



Revenue Architecture

Definitions, handoffs, measurement,
change.

CONTENTS

- 01 Definitions, and why they come before anything else
- 02 Handoffs, and where revenue leaks between teams
- 03 Measurement, and one number with one source
- 04 Change, and what happens when the business moves

Definitions, and why they come before anything else.

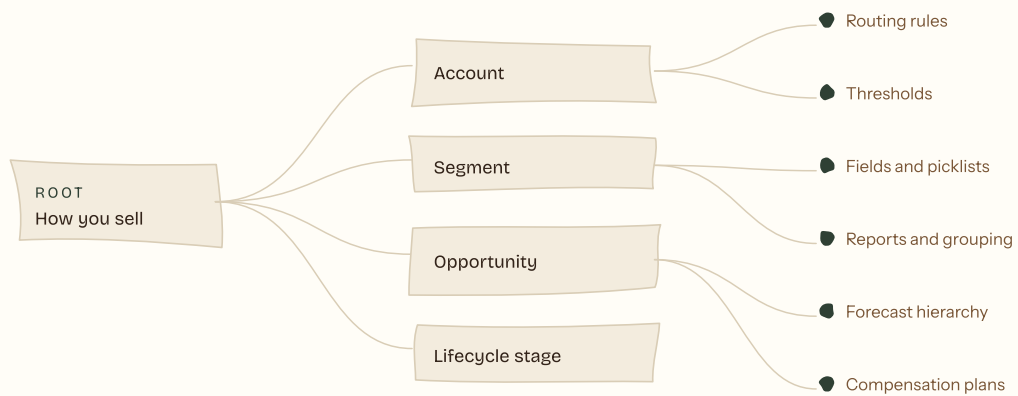
Revenue architecture is the small set of definitions that everything else inherits, plus the rules that keep those definitions true after other people start changing them.

Nearly every argument about a number turns out to be an argument about a definition. Marketing reports four hundred leads for the month. Sales reports sixty. Both are reading their own system correctly and both are right. Nobody ever wrote down what a lead is, so each team answered the question in the way that made sense from where they sat.

So the first work is settling a short list of terms in writing, in one place, each with a named owner. What an account is, and whether a parent and its subsidiaries are one record or several. What a segment is, and which field computes it rather than who assigns it by hand. What a qualified opportunity is, expressed as conditions that have to be true rather than as confidence. What a lifecycle stage means, and whether a record is allowed to move backward through one. What a renewal is, what churn is, and the date each one counts on.

Written down, these look too obvious to be worth a meeting. They are also the reason two teams reporting the same month produce two different numbers, and they cascade. Every field, every report and every compensation plan inherits from them. Change one a year in and you invalidate the history of everything downstream.

We do not invent these definitions. They come out of how you already sell, which is usually consistent in practice and undocumented. The work is writing down what is already true, finding the three or four places where practice disagrees with itself, and getting a decision on those from somebody with the standing to make it.



A CHANGE AT THE ROOT REACHES EVERY LEAF

ARTIFACT 1

What inherits from what. A change at the root moves through every branch above it, which is why these get settled first and revisited on a schedule rather than in the middle of a quarter.

Handoffs, and where revenue leaks between teams.

Revenue goes missing between teams more often than inside them. Each team is measured on its own step and nobody is measured on the seam, so the seam is where records go quiet.

A working handoff needs four things, and it needs all four.

An entry condition. What has to be true before the record moves. Written, checkable, and not a judgment call made in the moment by whoever is under quota pressure that week.

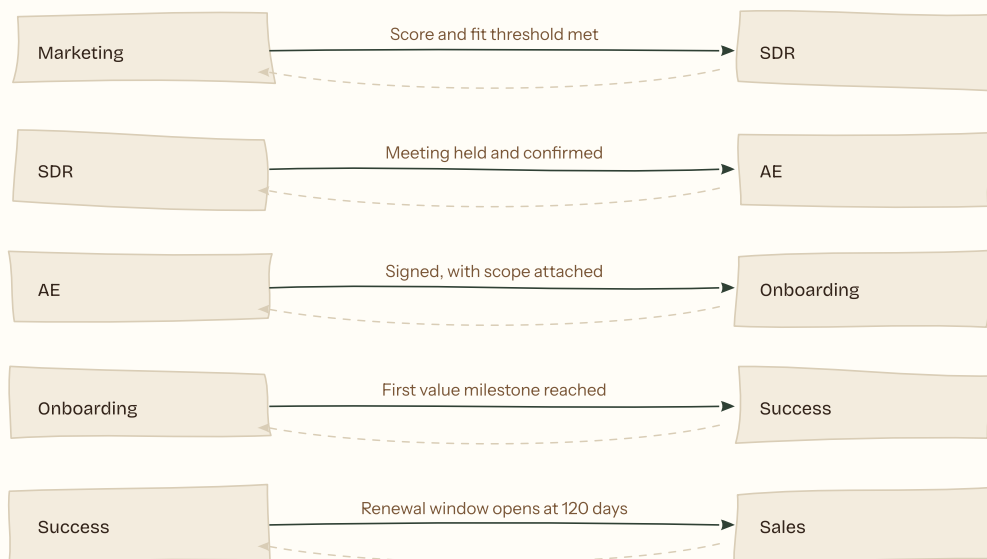
An owner on both sides. A person on each end rather than a queue. The handoff is not finished when the sender pushes it, it is finished when the receiver accepts it.

A time limit. How long the receiving side has, and what happens automatically when that time passes. Without this, a handoff has no failure state and nothing ever surfaces.

A way back. What happens when the receiving side rejects the record, where it goes, and who hears about it. A handoff with no reverse path becomes the place records disappear into, and the reject rate is one of the more useful numbers you will get out of this work.

The common shape we find is a handoff with an entry condition and none of the other three. Records move, nobody formally accepts them, and six weeks later somebody asks why conversion between two stages dropped. The answer is usually that it did not drop, and the records were never worked.

Every seam gets the same four. Marketing to SDR, SDR to AE, AE to onboarding, onboarding to customer success, and customer success back to sales at renewal. That last seam is the one most companies never build at all, which is why renewals arrive as a surprise.



SOLID, THE HANDOFF. DASHED, THE WAY BACK WHEN IT IS REJECTED.

ARTIFACT 2

Five seams, each with an entry condition, an owner on both sides, a clock, and a way back. The loop from customer success to sales is the one most often missing.

Measurement, and one number with one source.

A metric that can be computed two ways will be computed two ways, and the two answers will diverge in exactly the quarter you most need them to agree.

So every reported number gets a ledger entry. The definition in one sentence, the single system it is computed from, the person accountable for it, and how often it refreshes. A number that is not in the ledger does not go in a board deck. That rule sounds severe for about a month and then nobody misses the numbers it kept out.

One source per metric is what does the work here. Not one source per system, one per metric. Pipeline comes from the CRM. Recognized revenue comes from billing. Those two will never tie exactly, and they are not supposed to, because they measure different things at different moments in the same story. What matters is that everybody knows which is which and stops reconciling them by hand every month.

Definitions get versioned like anything else. When one changes, the change is dated and the previous series is kept, so a trend line cannot quietly rewrite its own history and make last year look like something it was not.

The test for whether this is working is short. Ask two people on different teams for the same number and see whether you get the same answer without either of them opening a spreadsheet first.

METRIC	DEFINITION	SOURCE OF TRUTH	OWNER	REFRESH
Pipeline created	Opportunity value with a create date inside the period	CRM	VP Sales	Hourly
Qualified opportunities	Opportunities that met the stage two exit condition	CRM	RevOps	Hourly
Marketing sourced pipeline	Pipeline where the first recorded touch is a campaign	CRM	Marketing	Daily
Recognized revenue	Invoiced amount recognized inside the period	Billing	Finance	Monthly
Net revenue retention	Recurring revenue from last year's customers over last year	Billing	Finance	Monthly

Pipeline and recognized revenue come from different systems on purpose. They are not expected to tie.

ARTIFACT 3

The metric ledger. Five rows, five columns, one source each. If a number cannot be given a row here, it is not ready to be reported on.

Change, and what happens when the business moves.

An architecture that only works for the business as it is today is a rebuild already scheduled, about eighteen months out, by somebody who does not know they are going to be doing it.

Four changes will arrive, and each one reaches further than people expect.

A **pricing change** touches the opportunity model, the quoting rules, every report on average deal size, and then the compensation plan, in that order.

A **new segment** touches the segment definition, routing, every threshold that was tuned per segment, and every report that groups by one.

A **new product** touches what an opportunity is, whether an expansion is a new record or a line on an existing one, and how churn gets counted when a customer drops one product and keeps another.

A **reorg** touches ownership on every open record, all five handoff seams, and the reporting hierarchy underneath the forecast.

The ripple is the whole point. A change made at the definition layer travels outward through fields, then reporting, then compensation, and the further out it travels before anyone notices, the more it costs to put right. Making the change deliberately at the center and letting it propagate is a fraction of the price of finding it at the edge in the fourth quarter, in a board meeting.

So the architecture ships with a change procedure attached. Who approves a definition change. What gets checked when one is approved. How the previous series is preserved. One hour a quarter, which is the same answer as the rest of this work, because it is the same problem.



ARTIFACT 4

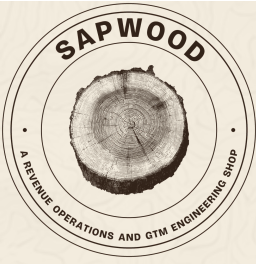
One change at the center, and the rings it moves through. Cost rises with every ring it crosses before somebody catches it, which is the argument for changing things at the middle on purpose.

Where this sits next to the other three.

Architecture is the layer the other guides inherit from. The CRM guide is how the definitions get expressed as records. The tech stack guide is how the tools around those records earn their licenses. The GTM engineering guide is what becomes possible once the first three hold.

They can be read in any order. They are usually done in that one.

CRM setup and rebuild · Tech stack implementation · Intent signal delivery



If two teams keep reporting different numbers, we can help.

Thirty minutes, no charge, no deck. Tell us which number people argue about most and who owns it today. We will tell you on the call whether this is an architecture problem, a CRM problem, or a definition somebody can settle this week without us.

hello@sapwood.io

SAPWOOD

REVENUE ARCHITECTURE

SEPTEMBER 2026

DENVER, COLORADO