



Build vs Buy Guide

What 67 Teams Told Us About Building It
Themselves



Executive Summary

AI has made building inexpensive. A tool that once took a quarter can now take an afternoon, and most GTM teams we talk to have someone who has already proved it. So when a CRM data problem comes up, "We'll build it" is a reasonable first answer.

We hear it on most of our calls. So we went back through every recorded sales and customer call where building came up. This guide includes information from **193 calls** with **67 companies**. Most of them are software companies with strong engineering teams. This guide is what they told us, plus what enterprise AI transformation leaders said when we asked them the same question.

Most of these teams build something, and they should. What separates the ones who got value from AI is where they drew the build line. They built the intelligence their business is judged on, and bought the layer underneath that has to be right every day.

BUILD IT WHEN:

Your judgment is the product.

-  ICP definitions and account priority
-  Deal strategy and messaging
-  Win-loss and churn models
-  Agents and copilots
-  The checks on whether AI output is any good

BUILD ON DATA YOU OWN IN YOUR SALESFORCE AND WAREHOUSE

BUY IT WHEN:

Reliability matters more than being different.

-  Capturing every email and meeting
-  Matching each one to the right contact, account and opportunity
-  Keeping the full history
-  Writing it into your own CRM

"Buy the dialer. Buy the CRM plumbing. Buy the systems where uptime, reliability, compliance, and scale matter more than uniqueness. There is no prize for vibe coding mediocre commodity infrastructure."



Kyle Norton

CRO, Owner.com | Source: "[Don't buy your GTM brain](#)," The Revenue Leadership Podcast

1

THE FIVE KINDS OF BUILD

"We'll build it" means five different projects.

Each one starts for a different reason and breaks in a different place. Know which one you are running before you put a price on it. Counts are companies in our sample, and some ran more than one.



THE STITCHED STACK 16 COMPANIES

HOW IT STARTS

Every team buys its own tool. Sequencing writes to Salesforce, the call recorder writes to Salesforce, the CS platform writes to Salesforce. Nobody approves a data pipeline. Three sensible integrations, signed months apart, add up to one.

WHERE IT BREAKS

Each tool writes its own version of the same event. One logs a meeting as a task, another logs it as an event, and the records never reconcile. Much of the history also sits in each vendor's own objects, so it leaves when the contract does.

"I'm pretty sure we have duplicate data writing into Salesforce today for calendar invites and meetings."

RevOps lead, fintech company



THE WAREHOUSE CLEANUP 21 COMPANIES

HOW IT STARTS

"We have a warehouse and data engineers. We'll fix it downstream." The largest group in the sample, and the most confident one, because the team really does have the skills.

WHERE IT BREAKS

You cannot clean data that was never captured, or match a meeting to a deal after the context is gone. Teams end up parsing, deduping and pushing records back into the CRM, using some of the most expensive people in the company on cleanup.

"Those resources aren't being used strategically."



Randy Likas
Head of GTM, Nektar



THE IN-HOUSE AI TEAM 8 COMPANIES

HOW IT STARTS

The company has a real AI team, an MCP stack and a warehouse. Buying anything that touches AI looks redundant.

WHERE IT BREAKS

The AI team is rarely pointed at go-to-market. It exists to give the whole company governed access to models, and GTM requests wait in line behind everyone else's.

"We do have a centralised AI team, but they're not really focused on go-to-market use cases. They're mostly there to make sure [we have] access to the technology and it's governed."

GTM leader, fintech infrastructure company



THE "USE WHAT WE ALREADY PAY FOR" BUILD 10 COMPANIES

HOW IT STARTS

An executive who already pays for an LLM, a call recorder and a CRM asks why the company should pay again. The champion has no number ready.

WHERE IT BREAKS

The team spends a few cycles assembling the answer from existing tools. Win-loss is the usual example: transcripts, CRM exports and raw email dropped into an LLM by hand, with a token bill nobody tracks. The gaps show up a quarter later.

"The pushback to them is: Okay, you figure a way to do it with what you have. And then they will waste some cycles and they'll realise it later."



Abhijeet Vijayvergiya

Co-founder and CEO, Nektar



THE BUILD ON TOP 11 COMPANIES

HOW IT STARTS

Same engineers, same ambition. These teams drew the line in a different place.

WHERE IT HOLDS

They build the agents, models and dashboards, and buy the capture and matching underneath. Mimecast built first, then bought. The full story is later in this guide.

"We need the data layer to feed everything we want to do."

GTM leader, cloud security company

What teams in the fifth group built on top.

None of these are products anyone sells. Each team built its own, on activity data it owns.

Fintech company	A signal that flags when nobody has spoken to a customer's CFO, their earliest churn warning. An agent that nudges a customer success engineer when an account goes quiet.
Mimecast	A churn-risk model with an agent on top, and a white-space agent that maps which product to sell into which account and feeds plays to BDRs.
Cloud security company	A revenue command center, and the agents they plan to run on it.
Contract software company	A reasoning model that tests whether sales engineer involvement in a trial predicts a close.
AI software company	Renewal forecasting, account health and churn-risk models, plus pre-meeting briefs written by two agents that check each other.

The Pattern: In the first four groups, the build absorbed the data problem and the team never got to the part they wanted to build. In the fifth, the data problem was someone else's job from day one, and every engineer hour went into work that set the company apart.

2 WHY AN INTERNAL BUILD LOOKS FREE

The build has no line item. The purchase does.

An internal build is paid for by people already on payroll. On a budget sheet it costs nothing, so anything you buy shows up as new spend against a zero, and loses.

The clearest example in our sample came from a compliance software company. Its GTM operations team had spent several quarters building its own activity reporting. When they looked at buying, the build had no cost attached, and the person who would have to take the purchase to leadership said so directly.

"Most of today's work the team has put in sweat equity and there were no dollars tied to it outside of the timing and effort."

GTM operations leader, compliance software company

"My biggest inertia that I'm going to face is: not any dollars being spent, we're going to spend dollars on this. Why?"

Same leader, on the case he would have to make to leadership

It gets more expensive to leave every month.

The longer a build runs, the more reports, dashboards and habits depend on it. Switching then means restating history and re-enabling the field, and that cost lands on top of the purchase.

"We are currently doubling down on everything that we've built internally. The thing that I worry about is the cost of having to work through that rip and replace... restating historical data, enabling the field."

RevOps leader, same company

And it stops where the work gets hard.

An internal build does what it was scoped for. At this company, partner activity accounted for close to a third of revenue and had never been tracked. Slack activity was on the plan and never shipped. Nobody chose to leave a third of revenue uncovered. It was never worth a sprint, and the team that built the system was the last to notice.

ASK THIS BEFORE YOU COMPARE PRICES:

What has the build already cost, and what is it keeping the team from doing next?

Same engineers, same ambition. These teams drew the line in a different place.

3

THE COST NOBODY ADDS UP

The model is the inexpensive part.

Most of the expense is in making customer data trustworthy enough for a model to act on, then keeping it that way as reps, meetings and tools multiply. Teams find the hard parts in roughly the same order, usually after the budget is approved.

"The LLM part is really not the most difficult part. The difficult part is building that trustworthy customer intelligence substrate underneath the models."



Randy Likas

Head of GTM, Nektar, on a call with a software company

1 Identity resolution

A buyer joins from a personal phone under a different name. Rule-based matching guesses, and when it guesses wrong, nothing raises an error.

2 Association

A transcript records what was said. It cannot tell you which opportunity it belongs to when two are open on one account, or when the buying group spans three subsidiaries.

3 History

A model with no history has no baseline. Restating the past is the work nobody scopes at the start, and it is the first thing a churn or win-loss model needs.

4 Compute

More reps, more meetings, more transcripts every quarter. Tokens, API calls and storage only move one way, and in-house the company carries every unit.

Then someone has to own it.

Every build needs a person who keeps it alive after launch. At a fraud-prevention software company, one internal build had turned a teammate's whole role into maintenance.

"Be careful what you build because you might have to maintain it."

RevOps leader, fraud-prevention software company

Engineers are not the constraint.

A fintech company told us its bench for GTM projects had grown from three people to as many as ten engineers. They bought the data layer anyway, and put the engineers on what sits on top.

"We now probably have, if needed, up to 10 engineers to build things."

GTM operations leader, fintech company

WHO CARRIES THE VOLUME RISK:

When you build, the company owns a consumption bill that grows with every rep and every meeting, for as long as the system runs. When you buy, that risk can sit with the vendor. Ask any vendor, including us, how pricing behaves as your data volume grows.

4

WORKSHEET

Add up your build.

Fill this in before the conversation reaches finance. Use annual figures and count what the build costs to run as well as what it cost to ship. The lines you cannot fill in are usually the finding.

COST LINE	WHAT TO COUNT	YOUR NUMBER
Building the first version	Engineers and ops people on it, months spent, fully loaded cost per person	
Keeping it running	The share of engineering and RevOps time that goes to fixes, schema changes, new fields and broken integrations after launch	
Downstream cleanup	Data engineering hours spent parsing, deduping and pushing records back into the CRM	
Model usage	Tokens for summarizing, matching and analysis, at today's volume and at next year's headcount	
CRM API calls and storage	API calls against your Salesforce limits, and the storage the activity history takes up	
History	The work to backfill and restate past activity so models have a baseline	
Enablement	Training the field each time the build or its reports change	
Coverage gaps	Revenue that flows through channels the build does not capture, such as partners, Slack or personal devices	
What the team would build instead	The agents, models or plays waiting behind this project, and what they are worth	
Annual cost to run the build		

OWNERS	WHERE THE DATA LIVES
Business owner: _____ Technical owner: _____ Backup if either leaves: _____	List each tool that writes activity to your CRM. Mark which records sit in your own objects and which sit in the vendor's objects and would leave with the contract. <i>“Tomorrow if you remove [those tools], the data goes with them... it’s on rent, it’s not permanent.”</i>  Abhijeet Vijayvergiya Co-founder and CEO, Nektar

5

CUSTOMER STORY

Mimecast built first, then bought.

1 BUILT

An internal GenAI app

An insights application on top of Mimecast's own data warehouse, for customer success managers and account managers. Call transcripts went in alongside CRM data.

2 BROKE

Thin insights

Emails were not matched to the right contacts and opportunities in Salesforce, so the app had transcripts without the relationships around them. The insights lacked context.

3 BOUGHT

The layer underneath

Nektar captured email and meeting activity, matched it to the right records and wrote it into Salesforce. The same app, fed matched data, produced much better insights.

Mimecast is a global cybersecurity company with a small AI transformation team that builds proprietary models for the business. It ran the whole build vs buy experiment, in order.

"We've used tools like Nektar to go and pull data from various places back into Salesforce. And it makes that contact information and buying committees and the emails that might go back and forth from our employees with our prospects or customers more accessible."

**Tim Seamans**

VP of AI & Business Transformation, Mimecast, on The Revenue Lounge podcast

\$10M

directly attributed expansion revenue in 6 months.

\$150M+

Identified in additional revenue.

Credit where it belongs.

"You can't say Nektar did that. These are their applications that drove that. But the data that helped to get there is everything that we piped into it."

**Randy Likas**

Head of GTM, Nektar

"We have a mixture of solutions that we've built for a proprietary advantage and solutions that we've bought."

**Tim Seamans**

VP of AI & Business Transformation, Mimecast

Read the full story [here](#).

6

FROM THE REVENUE LOUNGE PODCAST

Three operators on the same question.

We asked AI transformation leaders at three companies how they decide what to build. They run different businesses and landed in nearly the same place.

**PINO SORO**

Chief Business Transformation Officer, QuickBase

PILOTS ARE DESIGNED TO SUCCEED. SCALING IS WHERE THEY DIE.

"The thing about pilots is they're kind of designed to work, right? You take your best people, you take a really clean use case... I think the challenge is like when you try to scale it, it usually dies."

To scale, Pino says a build needs three things: data and context you can trust, a way to build once and reuse it, and "somebody to own it to make sure it still works six months later." QuickBase now puts a business owner's name and a technical owner's name on everything it builds, and documents how it was built.

"I'm sure we could build that. We're gonna go buy that... I don't want to take our data engineering resources and other resources to build something that somebody on the market could be doing better."

His test for data that AI can use has four parts: clean, connected, contextual and governed. Skip them and the failure speeds up.

"You point AI at bad data and you're... gonna get it way faster and it's gonna be more disruptive."

**AMRUTHA SURESH**

Head of GTM AI & Innovation, Asana

"COULD WE DO THAT INTERNALLY? YES, MAYBE, BUT SHOULD WE DO IT?"

"I probably am never going to build a CRM."

Her team builds the parts that encode how Asana works: the semantic layer, the context layer, governance for handoffs between people and agents, and multi-agent orchestration. In the gray zone, such as pulling insights from call transcripts, she models the economics first: number of reps, tokens to process and orchestration cost against the vendor's bill.

"This is not a one and done. You've got to be running optimizations, you've got to be releasing features, continuous improvements. So that does take... an invisible sort of a hidden cost."

She draws a hard line between one person running a tool on a laptop and a thousand-person team running it at scale, where security, permissions, edge cases, business rules and integration dependencies all show up at once.

**TIM SEAMANS**

VP of AI & Business Transformation, Mimecast

BUILD FOR ADVANTAGE. BUY WHAT EVERYONE WILL HAVE.

"If we're all using the same technology and adopting that is going to be a competitive advantage for a certain amount of time, a short amount of time. But essentially every tool, every software that you buy for business... is going to have generative AI built in."

Tim's team is small: a head of engineering and architecture who leads build vs buy decisions, a data science group building proprietary models, and program management overseeing "all of the pieces that we're building, buying, partnering on."

"One of the biggest challenges is going to be in data, right? And how do we get the right data in the right place at the right time to drive the right outcome?"

He also expects every vendor to compete for the same seat, which makes owning your data matter more each year.

"Every tool... is going to want you to use their tool and to do a thing, to build a thing, to be the platform of record."

BUILD FOR ADVANTAGE. BUY WHAT EVERYONE WILL HAVE.**They build**

What encodes how their business works: business definitions, proprietary models and the agents that use them. Pino calls this business context "the moat," because a competitor can copy tools far faster than years of business definitions.

They buy

Purpose-built systems that someone else runs at scale, where building would pull data and engineering teams off the roadmap they committed to.

They own

Everything they build, with a named person. All three treat internal builds as products with roadmaps and upkeep, never as one-off projects.

WHAT A MODEL CAN'T BUY

"It's a great model that knows everything about the world and nothing about your company. If you drop a really good model into a company with no context, it's confidently wrong."

**Pino Soro**

Chief Business Transformation Officer, QuickBase

Models are converging, and every team can buy the same one. What a model cannot buy is your history: who you spoke to, about which deal, and what happened next. That history has to be captured and matched correctly before any definition or agent on top of it can be trusted.

All three conversations are on [The Revenue Lounge](#), Nektar's podcast on AI & GTM, hosted by Randy Likas.

7

CHECKLIST

Five questions to answer before you build.

Take these into the room before the decision is made. A question you cannot answer yet tells you where the risk is.

1 Which parts of this are your judgment, and which just have to work every day?

Build where your definitions and decisions are the product. Buy where being reliable matters more than being different.

☐ We have split the project into those two layers

2 What will your team build on top, and what does it need underneath?

Name the agents, models or dashboards. For each, list the data it needs: which emails and meetings, matched to which contacts, accounts and opportunities, going back how far.

☐ Every planned use case has its data requirements written down

3 Who keeps it running a year from now?

Name a business owner and a technical owner. Decide what happens when the person who built it moves on.

☐ Both owners are named, and the build is documented

4 If you switched off a tool tomorrow, which records would stay?

Check which activity history sits in your own Salesforce objects and warehouse, and which sits in a vendor's objects and leaves with the contract.

☐ We know where every activity record lives and who controls it

5 What has the build already cost, and what is it stopping the team from doing next?

Count people, upkeep, tokens, API calls and storage, and the channels it does not cover. Use the worksheet in this guide.

☐ We have an annual cost to run the build, next to the price of buying

You build the intelligence. Nektar delivers the context.

Nektar captures every customer interaction across email, calendar, call transcripts and Slack, with no rep input. It matches each one to the right contact, account and opportunity and writes it natively into your own Salesforce, then keeps that data complete, current and correctly mapped on its own.

The history belongs to you. Your agents, models and BI tools build on it, and it stays when other tools leave. Nektar makes Salesforce safe for AI execution.

Talk to our team



Read the
Mimecast story



     

     