



Tech Stack Implementation

Inventory, configure, consolidate,
govern.

CONTENTS

- 01 Inventory, and what you are actually paying for
- 02 Configure, and what implemented really means
- 03 Consolidate, and where the overlap is
- 04 Govern, and what happens before the next tool

Inventory, and what you are actually paying for.

Start with a list almost nobody has. Most companies under 250 people cannot produce a current list of what they license, what each one costs, when it renews, and who owns it. Assembling that list takes about two days, and it changes the conversation before any work has been done.

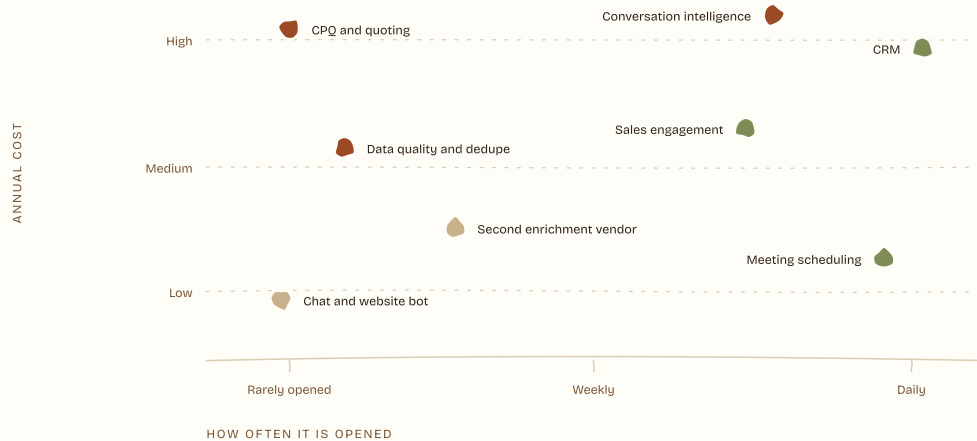
Four columns carry most of the weight. What it costs a year. When it renews, and how much notice the contract requires. Who inside the company is accountable for it by name. And how many people opened it last month.

That last column does the heavy lifting. Seat counts get bought on optimism and are rarely revisited afterward. A tool bought for a team of twelve is often still billing for twelve two years after four of those people moved on. Seat reconciliation on its own usually covers the first month of this work.

Then the numbers get plotted. Annual cost against how much the thing is actually opened. The tools at the top right are healthy: they cost money and people use them. The ones at the top left are the finding, expensive and untouched since the implementation call. The bottom half is mostly noise, cheap enough that it can wait for the next renewal.

We do not recommend cancelling anything in the first week. Low usage is often a symptom of a tool that was never configured rather than a tool nobody needed, and cancelling it just repeats the same purchase at a different vendor next year. Telling those two apart is the point of the next section.

One more thing comes out of the inventory that people do not expect. Shadow tools. Somewhere in the company there is a paid subscription running on a personal card, doing something important, known to two people. Inventories surface those, and surfacing them is worth the exercise on its own.



●	Conversation intelligence Recording calls nobody reviews	Configure High cost	●	CRM Used daily by everyone who sells	Keep High cost
●	CPQ and quoting Bought for a pricing model that changed	Configure High cost	●	Sales engagement Sequences running every morning	Keep Medium
●	Data quality and dedupe Installed, connected, never rule set	Configure Medium	●	Meeting scheduling Cheap, universal, invisible	Keep Low cost
●	Second enrichment vendor Overlaps the first at eighty percent	Cancel Medium	●	Chat and website bot Two logins in the last quarter	Cancel Low cost

ARTIFACT 1

A stack plotted by annual cost against how often it gets opened. The top left is where the money goes: expensive licenses that were connected and then left alone.

Configure, and what implemented really means.

Implemented is a word covering five separate states, and every vendor counts all five as a successful rollout.

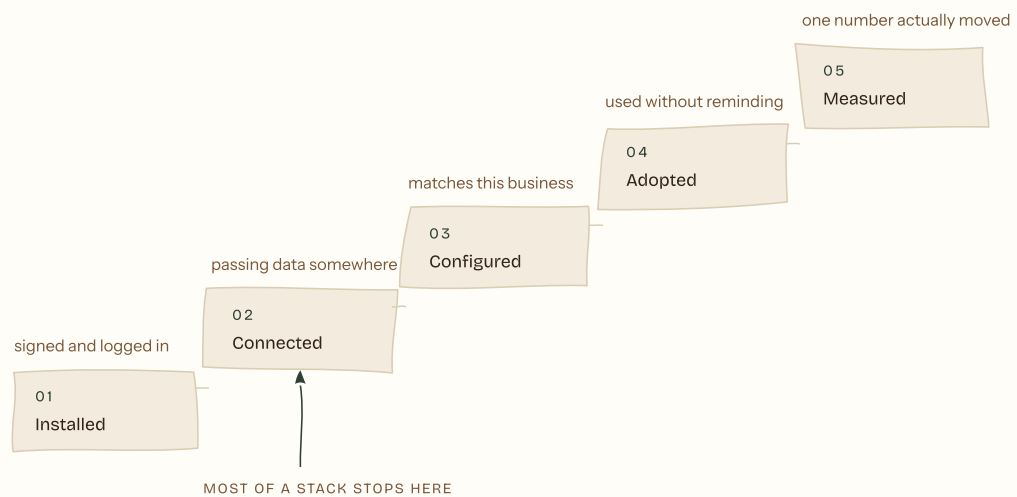
Installed. The contract is signed and somebody has logged in. **Connected.** It exchanges data with at least one other system. **Configured.** Its fields, stages and rules describe how this business works rather than how the demo worked. **Adopted.** The people it was bought for use it without being reminded in a team meeting. **Measured.** Somebody can say what changed because of it.

Most tools in most stacks stop at connected. They pass data, they bill monthly, and nothing downstream of them moved. A tool sitting at that step is indistinguishable from a tool nobody bought, except for the invoice.

The work of implementation is the last three steps, and none of it is glamorous. Mapping fields onto a model that already exists rather than accepting the vendor's defaults. Choosing which two of the twelve suggested workflows you actually want, and turning the other ten off. Sitting with the four people who will use the thing daily and watching where they get stuck, which is never where the documentation says they will. Then picking one number, writing it down, and checking it ninety days later.

We do this one tool at a time. Running three implementations at once means that when a number finally moves, nobody can say which of the three moved it, and you learn nothing you can reuse on the fourth.

Adoption is the step that gets skipped, and it is the one that decides whether any of the rest survives. A tool that is technically perfect and unused is the same as no tool, at full price. So the last thing we do on every implementation is sit with the team a fortnight after launch, watch what they are avoiding, and fix that instead of writing more documentation.



ARTIFACT 2

The five states a vendor will describe as implemented. The gap between the second and the third step is where most of a stack budget quietly sits.

Consolidate, and where the overlap is.

Overlap is the second finding, and it stays invisible until the capabilities get written down beside each other.

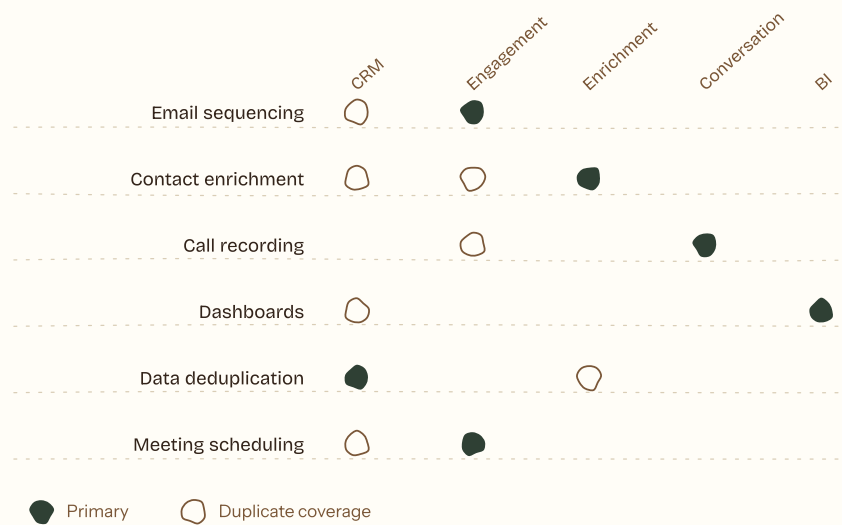
Draw a grid. Capabilities down one side, tools across the top, a mark in every cell a tool covers. In most stacks we read, three or four capabilities are covered twice and one or two are covered three times. This happens because different teams bought different tools in different years to solve the same problem described in different words, and each purchase was reasonable on the day it was made.

Duplication costs more than the second license. Two systems that both send sequenced email means two places a rep can send from, two sets of activity reporting, and two answers to the question of how many times an account was contacted. The license fee is the cheapest part of that.

The rule we use is short. For each capability, one tool is primary. The others either lose that capability in configuration or they leave the stack. Which one is primary is a business decision about where the data ought to live, and the answer is usually whichever system already owns the underlying record.

Turning a capability off inside a tool you are keeping is the move people forget. You do not have to cancel a contract to stop the duplication. Often the second tool is genuinely good at one thing and mediocre at four others, and switching the four off is enough.

This is also the part that generates the most argument, because somebody chose each of these tools and defended the purchase at the time. We keep the conversation on capabilities rather than vendors, which takes most of the heat out of it.



ARTIFACT 3

Capabilities against tools. Filled marks are the primary system for that capability, hollow marks are duplication. Six capabilities, five tools, and eleven marks where six would do.

Govern, and what happens before the next tool.

Stacks sprawl because buying is easy and removing is nobody's job. Governance is the small amount of process that changes the second half of that sentence.

Four questions get asked before anything new gets bought, and they get answered in writing.

Does a tool we already pay for do this? Most of the time something does it at eighty percent, and eighty percent of a capability you already own beats a hundred percent of one you have to implement.

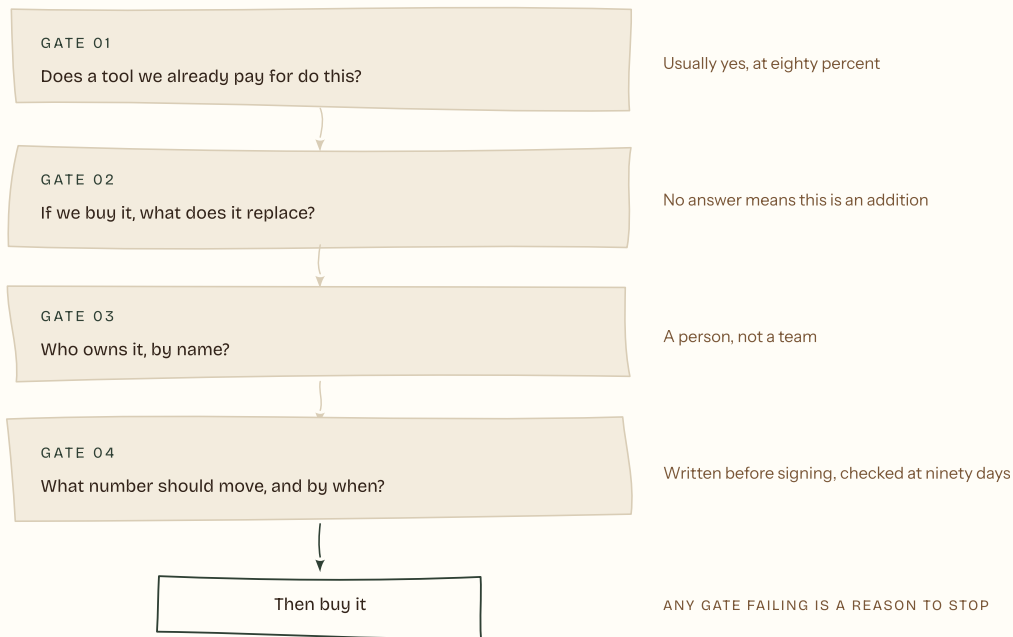
If we buy it, what does it replace? A purchase with no answer to this is an addition, and additions compound. Every tool added without one removed raises the cost of every future integration.

Who owns it, by name? A person, not a team. Somebody who will be asked about it at renewal and who has the standing to say it did not work out.

What number should move, and by when? Written down before the contract is signed rather than reconstructed afterward, and checked ninety days later whether or not anyone remembers to ask.

Then a renewal calendar. Every contract, its date, and its notice period, reviewed a quarter ahead rather than in the week the auto-renewal fires. Notice periods are where whatever leverage you have lives, and they expire quietly.

None of this is complicated. It is a spreadsheet and a recurring hour. It is also the difference between a stack that holds at nine tools and one that reaches nineteen inside three years, four of them doing the same job in slightly different words.



ARTIFACT 4

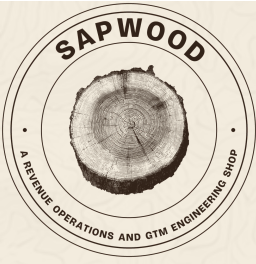
The four gates. Any one of them failing is a reason to stop, and stopping at the first gate is the outcome most of the time, which is the point.

Where this sits next to the CRM.

A stack question and a CRM question look similar from the outside and are answered differently. If the record itself is wrong, no amount of tooling around it helps, and the work belongs in the CRM. If the record is sound and the tools around it are half configured or duplicated, this is the work.

The inventory tells you which one you are looking at, which is why it comes first.

The CRM guide, if the record is the problem



If you are not sure what you are paying for, we can find out.

Thirty minutes, no charge, no deck. Tell us roughly how many tools you license and which ones you suspect nobody opens. We will tell you on the call whether this is an inventory, an implementation, or a conversation you should have with procurement instead of with us.

hello@sapwood.io

SAPWOOD

TECH STACK IMPLEMENTATION

SEPTEMBER 2026

DENVER, COLORADO