

Nurture a lead at exactly the stage they're at

One lifecycle attribute, then one workflow paired with one sequence. That's the whole structure — and it extends without ever being rebuilt.

THE PREREQUISITE

You can only nurture by stage if the stage is readable in your data. Before building anything, answer one question precisely: **what exactly makes a lead be in this situation rather than that one?** If no attribute carries that answer, find it or connect it. There is no shortcut past this step.

PART 1 — THE LIFECYCLE ATTRIBUTE (THE ENABLER)

Create a lifecycle attribute that is written by workflows only, never by your team.

- Give it a **default value**, so every record starts somewhere.
- **One workflow per transition**. Trigger it on the update of one specific attribute.
- Then a **switch that checks two things, not one**: is the new value the one that should move this record forward — and what is the record's *current* lifecycle?
- Then update the lifecycle value.

WATCH OUT — ROLLBACKS

The second check is what prevents them. A trigger can fire more than once, and the same action is often valid for two different stages. If the action that makes someone a client is also the action that makes them a repeater, a repeater will fall back to client every time it fires. **A repeater is a repeater forever**. Always read the current stage before writing the next one.

Why the CRM should own this: your team should be entering data nobody else can capture — the date they did something, the outcome of a call. Not a status. Set clear rules, and let the CRM cross-reference them and assign the stage itself.

PART 2 — ONE WORKFLOW + ONE SEQUENCE, SAME NAME

Nurturing is a pair. Give both the same name — Attio makes it obvious which is which, so there is nothing to disambiguate.

1. **Trigger** on the attribute update.
2. **Switch, immediately**. Before anything else. If the record matches none of your cases, nothing runs and nothing is consumed. This is the single biggest cost lever in the whole build.
3. **Two branches**. **ENROLL** — new value is the target status, plus your exclusions. **EXIT** — new value is one of the values that mean "stop".

EXCLUSIONS TO PLAN FOR IN THE ENROLL BRANCH

- Lost.
- Asked not to receive follow-ups, but still interested — not lost, just not to be emailed.
- A callback date set in the future — contacting them before that date makes no sense.

LIFECYCLE, OR A FINER ATTRIBUTE?

Default to the lifecycle. Drop to a finer attribute only when the message has to change *inside* a single stage — someone who started filling in their profile shouldn't get the same reminder as someone who never opened it. If the information can live in the lifecycle, put it in the lifecycle.

THE SEQUENCE

As personal as you can make it, as generic as it needs to be to keep working across everyone who enters it. Finding that balance is the actual work.

BUILD CHECKLIST

- | | |
|--|---|
| ☐ The stage is readable from an attribute — verified, not assumed | ☐ Team never edits the lifecycle by hand |
| ☐ Lifecycle attribute created, with a default value | ☐ Workflow and sequence share a name |
| ☐ One workflow per transition, each checking the current stage before writing | ☐ Switch placed first, before anything that consumes credits |
| ☐ No rollback possible on any transition | ☐ Enroll branch carries its exclusions; exit branch exists |