

# Build vs Buy

Why building internal CS tools is easy, but getting value from them isn't.

The companies getting ahead right now aren't building CS tooling, they're making their product AI-ready. That's the race worth winning.

## WHERE WE SEE BUILD VS BUY

### Scenario 1: Claude Vibe-Coding

*CROs are vibe-coding a CS tool in Claude over the weekend, and handing over to their CS team on Monday.*



AI tools like Claude are genuinely good at getting you an answer. Spin up a quick risk detection tool, automate a summary, prototype a scoring model...the barrier to building something that works is lower than it's ever been. For teams who want to move fast without a procurement process, that's a real advantage.

### Scenario 2: Internal Build

*Data teams are spending months building Machine Learning models that generate customer health scores.*



A data team can build strong health scores and ML models that are good at spotting signals. Doing this internally means you're in full control, modelling churn signals specific to your business, integrating directly with your existing stack, and iterating on your own timeline. If you're certain you have the resource, this is a compelling option.

- General-purpose AI tools like Claude are impressive, but they're designed to be useful for everyone, which is exactly the problem. They don't know your customers, your data, or your renewal process. **They aren't accountable for your revenue number.**
- The signals might be good, but a model is only as useful as what happens next, and the action layer, the escalation logic, the renewal tracking, that's where internal builds consistently stall. **You end up owning infrastructure, not business outcomes.**

## THE PROBLEM: GENERIC AI ANSWERS QUESTIONS, IT DOESN'T DRIVE OUTCOMES.

The hard part of CS isn't intelligence anymore. It's building something you can actually trust with your business outcomes. That will learn from every interaction across your customer base and get better over time. That is coordinated across your data, your systems and your team, and encodes specific playbooks so the right thing happens every time, not just when someone remembers to check.


Claude will tell you a customer is at risk. Hook will do that, determine next best steps, trigger outreach, assign the owner, track the response, and tie it to a renewal outcome. At scale.

# Build vs Buy

Why building internal CS tools is easy, but getting value from them isn't.

## THE HONEST COMPARISON

Every CS team needs three things: to know what's happening (intel), to know what to do (strategy), and to actually do it (action). Here's how building internally compares to Hook across all three.

	Build Internally	Buy  hook
<b>Gather Intel</b>	<p>You can build a health score. But without knowing which signals historically predict churn for your product, you're taking an educated guess.</p> <p>Static models degrade. Maintaining them needs ongoing engineering resource: resource that could be building your product.</p>	<p>ML trained on your historical data finds the signals that actually predict churn/upsell. LLMs surface sentiment from comms and tickets, contextualised by the ML layer.</p> <p>Hook's data science team keeps models sharp as your business evolves. No engineering overhead on your side.</p>
<b>Make a Plan</b>	<p>Pre-defined playbooks cover common situations. But they're generic by design: they can't adapt to what's actually happening in each account.</p> <p>CSMs still have to interpret signals and decide what to do. That's time that doesn't scale.</p>	<p>Hook generates bespoke Playbooks for every customer, specific steps based on that account's data, history and trajectory.</p> <p>Customer journeys adapt after every new interaction so CSMs always have a live, relevant next action.</p>
<b>Take Action</b>	<p>Insight without action. Drafting data-backed content and execution still land on your team.</p> <p>No reliable way to ensure playbooks are followed, escalations happen, or outcomes are being reached across the whole book of business.</p>	<p>Hook triggers outreach, assigns owners, tracks responses, escalates if nothing happens, without CSM effort and admin.</p> <p>Every action is tied to a renewal outcome. We're on the <i>hook</i> for your revenue number.</p>

## HOOK IS WHERE INSIGHT BECOMES REVENUE

The intelligence is mostly table stakes now. What separates CS teams that grow revenue from those that don't isn't insight, it's execution: at scale, reliably, personalised to every account.

That's the real advantage to using Hook. Not just to surfacing the signal, but to close the loop. From the first sign of risk to a confirmed renewal or upsell outcome. At a fixed cost, without adding headcount, and without your engineers maintaining the infrastructure to get there.