velaris

How to Build an AI-Ready CS Function: Fixing the Foundations

Table of contents

Introduction: Before you automate, make sure it works

A few years ago, you could probably get away with duct-taping your CS workflows together.

Your onboarding checklist may or may not have been in someone's Notion doc. Your health score looked awfully similar to a spreadsheet with made-up weights. Your team was scrambling to run QBRs off a last-minute deck assembled from random screenshots. And nobody really questioned it, since everyone was figuring it out as they went.

Then came AI.

Suddenly, you're being asked to scale. You're expected to move faster, be more predictive, eliminate manual work, and prove you're adding value. And of course, on top of all that, you're not allowed to lose that personal touch that made CS matter in the first place. The pressure's real. And honestly, you probably don't need more AI tools right now.

What you *do* need is a stronger foundation. Something AI can actually build on. Plugging automation into a broken process just makes the mess move faster. It's still a mess though.

You can't automate a health score if the inputs are unreliable.
You can't trigger alerts on milestones if you don't have a clear onboarding journey.
You can't personalize engagement if your segments are vague or outdated.

That's what this workbook is about.

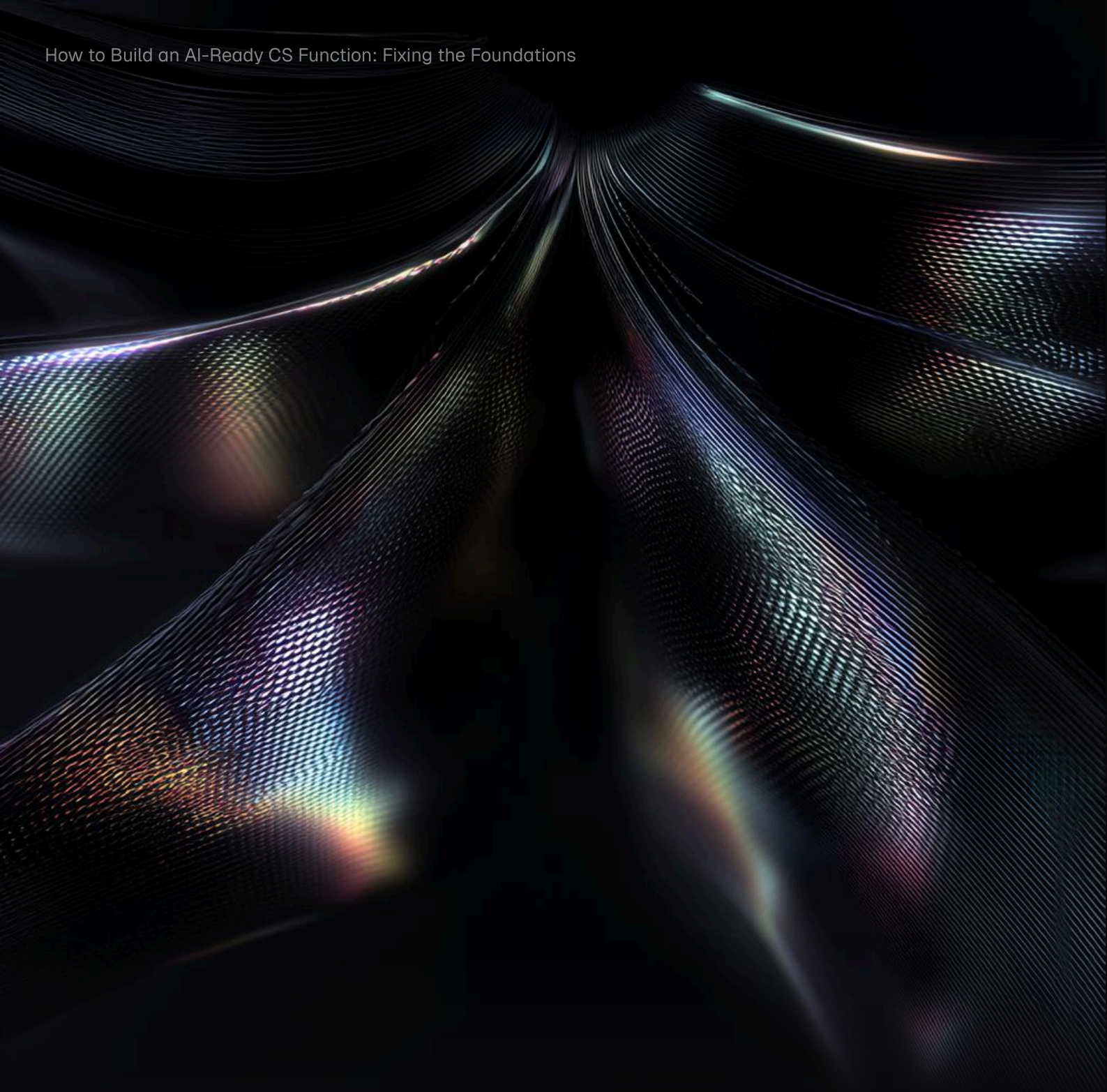
We won't give you long essays or theory. This is hands-on and practical: checklists, templates, and prompts that you can apply today.

We're going to help you:

- → Clean up your onboarding flows
- → Rebuild your data sources
- → Rethink segmentation and coverage
- → Set the groundwork for scalable, AI-enhanced workflows

You don't have to overhaul everything. But you do have to start somewhere.

AI might be the first wave of unparalleled change, but it won't be the last. Let's get your team ready for all of it.



CHAPTER 1

Audit your CS operating model

What do we mean by “operating model”?

Your operating model is the system that underpins how CS is run. It includes:

- → Customer segmentation and engagement models
- → Key processes like onboarding, renewals, and escalations
- → Team roles and ownership
- → Tooling and documentation
- → Visibility across the customer lifecycle

If these are inconsistent or not documented, AI will only accelerate the pain. So let’s clean it up first.

Self-Audit Checklist: What’s broken and what’s working

Core structure

- [ ] Do we have clearly defined customer segments (e.g. SMB, Mid-Market, Enterprise)?
- [ ] Is there a documented engagement model for each segment?
- [ ] Are roles and responsibilities clear across Sales, CS, and Support?

Repeatable workflows

- [ ] Is onboarding standardized?
- [ ] Do we have QBR/EBR templates, cadences, and ownership defined?
- [ ] Are renewals and expansions treated as repeatable motions?
- [ ] Are escalations and red accounts handled systematically?

Knowledge and training

- [ ] Can new team members easily follow our workflows?
- [ ] Do we have internal enablement docs, videos, or playbooks?
- [ ] Are success criteria clearly defined for each customer lifecycle stage?

Systems and visibility

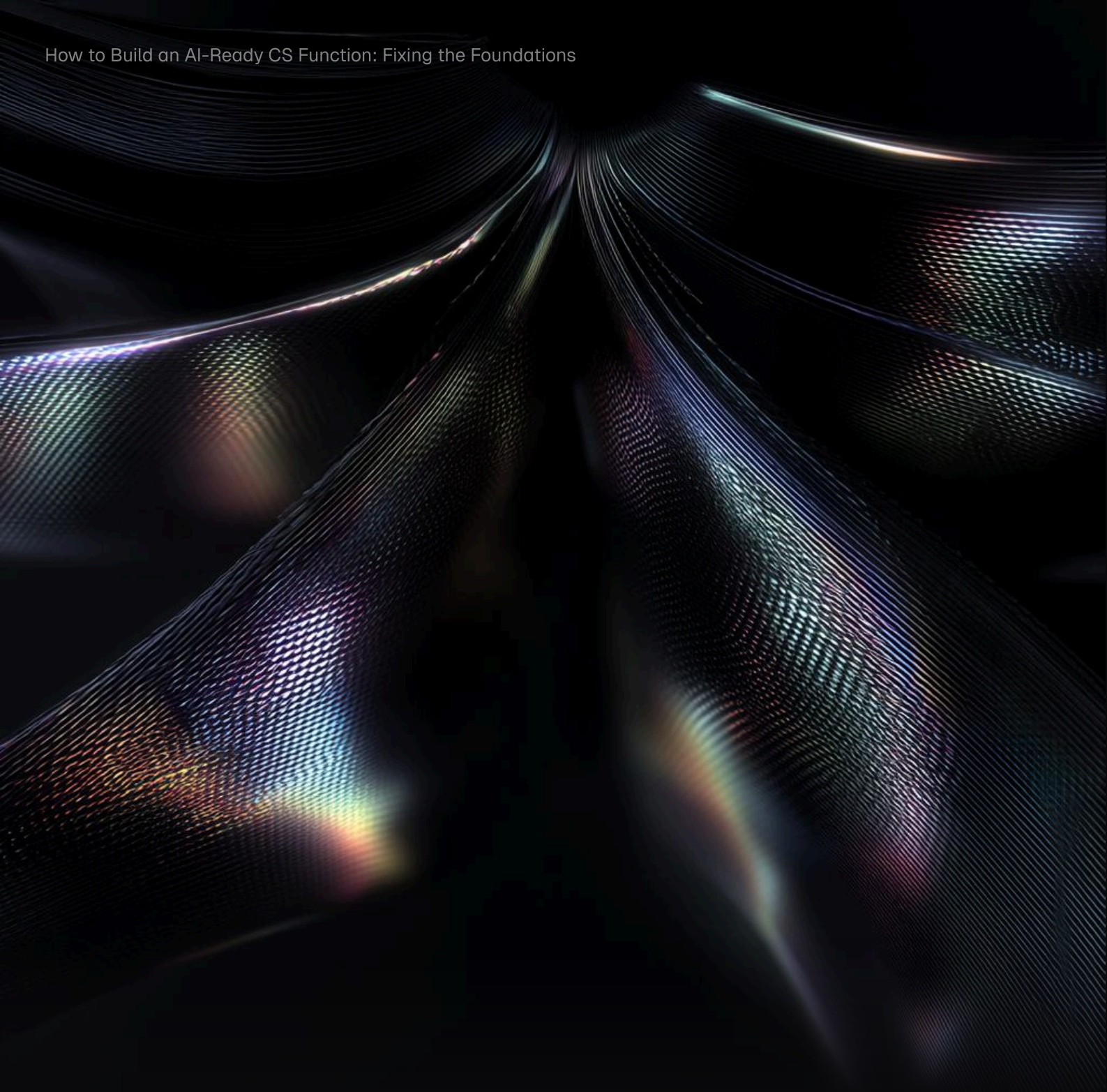
- [ ] Is the full customer journey visible in one place (e.g. CRM or CSP)?
- [ ] Do we track customer activity and lifecycle stages automatically?
- [ ] Is our customer data clean, trusted, and integrated across tools?

Reflection prompt

- [ ] What are the top 3 inconsistencies in how our team operates today?
- [ ] Where do CSMs spend the most time explaining internal processes instead of helping customers?
- [ ] What steps could we take this quarter to simplify or standardize our model?

Once you've audited your CS operating model and identified what needs fixing, the next step is to re-evaluate how you group and engage your customers.

In Chapter 2, we'll work on defining AI-ready segments, and making sure they actually drive action.



CHAPTER 2

Clean up your segmentation and journey logic

Worksheet: Define your AI-ready segments

Segmentation is the foundation for everything: playbooks, automations, health scores, even task prioritization. But many CS teams use inconsistent or outdated logic that's often based on what was easiest to implement. It may not actually reflect customer needs or business value.

Before you introduce AI into the mix, your segments need to make sense. That means thinking beyond just ARR or headcount. A good segmentation model reflects the potential of the account, not just its size.

This chapter will help you define meaningful segments and tie them to real outcomes.

Self-Audit Checklist: What's broken and what's working

Audit your current segmentation

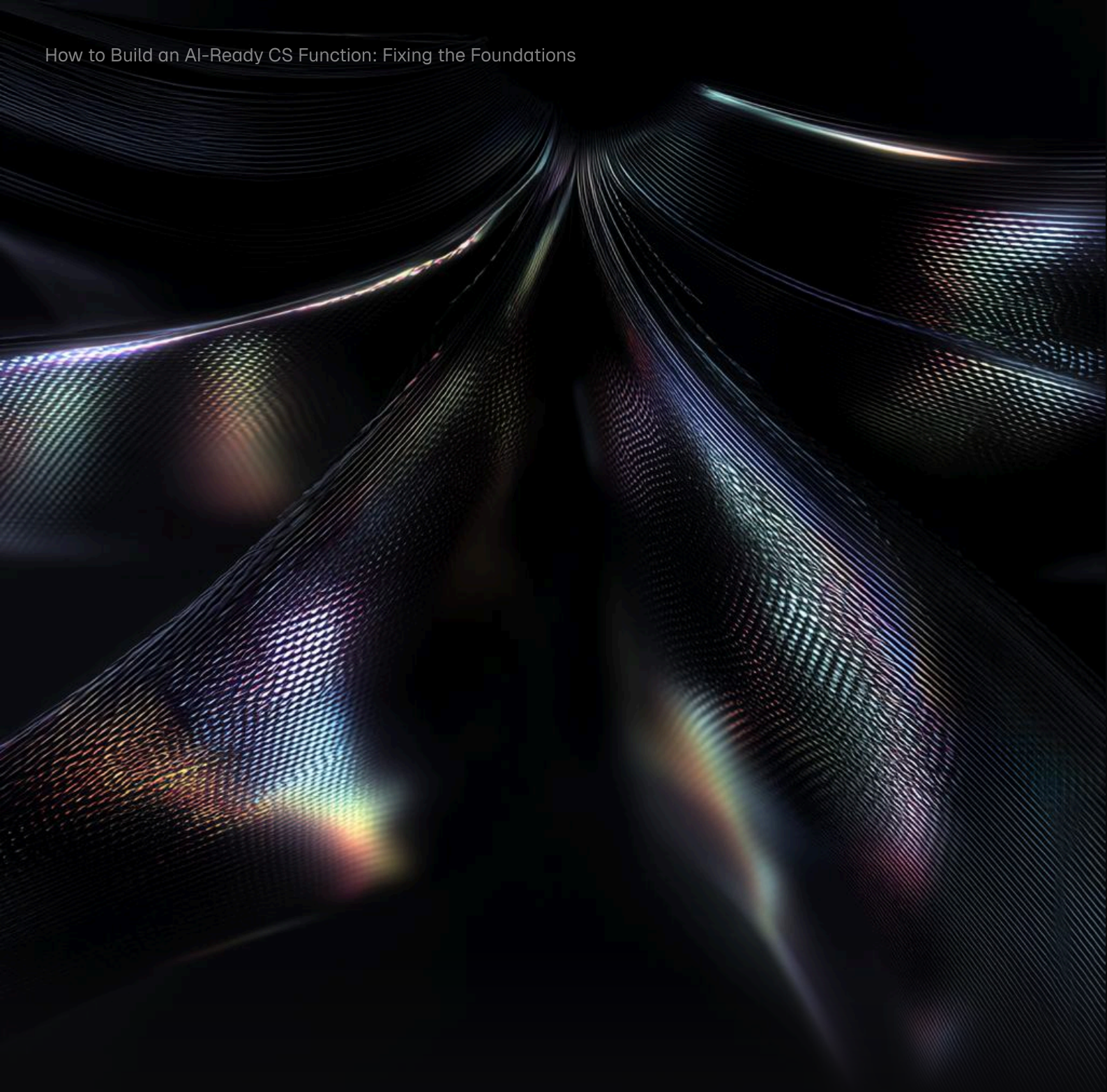
- [ ] What is our current segmentation logic (ARR, industry, size, region, etc.)?
- [ ] Are these segments still useful for driving outcomes, or just legacy definitions?
- [ ] Are they used consistently across CS, Sales, and RevOps?
- [ ] Are playbooks and workflows tied to these segments?

Define future-ready segments

- [ ] Which signals best indicate growth potential? (e.g. ICP fit, product usage, expansion intent)
- [ ] Which accounts need a high-touch experience vs. scaled engagement?
- [ ] How should we group accounts based on their needs, not just their size?
- [ ] What internal factors (e.g. team capacity, expertise) should influence our model?

Make segmentation actionable

- [ ] Is our segmentation logic documented and visible to everyone who needs it?
- [ ] Do we have workflows or automations triggered by segment?
- [ ] Can we personalize playbooks and comms based on segment?
- [ ] Is segmentation being used in forecasting, planning, and success measurement?



CHAPTER 3

Fix your data layer before you predict anything

If AI is going to tell you who's at risk, who's ready for expansion, or what play to run next, it needs to trust your data. But unfortunately, most CS data isn't AI-ready.

You don't need to fix everything overnight but you do need to know what's reliable, what's broken, and what needs cleaning before you layer AI on top.

Use this audit to assess your current data layer:

Assess your health score inputs

Your health score is the most visible metric in CS, and one of the easiest to get wrong. If it's based on outdated surveys or inconsistent tracking, it won't be useful to AI.

Audit prompts:

- [ ] Are our health score inputs manual, automated, or a mix?
- [ ] Are thresholds for each health component clearly defined and consistently applied?
- [ ] Are we over-relying on NPS or survey data without deeper engagement context?

Evaluate product usage tracking

Product data should tell a story about how your customers are progressing toward value.

Audit prompts:

- [ ] Are we tracking product usage at the feature level or just logins?
- [ ] Which actions correlate with retention, expansion, or success milestones?
- [ ] Are we capturing this data automatically, or relying on manual inputs?

Structure your sentiment signals

Sentiment is one of the most overlooked signals in CS, but it's also one of the richest when done right.

Audit prompts:

- [ ] Do we capture sentiment in structured ways (e.g., surveys, post-call summaries)?
- [ ] Are those signals stored in formats that AI can understand (text, tags, scores)?
- [ ] Do we treat qualitative feedback (emails, Slack threads) as first-class data?

Align data across tools

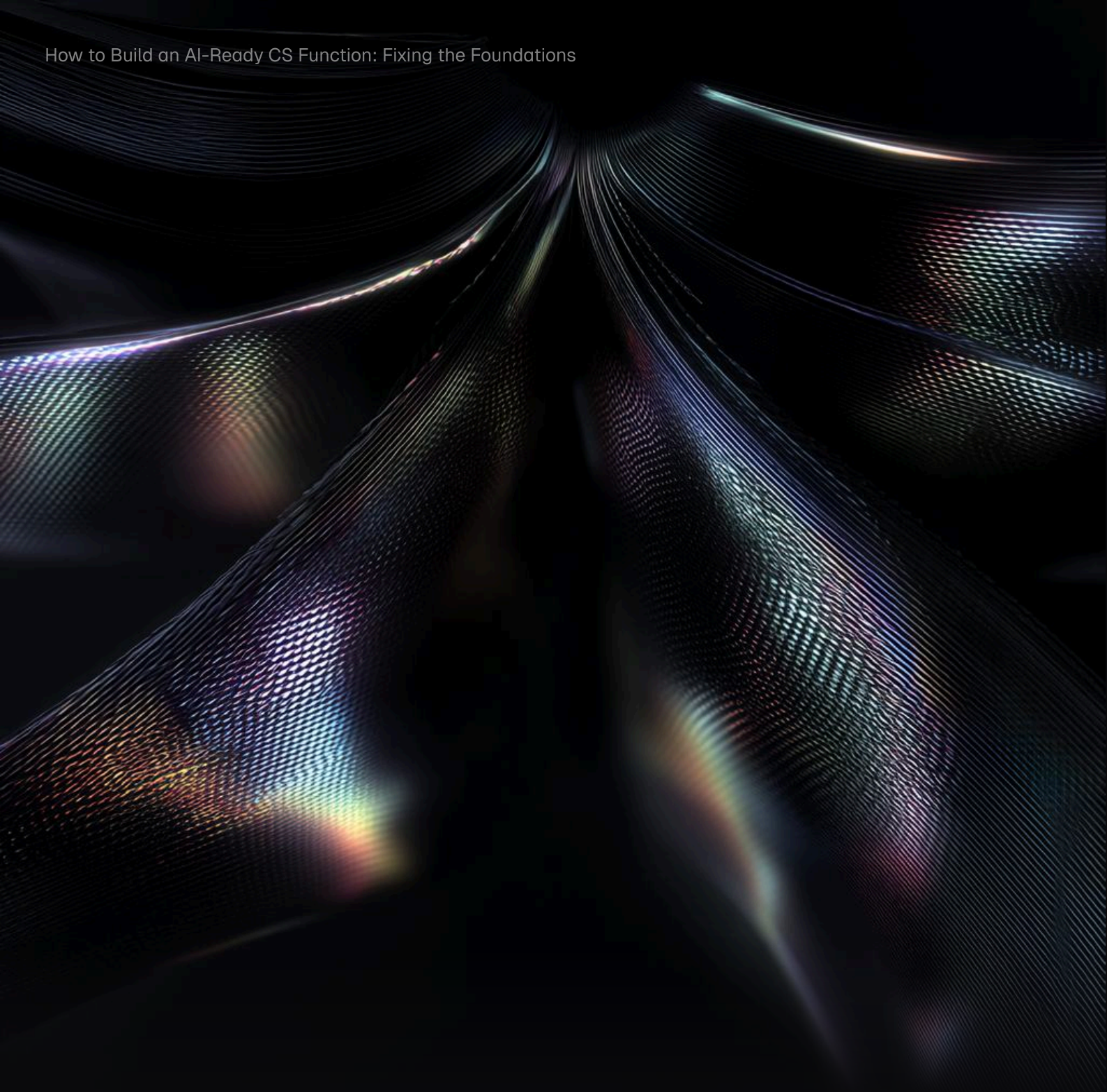
If your CRM says something different than your CSP or BI tool, AI won't know which version to trust. Consistency matters.

Audit prompts:

- [ ] Are customer fields (e.g., ARR, renewal dates) consistent across CRM, CSP, and BI?
- [ ] Do we have duplicate records or conflicting values across platforms?
- [ ] Are playbooks, success plans, and notes centralized or scattered?
- [ ] How frequently is data synced and updated across systems?

Before you try to predict churn, expansion, or adoption, make sure your data isn't lying to you. Even the best AI tools won't help if your foundation is flawed.

In Chapter 4, we'll talk about how to rethink the way your CS team works, and make sure AI supports their workflows.



CHAPTER 4

Align the team around change

Even with a clean stack and organized workflows, AI won't move the needle unless your team actually *uses* it. That's where alignment and change management come in.

A CS leader needs to build understanding, create ownership, and bake AI into the team's way of working.

This chapter gives you a checklist to assess whether your team is ready to make AI part of the day-to-day.

Get the team on board

Start by framing the *why*. If AI is introduced as a cost-saving play or a vague innovation goal, you'll lose buy-in fast. Position it as a way to amplify the strengths your team already has.

Checklist:

- [ ] Have we clearly communicated **why** AI is being introduced?
- [ ] Does the team understand how AI will support (and not replace) their work?
- [ ] Have we shared examples of how other teams (internal or external) are using it effectively?

Document what AI should follow

AI can only act on what it understands, so it's critical to have structured processes in place. If your QBR format or onboarding journey isn't documented well, an AI tool can't help at all.

Checklist:

- [ ] Are key workflows documented (onboarding, renewals, escalations)?
- [ ] Are playbooks written clearly enough that an AI agent could follow them?
- [ ] Is someone responsible for maintaining this documentation?

Enable the team to use AI

Don't assume people will "figure it out." Even simple tools require training, and your team needs space to learn and experiment.

Checklist:

- [ ] Is every CSM trained on the AI tools available to them?
- [ ] Have we designated an internal owner for AI enablement?
- [ ] Are team leads empowered to drive adoption within their pods or regions?

Reinforce adoption

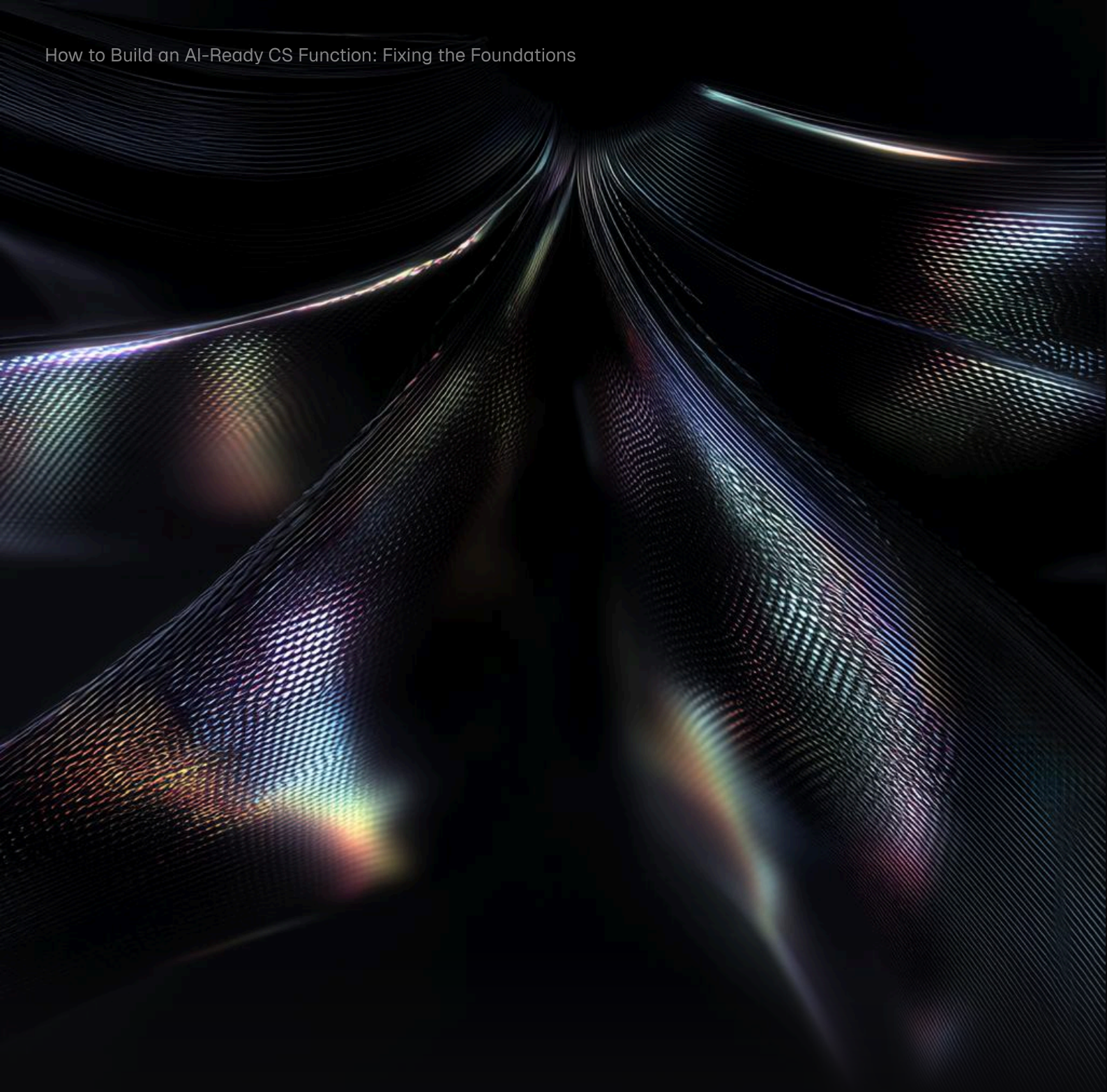
Tools alone don't create change, but habits do. Make sure there are spaces to share wins, flag issues, and continuously improve how AI is being used.

Checklist:

- [ ] Do we review AI usage or impact regularly in team meetings?
- [ ] Are we measuring adoption beyond "who has access"?
- [ ] Are there incentives or goals tied to using AI workflows?

Once your team understands why change is happening and is aligned on the direction you're heading, the next step is making that change executable.

In Chapter 5, we'll break your most critical CS workflows into modular, repeatable steps that make automation (and delegation to AI) possible.



CHAPTER 5

Modularize your CS workflows

A chaotic, linear workflow is one of the biggest blockers to AI adoption. If every onboarding or renewal follows a different path depending on who runs it, AI can't help clean that mess.

Break your key journeys into modular building blocks. This creates clarity for the team, consistency for customers, and structure for automation or AI to plug in.

This chapter walks you through how to modularize a single journey so you can apply the same structure to other workflows.

Workflow Builder

1. Choose one customer journey to focus on.

(We recommend onboarding, QBRs, or renewal. Start with the one you repeat most often.)

2. Break the journey into distinct steps.

List out each individual step, keeping them short and action-focused.

Example:

- → Kickoff meeting scheduled
- → Stakeholders identified
- → Technical setup completed
- → Training session delivered
- → Milestone 1 achieved

3. Label each step based on ownership.

Use one of the following three tags:

- → **Human-led:** Requires CSM or other human involvement
- → **AI-led:** Can be handled entirely by automation or AI
- → **Shared:** AI can assist, but a human still leads the task

4. Identify repeatable patterns.

For each step, ask:

- → Does this task happen across most customers?
- → Could it be templatized into a reusable playbook or scenario?

This is where workflows become scalable.

5. Define failure modes and fallback owners.

What happens if the step doesn't happen on time or as expected?

For each step, define:

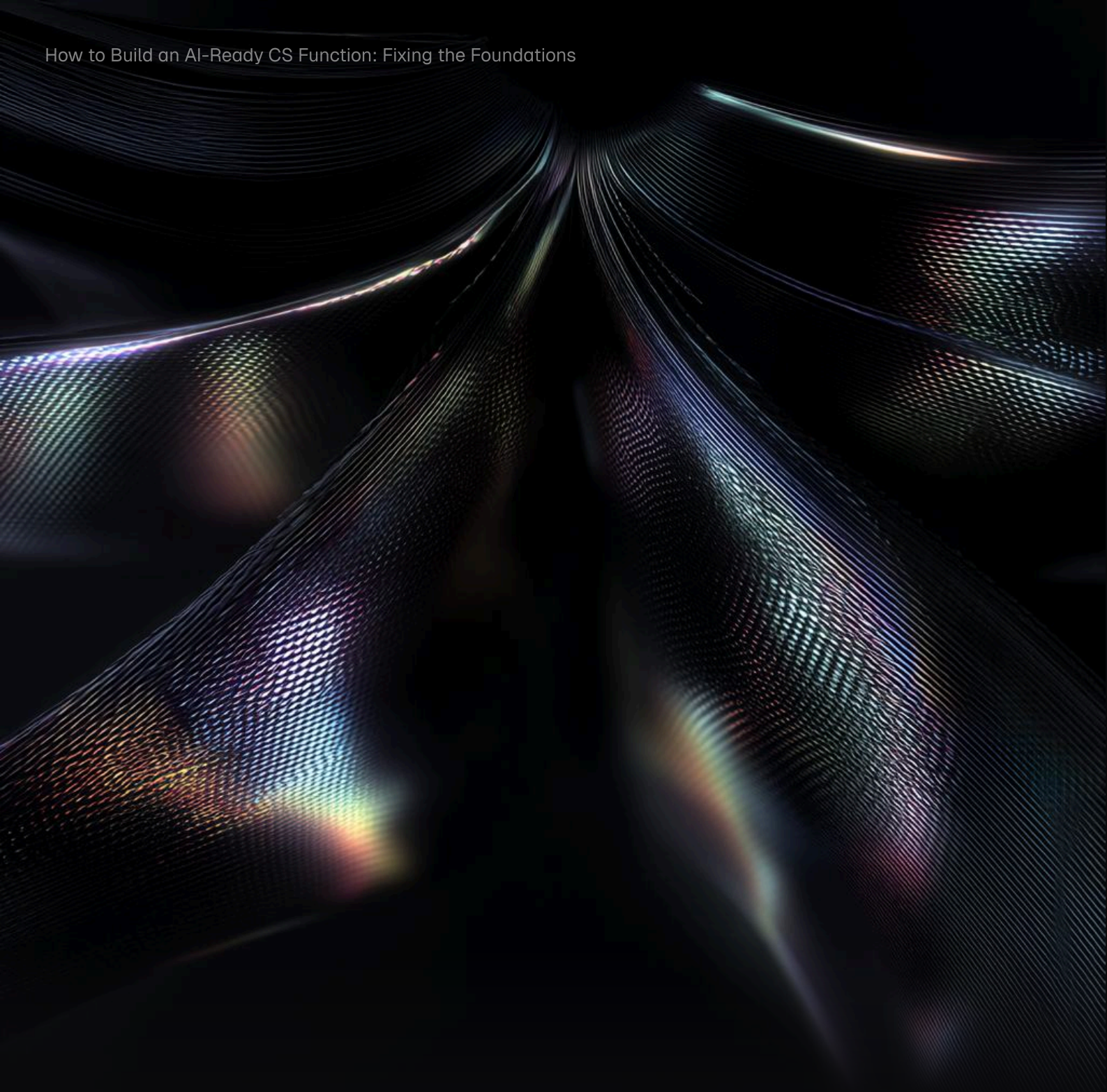
- → What does "failure" look like?
- → Who is responsible for following up or fixing it?

Example: Onboarding Workflow (simplified)

Step	Ownership	Repeatable?	Fallback Owner
Kickoff call scheduled	Shared	Yes	CSM
Welcome email sent	AI-led	Yes	Automation Owner
Key contacts confirmed	Human-led	Yes	CSM
Product configuration started	Shared	Varies	Solutions Engineer
First milestone achieved	Human-led	Yes	CSM

Once you've modularized your key workflows, you're ready to build automation and AI on top. But one of the biggest advantages AI offers is agents. And your team needs to be prepared to collaborate with them.

In Chapter 6, we'll talk about what it really means to design your CS function for agents.



CHAPTER 6

Prep for agents

Dashboards help you visualize insights. But nowadays, we have AI agents that go beyond just displaying information, and actually decide to do with it.

Which means your workflows need to move from human interpretation to machine execution.

This chapter helps you assess whether your CS function is ready to support autonomous or semi-autonomous agents.

Agent-Readiness Checklist

Use the questions below to evaluate whether your system is ready for task-oriented AI.

Ownership and logic

- → Do we have a clear owner for each CS task (human, AI, or shared)?
- → Are fallback owners defined for when AI fails or flags uncertainty?

Inputs and outputs

- → Are the required inputs for key workflows clearly defined?
- → Are the desired outputs from each task documented and trackable?

AI candidate tasks
Can an AI agent handle any of the following?

- → Customer check-ins (**e.g.** post-onboarding or risk alerts)
- → Expansion opportunity alerts (**e.g.** based on usage spikes or product signals)
- → Sentiment analysis (**e.g.** from call transcripts or support tickets)
- → Executive summaries (**e.g.** of meetings, accounts, or renewal risk)

Observability and feedback

- → Do we track the outcomes of AI-driven workflows?
- → Is there a feedback loop in place to correct or improve agent behavior?

Conclusion

If you've followed this workbook through all six chapters, you've created a structure that:

- → Makes customer data actionable
- → Segments intelligently and adapts based on value
- → Streamlines how work gets done, by both people and AI
- → Prepares your team to operate in a hybrid future

You now have:

- → A modular workflow map
- → Segmentation logic that can trigger automation
- → A clean data layer to power predictions
- → A change management plan that actually sticks
- → A foundation for agent-led execution

The next step is not to wait for everything to be perfect before you start. Pick one journey. Pick one agent. Pick one team. Start small. Iterate quickly.

And most importantly, stay grounded in the work CS is really here to do: building meaningful, long-term value for customers.

Building an AI-ready CS org won't be a one-off project, so you may have to return to the exercises in this workbook as your CS function grows and AI develops further.

But it'll be worth it, because with the right foundations, AI won't feel like disruption. It'll feel like progress.

Appendix

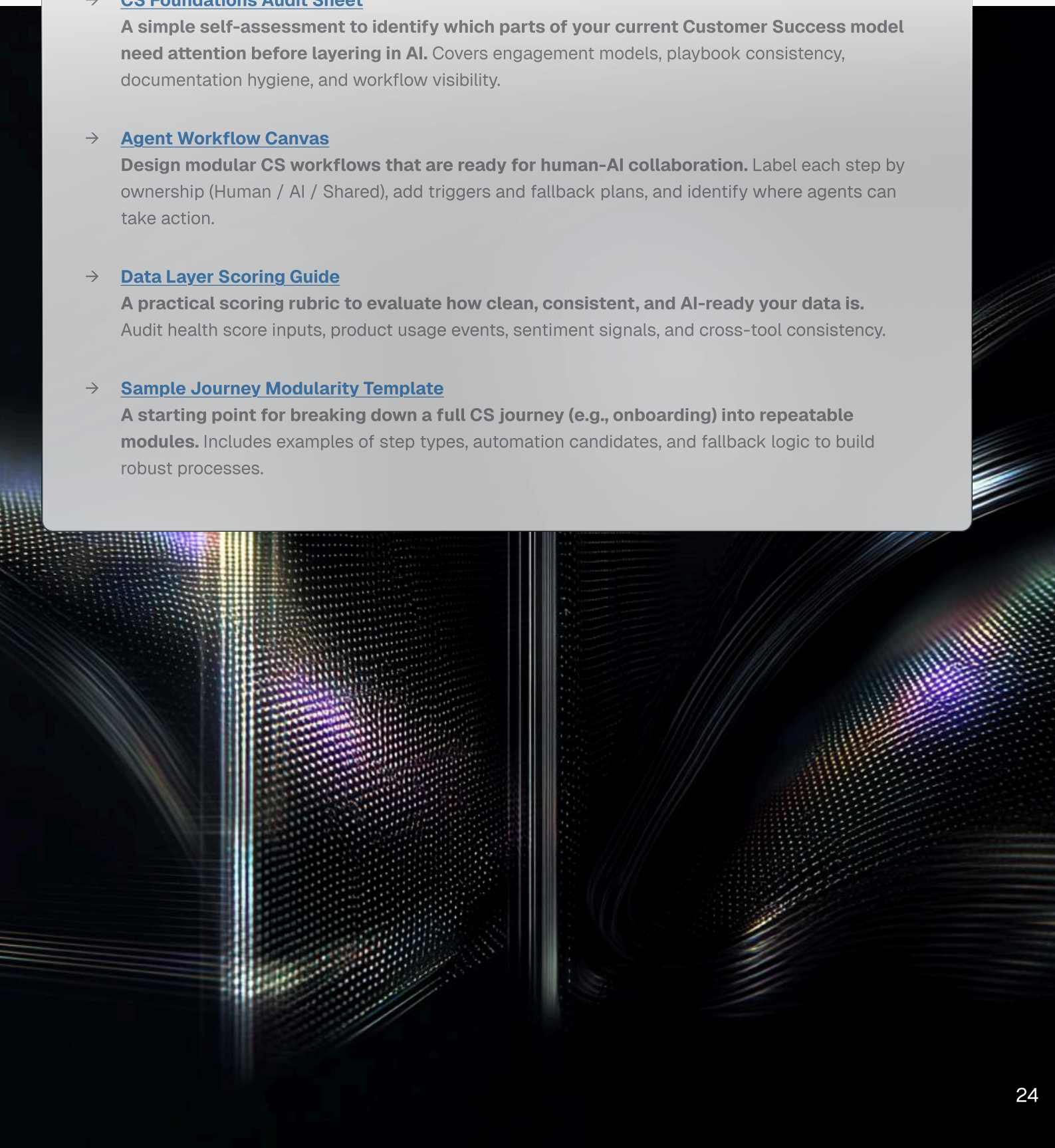
Practical Templates to Make Your CS Function AI-Ready.
Use the following resources to run workshops, align your team, and kick off your transformation with clear next steps.

→ **CS Foundations Audit Sheet**
A simple self-assessment to identify which parts of your current Customer Success model need attention before layering in AI. Covers engagement models, playbook consistency, documentation hygiene, and workflow visibility.

→ **Agent Workflow Canvas**
Design modular CS workflows that are ready for human-AI collaboration. Label each step by ownership (Human / AI / Shared), add triggers and fallback plans, and identify where agents can take action.

→ **Data Layer Scoring Guide**
A practical scoring rubric to evaluate how clean, consistent, and AI-ready your data is. Audit health score inputs, product usage events, sentiment signals, and cross-tool consistency.

→ **Sample Journey Modularity Template**
A starting point for breaking down a full CS journey (e.g., onboarding) into repeatable modules. Includes examples of step types, automation candidates, and fallback logic to build robust processes.

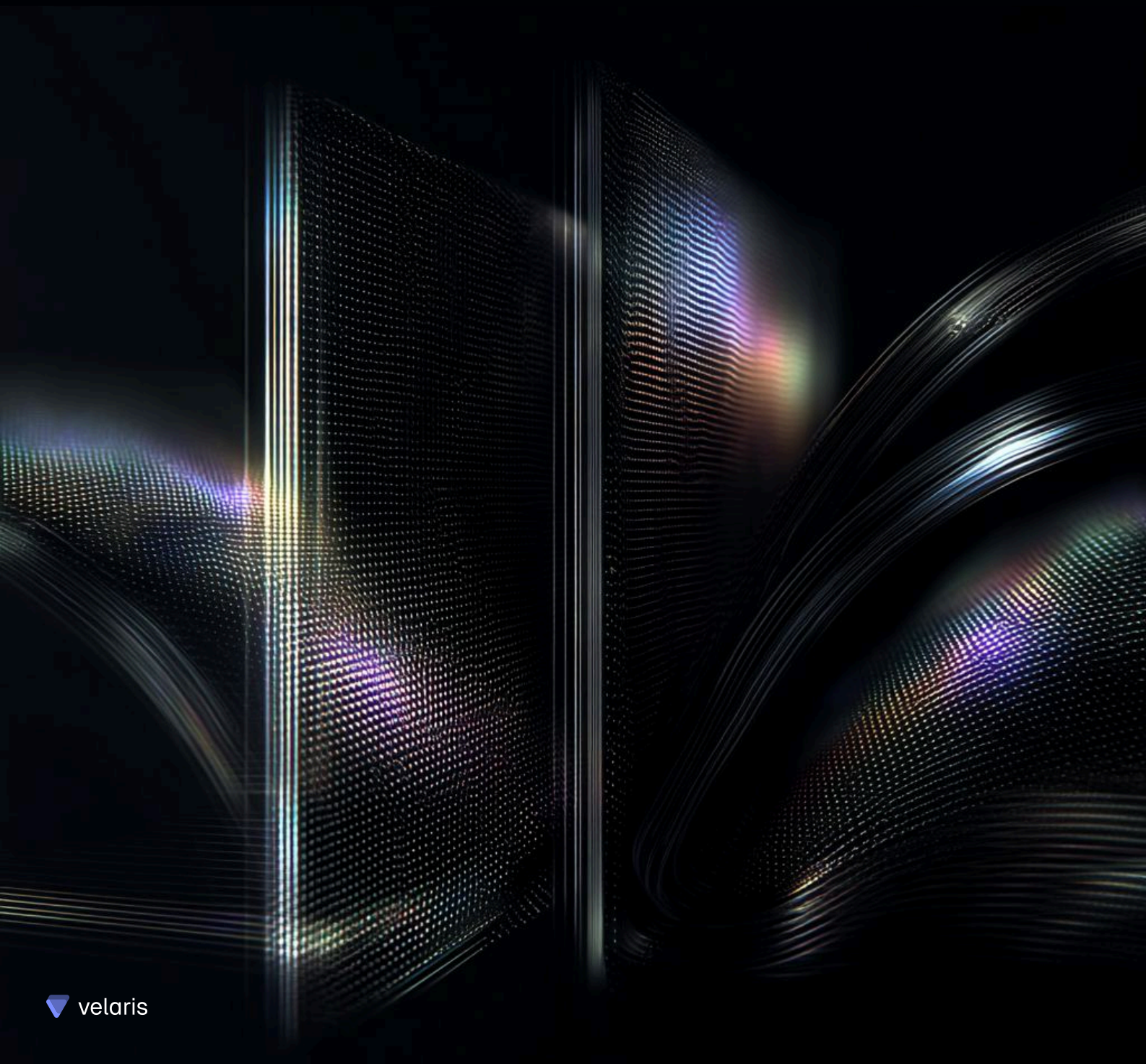


Brought to you by Velaris

Velaris is the intelligent Customer Success Platform built to help CS and post-sales teams gain control over operations, scale outcomes and prove their impact.

Want to see how we can help you?

Book a demo today ↗



velaris