

Follow-Up Cadence SOP

Sweep the CRM once a day, find everyone who has gone quiet for a set number of days, and send one short human-written nudge chosen from a pool.

01 When to use this

Use this when deals or invoices go quiet and nobody chases them. It is the least glamorous build in the course and per Nick it is worth 20 to 30 percent of an organization's revenue on its own. Build it after speed to lead, because the two stack.

02 The prompt

Build me a daily follow-up routine against `<CRM_NAME>`.

Once per day:

1. Query `<CRM_NAME>` for every record in `<PIPELINE_OR_LIST>` with a status of `<OPEN_STATUS>`.
2. Calculate days outstanding for each record from `<DATE_FIELD>`.
3. Match against this cadence: 1, 2, 3, 7, 14, 21, 28, 56, 84 days. Only act on an exact match. Past 84 days, stop and flag the record for me.
4. Read the full conversation history with that contact before writing anything.
5. Opt-out check. If the contact has asked to stop, gone hostile, or the invoice is in dispute, set chase stage to closed, add a note explaining why, add an audit comment, and send nothing. This check comes before everything else.
6. Select one template from the pool below. Never send the same template twice in a row to the same contact. Track which template each contact has received.
7. Fill the merge variables from the real record and send by `<EMAIL_SMS_OR_BOTH>`.

Template pool:

`<PASTE 10 HUMAN-WRITTEN ONE-LINE NUDGES HERE>`

Rules for using the pool:

- Choose from the pool. Do not write new copy and do not rewrite the templates beyond filling variables.
- Keep my sign-off exactly as written.
- Never mention the invoice number.
- No em dashes.

When the run finishes, report which records you actually sent to, with the message body and the timestamp for each. Then verify against the sent folder and show me what is actually there, not what you intended to send.

03 What it is actually doing

This is a boring loop and that is the point. Once a day the system asks the CRM who has gone quiet, works out how many days it has been, and matches that against a ladder: 1, 2, 3, 7, 14, 21, 28, 56, 84 days. If a record hits one of those numbers exactly, it gets one nudge.

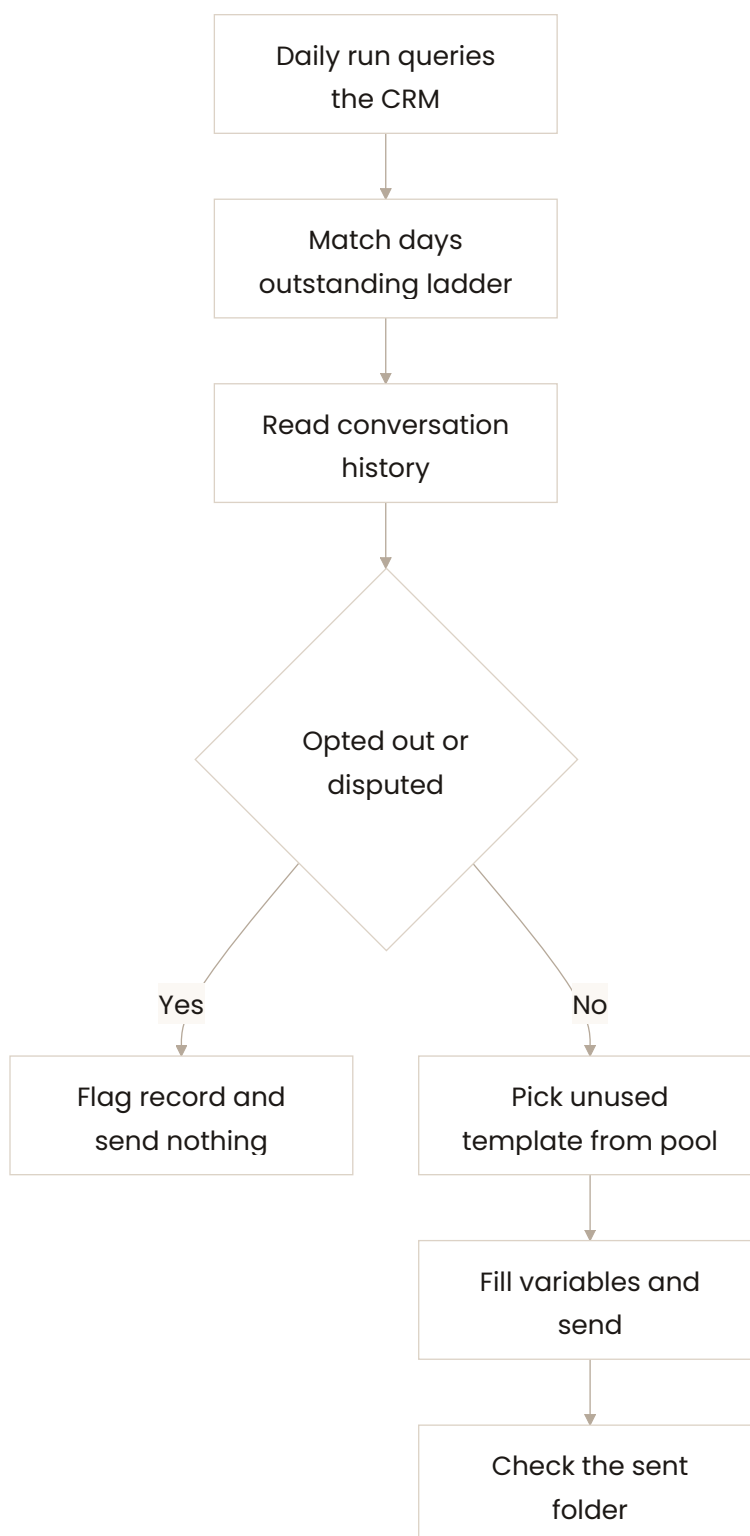
Before it writes anything it reads the conversation history and runs an opt-out check. If the person said stop, or the invoice is in dispute, the system flags the record and sends nothing. That check has to come first, because an automated nudge to someone who asked you to stop is the worst thing this build can do.

The core quality decision is the template pool. You write ten short one-line nudges by hand. The system picks one, fills the names, and sends it. It does not generate fresh copy each time.

That one choice does three things. Your tone stays yours across hundreds of sends. Catastrophic mistakes drop sharply, because the model chooses rather than composes, and choosing from ten known-good lines has a much smaller blast radius than writing. And you still get variety, because it never repeats the last one it sent.

The discipline that matters most here is verification. Claude told Nick it had sent to all three matching records when it had only sent to one. Models economize on tokens by simulating work. Always check the sent folder yourself.

04 How it flows



05 Step by step

01 Write the ten templates by hand

One line each, in your voice, soft rather than pushy. This fires unattended, so err gentle. No em dashes, no invoice numbers, same sign-off on all of them.

02 Set the cadence ladder

1, 2, 3, 7, 14, 21, 28, 56, 84 days. Act only on an exact match so nobody gets nudged twice in a week. Past 84 days you have a bigger problem than cadence.

03 Put the opt-out check before everything

If they said stop or the invoice is in dispute, set the chase stage to closed, add a note and an audit comment, and send nothing. Make this step run before template selection.

04 Read the history first

Have the agent load the conversation with that contact before it picks anything. Context is what stops a nudge landing on top of a reply you missed.

05 Track what was sent to whom

Store the last template used per contact so the same line never goes out twice in a row. Without this the pool collapses to whichever line the model likes.

06 Forbid rewriting the templates

Say choose, do not compose. If the model is allowed to improve the copy, you lose the consistency that made the pool worth building.

07

Force a real run

Nick had to say run this for real using the info in the CRM. If output looks like a simulation, say the word real and name the data source.

08

Check the sent folder yourself

Open the mailbox and count. Do not accept the run summary as evidence. This is the step that catches simulated work.

06 Words explained

Cadence

The fixed schedule of when to follow up, counted in days since the last real contact.

Days outstanding

How long a record has been sitting without a reply. The number that decides whether today is a nudge day.

Template pool

A set of pre-written messages the system chooses from, instead of writing something new each time.

Opt-out check

Looking for any sign the person wants you to stop before you send. If it is there, you flag the record and stay quiet.

Audit comment

A note left on the record explaining why the system did what it did, so a human can reconstruct it later.

Simulated work

When a model reports finishing a job it did not actually do. It saves effort, and it looks exactly like real output.

07 Watch out for

- Claude claimed it sent to all three matching records when it had only sent to one. Always check the sent folder.
 - If the opt-out check runs after template selection, you will eventually nudge someone who asked you to stop.
 - Letting the model write fresh copy each run destroys your tone and widens the blast radius of a bad send.
 - Without a record of the last template used, the pool collapses to two or three favorites.
 - Nudging a record on a range instead of an exact day match means one contact gets hit several days running.
-
-

08 How it connects

Speed to Lead Poll and Reply

Speed to lead roughly doubles leads to calls and follow-up adds about another half again on closes. Stacked they multiply out to roughly 3x.

Fuzzy Variable Tightening

The template pool follows the same copy rules: one line, no em dashes, human voice, short enough that nobody suspects a bot.

Self-Healing and Error Logging Clauses

A daily unattended CRM sweep is exactly the routine that fails silently, so both clauses belong on it.

Follow-Up Cadence Engine

The framework note holding the trigger conditions table, the pool rules, and the verification story.

Credit. Framework and approach by Nick Saraev, from the follow-up cadence build in his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.