited inside an AI-generated answer.

✓ Apply the ai-seo skill's core practices — clear claims, structured facts, and citable authority — to a real page.

✓ Recognize AEO, GEO, and LLMO as overlapping names for the same underlying discipline, and know which term shows up where.

✓ Decide, for your own content, when traditional SEO and AI-answer optimization point in the same direction and when they diverge.

A friend of Dana's mentioned, almost as an aside, that she'd asked Claude a question about scheduling software for her cleaning business and gotten back a paragraph that referenced Northstar Ops by name — before Northstar's homepage had ranked on the first page of Google for anything. Dana had no idea that was even possible to influence on purpose.

It is, and it's what the ai-seo skill is built around: optimizing content so it gets cited by LLMs and surfaced inside AI-generated answers, a discipline that goes by several overlapping names depending on who's writing about it — AEO (answer engine optimization), GEO (generative engine optimization), and LLMO (large language model optimization). They describe the same underlying shift from slightly different angles, and the skill treats them as one connected practice rather than three competing frameworks to choose between.

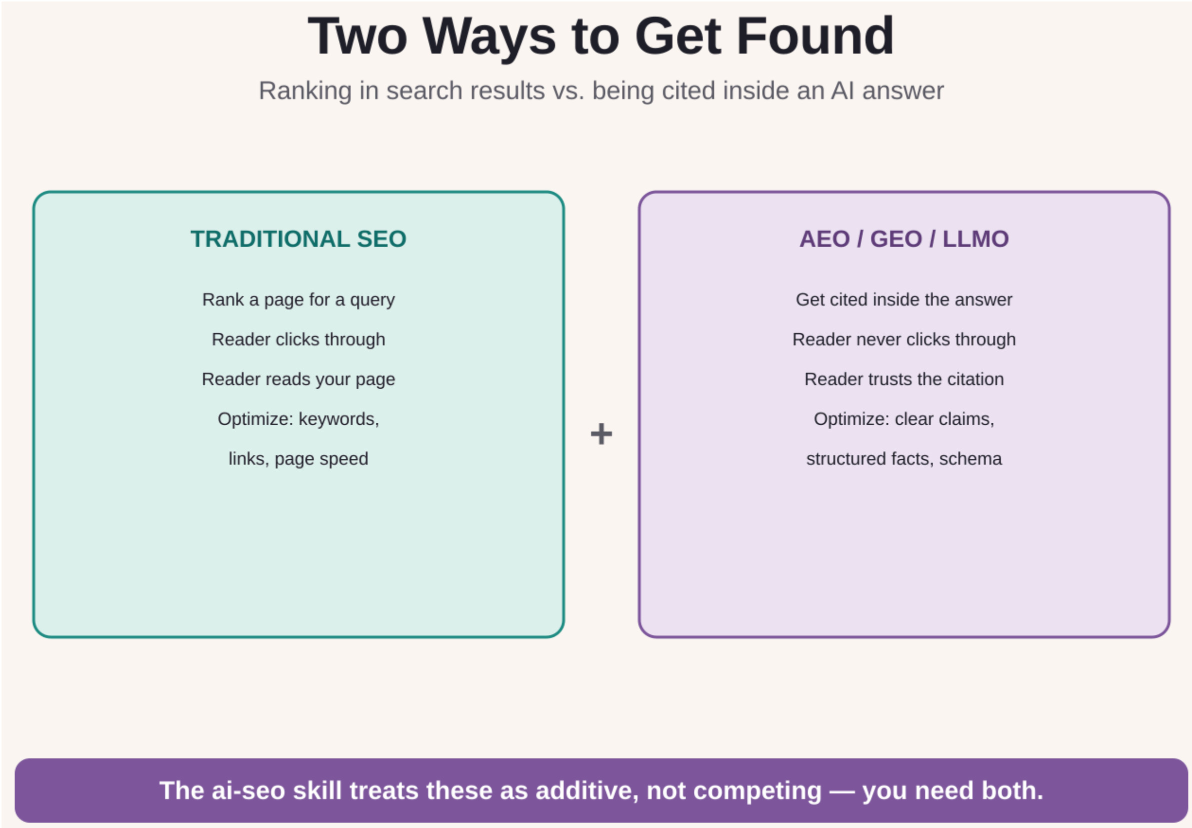

Figure 4. *Two funnels, not one: ranking for a query and getting cited inside an AI answer optimize for different moments.*

A Different Funnel, Not a Replacement

The habit worth breaking first is thinking of AI-answer visibility as a new version of SEO that makes the old kind obsolete. It's closer to a second, parallel funnel with a genuinely different shape. Traditional SEO optimizes for a click: rank for a query, earn the visit, let the page do the persuading once someone arrives. AI-answer optimization optimizes for a citation inside a response the reader may never click through from at all — which means the persuading has to happen inside the two or three sentences an AI system might quote or paraphrase, not on the page the reader may never see. Both funnels can run off the same underlying content, but they reward different things: SEO rewards depth, backlinks, and dwell time; AI-answer visibility rewards clarity, structure, and claims specific enough to be extracted cleanly.

That last point is where most existing content quietly fails. A paragraph that hedges, buries its actual claim in the third sentence, or requires the surrounding context to make sense is nearly impossible for a language model to lift cleanly into an answer, even if the information in it is accurate and well-researched. The ai-seo skill's most concrete, immediately usable practice is rewriting key claims into standalone, unambiguous statements near the top of a section — the kind of sentence that reads correctly whether or not the reader has seen anything before it. Dana's Northstar Ops “why we're different” section originally opened with three sentences of scene-setting before it ever stated a clear fact. The rewrite moved the load-bearing claim — "Northstar Ops assigns jobs to the nearest available truck automatically, which cuts average drive time between jobs by roughly a third" — to the first sentence, specific number included, and left the scene-setting for after.

Structured Facts and Earned Authority

The second lever, and the one that connects directly back to Chapter 5, is structure: schema markup, clear headings, and FAQ-formatted content give an AI system an explicit map of what a page claims and how those claims relate to each other, rather than forcing it to infer structure from prose. This is why the schema and ai-seo skills are meant to be used together — schema is the mechanism, ai-seo is the judgment about which claims deserve that mechanism applied to them. The third lever is authority signals that exist independently of your own site: being mentioned, quoted, or linked from other sources an AI system already trusts. This is slower to build and less directly controllable, which is exactly why Marcus's sixty-post content library at Ledgerline — built over more than a year, cited by industry newsletters and quoted in a few trade publications along the way — started paying off in AI citations almost as soon as the schema work made it legible, while Dana's brand-new three-page trade cluster is still building that kind of external trust from zero.

> **THE OPEN KNOWLEDGE FORMAT**
>
> As of this framework's most recent updates, ai-seo also covers Google's Open Knowledge Format (OKF), an emerging markdown specification for representing a site's content as an agent-readable bundle. It's early and evolving — worth watching, not yet worth over-investing in ahead of the fundamentals covered in this chapter.

The persuading has to happen inside the two sentences an AI might quote — not on the page the reader may never see.

WATCH FOR

- Treating AEO/GEO/LLMO as a reason to abandon traditional SEO. The two funnels share almost all of the same underlying content investment — they diverge in framing, not in foundation.
- Hedging claims to sound careful. A precise, specific sentence is both more citable and, done honestly, more persuasive than a vague, hedged one.
- Chasing AI-answer visibility before the technical and structural foundations from Chapter 5 are in place. Structure is the mechanism that makes any of this legible in the first place.

KEY TAKEAWAYS

- AEO, GEO, and LLMO are overlapping names for optimizing content to be cited inside AI-generated answers — the ai-seo skill treats them as one connected discipline.

- AI-answer visibility is a parallel funnel to traditional SEO, not a replacement — it rewards clarity and extractable claims where SEO rewards depth and dwell time.
- Rewriting key claims into standalone, specific, unhedged statements near the top of a section is the single most immediately actionable practice in this chapter.
- Structure (schema, headings, FAQ formatting) and external authority (citations, mentions) are the two levers that make claims both findable and trustworthy to AI systems.
- Both funnels can and should run off the same underlying content — you rarely need to choose between optimizing for one or the other.

EXERCISE

Exercise

Self-Reflection

Pick your single most important page. Read its opening paragraph and ask: if an AI system quoted only the first sentence, would it be an accurate, standalone, useful claim about your product?

Collaborate

Ask Claude to run the ai-seo skill against that page, rewriting its key claims for clarity and extractability, and to flag where schema markup from Chapter 5 would reinforce the rewritten claims.

Reflection

- Where has hedging crept into your copy in a way that actually undersells a genuinely strong claim?
- What's one piece of external validation — a mention, a review, a trade-publication quote — you could realistically pursue in the next quarter to build the authority signal this chapter describes?

REFLECTION

- *How would your homepage read differently if you assumed most readers would only ever see one sentence of it, quoted out of context?*
- *Which is currently truer of your content: that it's too vague to cite, or too unstructured to be found?*

WHAT'S NEXT

Getting found is only useful if what happens next converts. Part IV turns to the moment a visitor actually lands on your page — starting with the copy itself.

PART IV

Converting Visitors

Turn attention into signups, and signups into customers who stay.

CHAPTER SEVEN

Copy That Converts

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Use the copywriting skill to write page copy that leads with the reader's problem, not your product's feature list.

✓ Apply copy-editing to sharpen existing copy without starting from a blank page.

✓ Structure a homepage around a single, sequential argument rather than a grab-bag of features.

✓ Recognize when weak copy is actually a symptom of an unresolved positioning question from earlier chapters.

Dana's original Northstar Ops homepage opened with a sentence that took two careful readings to understand: "Northstar Ops is an intelligent field service management platform built for the modern trades." It was accurate. It was also the kind of sentence that could sit, unchanged, on the homepage of a dozen competitors.

The copywriting skill's first move, once it had read the product-marketing.md context file, was to ask Dana a question that had nothing to do with word choice: who is standing in front of this page, and what were they doing thirty seconds before they landed here. The honest answer was specific — a home-service business owner, probably searching at night after a day of runs, probably frustrated by a double-booking or a missed job that day, probably skeptical of software after trying something clunky before. That person does not need to be told Northstar Ops is "intelligent" or "modern." They need to see, in the first five seconds, that the exact frustration they're carrying is the frustration this page understands.

Leading With the Problem, Not the Product

Every skill in this framework that touches copy shares one underlying discipline: lead with the reader's situation before you lead with your solution. It sounds obvious stated plainly and is nonetheless the single most common mistake on early-stage SaaS homepages, because founders know their product intimately and default to describing it, while a visitor arrives knowing only their own problem and needs to be met there first. The rewrite the copywriting skill produced for Northstar Ops opened instead with: "Your dispatcher shouldn't have to remember which truck is closest. Northstar Ops does it automatically — no more double-bookings, no more fifteen-minute phone tag before a job starts." The product is still there, in the second sentence. But the door the reader walks through is their own frustration, not a category label.

This same discipline scales down to a single sentence and up to an entire page's structure. A strong homepage, in this framework's view, is not a list of features arranged by importance; it's a single sequential argument, the way a good salesperson would make a case in person — problem, why it matters, why existing solutions fall short, what you do differently, proof that it works, and finally the ask.

Dana's rebuilt homepage followed exactly that arc: the double-booking problem, why it costs real money and real customer trust, why a shared whiteboard and group texts can't scale past a few trucks, what Northstar Ops automates instead, a short quote from one of the eleven existing customers, and a single clear call to action. Nothing on the page was competing with anything else on the page for the reader's attention.

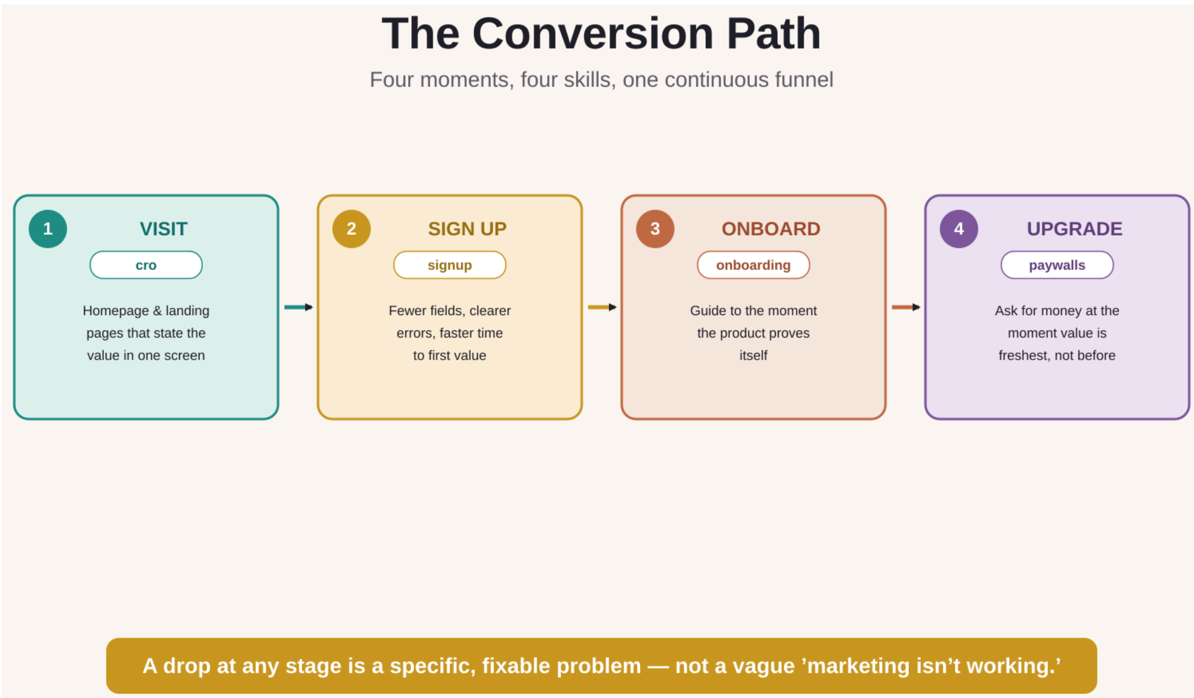

Figure 5. *The conversion path this chapter and the next both feed — copy earns the click; CRO and signup carry it through.*

Editing What Already Exists

Not every copy problem starts from a blank page. Marcus's situation at Ledgerline was the opposite of Dana's: plenty of existing copy across a growing site, written at different times by different people, with a voice that had drifted noticeably between the oldest pages and the newest ones. This is what the copy-editing skill is built for — refreshing and tightening copy that already exists rather than replacing it wholesale, catching drift in voice, cutting sentences that had accumulated hedge words and qualifiers over successive small edits, and flagging claims that had gone stale as the product changed. Run across Ledgerline's fifteen highest-traffic pages, it didn't produce a single new idea; it made the existing ideas land the way they were originally meant to, which turned out to move Ledgerline's trial-start rate more than any of the net-new pages published that same quarter.

THE METHOD

The Problem-First Method

1. Before writing a single word, describe the reader in one sentence: who they are, and what frustration brought them to this page thirty seconds ago.

2. Open with that frustration in the reader's own language, not a category description of your product.
3. Structure the page as one sequential argument — problem, stakes, why alternatives fall short, your approach, proof, ask — not a list of features competing for attention.
4. Cut every sentence that could plausibly appear on a competitor's page unchanged. If it's not specific to you, it's not earning its place.
5. Read the final draft out loud once. Where you stumble or want to skip ahead, the reader will too.

The visitor arrives knowing only their own problem. Meet them there first.

WATCH FOR

- Rewriting copy repeatedly without improvement because the underlying positioning is unresolved. If copy still feels generic after a few honest passes, the problem may be Chapter 2's context file, not this chapter's wordsmithing.
- Describing the product before establishing the reader's situation. Even strong writing fails if it opens in the wrong place.
- Treating copy-editing as a lighter version of copywriting. It's a distinct skill with a distinct job: preserving what works while cutting what's drifted, not starting over.

KEY TAKEAWAYS

- The copywriting skill's central discipline is leading with the reader's situation before describing your product — apply it at the sentence level and the whole-page level alike.
- A strong page reads as one sequential argument — problem, stakes, shortfall of alternatives, your approach, proof, ask — not a list of features.
- copy-editing is a distinct skill from copywriting: it refreshes and tightens what already exists rather than replacing it, and is often higher-leverage on a mature site than net-new pages.
- If a rewrite still feels generic after a few honest attempts, the real issue is very often unresolved positioning from Chapter 2, not word choice.
- A useful test for any sentence: could this appear unchanged on a competitor's page? If yes, it isn't done yet.

EXERCISE

Exercise

Self-Reflection

Write one sentence describing exactly who lands on your homepage and what they were doing thirty seconds before they arrived.

Collaborate

Ask Claude to run the copywriting skill on your homepage's opening section, using the reader description above. Compare the rewrite's first sentence to your current one.

Reflection

- Which sentences on your site could currently sit unchanged on a competitor's page?
- Does your homepage read as one sequential argument, or as a list competing for the reader's attention?

REFLECTION

- *Where has describing your product, rather than the reader's situation, quietly crept into your best copy?*
- *What's the single sentence on your site you're proudest of, and why does it work?*

WHAT'S NEXT

Great copy earns attention. Chapter 8 covers what happens next — turning that attention into a signup, and a signup into someone who actually activates.

CHAPTER EIGHT

CRO, Signup & Onboarding

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Diagnose a specific conversion drop-off point using the cro skill instead of guessing at page-wide fixes.

✓ Reduce signup friction with the signup skill without sacrificing the qualification a form provides.

✓ Design an onboarding flow with the onboarding skill that gets a new user to their first real value quickly.

✓ Know when a paywall or upgrade moment, via the paywalls skill, belongs in the flow at all.

The rebuilt homepage worked — traffic that used to bounce in seconds was staying, reading, scrolling to the bottom. And then, mysteriously, converting barely better than before. Something downstream of the good copy was quietly leaking visitors.

This is exactly the situation the cro skill is built to diagnose, and it's worth noticing what it didn't do first: it didn't suggest a redesign. Conversion rate optimization, done well, starts by finding the specific point in a specific flow where people are dropping off, because "improve conversions" is not a task, it's a symptom with several very different possible causes. Working through Northstar's analytics with Claude, the cro skill isolated the actual leak: visitors were reading the whole homepage, clicking "Start Free Trial," and then abandoning on the signup form itself, roughly forty percent of them, most within the first two fields.

From the Page to the Form

That finding handed the problem to the signup skill, which exists specifically for registration and trial-activation flows — a different discipline from general page CRO, because a form has its own specific failure modes: too many fields, unclear error states, no visible indication of how many steps remain, or fields that ask for information the product doesn't actually need yet. Northstar's original form asked for company name, number of trucks, current software (if any), full billing address, and a password meeting six separate complexity requirements, all before a visitor had done anything with the product at all. The signup skill's rewrite cut this to two fields — email and a password — and moved everything else (company size, current tools, billing) to after the account existed, spread across the first few minutes of actual product use instead of front-loaded as a wall before it.

The distinction worth holding onto here: reducing friction is not the same as reducing qualification. Northstar still needed to know a customer's truck count to size their eventual per-truck pricing correctly — that information didn't disappear, it just moved to a moment when the visitor had already committed by creating an account and had a reason to answer honestly, rather than a moment when any added friction risked losing them before they'd seen the product at all.

The First Ten Minutes

Fixing signup surfaced the next leak downstream: people who created accounts and then never came back. This is squarely the onboarding skill's territory — the post-signup experience, activation, and time-to-value — and its central question is blunt: what is the single action that, once a new user takes it, makes them meaningfully more likely to stick around, and how fast can the flow get them there. For Northstar Ops, that action was scheduling one real job on one real truck; users who did that in their first session converted to paid at a rate several times higher than those who logged in and left without it. The onboarding rebuild stripped away a five-step product tour nobody was finishing and replaced it with a single guided prompt: import your trucks, add one job, watch it get assigned automatically. Everything not in service of that one action got deferred or cut.

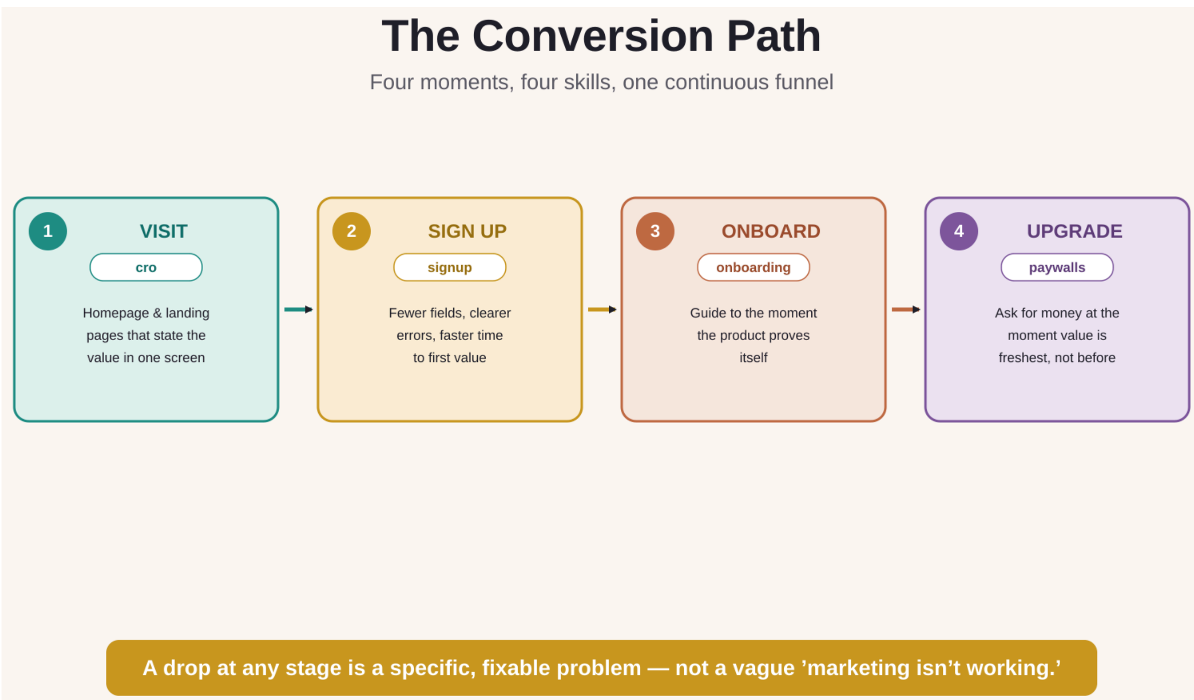

Figure 5. *The four moments this chapter and the last cover, end to end — a drop at any single stage is a specific, fixable problem.*

Marcus's version of this problem at Ledgerline looked different again, because Ledgerline sold to teams, not individuals: the activation moment wasn't one person doing one thing, it was getting a second team member invited and reconciling their first real account together, since the product's value was inherently collaborative. The onboarding skill's judgment held regardless of the specific action: find the one thing that predicts retention, and design the entire first session around removing every obstacle between signup and that thing, cutting anything — however polished — that doesn't serve it.

Where Paywalls Fit

Once activation was working, Dana faced a final question: where does the free trial actually end, and how should that moment feel. This is the paywalls skill's specific territory — in-app upgrade screens, upsell modals, and feature gates — and its core guidance runs against the instinct to gate as early as possible to protect revenue. The better-performing pattern, and the one Northstar Ops ended up with, was gating at the moment a user had already experienced real value and was reaching for more of it — in Northstar's case, the fourteenth day of trial, right as a shop's second week of scheduling was underway and the cost of going back to a whiteboard had become vivid and specific, rather than an abstract inconvenience it was on day one.

THE METHOD

The Leak-Finding Method

1. Before touching design, find the specific step in your funnel where the drop-off is largest — page, form, or first session. Fix that step, not the whole flow at once.
2. For form abandonment, ask which fields could move to after account creation without losing the information you actually need.
3. Name the single activation action that best predicts retention, and design the first session around reaching it as fast as possible.
4. Cut anything in onboarding — however polished — that doesn't serve the activation action directly.
5. Place any paywall at the moment value has already been felt, not at the moment friction is cheapest for you to add.

> *Reducing friction is not the same as reducing qualification — it's asking for the same thing at a better moment.*

WATCH FOR

- Redesigning an entire page or flow when the actual leak is one specific step. Diagnose before you rebuild.
- Cutting form fields you genuinely need instead of relocating them to a better moment in the relationship.
- Gating value before a user has felt any of it — the earliest possible paywall is rarely the best-performing one.

KEY TAKEAWAYS

- cro diagnoses the specific point in a specific funnel where visitors are dropping off — “improve conversions” is a symptom, not a task, until it's localized.
- signup reduces form friction by relocating unnecessary fields to after account creation, not by discarding the information the business genuinely needs.

- onboarding identifies the single activation action that predicts retention and strips the first session down to reaching it as fast as possible.
- paywalls perform better placed at a moment of felt value than at the earliest point that's cheapest to implement.
- Each of these four skills solves a different, specific leak — diagnose which stage is actually losing people before choosing which skill to run.

EXERCISE

Exercise

Self-Reflection

Sketch your current funnel from first visit to activated user in four or five steps. Guess, honestly, which step loses the most people — then check if you actually have the data to confirm it.

Collaborate

Ask Claude to run the cro skill on the step you suspect is weakest, then the matching skill (signup, onboarding, or paywalls) for a concrete fix.

Reflection

– What's the one action a new user could take that best predicts they'll stick around — and how many steps currently stand between signup and that action?
– Is your current paywall placed at a moment of felt value, or a moment that was simply easiest to build?

REFLECTION

- *Where have you been treating a specific, diagnosable leak as a vague, general "conversion problem"?*
- *What would change about your onboarding if you designed the first session around only one goal?*

WHAT'S NEXT

With the funnel itself in good shape, Part V turns outward — to the channels, campaigns, and lifecycle systems that fill the top of it.

PART V

Reaching, Nurturing & Keeping

Fill the funnel, stay in touch, and make growth compound.

CHAPTER NINE

Paid, Social & Content at Scale

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Decide, using the ads skill, whether paid media fits your stage before spending a dollar.

✓ Generate and iterate ad creative in volume with ad-creative without losing quality control.

✓ Build a social presence with the social skill that's sustainable for a solo founder, not a content treadmill.

✓ Use content-strategy to decide what to write before writing anything, so content compounds instead of accumulating.

Dana's six-hundred-dollar monthly budget made the ads skill's first question easy to answer honestly: not yet. Run properly, the skill doesn't open with campaign structures or bidding strategy — it opens by checking whether paid media is even the right move at your stage and budget, and for Northstar Ops at eleven customers, it wasn't.

This is worth dwelling on, because it cuts against a common instinct: paid media feels like the fastest lever to pull, and it's tempting to treat any marketing budget, however small, as an ads budget by default. The ads skill — covering Google, Meta, LinkedIn, and X campaigns — is built to talk a founder out of paid spend as often as it's built to help them run it well, because a few hundred dollars spread across a testing budget rarely generates enough data to learn anything, and is more likely to be quietly wasted than meaningfully informative. Dana's six hundred dollars went to the referral program from Chapter 4 instead. Marcus, eighteen months and a Series A later, had a genuinely different conversation: a real budget, a sales team that could handle qualified leads at volume, and unit economics well enough understood to know what a customer was worth and therefore what was safe to spend acquiring one. That's the actual gate this skill enforces — not company age, but whether you can afford to learn from the spend.

Creative at Volume, Without Losing the Thread

When Marcus's paid program was ready to scale past a first handful of test campaigns, the bottleneck stopped being budget and became creative — enough distinct headlines, descriptions, and ad variations to test meaningfully across audiences without every ad sounding like a copy-paste of the last one. This is precisely what the ad-creative skill exists for: generating and iterating ad creative in bulk, grounded in the same product-marketing.md context that keeps every variation on-message even at volume. The skill didn't just multiply one good ad into twenty; it generated genuinely different angles — one leading with the loss-aversion framing from Chapter 4's trial-conversion work, one leading with a specific number from a customer case study, one leading with the director-level sign-off objection directly — so the resulting test actually taught Marcus something about which argument moved his specific audience, rather than testing twenty shades of the same argument.

Social and Content, Sized to Reality

For Dana, the more relevant channels were social and content — not because they were free, exactly, but because they cost time she already had rather than money she didn't. The social skill's job is creating and scheduling content across LinkedIn, Twitter/X, Instagram, and the other platforms that fit a given audience, but its more important contribution for a solo founder is realism: it pushed Dana away from a five-post-a-day ambition that would have collapsed within a month, toward a rhythm she could actually sustain — three posts a week, mostly Dana answering real scheduling questions she'd fielded from customers that week, written in her own voice, on the two platforms where home-service business owners actually spent time, rather than a scattered presence across six.

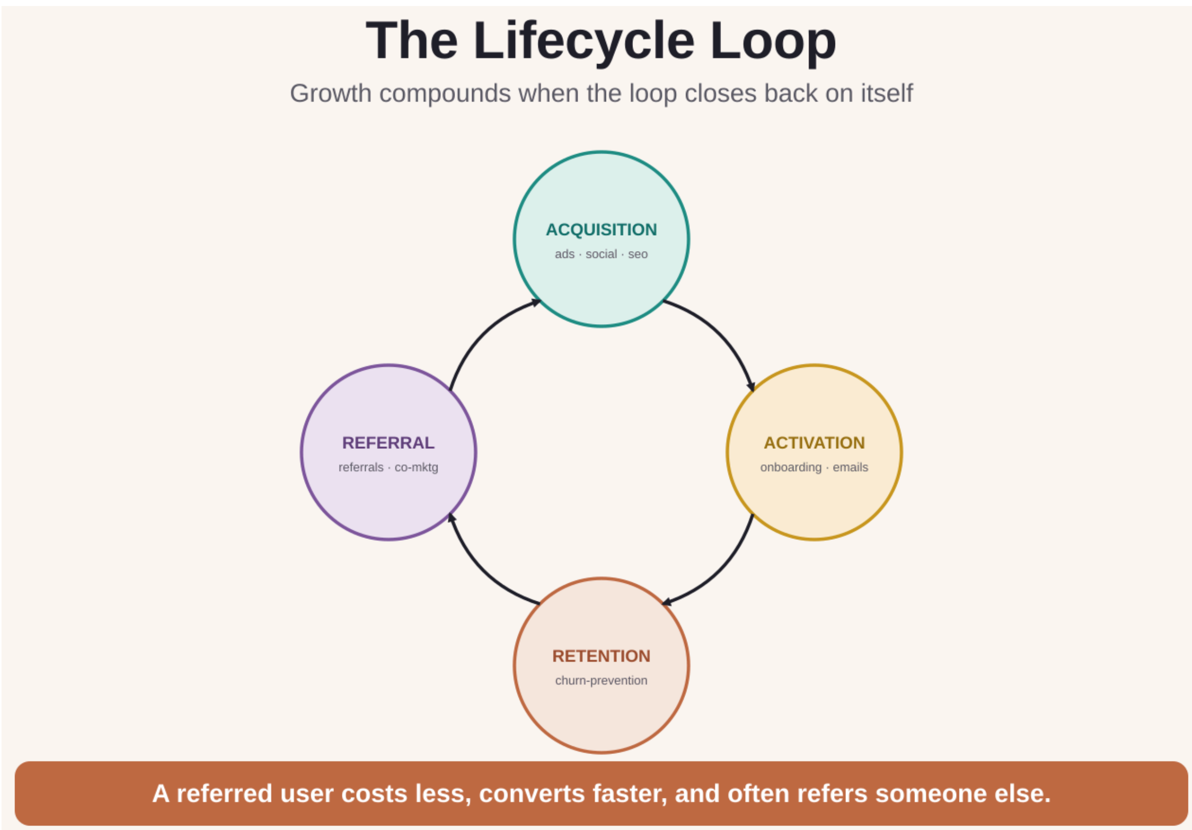

Figure 6. *Acquisition sits at the top of a loop that closes through activation, retention, and referral — this chapter fills the loop's entry point.*

The content-strategy skill sits one level up from any single piece of content, and its job is deciding what to write before anything gets written — which topics, in what order, covering which of a reader's questions, so that content compounds into a coherent body of expertise instead of accumulating into an unrelated pile of posts. For Northstar Ops, that meant mapping the specific questions each trade searched for (already surfaced back in Chapter 5's site-architecture work) against what Dana could speak to with genuine, field-earned authority, and sequencing them so each new piece linked back to and reinforced the ones before it, rather than treating every post as a standalone bet.

Paid media isn't the fastest lever until you can afford to learn from what it teaches you.

WATCH FOR

- Treating any available budget as an ads budget by default. If it's too small to generate a real learning signal, it's better spent elsewhere — or saved.
- Multiplying one ad into many near-identical variations instead of testing genuinely different angles. Volume without variation teaches you nothing new.
- Committing to a content or social cadence you can't sustain past the first enthusiastic month. A slower, consistent rhythm compounds; a burst-and-quit pattern doesn't.

KEY TAKEAWAYS

- The ads skill's first job is deciding whether paid media fits your stage at all — it's built to talk you out of underfunded tests as often as it's built to run well-funded ones.
- ad-creative generates genuinely different creative angles at volume, grounded in your context file, so tests teach you something instead of just multiplying one ad.
- social sizes content cadence to what's actually sustainable for the team you have, not an aspirational schedule that collapses within a month.
- content-strategy decides what to write before writing starts, sequencing pieces so they compound into real authority rather than accumulating as disconnected posts.
- The right channel mix in this chapter depends entirely on whether you have money to test with or time to invest instead — there's no universal right answer.

EXERCISE

Exercise

Self-Reflection

Honestly assess your current marketing budget: is it large enough to generate a real learning signal from a paid test, or would it be better spent on something time-based instead?

Collaborate

Ask Claude to run the content-strategy skill to sequence your next six pieces of content around the real questions your customers ask, using your site-architecture work as a starting point.

Reflection

- What social or content cadence could you sustain for a full year without burning out, and is that the cadence you're currently attempting?
- If you ran one paid test today, what specific question would you want it to answer?

REFLECTION

- *Where have you been chasing a channel because it's visible, rather than because it fits your current resources?*
- *What's one piece of content you could write this week that only you, specifically, could write with real authority?*

WHAT'S NEXT

Filling the funnel is only the first half of reaching people well. Chapter 10 covers what happens after someone gives you their attention once: email, SMS, and the lifecycle systems that keep the relationship going.

CHAPTER TEN

Email, SMS & Lifecycle

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

- ✓ Build an automated lifecycle email flow with the emails skill that does more than announce features.
- ✓ Know when SMS earns its place in a lifecycle strategy, and when it's an intrusion.
- ✓ Write B2B cold outreach with cold-email that gets replies instead of getting deleted.
- ✓ Connect lifecycle messaging back to the activation action defined in Chapter 8.

Northstar Ops had one email: a receipt, sent automatically by the billing system, that nobody had ever looked at twice. Everything else — the emails skill pointed out almost immediately — was a missed opportunity to do the one thing lifecycle email is actually for: meeting a user at the specific moment they need a nudge, not announcing things at the company's convenience.

The welcome sequence the emails skill helped build wasn't built around introducing Northstar Ops's features. It was built around the activation action defined back in Chapter 8 — scheduling one real job on one real truck — with every email in the sequence designed to move a stalled user one step closer to that action if they hadn't reached it yet, and to go quiet if they had. A user who signed up and imported their trucks but hadn't scheduled a job within a day got a short, specific email: a screenshot of exactly what automatic assignment looks like, with an invitation to try it on one real job, not a generic "how's it going" check-in. A user who'd already scheduled three jobs got nothing from that sequence at all, because they didn't need it — they needed the next thing, which was an email two weeks later about the per-truck pricing tiers, timed to when a growing shop would plausibly be adding a truck.

Timed to the Moment, Not the Calendar

This is the discipline the emails skill enforces across every flow it builds, whether it's a welcome sequence, a re-engagement campaign for lapsed users, or a feature announcement: trigger on user behavior and lifecycle stage, not on a fixed calendar. A drip campaign that sends the same five emails to every new signup on the same schedule, regardless of what they've actually done, treats a user who's fully activated the same as one who's stuck on the first screen — which means it's wrong for at least one of them, usually both. Dana's rebuilt flow branched based on actual product usage, checked daily, which took more setup than a fixed sequence but meant every email a user received was relevant to where they actually were, not where the calendar assumed they'd be.

SMS: Earned, Not Assumed

Dana considered adding SMS on top of email and, working through the sms skill, mostly decided against it — which is itself the useful outcome of running the skill properly. SMS is a higher-intrusion channel than email; it interrupts in a way email doesn't, and it earns its place only for messages genuinely time-sensitive enough to justify that interruption. For Northstar Ops, that turned out to be exactly one use case: an alert when a job's assigned truck was running more than fifteen minutes behind, sent to the dispatcher, not a marketing message at all. Marcus's Ledgerline, selling a very different kind of urgency — month-end close deadlines that finance teams genuinely cared about being alerted to — found a real marketing use for SMS in reconciliation-deadline reminders, which worked precisely because the urgency was real and specific, not manufactured to justify using the channel.

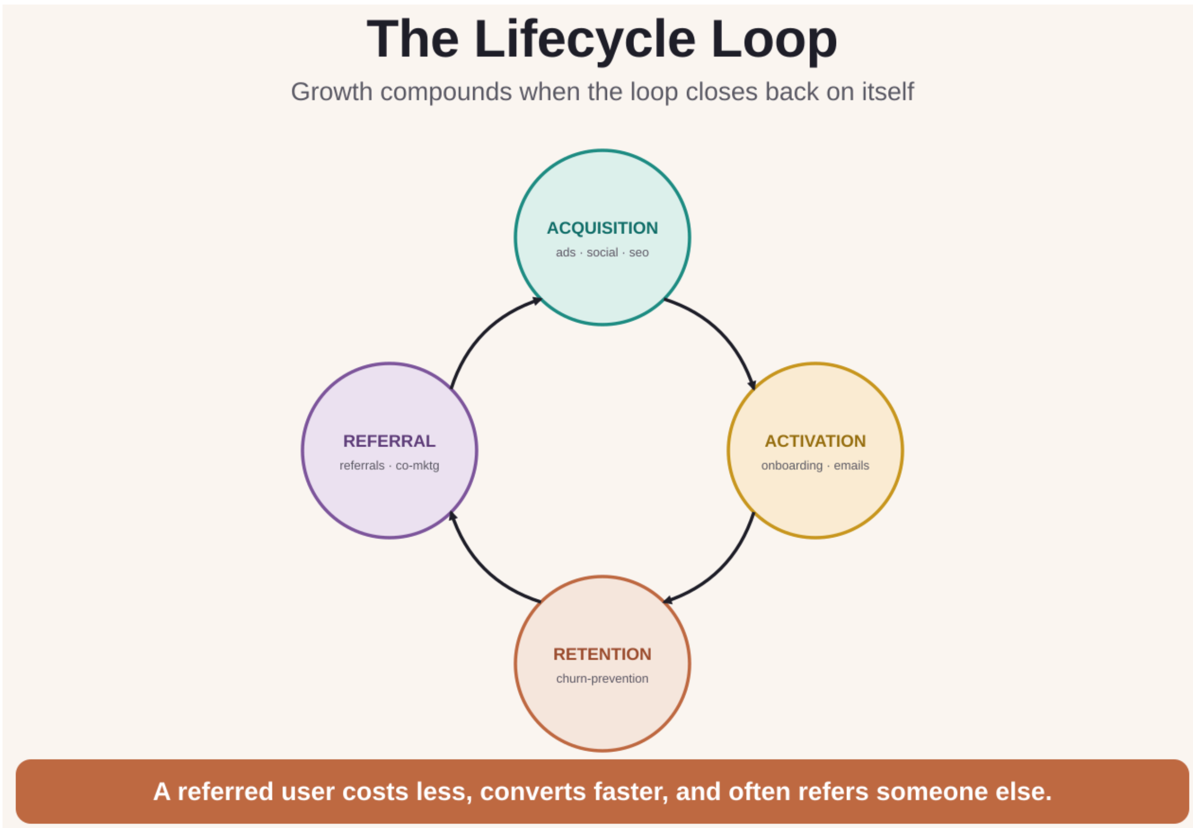

Figure 6. *Retention and referral — the two stages lifecycle messaging serves most directly, closing the loop back to acquisition.*

Cold Outreach as a Lifecycle Skill

Marcus's other major use of this part of the framework was upstream of any of Dana's lifecycle work: cold-email, for B2B outreach to finance leaders at mid-market fintechs who'd never heard of Ledgerline. The skill's core discipline runs directly counter to the instinct to write a compelling pitch — it optimizes for getting a reply, which is a narrower, more specific goal than being persuasive in the abstract. Ledgerline's highest-performing cold sequence wasn't the one that explained the product best; it was three short, specific sentences referencing a genuine, researched detail about the recipient's company (a recent funding round, a public statement about scaling finance operations) followed by one direct, low-commitment question — not “want to hop on a call” but “is reconciliation timing something your team's actively trying to fix this quarter, or is that further out?” The second version got answered. The first version, however polished, mostly got ignored.

THE METHOD

The Trigger, Not Calendar Method

1. Define what “stuck” looks like at each stage of your funnel — signed up but not activated, activated but not paid, paid but going quiet.
2. Build one short, specific email for each stuck state, addressing exactly what would move that specific user forward.
3. Suppress messages for users who've already moved past the state a message addresses — relevance requires knowing when to say nothing.
4. Before adding SMS, name the specific, time-sensitive use case that justifies the interruption. If you can't name one, don't add the channel.
5. For cold outreach, write for a reply, not a pitch — end with one specific, low-commitment question a recipient could answer in five words.

A drip campaign that ignores what a user has actually done is wrong for someone on every single send.

WATCH FOR

- Fixed-schedule email sequences that treat every user identically regardless of what they've actually done in the product.
- Adding SMS because it's available, not because a genuinely time-sensitive use case justifies the higher intrusion.
- Cold email optimized to sound impressive rather than to earn a reply — polish and reply rate are only loosely correlated.

KEY TAKEAWAYS

- Lifecycle email should trigger on user behavior and stage, not a fixed calendar — relevance requires knowing when to send nothing.
- The emails skill's flows should tie directly back to the activation action defined during onboarding, nudging stuck users specifically toward it.
- SMS earns its place only for genuinely time-sensitive messages; most marketing use cases don't clear that bar and shouldn't force the channel.
- cold-email optimizes for replies, not persuasiveness — short, specific, researched messages ending in a low-commitment question consistently outperform polished pitches.
- Every lifecycle message should be traceable to a specific user state — if you can't say who it's for and why now, it's probably not ready to send.

EXERCISE

Exercise

Self-Reflection

List the three most common "stuck" states in your funnel — places users pause without progressing. For each, write what you currently do, if anything, to help them past it.

Collaborate

Ask Claude to run the emails skill to build one triggered email for your most common stuck state, tied directly to your activation action from Chapter 8.

Reflection

– Is there a genuinely time-sensitive use case in your business that would justify SMS, or would it just be another interruption?

– If you sent cold outreach today, would the goal be to sound impressive or to get a reply — and does your current draft reflect that choice?

REFLECTION

- *Where has a fixed-schedule email or content habit outlived its usefulness because it's easier than building something triggered?*
- *What's the shortest, most specific message you could send that would genuinely help someone currently stuck in your funnel?*

WHAT'S NEXT

Lifecycle messaging keeps individual relationships healthy. Chapter 11 zooms out to the loops — referral, community, retention — that make growth compound across your whole customer base.

CHAPTER ELEVEN

Growth Loops & Retention

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Design a referral program with the referrals skill that fits how your specific customers already talk to each other.

✓ Build a free tool with free-tools that generates real leads instead of just goodwill.

✓ Reduce churn with churn-prevention by fixing causes, not just building better cancellation flows.

✓ Recognize community-marketing and co-marketing as leverage on relationships you already have, not new relationships to build from zero.

Twenty-five paying customers, the outcome Dana had named back in Chapter 3, arrived in week eleven of the ninety — and referrals had brought in more than half of them. The loop the framework had been building toward, one chapter at a time, had actually closed.

The referrals skill's contribution to that number wasn't a generic “refer a friend, get $50” program — it was the specific, community-aware version built back in Chapter 4, refined further here: a referral flow that made it easy for a Northstar customer to recommend the product inside the trade associations and contractor groups where they were already active, with the reward structured as recognition inside that specific community rather than a discount that meant nothing to someone else's bottom line. The skill's underlying judgment, generalizable well past Northstar's specific case, is that a referral program works best when it rides on communication that would have happened anyway — tradespeople talking to other tradespeople about tools that work — rather than trying to manufacture sharing behavior from nothing.

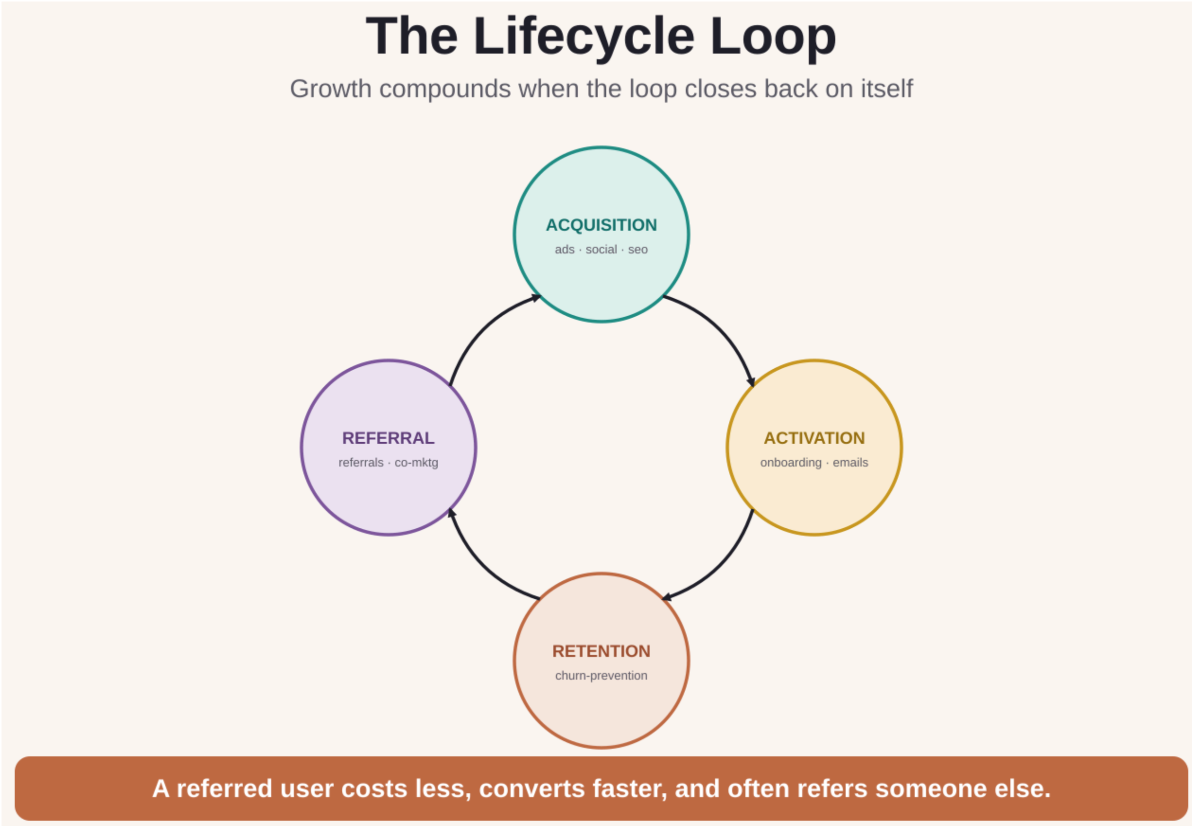

Figure 6. *The lifecycle loop, complete — referral feeds back into acquisition, and the whole system compounds.*

A Free Tool as a Lead Magnet

Separately, working through the free-tools skill, Dana built a small, genuinely useful calculator — how much a shop was likely losing per month to double-bookings and inefficient routing, based on truck count and average job value — and put it on its own page, unlocked, no email wall. The free-tools skill's guidance ran against Dana's first instinct to gate it behind a signup form: a truly useful free tool earns backlinks, gets shared inside exactly the trade communities the referral program was already reaching, and builds trust before ever asking for anything, which compounds into more qualified leads over time than a smaller number of gated form-fills would have. The calculator became, unexpectedly, one of Northstar's better-performing pages for organic search within two months, exactly the kind of SEO-and-lead-generation overlap this framework is built to notice and exploit.

Retention: Fixing Causes, Not Just Cancellation Screens

Growth loops only compound if the bucket isn't also leaking, and Northstar's early churn — two of the first thirteen customers canceling within two months — sent Dana to the churn-prevention skill. Its first move was diagnostic, not tactical: understanding why those two customers actually left before building anything to stop the next one. Both cancellations traced back to the same root cause — a confusing multi-truck setup flow that had, in effect, never been fixed since Northstar's earliest days — which meant the fix wasn't a save offer or a discount at the cancellation screen, it was going back and repairing the onboarding flow from Chapter 8 for the specific case of a shop with more than three trucks. The skill's guidance here is worth holding onto generally: a well-designed cancellation flow with a save offer treats a symptom, and can even mask a real problem long enough for it to quietly cost more customers before anyone traces it back to its actual cause.

Marcus's churn problem at Ledgerline was different in kind, not just scale: failed payments on annual renewals, not dissatisfaction. The churn-prevention skill's dunning and payment-recovery guidance — retrying failed cards on a specific cadence, emailing the actual billing contact rather than the original signup contact who'd often moved teams by renewal time — recovered a meaningful share of what had been quietly categorized as churn but was really just a broken payment-retry process. Not every churn problem is a product problem; some of it is closer to plumbing, and the skill's value is telling the two apart before proposing a fix aimed at the wrong one.

Community and Co-Marketing: Leverage on What You Already Have

The last two skills in this chapter are the ones easiest to underuse, because they don't feel like “marketing” in the traditional sense. community-marketing is about building and leveraging an actual community around the product — for Northstar, this ended up being a modest private group for the growing base of trade-association-connected customers, a space where they were already inclined to help each other, which Dana simply gave a home rather than trying to build engagement from a cold start. co-marketing, meanwhile, is about finding and running joint campaigns with complementary partners — for Dana, a co-branded guide with a payroll software company serving the same home-service trades, neither one a competitor, both reaching an overlapping but not identical audience, splitting the work of creating something genuinely useful to both customer bases.

THE METHOD

The Loop-Closing Method

1. Build referral mechanics around communication that would happen anyway, rather than trying to manufacture sharing from nothing.
2. Keep free tools genuinely free and useful — trust compounds into more qualified leads than a smaller number of gated form-fills.
3. Diagnose the actual cause of a cancellation before building anything to prevent the next one — a save offer treats a symptom, not a cause.
4. Separate product-driven churn from process-driven churn (failed payments, billing contact drift) — they need entirely different fixes.
5. Look for community and co-marketing opportunities inside relationships you already have before building new ones from scratch.

A referral program works best riding on conversations that would have happened anyway.

WATCH FOR

- Gating a free tool behind a signup form out of habit — it often costs more in reach and trust than it gains in leads.
- Treating every cancellation as a save-offer opportunity instead of asking what actually caused it.
- Skipping community and co-marketing because they feel less controllable than paid or email — they're often the highest-trust channel available to an early-stage company.

KEY TAKEAWAYS

- referrals works best when it rides on communication your customers would have anyway, structured around what your specific community values, not a generic discount.
- free-tools performs better ungated — the trust and reach a genuinely useful free tool earns usually outweighs a smaller number of gated leads.
- churn-prevention starts with diagnosing the actual cause of cancellations before building a fix; a save offer at the cancellation screen treats a symptom.
- Product-driven churn and process-driven churn (like failed payments) require entirely different fixes — telling them apart is the skill's first job.
- community-marketing and co-marketing are leverage on relationships and trust you already have, not new relationships you need to build from zero.

EXERCISE

Exercise

Self-Reflection

List your last five cancellations, if you have them, and write one honest sentence for each about the actual cause — not the reason given on the cancellation form, if those differ.

Collaborate

Ask Claude to run the churn-prevention skill against that list, looking for a shared root cause across multiple cancellations before proposing any fix.

Reflection

- What conversation is already happening between your customers that a referral program could simply make easier, rather than trying to create from nothing?
- Is there a genuinely useful free tool you could build from something you already know how to calculate or explain?

Stretch Goal

Identify one non-competing company serving an overlapping audience and sketch a single piece of co-marketing content you could realistically build together.

REFLECTION

- *Where have you been building growth mechanics from scratch that could instead ride on relationships or conversations that already exist?*
- *What would change if you treated every cancellation as a diagnostic question rather than a save-offer opportunity?*

WHAT'S NEXT

With acquisition, conversion, and retention all working together, Part VI closes the book with the moment that pulls it all into public view: launch day, earned media, and the handoff to sales.

PART VI

Launch & Systemize

Bring it into public view, hand it to sales, and make it run without you.

CHAPTER TWELVE

Launch Day, PR & the Sales Handoff

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

- ✓ Plan a launch with the launch skill that has a clear moment, not just a vague sense of momentum.
- ✓ Use public-relations to pitch journalists and newsjack relevant stories without a PR agency.
- ✓ Hand qualified leads to sales cleanly using revops and sales-enablement.
- ✓ Build a prospecting list and competitor understanding to support a growing sales motion.

Northstar Ops had been quietly live for four months by the time Dana asked Claude to help plan an actual launch — which felt backwards until the launch skill explained why it wasn't. A launch isn't the day a product goes live. It's a deliberately chosen moment to concentrate attention that would otherwise arrive gradually, and four months in, with real customers and a real case study, Dana had far more to concentrate attention around than she'd had on day one.

The launch skill's planning process resembled the marketing-plan skill's from Chapter 3 more than it resembled a checklist: what's the specific news, who's the specific audience for that news, and what's the specific action you want them to take in the narrow window when attention is highest. For Northstar Ops, the news wasn't “we exist” — it was “eleven home-service businesses cut their double-bookings to zero,” a claim with a number and a proof point behind it, aimed squarely at the trade-association audiences the referral program had already been building relationships inside. The plan built around that: a coordinated post across the private community from Chapter 11, a short case-study page timed to go live the same morning, and outreach to two trade-association newsletters Dana already had a relationship with, all converging on a single Tuesday rather than trickling out over a month.

Earning Coverage Without an Agency

The public-relations skill covers the same territory a PR agency would, scaled to what a founder can realistically do alone: pitching journalists directly, responding to reporter queries through services like HARO successor platforms and Qwoted, and newsjacking — injecting a company's specific point of view into a trending story rather than waiting for a reporter to come looking. Most teams that attempt PR at all lean on only one of these modes and skip the other three, usually reactive pitching, because it's the one that requires the least initiative and the most luck. Dana's actual coverage — a mention in a regional trade publication — came from the newsjacking mode: a widely discussed story about labor shortages in the skilled trades broke the same week, and Dana pitched a specific, data-backed angle Northstar was positioned to speak to (how much dispatcher time inefficient scheduling was costing already-short-staffed shops), rather than pitching Northstar Ops as a story in its own right, which is a much harder sell to a reporter with no reason yet to care.

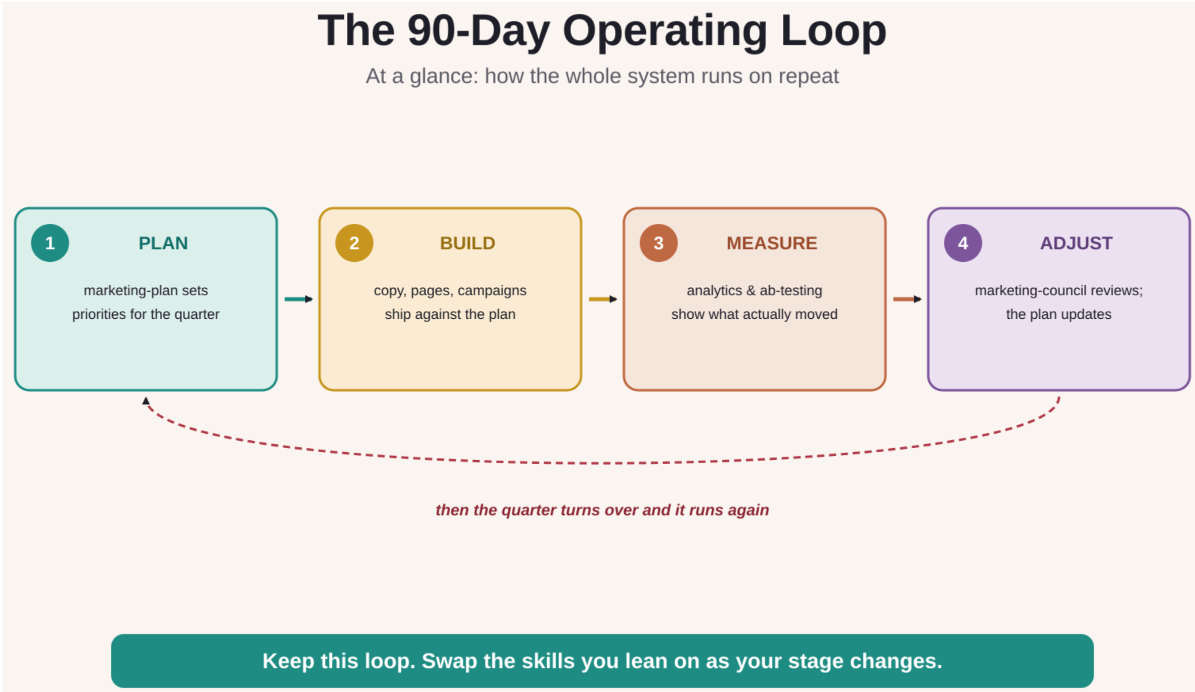

Figure 7. *The operating loop this chapter's launch work feeds into — plan, build, measure, adjust, and the sales handoff sits inside 'adjust.'*

Handing Leads to Sales Without Losing Them

As Northstar's inbound grew past what Dana could personally qualify and follow up with by hand, the revops skill became relevant for the first time — not because Northstar had a sales team yet, but because lead lifecycle management, scoring, and routing matter the moment there's more inbound than one person can track by memory. The skill helped define, concretely, what actually counted as a qualified lead (truck count above a threshold, a specific trade the product served well, a signal of active pain rather than idle browsing) and where each one should go, which for Northstar still meant Dana's own inbox, but organized and prioritized rather than an undifferentiated pile.

Marcus's Ledgerline, with an actual sales team by this point, used the same skill for a genuinely different problem: leads were reaching sales, but arriving without the context marketing had already gathered — which content they'd engaged with, what specific pain point had shown up in a demo request form, whether they'd already read the case study most relevant to their industry. Fixing that handoff was revops's job, and once it was fixed, sales-enablement picked up from there — building the one-pagers, the objection-handling documents, and the demo scripts that let Ledgerline's sales team have a sharper first conversation because marketing's context arrived with the lead instead of getting rebuilt from scratch on every call.

Underneath both of these sat two more specialized skills worth knowing exist even if you use them only occasionally: competitors, for building comparison and alternative pages that help prospects who are actively evaluating options (and who search for exactly that comparison) find you instead of only the incumbent; and prospecting, for building a qualified outbound list across SaaS, general B2B, or local SMB motions when inbound alone isn't generating enough volume to hit a growth target. Neither Dana nor Marcus needed both immediately, but both skills exist in the framework precisely because a growing company eventually does.

THE METHOD

The Concentration Method

1. Name the specific news for your launch — not “we exist,” a claim with a number or proof point behind it.
2. Identify the specific audience most likely to act on that news, and concentrate your outreach on them rather than broadcasting broadly.
3. Converge every launch element — content, outreach, community posts — on a single moment rather than trickling them out.
4. For PR, look for a trending story you can add a genuine, specific point of view to before trying to pitch your company as a story on its own.
5. As inbound grows, define what counts as a qualified lead explicitly, and make sure the context behind each lead travels with it to whoever follows up.

A launch isn't the day a product goes live. It's a chosen moment to concentrate attention that would otherwise arrive gradually.

WATCH FOR

- Launching on day one, before there's a real proof point or number to build the news around.
- Pitching your company as the story instead of finding the trending story you have a genuine, specific angle on.
- Letting inbound leads arrive at sales — or at your own inbox — without the context marketing already gathered about them.

KEY TAKEAWAYS

- A launch concentrates attention around a specific, proof-backed piece of news at a chosen moment — it doesn't have to coincide with a product's first day live.
- Effective PR for a founder usually means newsjacking a trending story with a genuine angle, which requires far less luck than pitching a company as a story on its own.
- revops defines what counts as a qualified lead and makes sure it's routed with context, whether “sales” is a full team or just a founder's organized inbox.

- sales-enablement lets a sales conversation start from the context marketing already gathered, instead of rebuilding it from scratch on every call.
- competitors and prospecting exist for when inbound alone isn't enough — comparison pages capture active evaluators, prospecting builds volume deliberately.

EXERCISE

Exercise

Self-Reflection

Name one specific, provable piece of news about your product right now — a number, a result, a milestone — that could anchor a real launch moment.

Collaborate

Ask Claude to run the launch skill to plan a concentrated launch moment around that news, and the public-relations skill to identify one current trending story you could pitch a genuine angle on.

Reflection

- What context do your leads currently arrive with, and what context is getting lost between marketing and wherever they go next?
- Is there a comparison or alternative page you're missing that active evaluators are searching for right now?

REFLECTION

- *Where have you been treating ongoing marketing activity as a substitute for ever choosing a concentrated moment to launch something?*
- *What would change about your sales conversations if every lead arrived with the context marketing already knows about them?*

WHAT'S NEXT

The final chapter pulls every skill in this book into one running system — the operating loop that keeps working long after the launch is over.

CHAPTER THIRTEEN

Running It as a System

BY THE END OF THIS CHAPTER YOU'LL BE ABLE TO

✓ Distinguish strategy work worth keeping in-house from execution worth delegating, using this framework's team-and-agency model.

✓ Recognize the pi-shaped marketer pattern and apply it to your own or your first hire's skill development.

✓ Set up a recurring marketing-loop so routine work runs on a schedule instead of depending on memory.

✓ Complete the book's second bookend artifact: a 90-day operating loop built from your own, now-mature product-marketing.md.

Twelve chapters after Dana first opened Claude and got a competent, generic homepage back, Northstar Ops had twenty-nine paying customers, a product-marketing.md file three times the length of its first draft, and a question that hadn't existed in week one: what happens when this all needs to keep running without Dana personally driving every skill by hand.

That question is what this closing chapter, and the team-and-agency-model framework underneath it, exists to answer. Its starting principle is a clean split: keep strategy in-house, and treat execution as something you can responsibly outsource — to a contractor, an agency, a future hire, or, for a long stretch of an early company's life, to Claude running the skills this book has walked through. Strategy is the judgment about which growth pattern fits your economics, which customer language belongs in your context file, which channel is worth testing this quarter; it depends on an intimacy with the business that's expensive to hand off and cheap to keep, because you already have it. Execution — writing the fifth variation of an ad, formatting schema markup across sixty blog posts, drafting a welcome email — is valuable but replicable, and is exactly where an agent, a contractor, or eventually an employee earns their keep.

Three Functions, One Marketer

The framework names three core functions any marketing effort eventually needs, regardless of team size: Growth, the demand engine — paid, SEO, lifecycle, everything in Parts III through V of this book; Product, the story engine — positioning, messaging, the context file from Chapter 2 that everything else depends on; and Content, the trust engine — the SEO, social, and authority-building work from Chapters 5, 6, and 9 that compounds slowly and pays off over the longest horizon of the three. A solo founder is, whether they've named it or not, running all three functions simultaneously, which is exhausting and also exactly why the framework's pi-shaped marketer pattern matters: rather than being deep in one function and shallow everywhere else (T-shaped, the traditional model for a specialist hire), the highest-leverage person in an early marketing effort is deep in two functions that compound together — for Dana, that turned out to be Product and Growth, with Content handled largely by Claude running the content-strategy and copywriting skills against a well-built context file, and picked up in more depth only once Northstar could justify a real hire.

This is also the frame for thinking about your first marketing hire, whenever that becomes affordable: not “find someone who does everything I've been doing,” but “find someone whose deep two functions complement mine,” so the combination covers all three at real depth rather than three functions covered at a shallow, generalist level by two people redundantly. Title progression in this framework follows the same logic — Manager, Lead, Director, VP, Chief — with each step reflecting a widening of which functions a person owns and coordinates, not simply years of tenure.

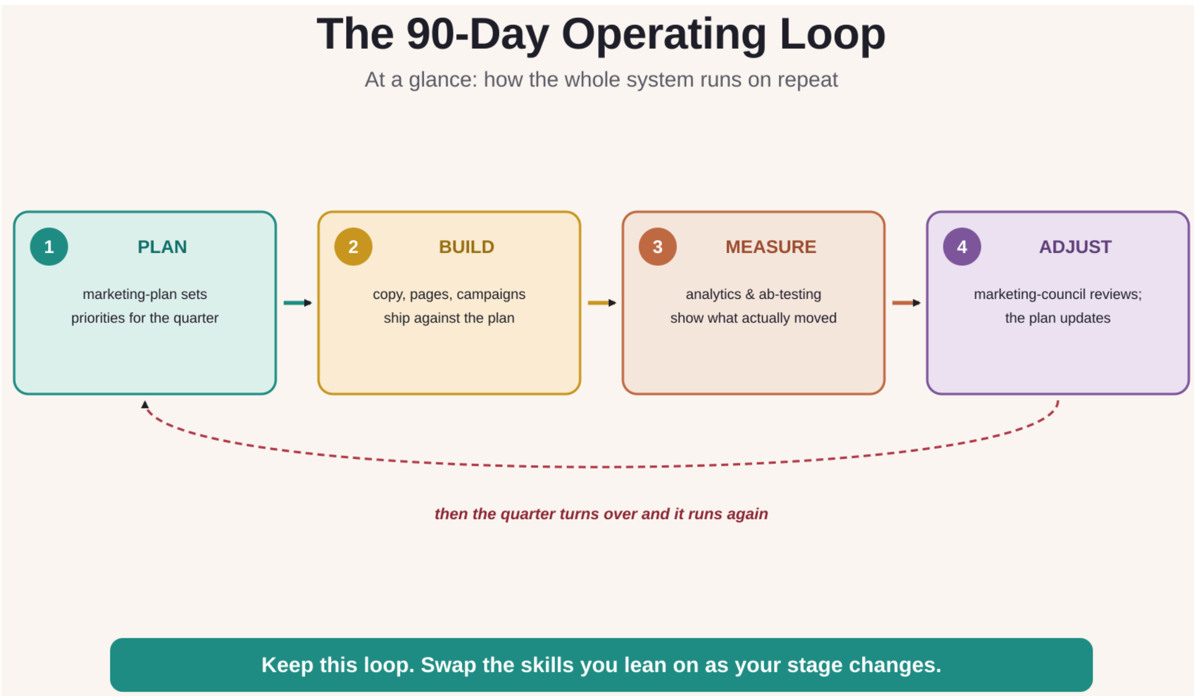

Figure 7. *The 90-day operating loop, at a glance — plan, build, measure, adjust, on repeat, with the skills from every part of this book slotting into each stage.*

The Loop That Runs Without You

The marketing-loops skill formalizes the last piece: a recurring, self-running workflow that an agent runs on a schedule, rather than a task someone has to remember to kick off. For Northstar Ops, the loop that mattered most was simple and unglamorous — a weekly check of signup-to-activation conversion, a monthly SEO audit re-run against the growing site, a quarterly full re-engagement of the marketing-plan skill to check whether the growth pattern from Chapter 3 still fit a company that now looked different than it had at eleven customers. None of this required Dana to remember to do it; it required setting the cadence once and letting Claude, and eventually a hire, keep it running. This is the final, quiet payoff of the context file from Chapter 2: a marketing-loops check that runs monthly is only useful if it's reading a context file that's still accurate, which is exactly why keeping that file current is the one piece of marketing work that never fully hands off, no matter how much of the rest of this system eventually does.

THE MARKETING COUNCIL, BRIEFLY

One more skill worth knowing exists: marketing-council, a simulated board of expert perspectives you can convene for a genuinely hard call — a positioning pivot, a pricing change, a launch decision with real stakes — when you want several specialist viewpoints in tension with each other before committing, rather than a single skill's single recommendation.

Keep strategy in-house. Let execution scale — to Claude, to a contractor, eventually to a team.

The Second Bookend Artifact

The exercise that closes this chapter, and this book, is the one it opened with in Chapter 2, grown up: instead of a first, thin product-marketing.md built in twenty minutes, build the 90-day operating loop that will keep running on top of it — what gets checked weekly, what gets re-run monthly, what gets fully revisited quarterly, and which skills from this book's thirteen chapters slot into each stage. Dana's finished version, reproduced in spirit in the At a Glance page that follows this chapter, isn't more sophisticated than what a much larger company would run; it's simply honest about what one person, working with Claude, can actually sustain — which was, in the end, the same lesson every chapter in this book kept returning to in a different form.

WATCH FOR

- Trying to stay deep in all three functions indefinitely. The pi-shaped pattern is a relief, not a limitation — pick your two, and let the skills in this book cover the third.
- Building a beautiful operating loop and then not keeping the context file underneath it current. The loop is only as good as what it's reading.
- Hiring your first marketer as a replica of yourself instead of a complement to your two deep functions.

KEY TAKEAWAYS

- Keep strategic judgment in-house; treat execution — much of which this book's skills handle directly — as safe to delegate, whether to Claude, a contractor, or a future hire.
- Every marketing effort needs three functions covered: Growth (demand), Product (story/positioning), and Content (trust) — a solo founder runs all three; a first hire should complement, not replicate.
- The pi-shaped marketer pattern — deep in two functions, not shallow in all three — is the realistic model for an early-stage team of one or two.

- marketing-loops turns routine checks (conversion, SEO, plan review) into a running schedule instead of a task depending on memory.
- Your product-marketing.md is the one piece of this system that never fully hands off — everything else scales only as well as that file stays current.

EXERCISE

Exercise

Self-Reflection

Of the three functions — Growth, Product, Content — name the two you're genuinely strongest in, and be honest about which one you've been neglecting.

Collaborate

Ask Claude to help you set up a marketing-loops schedule: what gets checked weekly, monthly, and quarterly, and which skill from this book handles each check.

Reflection

- If you hired one marketing person tomorrow, which function should they be deep in to complement, not duplicate, your own strengths?
- What would it take to make reviewing and updating your product-marketing.md a genuine, recurring habit rather than a one-time exercise from Chapter 2?

Stretch Goal

Write your own version of the At a Glance page that follows this chapter — one page, your company's real loop, the skills you actually plan to keep running.

REFLECTION

- *What does your business look like in a year if this loop runs quietly in the background, the way you've now set it up to?*
- *Which chapter in this book do you expect to return to most often — and what does that tell you about where your business's real leverage sits right now?*

WHAT'S NEXT

The At a Glance page ahead compresses all thirteen chapters onto one page. Keep it somewhere you'll actually see it.

AT A GLANCE

The Marketing Skills Framework

Pin this page. Whatever the specific problem in front of you, it maps to one of seven places to look — the same seven categories from Figure 1 back in Chapter 1.

KEY TAKEAWAYS

- SEO & Content — seo-audit, ai-seo, site-architecture, programmatic-seo, schema, content-strategy, aso: getting found, in search and in AI answers alike.
- CRO — cro, signup, onboarding, popups, paywalls: turning attention already earned into activated, paying customers.
- Content & Copy — copywriting, copy-editing, cold-email, emails, social, video, image, sms: everything with a reader on the other end.
- Paid & Measurement — ads, ad-creative, analytics, ab-testing: spending money and attention deliberately, and knowing what it bought.
- Growth & Retention — referrals, free-tools, churn-prevention, community-marketing, co-marketing: making growth compound instead of resetting every month.
- Sales & GTM — revops, sales-enablement, launch, pricing, competitors, prospecting, public-relations, offers, directory-submissions: the moment marketing meets revenue.
- Strategy — marketing-plan, marketing-ideas, marketing-psychology, marketing-council, marketing-loops, customer-research, product-marketing: the judgment layer everything else depends on.

REFERENCE

Glossary of Skill Names

A quick-reference index of every skill named in this book, in the order its category first appears. Full, current documentation for each lives in the framework's own repository.

product-marketing — the shared context file every other skill reads first; positioning, audience, and voice.

marketing-plan — a fractional-CMO-style engagement that sizes a plan to your real stage and budget.

marketing-ideas — a filtered library of proven tactics, matched to your stage and growth pattern.

marketing-psychology — mental models and behavioral science applied deliberately to pages and campaigns.

offers — the thing you actually sell — value framing, bonuses, guarantees — separate from the copy describing it.

pricing — packaging and monetization strategy, modeled against your real customer mix.

seo-audit — technical and on-page SEO diagnosis for existing pages.

site-architecture — page hierarchy, navigation, and URL structure, planned deliberately.

programmatic-seo — template-and-data-driven pages at scale, used only where each page earns its existence.

schema — structured data that helps both search engines and AI systems parse a page's claims.

ai-seo — optimizing content to be cited inside AI-generated answers (AEO/GEO/LLMO).

copywriting — writing marketing page copy that leads with the reader's situation.

copy-editing — refreshing and tightening copy that already exists.

cro — diagnosing and fixing the specific point where a page or form is losing visitors.

signup — reducing friction in registration and trial-activation flows.

onboarding — designing the first session around the action that predicts retention.

paywalls — placing upgrade prompts at a moment of felt value.

ads — deciding whether, and how, to run paid campaigns across major platforms.

ad-creative — generating and iterating ad creative in volume without losing the thread.

social — sustainable content and scheduling across social platforms.

content-strategy — sequencing what to write so content compounds into real authority.

emails — triggered, lifecycle-stage-aware automated email flows.

sms — high-intrusion messaging, used only where urgency genuinely justifies it.

cold-email — B2B outreach optimized for replies, not polish.

referrals — referral mechanics that ride on communication that would happen anyway.

free-tools — ungated tools that build trust and organic reach.

churn-prevention — diagnosing the real cause of cancellations before building a fix.

community-marketing — building and leveraging community around the product.

co-marketing — joint campaigns with non-competing, complementary partners.

launch — concentrating attention around a specific, proof-backed moment.

public-relations — earning coverage through direct pitching, reactive queries, and newsjacking.

revops — lead scoring, routing, and lifecycle management from marketing to sales.

sales-enablement — collateral — one-pagers, objection docs, demo scripts — for a sharper sales conversation.

competitors — comparison and alternative pages for active evaluators.

prospecting — building qualified outbound lists across SaaS, B2B, and local motions.

marketing-council — a simulated panel of expert perspectives for genuinely hard calls.

marketing-loops — recurring, self-running workflows that keep routine marketing work on schedule.

ACKNOWLEDGMENTS & INDEPENDENCE NOTE

Where This Framework Came From

The marketing skills framework this book teaches — the seven categories, the forty-plus individual skills, the product-marketing context pattern, and the budget and growth-pattern models referenced in Chapters 3 and 13 — is the work of Corey Haines, released as the open-source, MIT-licensed project Marketing Skills for AI Agents. It is built for Claude Code and compatible AI coding agents, and it is, as of this writing, freely available for any founder to install and use directly, without needing a book at all.

This book is an independent field guide, not an official product of that project or of Anthropic. Dana Whitfield, Marcus Odei, Northstar Ops, and Ledgerline are fictional — invented to make the framework's ideas concrete, not case studies of real companies. Every skill name, category, and framework concept referenced by name in these chapters corresponds to a real, currently maintained part of the open-source project; the specific product-marketing.md content, homepage copy, email sequences, and plans built inside this book's narrative are original illustrations, not reproductions of the project's own documentation.

If this book has done its job, the most useful next step isn't rereading it — it's installing the framework, running the product-marketing skill for real, and letting the rest follow the way it did for Dana, one honest conversation with Claude at a time.

✦