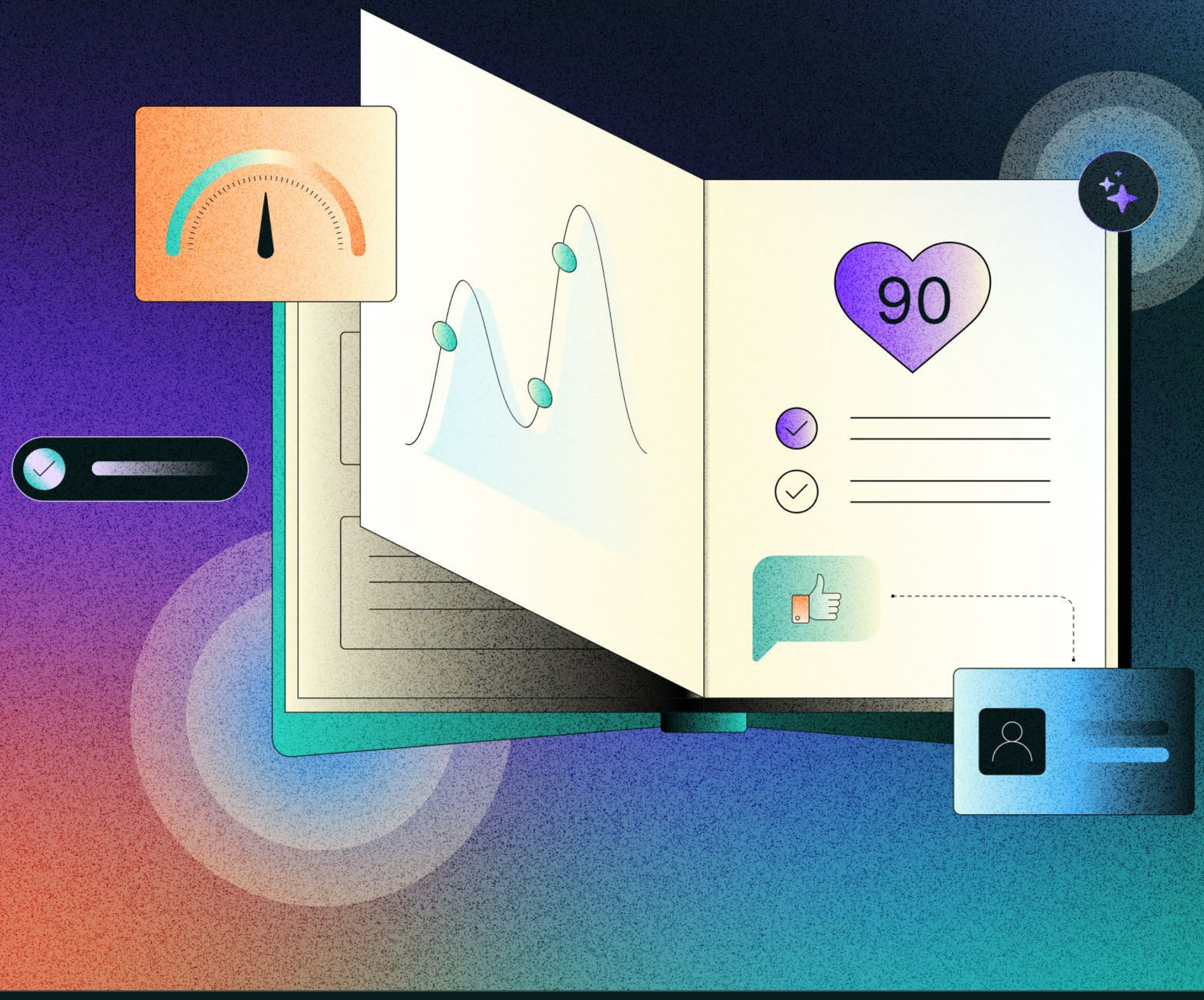

Vitally + ScaleUp

The Modern CS Scaling Playbook

A practical guide to hiring, structuring, and growing a Customer Success team that scales with AI

Contents

INTRODUCTION

Most CS teams don't build their first playbook on purpose

They build it by accident, one Tuesday at a time, usually right after something goes wrong. A founder is still running onboarding calls. A customer churns and nobody can say exactly why. A renewal gets missed because it was tracked in a spreadsheet nobody openeD that week.

This guide exists so you do not have to learn all of that the hard way.

It is built from real conversations Nina Wilkinson and Linnea Olson of ScaleUp CS ran with Vitally's community over the past year, covering the questions CS leaders ask most often: when to make that first hire, how to keep CSMs and AMs from stepping on each other, how to segment a book of business and build a leveling framework people trust, and how to bring a team through onboarding without losing the thread. We pulled the frameworks, the numbers, and the hard-won advice from those sessions and organized them into one guide you can work through in order, or jump into wherever your team happens to be.

A word on timing. Every section here assumes AI is already changing what a CSM's day looks like, not as a future trend but as something happening in your inbox right now. The teams getting the most out of this moment are not the ones with the fanciest AI stack. They are the ones who already know when to hire, how to structure the team, and what good looks like at each stage, so that when AI frees up time, there is a clear answer for what to do with it.

> That question, what a team does with the time it gets back, is where this guide ends up.

It is also where we think the next few years of Customer Success get interesting: not just faster individuals, but organizations that get smarter with every customer they work with. More on that at the end.

For now, let's start with the question nearly every founder asks us first.

PART 1

When to Hire Your First CSM

Hire on complexity, not volume. Then look for a generalist who can build the plane while flying it.

There is no magic number

Founders often want a threshold: hire at this ARR, or after this many customers. There isn't a clean one, and chasing one is part of why so many teams hire at the wrong time. Hire based on complexity, not volume. A company with 15 customers and a complicated onboarding can need a CSM before a company with sixty customers and a simple product does.

Two mistakes account for most of the pain we see.

- **Hiring too early.** Product-market fit is still shaky, onboarding isn't repeatable yet, and you don't fully know who your customers are. The new hire burns cash on overhead before there is a real system for them to run.
- **Hiring too late.** The more common mistake, and the more expensive one. Churn climbs while leadership stays heads-down on product, and nobody is left with the bandwidth to keep customers on track.

Five triggers worth checking now

Founders often want a threshold: hire at this ARR, or after this many customers. There isn't a clean one, and chasing one is part of why so many teams hire at the wrong time. Hire based on complexity, not volume. A company with 15 customers and a complicated onboarding can need a CSM before a company with sixty customers and a simple product does.

Two mistakes account for most of the pain we see.

- Your founder is spending 25% or more of their time on post-sale work: onboarding calls, check-ins, ad hoc support.
- You can't explain why customers churn. If a customer leaves and nobody on the team can say why, there's no CSM tracking risk or running exit conversations.
- Renewals and upsell conversations are getting missed because nobody owns a clear view of what's coming up.
- You're in the range of 15–25 high-touch customers, generally landing around $200K–$500K in ARR. Treat this as a rough planning input for a board conversation, not a rule.
- Monthly gross revenue churn is running above 5%. At that rate, the math to reach durable, IPO-scale ARR gets very hard.

Check off what's already true for your team.

- [ ] Basic onboarding steps are documented, even roughly
- [ ] You can describe what "successful in the first 30 days" looks like for a customer
- [ ] You have a stable, repeatable onboarding flow
- [ ] You have some way (even a spreadsheet) of tracking renewal dates
- [ ] Your founder is genuinely willing to hand off post-sale ownership

Checked three or four?
You're in a good spot to hire and set that person up to succeed.

The founding CSM profile: hire a generalist, not a specialist

Your first CSM needs to be comfortable in ambiguity and able to build the plane while flying it. Almost nothing they set up in year one will still be the process in year three, and that's fine. What matters is that they can get something workable in place now.

A useful starting profile: three to six years in a customer-facing role (CS, account management, or solutions consulting), someone seasoned enough to juggle competing priorities without needing to be managed closely.

MUST HAVE

Onboarding expertise

Gets a customer to the moment the product clicks, systematically, not by luck.

MUST HAVE

Commercial acumen

Comfortable with renewal timing, seat expansion, and pricing without being a sales specialist.

MUST HAVE

Building a scalable onboarding playbook

Creates structure from nothing, and uses AI to draft the first version of a framework.

Plan their time roughly 80/20: 80% on account management and onboarding, 20% on building the systems and playbooks that let the function scale later.

PART 2

Building a Scalable Onboarding Playbook

The churn window closes long before month eleven. Build a system that doesn't depend on who happens to be free.

The silent revenue multiplier

Most teams treat onboarding as a cost center to get through quickly. The data says otherwise.

75%

of customers who churn do it in the first 45 days, not at month 11.

63%

say onboarding determines whether they stay subscribed at all.

86%

stay loyal longer when onboarding and ongoing education are done well.

If a Salesforce integration or API connection stalls during that window, NRR is dead on arrival before a CSM ever gets a real chance at the account. That makes onboarding a go-to-market and revenue problem, not just a product problem. It's also the cheapest lever available: improving it doesn't require discounting, just better structure.

The chaos phase, and how to know you're in it

Nearly every team starts here. Sales closes a deal, the team celebrates, and then the founder jumps in because they know the customer's CEO, or whoever happens to have calendar availability that week, an intern, someone from marketing, ends up running the implementation. The customer is left piecing together how to use the product on their own, because nobody owns getting them there.

The fix isn't heroics from one talented person. It's a repeatable system that doesn't depend on who happens to be free that day.

Five onboarding mistakes that cost the most

- **The broken handoff.** Sales moves to the next deal (which is correct, that's their job) but the context they gathered, why the customer bought, what they're trying to solve, doesn't make it to the onboarding team. The customer has to repeat themselves and immediately feels it.

- **No shared definition of success.** The team tracks feature setup; the customer cares about the business outcome that feature enables. When those aren't explicitly connected, customers finish onboarding unsure whether it was worth their time.

- **Haphazard, inconsistent experience.** Two customers onboarding with two different team members get two different experiences. Standardize the core path first, then layer in personalization.

- **Flying blind.** Without metrics, a team can feel like onboarding is going well and have no actual way to know. Given how fast the churn window closes, that's an expensive blind spot.

- **Feature training instead of value.** Walking a customer through settings without explaining why it matters produces compliance, not adoption. They won't use the feature later and won't remember why they should have.

Map the journey around value, not a checklist

A checklist tells a customer what to click. It doesn't tell them why. Reframe each step around the benefit it unlocks ("connect your email so meeting notes sync automatically," not "go to settings and connect email"), and get the customer to a first moment of real value as early as possible. If it takes seven steps before they feel any benefit, most will disengage before they get there.

Two details worth planning for explicitly:

- **Complex implementations.** If an integration needs someone from the customer's data team and your onboarding contact is only in finance, that gap needs to be caught during handoff, not discovered mid-project.

- **Segment the journey.** Paid vs. free onboarding, and the depth of the journey itself, should flex by customer size and complexity. A self-serve customer doesn't need (or want) the same path as an enterprise account with a technical integration.

Fix the sales-to-onboarding handoff at the source

This is one of the most consistently painful handoffs in any post-sale org, and it's fixable with structure rather than better intentions. Build a buyer map into the sales process itself: who on the customer side needs to be looped in, what they bought and why, and anytimeline expectations, captured before the deal closes, not reconstructed afterward.

This is also one of the clearest places AI earns its keep rather than just being a feature checkbox. Instead of a rep filling out a form after the fact (which happens inconsistently at best), a system can read the sales call recording and the CRM data the moment a deal closes and draft the handoff brief itself: the customer's core problems, how the product solves them, likely roadblocks, and timeline. The AE reviews it in a few minutes to correct anything off and sends it on.

> The CSM's first question stops being "tell me about your business," a question the customer answered in detail two hours earlier for the AE.

Hire the onboarding architect, not a promoted support rep

A strong support rep with great CSAT scores is not automatically the right first onboarding hire. Support tends to be reactive (answer the ticket, resolve it, move on) and empathy-led, which is genuinely valuable, but it isn't the same skill as proactively driving a customer through a structured, milestone-based journey.

Look instead for a product educator with real project management rigor: someone comfortable pushing a customer toward outcomes (not just being liked by them), fluent across whatever systems they'll live in day to day (Asana, Jira, a CSP), and, ideally, technically comfortable enough to understand how data moves between systems even if they can't write code themselves.

As the function matures, look for the same data fluency described in Part One: someone who reads dashboards, spots patterns across customers ("three accounts are all getting stuck at this same milestone, why?"), and brings that back to product instead of treating each stuck customer as a one-off.

Use onboarding data to plan headcount, not just customer count

Speed-to-close and completion metrics are a more reliable capacity signal than customer count alone. As a rough planning guide: if one onboarding owner is running upward of 15 onboardings a quarter, that's a hiring signal. A team that's found a repeatable motion at 6 to 8 customers per owner per quarter usually has room to absorb one or two more before it needs to add headcount, especially once automation and AI are handling more of the admin load.

Keep onboarding and CS on the same go-to-market team. A poor onboarding experience becomes a CS problem the moment it's over, and CSMs need visibility into what customers struggled with early on. Structuring them separately, without a shared reporting line or regular sync, tends to recreate the same handoff problem this section opened with.

Build the onboarding engine: three pillars

- **Identify your sticky features.** Compare customers who renewed against customers who churned. What did the renewals consistently adopt that the churns never touched? Those features belong in the onboarding checklist and in the health score, not just in a nice-to-have tour.

- **Turn the path into playbooks and templates.** Get it out of one person's head and into a documented, repeatable flow so the experience doesn't depend on who's running it. Get it out of Slack too: it feels lightweight early on but becomes unsearchable and unreportable fast. A CSP or shared system that triggers automatically off the closed-won event is worth the setup cost.

- **Automate what data already knows.** If a customer has already connected an integration, that task shouldn't sit open on a checklist waiting for a human to check the box. Let data triggers close it. That time back is what lets a small team behave like a bigger one.

Managing the machine once it's built

- **Clear task ownership.** Every step in the journey needs one accountable owner, sales, the onboarding lead, or a CSM handling the kickoff, so nothing sits unclaimed when a customer starts to lag behind the journey map.
- **A real feedback loop to product.** When a team member notices customers struggling with the same feature or step, document it with enough specificity (what broke, how many customers, how much ARR is affected) that product can actually act on it. Onboarding is often the earliest, clearest signal a product team gets.

The 90-day onboarding build, in order

STEP 1

Document and standardize

Out of someone's head and onto paper, even roughly.

STEP 2

Implement and test

Run it for real. See where customers actually get stuck.

STEP 3

Automate and scale

Layer in the data triggers and playbooks that carry the load.

STEP 4

Assess and adapt

This step never really ends. Revisit as things change.

PART 3

CSMs vs. AMs: Building Swim Lanes Without Stepping on Toes

When two teams both think they own the renewal, the customer finds out first. Name the owner, then fix the comp.

This isn't a competition.
It's an org design problem.

One real example from the field: a customer got three separate meeting invites in one week, one from a CSM booking a QBR, one from an AM wanting to talk renewal, and a third nobody could explain. That customer was a C-suite decision maker, and the email that came back asked, plainly, to have it sorted out.

The instinct is to blame the people involved. It's rarely that. It's usually a structural gap: nobody defined who owns what, so both teams reasonably assumed they should step in.

Two roles, two different jobs

	CSM · THE HEALTH GUARDIAN	AM · THE GROWTH CATALYST
Mandate	Defends what's already been won	Grows what's already working
Day to day	Watches usage data, health signals, renewal risk	Maps stakeholders, tracks white space, negotiates terms
Owns	Adoption and value realization	Closing what the CSM surfaces
Scorecard	**GRR**	**NRR**

They're different roles feeding the same revenue engine, not competing ones.

Use a RACI to make it concrete

A RACI matrix names who is Responsible (doing the work), Accountable (owns the outcome, good or bad), Consulted (weighs in before a decision is made), and Informed (learns about it after). The distinction between Consulted and Informed is where most teams get sloppy, and it's usually the source of friction: Consulted means your input can change the outcome. Informed means you find out once it's already decided.

ACTIVITY	CSM	AM
Onboarding, EBRs, health monitoring	**Responsible**	Informed
Commercial negotiation	Consulted	**Responsible & Accountable**
Expansion opportunity (CSQL)	**Responsible (surfaces it)**	**Accountable (closes it)**

One thing worth naming directly: Responsible and Accountable can be the same person, but two people should rarely both be Accountable for the same number. Shared accountability tends to collapse into no accountability.

A handoff timeline, tied to milestones, not dates

01

Onboarding handoff

The AE brings the CSM and AM in together (even a short Slack thread works) so everyone hears the same story about who the customer is and why they bought.

02

Value realization

The CSM leads. The AM stays in the background unless a CSQL (a customer success qualified lead, essentially an expansion opportunity the CSM has validated) comes up.

03

Renewal period

The AM steps forward to lead commercial conversations. The CSM is consulted, not sidelined; they carry the account history and relationship context the AM needs.

Then it loops. The CSM picks ownership back up for the next adoption and value- realization cycle.

The CSQL playbook: make the incentives match the goal

A pattern we see often: CSMs generate great expansion leads, hand them to AMs, and then nothing happens, because the AM's comp is tied to new logo revenue, not existing-account expansion. The fix isn't a policy memo. It's comp.

- Comp CSMs on GRR, logo retention, and CSQL volume, ideally as a spiff (a set bonus per qualified lead) rather than a hard quota. Hard quotas tend to drag down lead quality.
- Comp AMs partly on CSQL deals closed, not only on dollar amount. Otherwise one large expansion can mask a dozen smaller opportunities that never got worked.
- Set a dollar threshold under which the CSM can close small upsells directly (a couple ofextra seats shouldn't trigger a full sales process). Above that threshold, route to the AM.

How the split evolves with ARR

STAGE	STRUCTURE
Under ~$1M ARR	One hybrid CSM handles both health and growth. There isn't headcount to split it yet.
~$3M ARR	An AM function starts to emerge, taking strategic commercial focus off the CSM.
Growth stage	Dedicated, specialized roles, GRR and NRR tracked separately, formal RACI in place.
$10M+ ARR	Fully specialized teams with defined automations and programs supporting each role.

As AI absorbs more admin work, the skill CSMs increasingly need on top of the fundamentals is commercial fluency: reading usage data well enough to tell the ROI story before an AM ever enters the room, and staying comfortable (not panicked) when a customer asks about pricing.

If you run a PLG motion

The comp and ownership model above assumes sales-assisted expansion. If customers can self-serve upgrades in-product, treat it as a separate lane: cap what's available for self-serve (a small number of seats, or a credit ceiling) and route anything larger through a human conversation. Self-serve upsells generally shouldn't count toward a CSM's CSQL comp, since they didn't create the lead. Track the volume anyway; it's a useful signal for account health and NRR even without a direct payout attached.

TRY THIS Draw your swim lanes

Pull your CS and AM leads into one working session and fill in a RACI for these five moments: onboarding handoff, EBRs, commercial negotiation, expansion opportunities, renewal conversations. For each row, agree on one Responsible and one Accountable owner before you leave the room. If you can't agree, that's the friction point to fix first.

PART 4

Segmentation, Leveling, and Career Paths

Segment the book first. Then build a leveling framework people can see themselves in.

Why this matters more than it looks like it should

Without a clear framework, promotion cycles start rewarding tenure over skill, and nothing kills a good performer's motivation faster than watching that happen to a peer. Book assignments start ignoring account complexity. And your best individual contributors, the ones who could become future team leads, leave because they can't see a path forward.

Start with segmentation, not leveling

Segmentation comes first because it's the foundation everything else sits on. Three inputs are usually enough: ARR, seats or users, and product footprint (how many product lines or integrations a customer touches). You don't need all three on day one. Start with ARR, add the others as your data supports it.

TIER	ROUGH ARR	MODEL
SMB	Under $20K	Digital / tech touch, higher account load
Mid-market	Mid-range	Blended touch, moderate load
Enterprise	Higher	High touch, lower load per CSM
Strategic	Top accounts	White-glove, lowest load per CSM

Treat this as a guideline, not a law. Build in an override process for accounts that don't fit (a large, low-maintenance customer, or a small, high-touch one), and date-stamp any override so it gets revisited rather than sitting there permanently.

Once the basics are working, consider adding a fourth axis: region, timezone, or language. Staffing a book so customers get help during their own working hours, and so a team member is actually equipped to communicate in the customer's preferred language or style, can matter as much as ARR for both satisfaction and retention. Treat it as an optional layer, not a replacement for the three inputs above.

One diagnostic worth running if a specific tier is churning more than the others (mid-market is a common offender): look at what's actually happening in that segment's touch cadence before assuming the segment itself is the problem. High churn in one tier is often a sign the team is being reactive instead of proactive there, not that the segment boundaries are wrong.

A second, complementary layer is journey stage (onboarding, adoption, renewal, nurture). It's less useful for assigning books and more useful for understanding what kind of work a CSM is doing and where the coaching or automation needs to focus.

The leveling framework: IC1 through IC6

A useful starting band structure runs from IC1 to IC6 (expand it further if your org needs more granularity).

- **IC1–IC3: mastering the fundamentals.** Owning your own portfolio, running renewal conversations, building product and domain knowledge.
- **IC4 and beyond: execution gives way to influence.** Less about doing the tactical work well (that's assumed by now) and more about shaping how the team or function operates.

The jump from IC3 to IC4 tends to be a genuinely different job, not more of the same job at a higher level.

TIER RANGE	STRUCTURE
IC1–IC3	Self and peers: own portfolio, informally share knowledge
IC4	Team: shape norms, set the example ("this is how I run a QBR")
IC5	Function: influence how CS operates beyond your own segment
IC6	Company: shape direction for other departments, e.g. a standing seat at the product roadmap table

Developing vs. proficient: a half-step, not a leap

Every promotion is an entry point, not a finish line. When someone moves up a tier, they start over as "developing" in that tier and grow into "proficient" before they're ready for the next one. That gives you a built-in half-step: rather than jumping someone from a full IC3 to a full IC4, you can promote a developing IC3 into a proficient IC3 first. It moves faster than a full-tier jump and still feels like real progress to the person receiving it.

Rough time-in-seat guidance: 6 to 12 months minimum per half-step at IC1–IC2, stretching to 3 to 5 years per half-step by IC4–IC6. Time-in-seat is a floor, not a target. It has to be balanced against demonstrated proficiency and whether the business actually has room and budget for the promotion.

> Pay inequity between people at the same demonstrated level is one of the fastest ways to lose trust across a team.

Evaluating readiness without recency bias

Build a monthly performance view that separates the data (GRR, CSQL activity, adoption metrics, percentage of book engaged) from a manager's qualitative read, then combine both at review time. That keeps the evaluation from being dominated by whatever happened in the last 30 days, which is the easiest bias to fall into. Add a self-evaluation against the competency matrix, evidence like call recordings or project artifacts, and for IC4+ roles, input from cross-functional partners in product, sales, and support.

Career paths that aren't just "become a manager"

Not every strong CSM should manage people, and that's not a knock on them. It's a different skill set. Build real alternatives.

Management track

For people who genuinely want to develop others.

IC / principal track

Deeper on strategic accounts and company-wide frameworks, without people management.

Lateral moves

Into AM, product, enablement, or CS ops. Not consolation prizes.

A CSM who moves into product often brings a customer lens that changes how the team scopes problems. Tie compensation to distinct bands within each tier (developing vs. proficient), not to individual negotiation.

PART 5

The Capacity Question: What AI Actually Gives Back

AI helps with capacity, not discipline. What fills the space it opens up is a leadership decision.

Capacity, not discipline

A CS leader running a team of ten or more CSMs should be reviewing calls every week, coaching on objection handling, catching where customer sentiment is turning. Most already know that. What they don't have is the bandwidth. Reviewing two or three calls a week at double speed, which strips out tone and nuance in the process, isn't a discipline problem. It's a capacity problem, and it's exactly the kind of problem AI is actually good at solving.

That framing matters because it sets expectations correctly. AI doesn't make a leader a better coach. It gives an already-disciplined leader the bandwidth to do the coaching they already knew they should be doing.

The numbers so far

5–10 hrs

reclaimed per CSM per week, depending on team structure. Close to a quarter of an additional CSM.

50/50 → 20/80

the admin-to-strategic split, as AI absorbs the paperwork around the conversation.

~30%

of weekly time reclaimed by Triptease's CS team after implementing anAI copilot.

Industry estimates put CSM bandwidth up 25–50% by the end of this year as AI tools mature. None of that is a replacement story. It's a capacity story, and capacity by itself isn't the interesting part.

A concrete way to put that capacity to work: coaching

Call review is one of the clearest examples of a discipline every CS leader already knows they should maintain and rarely has the hours for. One workable pattern: score every call automatically against a fixed rubric, then get a short digest instead of trying to sit through recordings. A four-competency rubric, scored on a three-point scale, covers most of what actually matters on a customer call.

Discovery and business intelligence

Did the CSM surface real business context, not just talk through the product?

Communication and outcomes

Was value quantified and tied back to whatthe customer is trying to achieve?

Risk identification and mitigation

Was risk surfaced and documented before it needed to be escalated?

Stakeholder engagement and depth

Is the relationship multithreaded, or does it live with one day-to-day contact?

Run every call through that rubric automatically and route the output into a short weekly digest, a Monday-morning summary of what's worth celebrating and what's worth coaching, and the review time drops from hours to about five minutes. The value isn't the automation itself. It's that a leader stops guessing which calls are worth their limited attention.

The same underlying pattern, catching a signal early and routing it to the person who can act on it, applies before onboarding even starts. Risk signals often surface during the sales cycle itself (a mentioned budget constraint, a competing internal priority) and get lost the moment the deal closes. Pulling those quotes out automatically and surfacing them to the CSM means the first customer conversation can address the risk directly instead of discovering it three months later at renewal.

The real question: what fills the space

Removing admin work doesn't automatically make a team more strategic. It just makes room. What a CSM does with five extra hours a week is a leadership decision, not something that happens on its own. Left undirected, reclaimed time tends to get absorbed by more accounts per CSM, which is a reasonable business choice but not a particularly ambitious one.

> AI doesn't replace the human on the call. It decides which calls, which accounts, and which risks deserve that human's attention first.

The more interesting use of that time is deeper account strategy: catching the risk nobody flagged, spotting the expansion opportunity a rules-based system would have missed, building the relationship that turns a renewal into a reference. That's judgment work, and it's exactly the work that was hardest to get to when a CSM's day was full of meeting prep, note-taking, and status updates.

A practical filter for tooling

Before buying anything, document your process manually first. The most common mistake we see is over-tooling too early: a team buys a $50K platform, nobody adopts it, and it becomes tech-graveyard software nobody wants to talk about in the budget review. Build the manual muscle, then automate what you've proven out. When you are ready to evaluate a tool, look for four things.

- It centralizes your actual data (CRM, product usage, support, sentiment) rather than adding another screen to check.
- It has a flexible, native health-scoring engine, not a rigid one you have to bend your business around.
- It automates the admin slog specifically, meeting notes, data entry, so the CSM's time goes to the conversation, not the paperwork around it.
- Your CSMs were part of evaluating it. Adoption follows involvement; a tool nobody asked for tends to sit unused.

RED FLAGS

Enterprise-only pricing that assumes you need a dedicated admin team to run it, and any implementation timeline stretching past 90 days. Early-stage teams don't have the runway for that.

Where this is heading

Here's the pattern underneath everything in this guide. Yesterday, CS teams mostly created and managed customer data by hand. Today, AI is starting to interpret that data for individual CSMs, drafting the follow-up, flagging the risk, summarizing the call. Tomorrow, the more interesting shift isn't that any one person gets faster. It's that what one CSM learns about a customer stops staying with that one CSM.

Think about what already happens informally on a good team: a CSM figures out that customers who adopt a second product line renew at a meaningfully higher rate, and that insight quietly reshapes how the whole team runs quarterly goals. Right now, that kind of discovery spreads by word of mouth, a Slack message, a team meeting, if it spreads at all. The next stage of Customer Success is making that spread automatic: every customer interaction becomes something the whole organization can learn from, not just the individual sitting closest to it.

That's the direction we're building toward at Vitally, and it's a bigger conversation than this guide. If the frameworks here get your team's foundation solid, that's exactly the position you want to be in when the next shift arrives.

IN CLOSING

Where this leaves you

Five questions, one team.

01 Do you know your hiring triggers, or are you still guessing at a magic number?

02 Could a new hire run your onboarding process from documentation alone, or does it still live in one person's head?

03 Do your CSMs and AMs have a RACI they'd both point to the same way?

04 Would your team recognize their own leveling framework if you handed it to them cold?

05 Do you have a real answer for what your team does with the hours AI gives back?

None of these require a big rebuild. Pick the one section above that maps to your most immediate pain, run the exercise, and revisit the rest as you grow into them. Onboarding, segmentation, and leveling in particular are meant to be revisited, not set once and left alone.

ABOUT

Vitally

Vitally is the AI-powered workspace for Customer Success teams. CSMs use Vitally for the clarity, automation, and AI-driven insights needed to move faster, stay aligned, and drive customer outcomes at scale.

Vitally

Request a demo

Scaleup

ScaleUp CS helps B2B SaaS companies build and scale post-sale organizations, from the first CSM hire through segmentation, leveling, and comp design.

ScaleUp

Work with ScaleUp CS