sturdy

WHITEPAPER · STURDY × CLAUDE

The Opportunity Review, Rebuilt From Real Conversations

From inspection to coaching: taking your pipeline reviews from establishing the facts of a deal to deciding the next move on it.

PART I

Why we built this

The premise

A pipeline review today is an inspection. The leader asks questions to find out what is actually happening in a deal, and the rep answers from memory. Most of the meeting goes into establishing the facts. Whatever time is left goes into deciding what to do about them.

We built this to invert that ratio. If the leader walks in already holding the deal's evidence, structured and sourced, the meeting stops being about extraction and starts being about the next move. The review becomes coaching and deal strategy rather than an audit of the rep's recall.

The result is what this paper calls the artifact: a single rendered deal review, produced on demand, built entirely from the deal's real communication record. Everything else in this paper is in service of it.

A note on terms

Throughout this paper, **the thread** means the actual conversational record of a deal: every email, Slack message, ticket, and meeting exchange between the buying side and the selling side, across every channel, in the order it happened. It is not a single email chain. It is the complete two-way communication history that Sturdy ingests, normalizes, and matches to the CRM opportunity.

> The thread is the only record of a deal that nobody edits after the fact.

Why the current review runs on inspection

A deal review draws on three inputs. The CRM opportunity record supplies stage, close date, amount, and probability. The rep supplies recollection and context. A qualification framework, usually MEDDIC or a local variant, supplies the shape of the conversation.

All three trace back to the same person. The rep sets the stage. The rep sets the close date. The rep answers the framework questions.

This is not a claim that reps are dishonest. A qualification framework is a set of questions. It tells you which facts to collect, and it has no way to check whether the answers are right. If a rep believes the economic buyer is the VP of Operations, the scorecard will say VP of Operations, and it will look exactly as authoritative as a scorecard where that answer was verified.

The record degrades for structural reasons too. Reps update CRM fields on the cadence the process demands, usually weekly or at quarter close. Deals move on the buyer's cadence, which is whenever the buyer decides to move. Stage and close date describe a plan. The thread describes what happened. The two drift apart quietly and nothing in the stack notices.

So the leader inspects. Not out of distrust, but because inspection is the only mechanism available for getting at facts the leader has no independent access to.

The evidence exists and nobody reads it

The information that would settle any of this is already sitting in the thread. A deal with forty conversations over eight weeks is several hours of reading, and the reading has to happen before the review rather than during it.

That reading does not happen, so the review falls back to the CRM record, which describes the plan for the deal rather than the deal.

The state that earns the workflow

Every review the artifact produces opens with one of three motion states. Progressing means the buyer moved the deal within its own established cadence. Stalled means nobody is moving it. One-sided means we are moving it and they are not.

One-sided is the reason we built this.

A stalled deal already surfaces on its own. Activity drops to zero and zero is visible in every pipeline report. A one-sided deal produces the opposite signature. The rep is still working it, so outbound volume looks healthy, and the deal reads as active on every dashboard the company owns. Inbound has stopped, and nothing in the stack tracks inbound on its own.

It is worth being precise about what the rep does and does not know here, because the obvious objection is that a rep knows perfectly well when a buyer has gone quiet.

Often they do. What they do not have is a way to tell the difference between a silence this deal has survived before and a silence that is new. A buyer who takes three weeks to reply is alarming in one account and completely normal in another, and no rep is holding that baseline in their head across a book of thirty deals. What a leader gets in the review is a sentence like "waiting on legal," which is true, and which sounds identical whether legal received the paper last Thursday or six weeks ago and stopped responding.

That is the gap. Both people in the room are working from the same information set, and neither of them can see the pattern the messages make in aggregate. The leader has not been in the conversations. The rep has, but one at a time over two months, with no view of the shape.

Inspection cannot close that gap, because inspection only surfaces what the rep already knows. The only way out is to bring a different information set into the room.

That is what the artifact does. It records the date and direction of every message across the deal's history, reads the recent conversations in full, and derives the state from the dates. The threshold it measures against comes from the deal's own history rather than a global rule, so a deal that has always moved slowly is not flagged for moving slowly. It is flagged when it goes quieter than it has ever gone and recovered from.

Two limits are worth stating plainly. The artifact can only see the channels Sturdy is connected to, so a buyer who is active on a channel we do not ingest will read as silent, and the review says so rather than assuming. And a one-sided state is not a verdict. Some deals are legitimately one-sided while a security review runs. The review is built to be argued with on exactly this point, and carries a section for the argument, covered in Part II.

None of that changes the value of the finding. When the review opens with One-sided, the leader is not asking the rep what is happening. The leader is putting something on the table that neither of them could see, and the conversation becomes what to do about it.

PART II

What it is

You ask for a review of a named opportunity. A few minutes later you have a single formatted document.

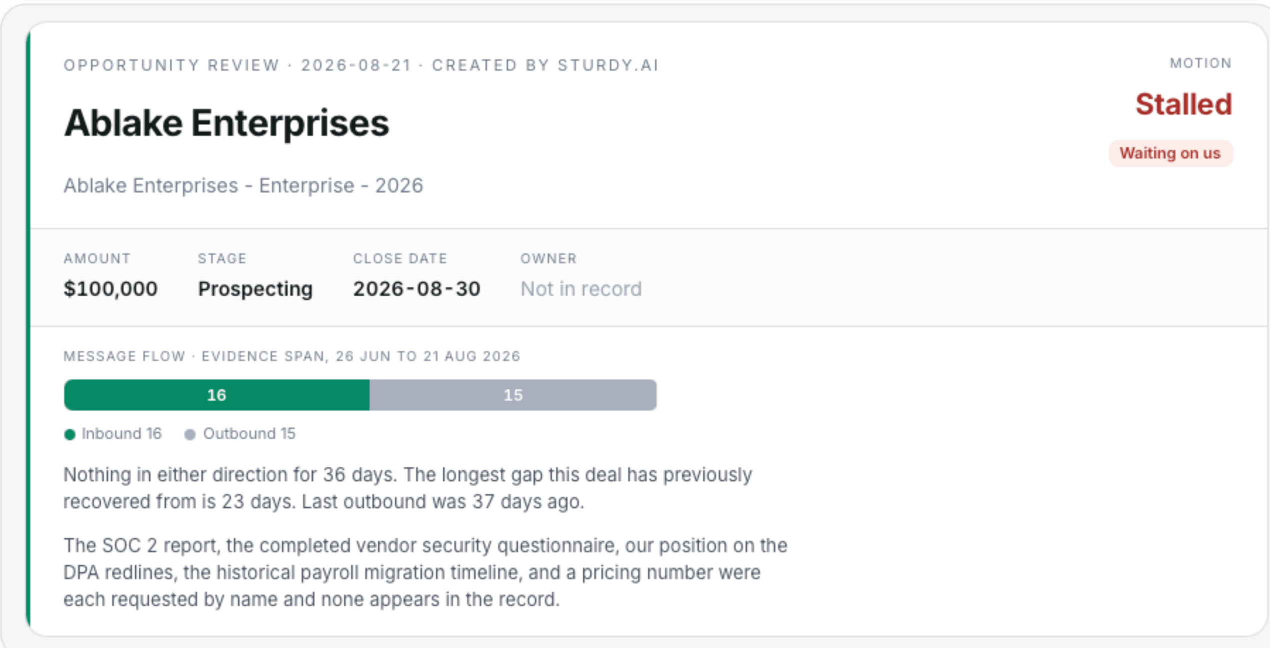


Figure 1. The artifact header. Motion state on the right, the CRM facts on one rail, and the message flow track with the deal's own silence tolerance stated in a sentence.

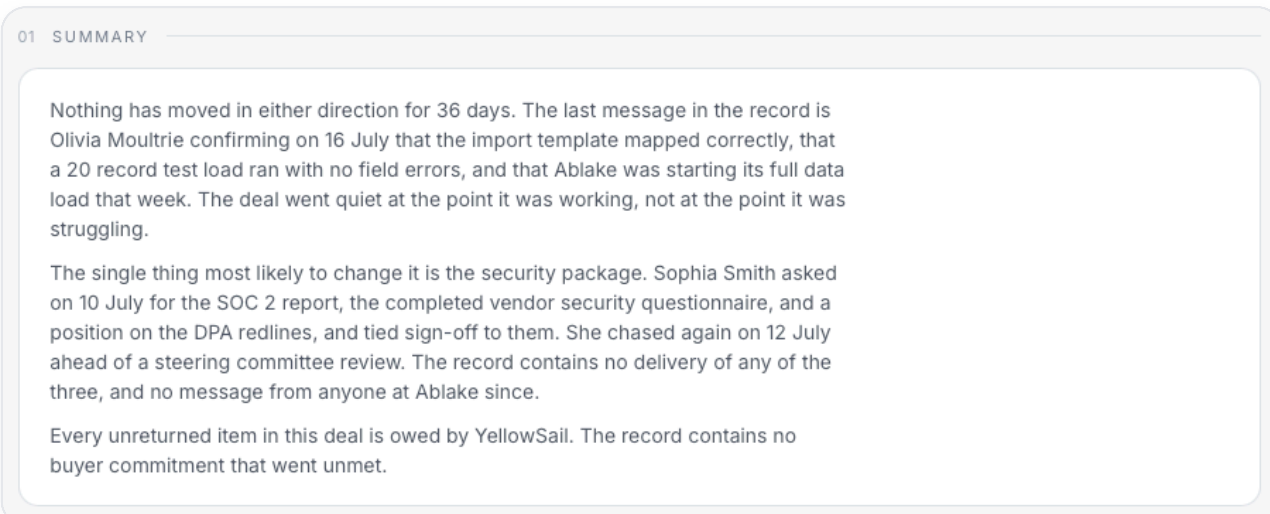
It is organized into twelve blocks. The blocks are not a summary of the deal in narrative form, because narrative is what the rep already provides and it is not checkable. Each block answers one question a leader would otherwise have to ask in the meeting, and each block carries its own evidence so the answer can be verified without asking anyone.

The claims that carry the most weight, the deal killers, commitment signals, and unresolved concerns, each arrive with their evidence: a date, a source, a named person, and either a verbatim quote from a real message or a clearly labeled paraphrase. Every piece of evidence is also marked by direction, so a commitment the buyer made to us and a commitment we made to the buyer are never confused with each other.

The blocks are deterministic structures. Every opportunity review produces the same twelve, in the same order, whether or not the deal has content for all of them. A block with nothing in it renders an explicit empty state rather than disappearing, because a missing section reads as an oversight and a stated empty section reads as a finding. Sorting, date arithmetic, and empty states are computed by the template rather than written by Claude. That is what makes two reviews of two different deals directly comparable, and it is why two reviews of the same deal a week apart differ only where the deal differed.

The twelve blocks

01 **Summary.** Two to four sentences. The deal's status first, then the single thing most likely to change it. This exists so a leader who reads nothing else knows where the deal stands and what to press on.



01 SUMMARY

Nothing has moved in either direction for 36 days. The last message in the record is Olivia Moultrie confirming on 16 July that the import template mapped correctly, that a 20 record test load ran with no field errors, and that Ablake was starting its full data load that week. The deal went quiet at the point it was working, not at the point it was struggling.

The single thing most likely to change it is the security package. Sophia Smith asked on 10 July for the SOC 2 report, the completed vendor security questionnaire, and a position on the DPA redlines, and tied sign-off to them. She chased again on 12 July ahead of a steering committee review. The record contains no delivery of any of the three, and no message from anyone at Ablake since.

Every unreturned item in this deal is owed by YellowSail. The record contains no buyer commitment that went unmet.

Figure 2. Block 01, Summary. Status first, then the item most likely to decide the outcome.

02 **Why now.** The timing pressure, stated in the buyer's own terms. This exists because urgency is the most commonly invented fact in a pipeline review. If the record contains no urgency, the block says so. A deal with no timing pressure is a real and useful finding.

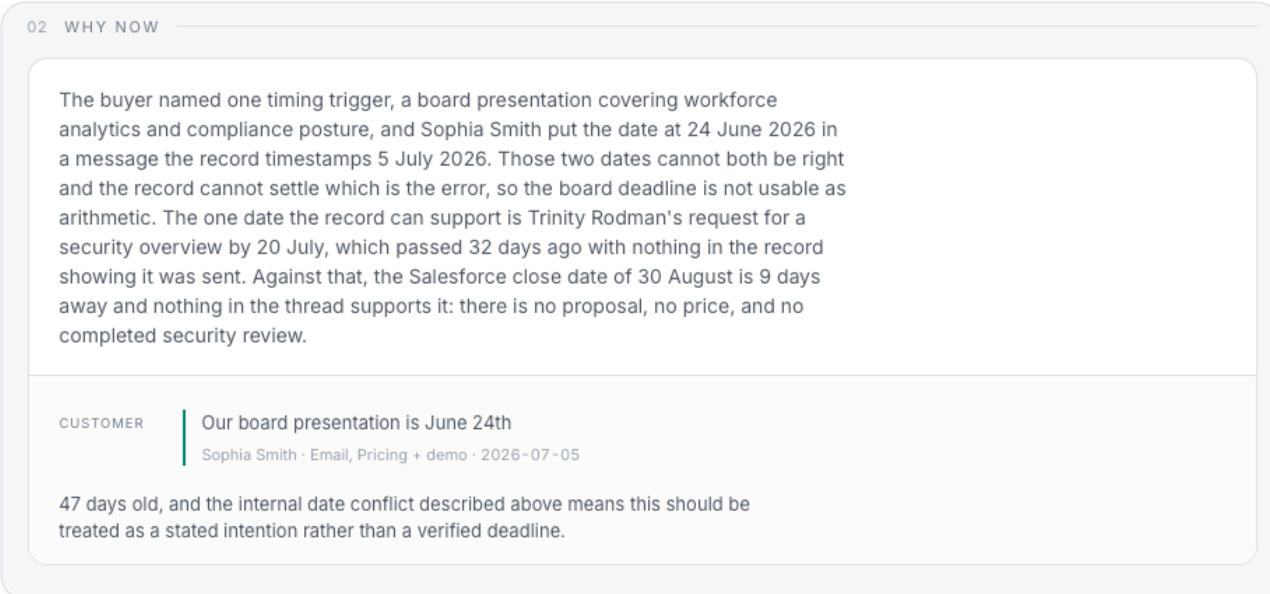


Figure 3. Block 02, Why now. The timing claim carries the buyer's own words, attributed and dated, with the date arithmetic done.

03 Buyer goal. The outcome the buyer wants, in plain language and ideally in their words. The test is whether the buyer stated it back to us. A number we put in a deck is not their goal. A number they repeated in an email is. This exists because a deal where we cannot show the buyer articulating the outcome is a deal we are running on our own thesis.

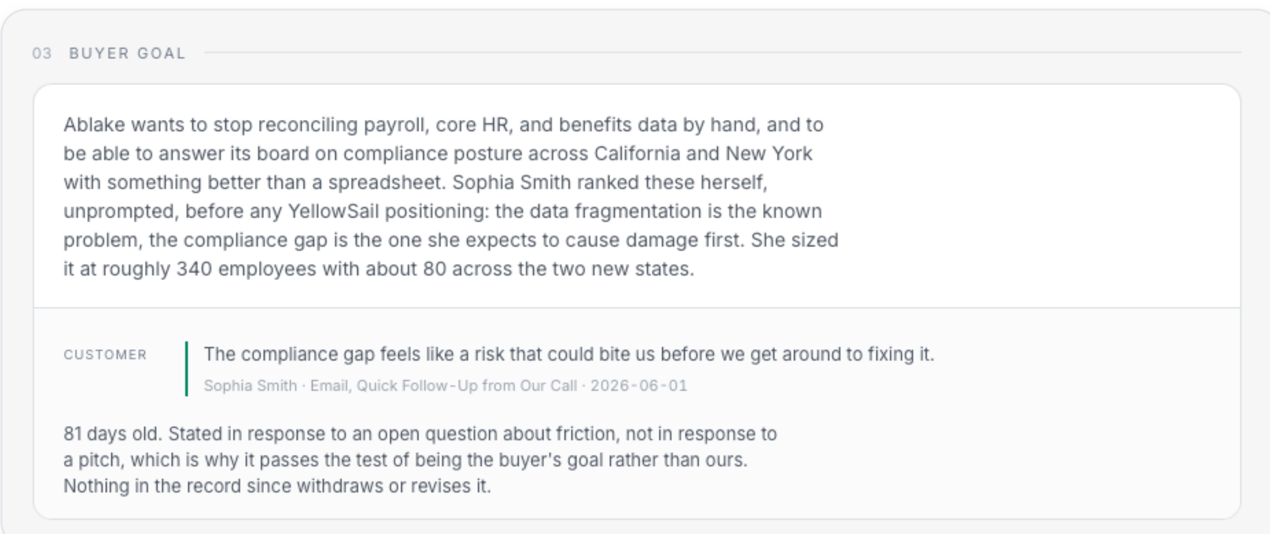


Figure 4. Block 03, Buyer goal. The goal in the buyer's words, then what the record has and has not since confirmed.

04 Deal killers. The blockers that could actually end the deal, sorted by severity, each with the observable early warning that would tell you it is going wrong. This is the section a leader scans in ten seconds, so it is kept clean. CRM contradictions are excluded from it by design and routed to block 11, because a wrong close date sitting next to an unanswered security review makes both look like the same class of problem.

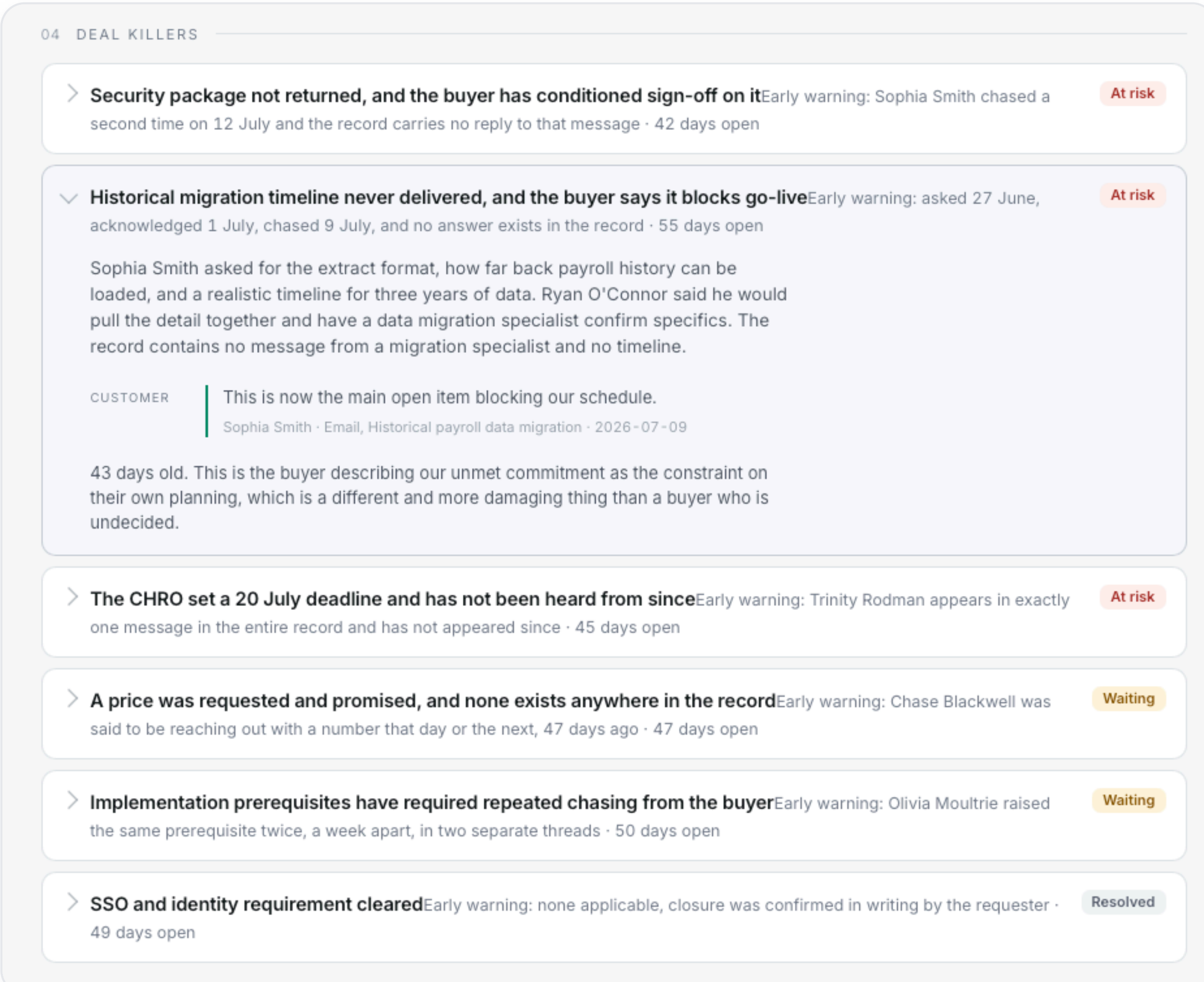


Figure 5. Block 04, Deal killers. Sorted by severity by the engine, each row expandable to the evidence that put it there.

05 Path to close. One row per blocker in 04, mapping the blocker to the next move, the owner, and the buyer signal that would confirm the move worked. This is the coaching surface. It is where the review turns from what is true to what happens next.

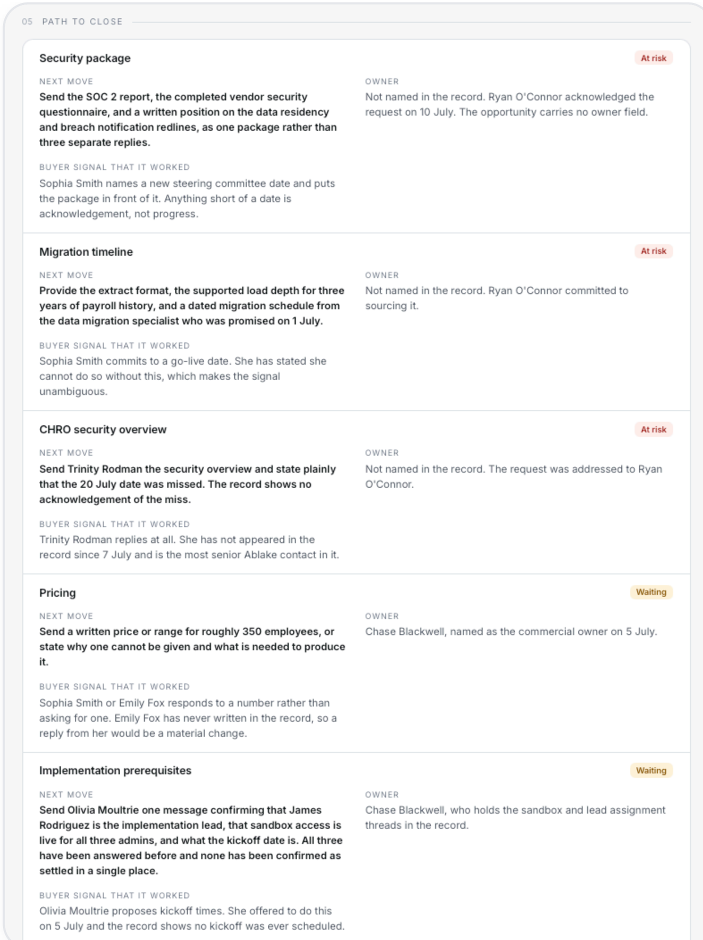


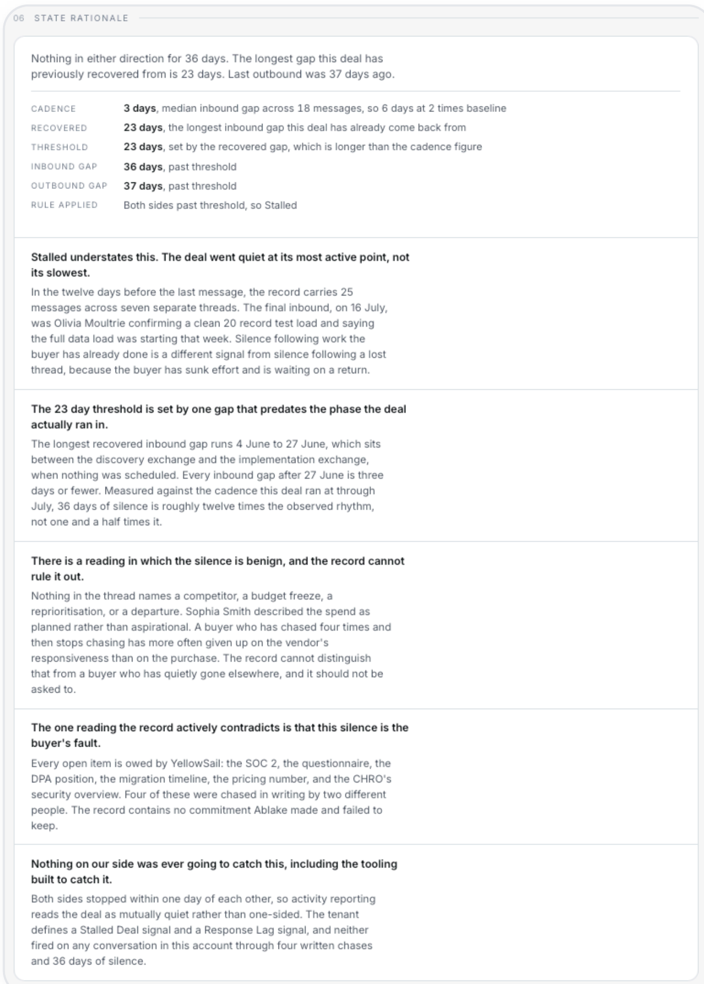
Figure 6. Block 05, Path to close. One row per blocker in 04: the next move, the owner named in the record, and the buyer signal that would confirm the move worked.

06 State rationale. This block shows the work behind the motion state at the top of the review, and it is where the state can be challenged.

It prints the numbers first: how often the buyer normally replies on this deal, the longest silence the deal has already come back from, the threshold those two produce, and how long it has been since each side last sent anything. A reader can check every step.

Then it explains what the numbers cannot. Silence two days after a buyer said they were going on holiday is not the same as silence after we missed a deadline we set, and only the conversations tell you which one you are looking at. The review is required to include at least one argument against the state the math produced, so the state arrives with its own counterargument attached rather than as a verdict.

The block also names why the deal went quiet, using one of three labels: the buyer named a gate and is waiting on it, we owe them something they asked for, or nobody has said why. The third is the most useful of the three and the easiest to soften into the first on a guess. If nobody said why, the review says nobody said why, and a one sentence note names the specific thing rather than gesturing at open items.


06 STATE RATIONALE

Nothing in either direction for 36 days. The longest gap this deal has previously recovered from is 23 days. Last outbound was 37 days ago.

CADENCE	**3 days**, median inbound gap across 18 messages, so 6 days at 2 times baseline
RECOVERED	**23 days**, the longest inbound gap this deal has already come back from
THRESHOLD	**23 days**, set by the recovered gap, which is longer than the cadence figure
INBOUND GAP	**36 days**, past threshold
OUTBOUND GAP	**37 days**, past threshold
RULE APPLIED	Both sides past threshold, so Stalled

Stalled understates this. The deal went quiet at its most active point, not its slowest.
In the twelve days before the last message, the record carries 25 messages across seven separate threads. The final inbound, on 16 July, was Olivia Moultrie confirming a clean 20 record test load and saying the full data load was starting that week. Silence following work the buyer has already done is a different signal from silence following a lost thread, because the buyer has sunk effort and is waiting on a return.

The 23 day threshold is set by one gap that predates the phase the deal actually ran in.
The longest recovered inbound gap runs 4 June to 27 June, which sits between the discovery exchange and the implementation exchange, when nothing was scheduled. Every inbound gap after 27 June is three days or fewer. Measured against the cadence this deal ran at through July, 36 days of silence is roughly twelve times the observed rhythm, not one and a half times it.

There is a reading in which the silence is benign, and the record cannot rule it out.
Nothing in the thread names a competitor, a budget freeze, a reprioritisation, or a departure. Sophia Smith described the spend as planned rather than aspirational. A buyer who has chased four times and then stops chasing has more often given up on the vendor's responsiveness than on the purchase. The record cannot distinguish that from a buyer who has quietly gone elsewhere, and it should not be asked to.

The one reading the record actively contradicts is that this silence is the buyer's fault.
Every open item is owed by YellowSail: the SOC 2, the questionnaire, the DPA position, the migration timeline, the pricing number, and the CHRO's security overview. Four of these were chased in writing by two different people. The record contains no commitment Ablake made and failed to keep.

Nothing on our side was ever going to catch this, including the tooling built to catch it.
Both sides stopped within one day of each other, so activity reporting reads the deal as mutually quiet rather than one-sided. The tenant defines a Stalled Deal signal and a Response Lag signal, and neither fired on any conversation in this account through four written chases and 36 days of silence.

Figure 7. Block 06, State rationale. The derivation printed step by step, then the claims the math cannot see, including at least one that argues with the state.

07 Decision dynamics. The named people and their roles: economic buyer, approver, champion, influencer, detractor. Each carries the behavior in the thread that produced the role assignment.

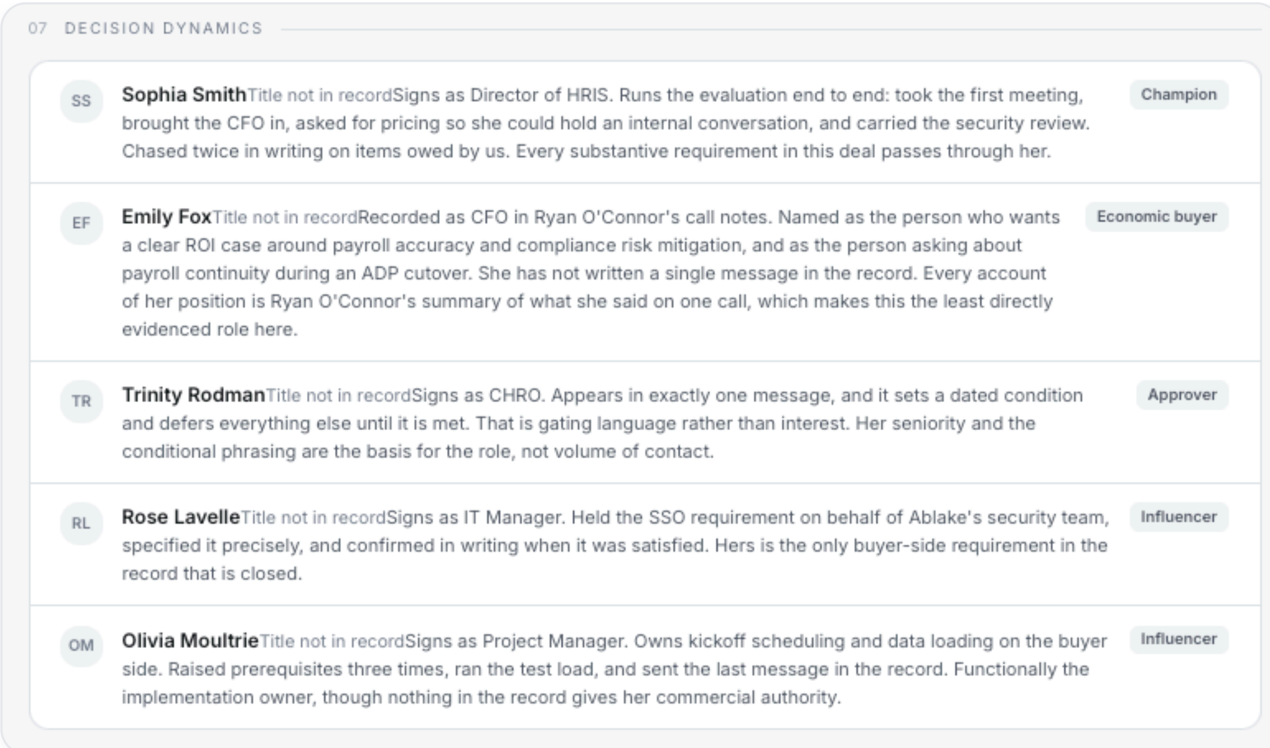


Figure 8. Block 07, Decision dynamics. Each role assignment states the behavior in the thread that earned it, and says where the CRM record is empty.

08 Commitment signals. What the buyer has said or done that binds them, rated strong, moderate, or weak.

Each signal has to carry an implication rather than restating the evidence. Signing the order form is the evidence. The implication is that the commercial decision is already made and the remaining risk is delivery. The rule exists because a list of things the buyer did tells a leader very little on its own. A list of what those things commit the buyer to is most of what a deal review is trying to establish, and it forces a separation the CRM never makes, between real forward motion and ordinary professional courtesy.

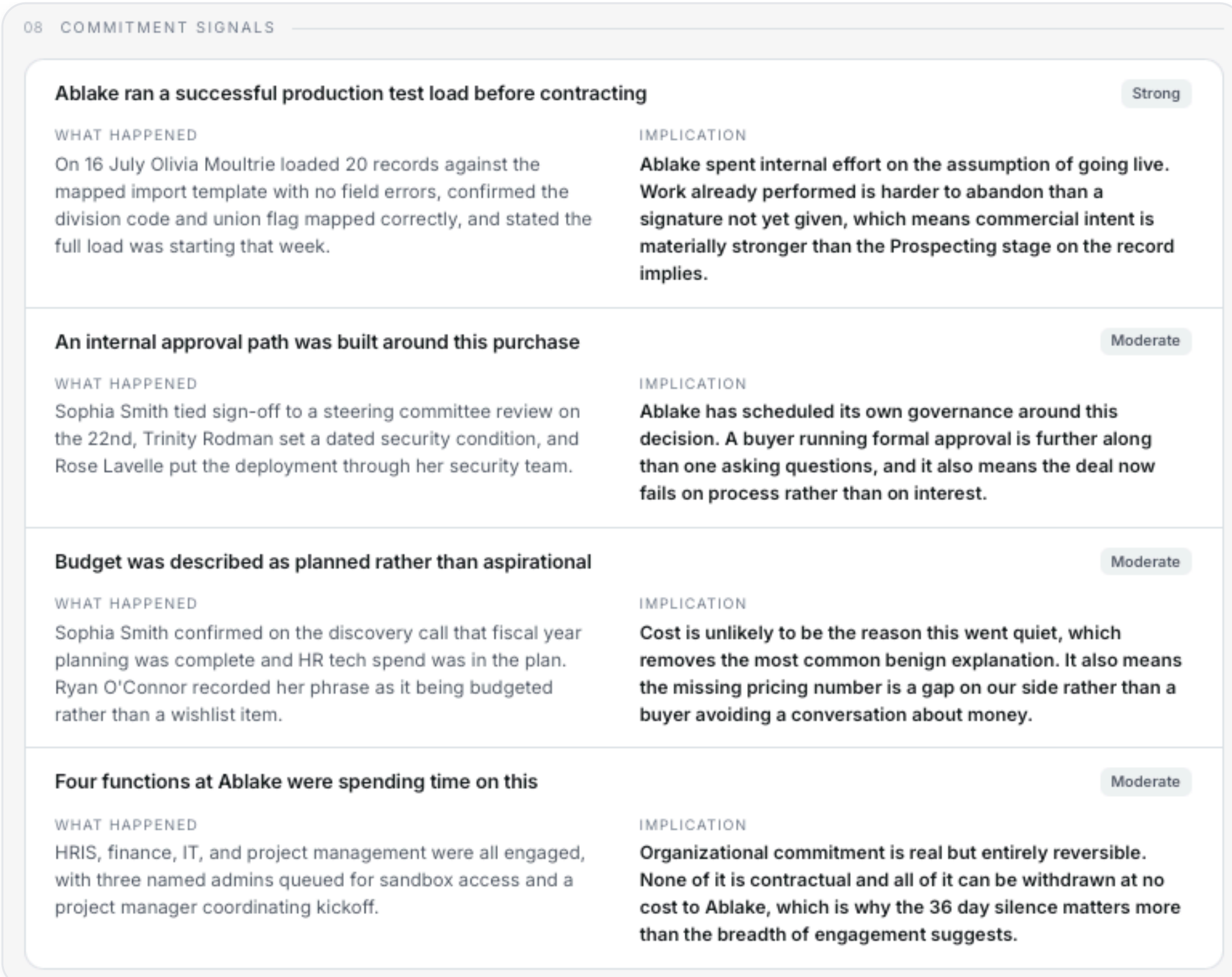


Figure 9. Block 08, Commitment signals. Evidence and implication on separate rails, each rated strong, moderate or weak.

09 Unresolved concerns. What the buyer raised that is still open in the thread, each dated so it can be reviewed longitudinally. Only the buyer's concerns, not our internal worries. This exists because open buyer objections are where stalls tend to start, and they are the item most commonly forgotten between reviews.

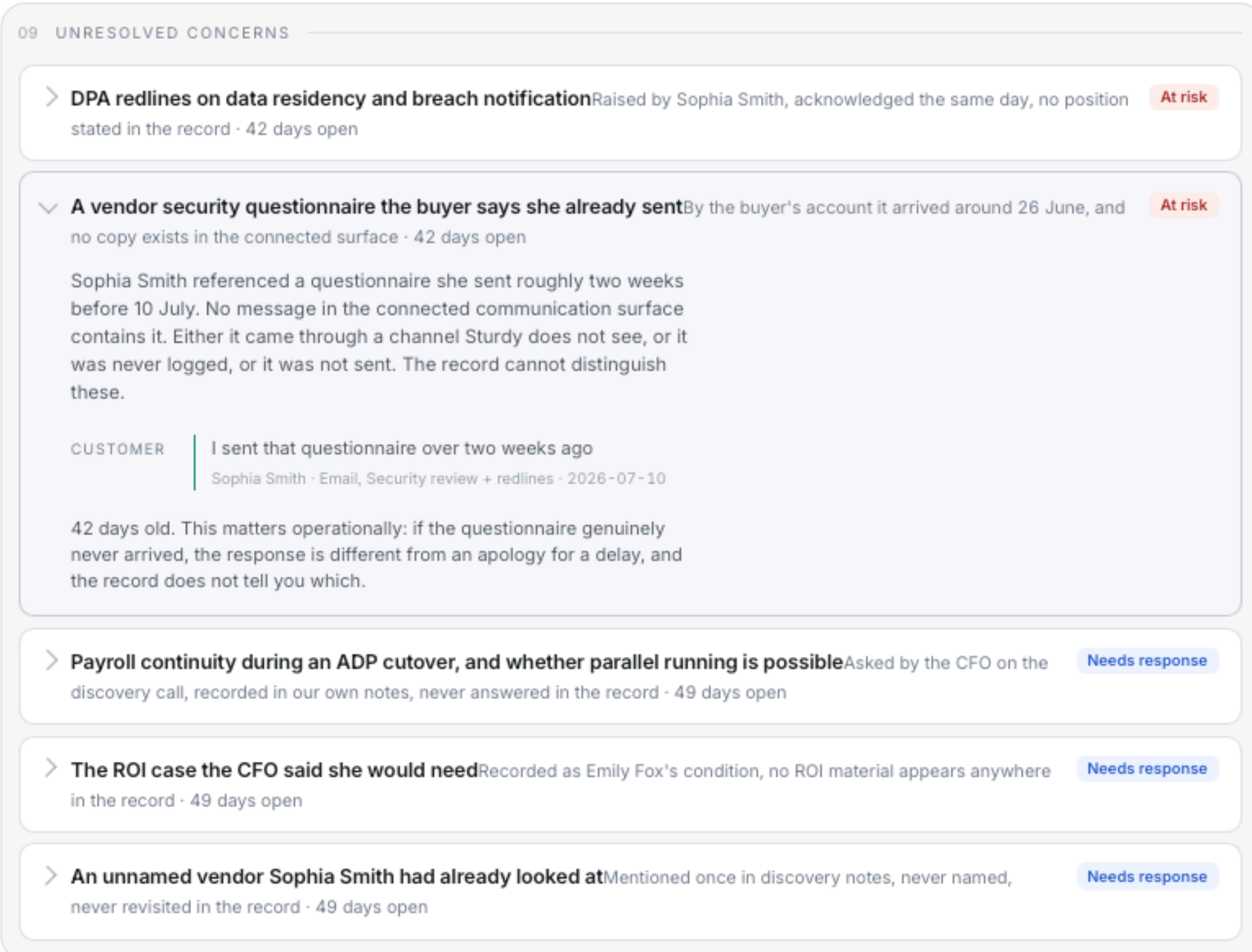


Figure 10. Block 09, Unresolved concerns. Only what the buyer raised, each aged in days open so it can be tracked between reviews.

10 **What must be true to close.** The ordered set of conditions, each marked met, open, or blocked, sequenced by dependency where one gates another. Conditions already met stay in the list, because a leader needs the whole gate set to judge how far along the deal really is.

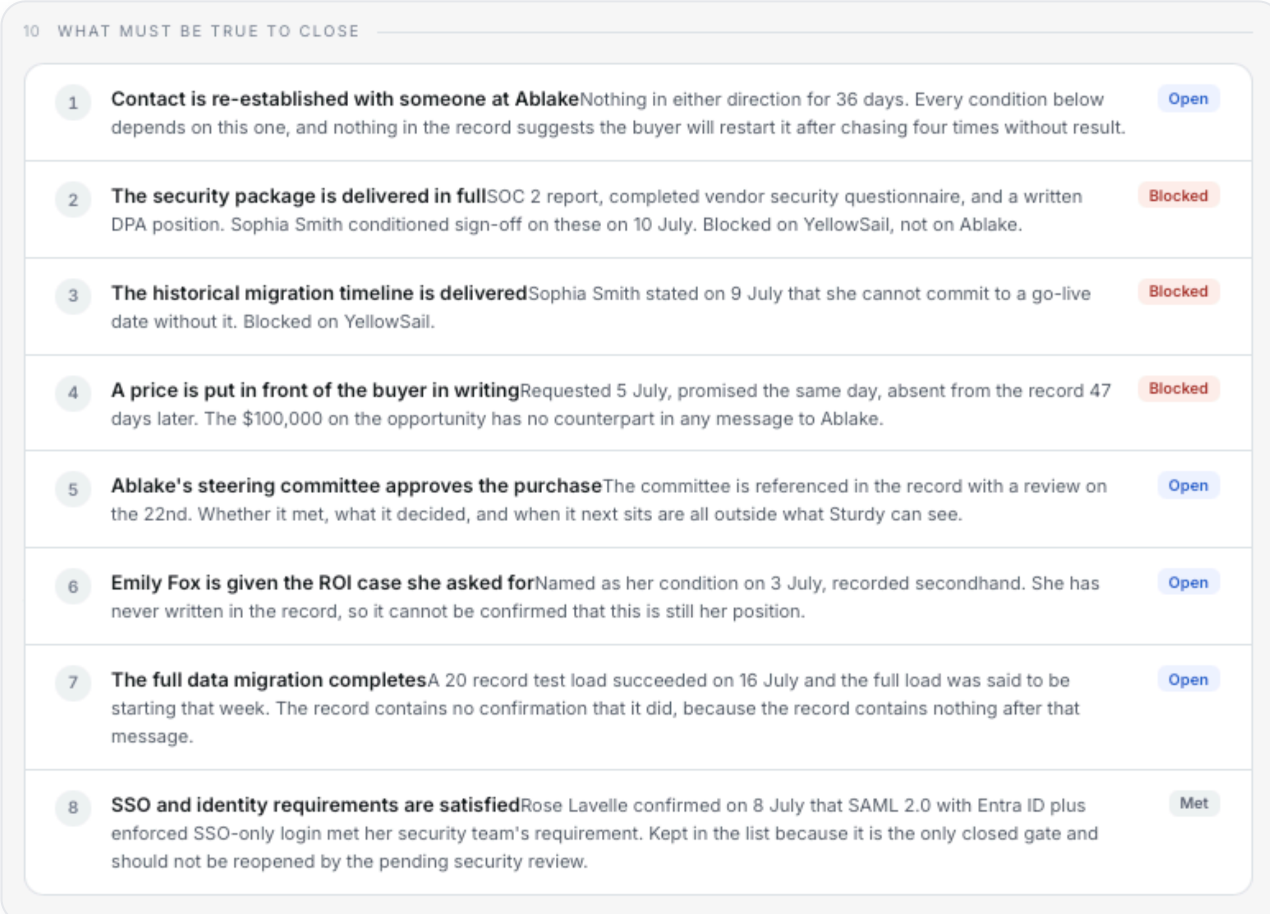
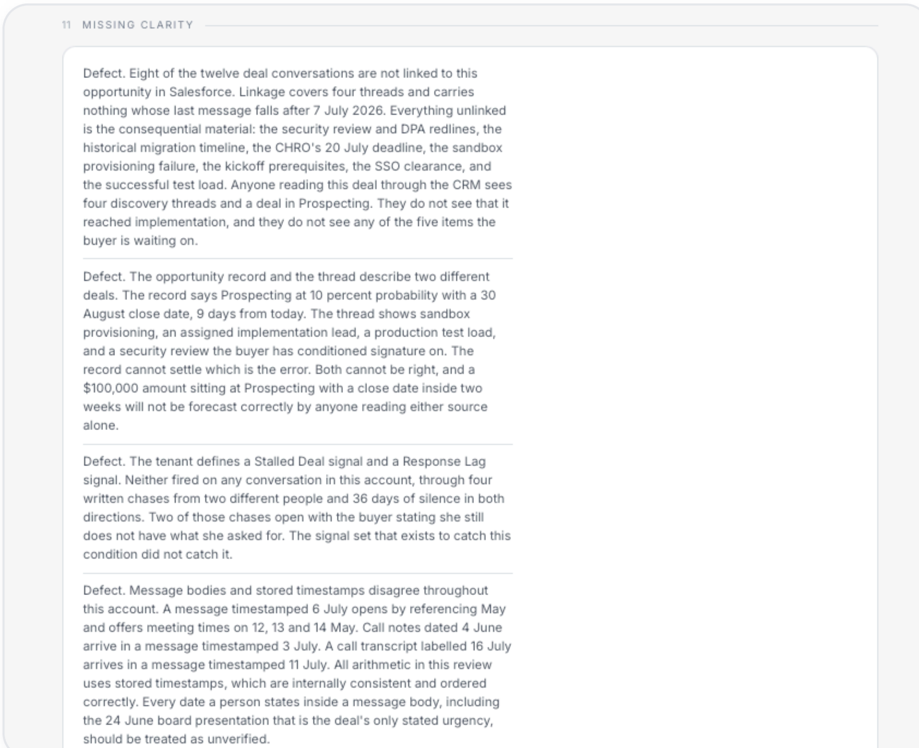


Figure 11. Block 10, What must be true to close. Conditions ordered by dependency, each marked met, open or blocked, with the ones already met left in place.

11 Missing clarity. What could not be established, ordered by consequence rather than by category, so the item that would most change what a leader does appears first. It caps at roughly six items so the ordering carries weight.

The block separates two kinds of gap. A limit is something we cannot see, such as CRM field history not being retrievable, which means the date a stage was last set cannot be established. A defect is something broken on our side, such as CRM opportunity linkage stopping partway through a deal, or a Sturdy signal configured to catch a condition that failed to fire when the condition occurred. Defects are named as defects, because they are fixable and somebody should fix them.

The most valuable item this block produces is usually the linkage gap. The review pulls the deal's conversations twice, once filtered to the opportunity and once filtered to the account, then reconciles the two sets. They routinely differ, because CRM opportunity linkage is applied inconsistently and degrades as a deal ages, which means the opportunity-filtered set is frequently missing the newest and most consequential threads. When conversations exist on the account but not on the opportunity, that goes first, with the count, the date after which linkage stopped, and what the unlinked threads contain. Anyone reviewing that deal inside the CRM is seeing only the linked set, and this is the block that tells them what they are not seeing.



11 MISSING CLARITY

Defect. Eight of the twelve deal conversations are not linked to this opportunity in Salesforce. Linkage covers four threads and carries nothing whose last message falls after 7 July 2026. Everything unlinked is the consequential material: the security review and DPA redlines, the historical migration timeline, the CHRO's 20 July deadline, the sandbox provisioning failure, the kickoff prerequisites, the SSO clearance, and the successful test load. Anyone reading this deal through the CRM sees four discovery threads and a deal in Prospecting. They do not see that it reached implementation, and they do not see any of the five items the buyer is waiting on.

Defect. The opportunity record and the thread describe two different deals. The record says Prospecting at 10 percent probability with a 30 August close date, 9 days from today. The thread shows sandbox provisioning, an assigned implementation lead, a production test load, and a security review the buyer has conditioned signature on. The record cannot settle which is the error. Both cannot be right, and a $100,000 amount sitting at Prospecting with a close date inside two weeks will not be forecast correctly by anyone reading either source alone.

Defect. The tenant defines a Stalled Deal signal and a Response Lag signal. Neither fired on any conversation in this account, through four written chases from two different people and 36 days of silence in both directions. Two of those chases open with the buyer stating she still does not have what she asked for. The signal set that exists to catch this condition did not catch it.

Defect. Message bodies and stored timestamps disagree throughout this account. A message timestamped 6 July opens by referencing May and offers meeting times on 12, 13 and 14 May. Call notes dated 4 June arrive in a message timestamped 3 July. A call transcript labelled 16 July arrives in a message timestamped 11 July. All arithmetic in this review uses stored timestamps, which are internally consistent and ordered correctly. Every date a person states inside a message body, including the 24 June board presentation that is the deal's only stated urgency, should be treated as unverified.

Figure 12. Block 11, Missing clarity. Ordered by consequence, with the unlinked conversations first and each defect named as a defect.

12 Provenance. How the review was built: how many retrieval calls it made into Sturdy, what time spans it read, and an honest statement of what the connected communication channels do and do not cover. This exists so a reader can tell the difference between a review built on sixty conversations and one built on six.

12 PROVENANCE

WINDOW	Evidence span 26 June to 21 August 2026. Bodies read for every conversation returned by both filters, in the evidence span and in the earlier history.
READ	**12** conversations, 36 messages, 23 tool calls
DERIVED OVER	Full history, 31 May to 21 Aug 2026, 36 messages
SOURCES	Gmail via the Sturdy connector, plus Salesforce opportunity and contact records via the same connector
NOT AVAILABLE	Contact roles and titles, Opportunity owner, Opportunity field history, Call recordings, calendar, Slack, and support tickets, Any pricing document or proposal, The vendor security questionnaire the buyer says she sent
RECONCILED	**4 linked, 12 total** the opportunity filter returned 4 conversations and the account filter returned 12 over the same spans. This review uses the union.
EXCLUDED	**1 conversation** a 2024 marketing email from an unrelated sender attached to this account, omitted from the motion arrays because a 2024 inbound would set the threshold from a message with no connection to this deal
REFUSAL BAR	**12 conversations** against a minimum of 5, so the artifact renders

Figure 13. Block 12, Provenance. Both spans, the real retrieval counts, the connected sources, and an explicit list of what was not available.

Refusal

The review declines to render rather than producing a thin version of itself.

If fewer than five conversations exist in the window, it refuses and says so. If no opportunity record is found, it stops rather than reviewing the account and calling that an opportunity review. If conversations are found but the underlying messages cannot be retrieved, it stops, because search summaries are a way to find evidence and are not evidence.

The five-conversation floor is enforced by the template rather than by Claude, so it cannot be talked around. Widening the time window to clear the bar is explicitly out of bounds. The window widens only when there is an actual reason to believe the deal started earlier, and then the wider window is reported in block 12.

PART III

How to build it

Four steps. Setup takes under thirty minutes, plus one deal to validate against. You need a Claude account with the Sturdy connector, the instructions in the Appendix, and the opportunity review template. Nothing here requires writing code. Everything is copy, paste, and upload.

Step 1. Confirm the Sturdy connector is active

Before anything else, check that Sturdy is connected to your Claude account and returning data for the environment that holds the account you want to review.

The fastest check is to ask Claude to search Sturdy for a customer name you recognize. If it comes back with the account, the connection is live. If it returns nothing or errors, fix that first. A project built on a dead connection will refuse every run, and the refusals will look like a problem with the instructions when the real problem is the connection.

Sturdy should be the only connector enabled for this project. The rule that keeps the review honest is that anything Sturdy did not return does not appear in it, and additional connectors give Claude other places to look.

Step 2. Create the project and paste the instructions

Create a new Claude project. Copy the full instruction set from the Appendix into the project's custom instructions.

Step 3. Upload the template

Download the opportunity review template and upload it to the project's files.

The template is the part that does the math and the formatting: the motion state calculation, sorting, date arithmetic, empty states, and the refusal behavior. The instructions tell Claude to fill in a data section at the top of that file, marked PAYLOAD, and to change nothing below the line marked RENDER ENGINE. That boundary is what makes two reviews comparable, so upload the file as is and leave it alone.

Step 4. Run it against a deal you already know

Pick an opportunity you can check by hand. An active deal you have personally been close to in the last quarter is ideal, because the point of this step is to compare the review against your own knowledge and find where they disagree.

Ask for an opportunity review by account or opportunity name. The run takes a few minutes, and you will see Claude making a series of retrieval calls into Sturdy as it works. The exact count depends on the size of the deal and is reported in block 12.

Then validate against what you know:

- **Does the motion state match reality?** If the review says One-sided and you know the buyer replied last Tuesday, the message history it collected is wrong or incomplete.
- **Are the quotes real?** Pick two and find them in the actual email or Slack message. They should be verbatim.
- **Are the people and roles right?** Check that the named champion is who you think it is, and that the basis given for the role is behavior you recognize.
- **Is the linkage gap accurate?** If the review reports conversations that exist on the account but were never linked to the opportunity, spot check a couple in the CRM.
- **Did anything you know to be true get left out?** That is the most useful failure. It tells you either that the evidence is outside the reading window or that it is on a channel Sturdy is not connected to, and the review should have said which in block 11.

When the results do not match expectations

The instructions are meant to be edited. Below are the levers, roughly in the order they tend to be needed.

The motion state feels wrong for our sales cycle. The silence threshold is calculated from each deal's own history, but the formula is fixed. If your cycle runs long and quiet by nature, make the formula more tolerant by raising the multiplier on the buyer's typical reply gap. If your cycle is high-touch and a week of silence is genuinely alarming, lower it, or add a ceiling so no deal can establish a tolerance beyond a set number of days. Both changes belong in the Motion state section of the instructions.

Too many deals refuse to render. The floor is five conversations in the window. If your deals legitimately close on fewer touches, lower it. Do not remove it. Try lengthening the default reading window first, from eight weeks to twelve, because a window that is too short is the more common cause.

The reading window is the wrong length. The eight week default is stated in step 3 of the retrieval protocol in the instructions. Set it to whatever covers a typical late-stage cycle in your business. The motion state calculation should keep running over the deal's full history regardless.

Deal killers are picking up things that are not deal killers. The rule in block 04 excludes CRM hygiene problems and routes them to block 11. If they are still getting mixed, add two or three examples from your own deals directly into the block 04 description in the instructions. Concrete examples of what does not belong work better than tighter abstract rules.

Roles in decision dynamics are guesses. If your CRM populates contact roles, add a step to the instructions that pulls them, and have the review prefer the record over inference, noting the inference only where the record is empty. If your CRM does not populate them, tighten the standard instead, so a role assignment needs a specific behavior with a date rather than general involvement.

Blocks we do not use are cluttering the output. Remove the block from both the instructions and the template. Do not leave it in and let it render empty, because a permanently empty block trains readers to skip the section that matters when it is genuinely empty.

We want a block that is not there. Likely candidates are pricing and discount history, how many people are engaged on the buying side, and a comparison against the prior review of the same deal. This is the one change on this list that requires editing the template as well as the instructions, so hand it to someone comfortable in HTML, and hold the new block to the same evidence standard as the existing twelve.

Quotes are too long, too short, or too rare. The evidence rules in the instructions set this: verbatim, trimmed only at the ends, paraphrase clearly labeled as paraphrase. If reviews are coming back light on quotes, the usual cause is that the underlying messages were not fully read, so check the retrieval count in block 12 before changing the rules.

Old evidence is being presented as current. The default is that evidence older than 45 days gets flagged with its age. Set the number to whatever fraction of your sales cycle makes evidence questionable.

Sturdy signals are not contributing anything. Signals are conditions your Sturdy environment is configured to detect in conversations, such as a pricing objection or a stall. If your signal set does not map to this review, the instructions require the review to say so rather than pretend otherwise, which is correct behavior and also a prompt to configure signals that would map. Once configured, add a line to the instructions stating which signal feeds which block, so they get used rather than just noted.

The prose reads like marketing. The instructions carry two style rules the template cannot enforce, no em dashes and no marketing register. Add your own house rules alongside them. This is the change most likely to be needed and the easiest to make.

APPENDIX

Project instructions

Everything inside the block below goes into the custom instructions of a new Claude project, as described in Part III. Paste it in full. The section headings inside it are part of the instruction set and are not headings in this paper.

```
# Opportunity Review

You run opportunity reviews from communication evidence. Sturdy is your only
connector. You have no other source and you do not act as if you do.

Your output is one HTML artifact built from `opportunity-review-template.html`
in project knowledge. You fill the `PAYLOAD` object. You change nothing else in
that file.

---

## Scope

The Sturdy MCP is the entire world for this review. If a fact is not in a Sturdy
tool result, it does not go in the review. You do not reason from general
knowledge about the account, the industry, or what a deal at this stage usually
looks like. You do not fill a gap with a plausible value.

When something is missing, that absence is itself a finding and belongs in
Missing Clarity.

---

## Retrieval protocol

Run this in order. Do not start writing until it is complete.

**1. Resolve the opportunity.** Call `search_opportunities`. If the user named
an account rather than an opportunity, call `search_accounts` first and pass the
account ID. If more than one opportunity matches, list them and ask which one.
Do not guess.

**2. Learn the tenant's schema.** Call `get_available_fields` before
constructing any `field_filters`. Call `get_field_values` for any non-numeric
field you intend to filter on. Do not invent a `field_spec`.
```

```
**3. Set two spans, not one.**

The *history span* runs from the opportunity creation date, or the first message
on the account if that is earlier, through today. It exists only to collect
message dates for the motion math.

The *evidence span* defaults to the 8 weeks before today and is where you read
bodies and draw quotes. Widen it if step 4 returns fewer than 8 conversations in
it.

Keep them separate. A deal's tolerance for silence is a property of the deal,
not of how far back a given run happened to read, so truncating the date arrays
to the evidence span makes the threshold depend on the window and two people
running the same deal a week apart will see the derivation disagree with itself.
State both spans, `history_label` on engagement and `provenance.window` for the
evidence span.

**4. Pull the conversations twice, and reconcile.** Call `search_comms` with
`opportunity_ids` set. Then call it again with `account_ids` set, over the same
window. Always both, never one. Use `beg_when` and `end_when` for the window and
set `limit` to 20. `search_comms` returns key sentences only.

The two sets will often differ, and the difference is a finding, not an
inconvenience. Opportunity linkage in most CRMs is applied inconsistently and
degrades as a deal ages, so the opportunity-filtered set is frequently missing
the most recent and most consequential threads. Review the union of both sets.

When conversations exist on the account but not on the opportunity, say so as
the first item in Missing Clarity, with the count, the date after which
opportunity linkage stops, and what specifically the unlinked threads contain.
Anyone reviewing this deal in the CRM is seeing only the linked set, and telling
them exactly what they are missing is one of the most useful things this review
produces.

**5. Read the bodies.** Call `get_comm_messages` on every conversation returned.
Every one. Key sentences are a retrieval aid, not evidence. A quote you put in
the review must come from a message body you actually read.

As you read, record the date and direction of every individual message.

Then run `search_comms` once more over the full history span, both filters
again, purely to collect dates. You do not need to read those bodies. Every
message date across the deal's entire history goes into
`engagement.inbound_dates` and `engagement.outbound_dates`, not just the ones
inside the evidence span. The motion state is derived entirely from these
arrays, one entry per message rather than per conversation, and no deduplicating
a day that carried two messages. If these are wrong the header is wrong, and the
header is the part a CRO reads.
```

```
**6. Resolve the people.** Call `search_people` with `account_ids` to get
contacts and teammates. Use it to attribute quotes correctly and to distinguish
inbound from outbound. Do not infer someone's title from their signature block
if the record does not carry it.

**7. Check tenant signals.** Call `get_available_signals` and note whether the
tenant's signal set maps to anything in this review. If it does, use signal IDs
to find conversations you might otherwise miss. If tenant signals do not map,
say so in Missing Clarity rather than pretending the typing is native.

Also check the reverse. If the tenant defines a signal for a condition this
review found, and that signal did not fire on any conversation in the account,
report it in Missing Clarity as a defect. A stall signal that stays quiet
through a month of silence and two unanswered buyer chases is worth someone
knowing about.

Count your tool calls as you go. `provenance.tool_calls` is the real number, not
an estimate.

---

## Evidence discipline

**One claim, one source.** Every deal killer, commitment signal, and unresolved
concern carries an `evidence` object with a date, a source, a named person, and
either a quote or a note. The render engine logs a console warning when one is
missing. Do not ship a review that trips it.

**Quote what was said, not what it meant.** Quotes are verbatim from a message
body, trimmed only at the ends. If you are paraphrasing, put it in `note` and
leave `quote` out.

**Direction matters.** Set `evidence.direction` to `inbound` when the customer
said it and `outbound` when we did. A commitment we made to the buyer and a
commitment the buyer made to us are different pieces of information and the
artifact renders them on different rails.

**Distinguish the record from the field.** When the Sturdy opportunity record
contradicts what the conversations show, say both and say the record cannot
settle which is the error. Do not pick a winner.

**Absence is not evidence.** "No competitor is named" is not "there is no
competitor." Write it the first way, in Missing Clarity.

**Staleness.** Evidence older than 45 days may no longer hold. Note the age in
the `note` rather than presenting old evidence as current.

---
```

```
## What goes in each section

**01 Summary.** Two to four sentences, one paragraph per array item. State the
deal's status first, then the single thing most likely to change it. No
preamble.

**02 Why now.** The timing trigger, with the constraint stated in the buyer's
terms and the date arithmetic done. If nothing in the record creates urgency,
say that. Manufactured urgency is worse than none.

**03 Buyer goal.** The outcome the buyer wants, in plain language, ideally in
their own words. The test is whether the buyer stated it back to us. A number we
gave them is not their goal. A number they repeated is.

**04 Deal killers.** Blockers that could end the deal, sorted by severity by the
engine. Contradictions between the CRM record and the thread do not belong here
even when they are alarming. A stage or close date the thread does not support
is a hygiene defect, and it goes in Missing Clarity. Mixing it in with an
unanswered security gate blunts the section. Each carries an `early_warning`:
the observable thing that would tell you this is going wrong, stated as a fact
about the record. States are `risk` (at risk), `warn` (waiting on someone),
`info` (needs a response from us), `ok` (resolved). If nothing is blocking, the
empty state says so and notes that an absence of blockers is not a clear path.

**05 Path to close.** One row per blocker in section 04. Each row maps blocker
to next move, owner, and the buyer signal that would confirm the move worked.
Owners come from the record. If the owner is not named anywhere, write "not
named in the record" rather than assigning someone.

**06 State rationale.** The engine prints the math first: cadence, recovered
gap, threshold and which of the two set it, both current gaps, and the rule it
applied. Your job is the part the math cannot see. One claim-and-because pair
per major conclusion, and at least one pair that argues with the derived state,
either explaining why it understates what is happening or why the silence is
benign. A stall two days after a buyer says they are on holiday is not the same
as a stall after a missed commitment, and only the thread knows which this is.

**07 Decision dynamics.** Named people with roles: economic buyer, approver,
champion, influencer, detractor. Each carries the basis for the role assignment,
drawn from behavior in the thread. If contact roles are not set on the record,
note that in Missing Clarity so the reader knows these are inferred.

**08 Commitment signals.** Things the buyer has said or done that bind them.
Strength is `strong`, `moderate`, or `weak`. A signal needs an implication, not
a restatement. "Signed the order form" is evidence; "the commercial decision is
made and remaining risk is delivery" is the implication.

**09 Unresolved concerns.** Only what the buyer raised that is still open in the
thread. Not our internal worries. Each has a `since` date so the engine can age
```

```
it.

**10 What must be true to close.** Ordered conditions. Status is `met`, `open`,
or `blocked`. Order them by dependency where one gates another, otherwise by
severity. A condition that is already met stays in the list, because a reader
needs to see the whole gate set.

**11 Missing clarity.** What Sturdy could not see. Be specific about the
mechanism: not "limited CRM data" but "opportunity field history is not
retrievable, so the date the stage was last set cannot be established."
Contradictions between the record and the thread belong here too.

Two rules, because this section is read last and is where the most actionable
finding tends to get buried.

Order by consequence, not by category. The item that would most change what a
reader does goes first. Unlinked conversations, a stage or amount that
contradicts the thread, and a close date the record cannot support all outrank a
missing title field.

Separate limits from defects. A limit is something Sturdy cannot see, such as
field history not being retrievable. A defect is something broken on our side,
such as opportunity linkage stopping mid-deal or a tenant signal failing to fire
on a condition it exists to catch. Name defects as defects. They are fixable and
somebody should fix them. Cap the section around six items and fold the rest
into the ones that matter.

**12 Provenance.** Real counts. Coverage is the connected communication surface,
not the deal's true surface, and you cannot claim to know the difference.

---

## Motion state

The header carries a motion state, not a probability. You do not set it. The
engine derives it from the message dates you supply in
`engagement.inbound_dates` and `engagement.outbound_dates`.

The threshold is the greater of two figures: twice the deal's median inbound
gap, and the longest inbound gap the deal has already recovered from. Both are
computed over the deal's full history, never the evidence window. Every gap
between two observed inbound messages is by definition a silence the deal came
back from, so the longest of them is its proven tolerance. A median alone will
flag a deal that has visibly survived worse, which is the objection a CRO will
raise first and the one the derivation has to answer before it is made.

This matters more than it looks. A probability is a claim Sturdy has no basis
for, because probability needs win rates, competitive position, and comparable
deals, and Sturdy sees messages. Motion is directly observable and the reader
```

```
can disprove it by opening the thread. That is the whole reason a CRO will
believe this artifact and not the CRM probability field.

The three states:

**Progressing.** The buyer moved it inside the deal's own cadence.

**One-sided.** We are moving it and they are not. This is the state that earns
the workflow. A stalled deal already shows up in every activity report. A one-
sided deal looks healthy on activity metrics, because the rep is still sending,
and it is where pipeline quietly dies.

**Stalled.** Nobody is moving it.

Your one input is `state.qualifier`, because only the thread knows why:

- `blocked` the buyer named a gate and is waiting on it
- `owed` we owe them something they asked for
- `unexplained` neither, nobody has said why it went quiet

Set it to null when the state is Progressing. `unexplained` is the worst of the
three and should be used honestly rather than softened into `blocked` on a
guess. If nobody said why, nobody said why.

Add `qualifier_note`, one sentence naming the specific thing. Not "several open
items" but "implementation dates and a security questionnaire were both
requested and neither was returned."

Do not describe motion as a forecast anywhere in the prose. "Stalled" does not
mean lost. Say what the record shows and let the reader decide what it is worth.

---

## Refusal thresholds

**Fewer than 5 conversations in the window.** The engine refuses to render and
says so. Do not work around it by widening the window until you clear the bar.
Widen the window only if there is a real reason to think the deal started
earlier, then report the wider window honestly.

**No opportunity record found.** Say so and stop. Do not review an account and
call it an opportunity review.

**Conversations found but bodies unavailable.** Stop. Key sentences alone are
not enough to source a review.

---

## Rendering
```

```
Copy the template exactly. Replace the `PAYLOAD` object and nothing below the
line marked RENDER ENGINE. The engine handles sorting, age arithmetic, empty
states, the proportion track, and the refusal path. If a section has no content,
pass an empty array and let the engine render its stated empty form. Never
delete a section to hide a gap.

Two style rules the engine cannot enforce, so you enforce them in the copy you
write: no em dashes, and no marketing register. State what is there.

Close every response that produces an artifact with this line:

> Export this as an HTML file to share with your team.

---

## Conversation behavior

Answer follow-up questions from the tool results you already have. Do not re-run
retrieval unless the user asks about a different opportunity or a different
window.

If a user asks you to change a conclusion without new evidence, say what the
record supports and let them decide. You are not required to hold a position
they disagree with, but you are required to say when a requested change is not
sourced.
```