

THE LIVECRM MANIFESTO

Clean First, Then Always On

Why your CRM keeps falling back into disrepair, and the system that keeps it a live copy of your market.

- 1 Define
- 2 Analyze
- 3 Purge
- 4 Enhance
- 5 Maintain

Jaime Muirhead, Founder and CEO, LiveCRM

Why LiveCRM exists

Every business needs the companies and people it can sell and market to inside its CRM and marketing automation. Most do not have them. The records are missing, duplicated, or so out of date that nobody trusts them.

Marketing automation was built on the assumption that you did. The original promise was always on: nurture your market continuously, warm people up, turn them into marketing qualified leads, and only call or email the ones who were ready. It worked because the database was the asset. You grew it, you kept it, and every campaign drew on it.

Over the last five years that changed. Inbound slowed down, outbound took over, and a new habit formed: build a fresh list for every campaign. A whole category of list-building tools grew up around it. My firm belief is that this happened because CRM data decayed. When the database stopped being trustworthy, teams stopped nurturing it and started rebuilding it, one campaign at a time.

List building never made much sense. Companies are not opening a new market every quarter. They run different plays and different campaigns, but their TAM only changes when they build a new product or go after a new market. What moves constantly is the people inside it. They change jobs, get promoted and leave, while companies grow, merge and rebrand. Keeping up with that movement is the real challenge, and a list rebuilt for each campaign only takes a snapshot of a market that will have moved again by the next one. And the channels growing fastest now, display, connected TV and every other audience you target by account, all draw on that same database. Always on matters more today than it did when marketing automation first promised it.

I have spent 15 years in CRM data operations, building RingLead and then running ZoomInfo's data business. Keeping a CRM clean has always come down to five steps: define, analyze, purge, enhance, maintain. The steps were never the problem. What was missing was a system that could run it end to end, and keep running it, without a

person stitching five tools together every quarter.

That is why LiveCRM was built. It is the framework as an engine: every step a tool, and every tool sharing the same rules, the same matching and the same audit trail. It runs in one order that matters more than anything else in this guide.

Clean first. Then always on.

02

Signals change the play, not the database

The strongest argument for list building is signals. Outbound today runs on intent, hiring, funding, job changes and website visits, and each signal seems to call for its own list. That part is right: a signal should change the message, the timing and the approach.

It should not change who is in your database. The company showing intent this week was in your TAM last week. The buyer who just changed jobs was someone you should already have known. A signal is a reason to run a different play on records you already hold, not a reason to go and buy them again.

When those records live in your CRM and are nurtured all along, a signal lands on someone with history: an owner, past touches, a known role, a place in an account. When every signal starts a new list, all of that is thrown away, and the same people are bought, uploaded and deduplicated again, campaign after campaign.

**The answer to signals is
segmentation, not list building.**

Your TAM stays in your CRM, always on and always nurtured. Each play is a segment of it, drawn from records you keep fresh rather than from a list you rebuild.

03

The framework was right. The tooling made it a project.

Every step of CRM hygiene has been sold as its own product. A dedupe tool for purge. An enrichment tool for enhance. A routing tool for assignment. A spreadsheet for analysis. Each one reads the CRM its own way, matches records its own way, and writes back on its own schedule.

So the work runs as a list and a button:

- The list is built on whatever looked incomplete, not on whether the records can be matched.
- Duplicates are merged in one tool while another is still creating them.
- Enrichment is written back without asking whether the company returned is the company on the record.
- New records land with no owner, because routing lives somewhere else.
- The run ends when the write ends. Nobody remembers what was asked last time.

Then it is repeated a quarter later, on a new list, from scratch. That is why teams feel they pay for the same records over and over, and why the CRM never seems to get cleaner. The problem was never a missing step. It was that no system ran the steps together.

04

Where it breaks

The same failures show up in almost every CRM. Each has a specific cause, and none of them is bad data.

FAILURE	WHAT HAPPENS	WHY IT HAPPENS
Duplicates that come back	A cleanup merges records, and within weeks there are new ones	Records are created and imported without being checked against the CRM with the same matching rules
Bad merges	The wrong value survives a merge, or two different companies are merged	Survivorship and matching rules are set per tool, or not at all
Formats that never match	"IBM" and "International Business Machines", or "CA" and "California", are treated as different	Values are compared as raw text, with no normalization and no ids
The job change overwrite	Someone who left gets their new title, email and phone written onto their old contact	The provider's company is never compared to the account on the record before writing
Enriching bounced contacts	A contact whose email already bounces gets filled in instead of flagged	Nobody verifies before enriching
Blind overwrites	Good CRM values are replaced by worse ones	There are no per-field rules for which value wins
Paying to re-ask	Records that never matched are sent again, run after run	Nothing remembers what was asked and what came back
Hierarchy confusion	A contact is tied to a parent, subsidiary or branch that is not theirs	The provider's company and the CRM's account structure are never reconciled

FAILURE	WHAT HAPPENS	WHY IT HAPPENS
Orphaned ownership	New records land on the integration user	Routing is someone else's tool and someone else's problem
No way back	A bad merge or batch cannot be undone	Writes are not recorded with what they replaced

Every row is a decision that should be made before anything touches your CRM. When each step is a separate list and a separate button, none of them are made.

05

Why the work in between is hard

The work between raw data and a trustworthy CRM is genuinely difficult, which is why most tools skip it. It is six separate problems, and each one needs its own logic.

Two matches, not one.

The provider answers who this person or company is, using whatever identifiers you send. Then you have to answer a different question: is the company the provider returned the same company as the account this contact sits on? The provider cannot answer that for you, because it has never seen your CRM. That second match needs your own matching rules, run against your own data.

Mapping is not a lookup table.

A provider returns fields your CRM may not have, fields that hold lists, and fields whose meaning differs from yours. Each one needs a home, a format and a decision about what to do when there is nowhere to put it.

Normalization makes values comparable.

Company names, states, countries, phone numbers, websites and job titles arrive in dozens of formats. Until they share one, every match, merge, segment and routing rule works on noise.

Survivorship decides which value wins.

For every field, someone has to decide whether the provider's value replaces yours, fills yours only when it is blank, or never touches it. Without that decision, enrichment overwrites whatever a rep typed yesterday.

Hierarchies decide where a record belongs.

A provider may return a parent company, a subsidiary or a headquarters. Your CRM may model those as one account, several, or not at all. Attaching a contact to the right one takes matching, not guessing.

Assignment decides who owns what is new.

A record without an owner is a record nobody works. Routing has to follow your rules, on values that may have only just arrived from enrichment.

06

Clean first: the five steps as one system

LiveCRM runs the five-step framework as tools that share one set of rules. Each step below is a tool you can run on its own, and every one of them matches, decides and writes the same way.

STEP	WHAT IT REQUIRES	HOW LIVECRM RUNS IT
Define	Your total addressable market (TAM) at the company and persona level, with your ideal customer profile as its core; matching, survivorship, naming and assignment rules	The Rules tab holds every rule as a template: matching, survivorship and update rules, normalization rules, assignment rules. Your TAM definition sets the audience every later step works on. Every tool reads the same rules.

STEP	WHAT IT REQUIRES	HOW LIVECRM RUNS IT
Analyze	Duplication rate, completeness against your market, completeness of records, validity	The AI Readiness Scan measures your own CRM and names the runs that would fix what it finds.
Purge	Remove duplicates, and the records outside your market or out of date	Deduplication matches with your rules and merges with your survivorship rules. Merges go to review, and each one can be reversed. Mass Update flags what no longer fits.
Enhance	Fill gaps, add net-new records, standardize, segment, score, assign	Enrichment, Create, Normalization, Segmentation, Scoring and Assignment, each a tool, all staging their results before anything is written.
Maintain	Repeat analyze, purge and enhance on a cadence, with triggers	Always On keeps the records inside your TAM fresh. Subscription adds the ones you do not have yet. Both run on a schedule or a trigger, and write once.

The order is the point. Define and analyze tell you what clean means for your CRM. Purge and enhance get you there. Maintain keeps you there.

07

Why clean has to come first

Always-on enrichment on a dirty CRM does not fix it. It makes the mess permanent, faster. Every problem the cleanup would have removed gets copied into every run that follows.

- Duplicates get enriched twice. You pay for both, and the two records drift apart with different values.

- Unnormalized names break matching. A company spelled three ways cannot be compared to what the provider returns, so real employers read as moves and moves go unnoticed.
- Accounts that were never resolved hold no ids. Without a company id on the account, every contact check falls back to comparing names.
- Routing fires on blanks. Assignment rules built on industry, state or employee count misroute every record where those fields are empty or inconsistent.
- Bad merges compound. A wrong value that survived a merge becomes the value every later run protects.

So the first run is a cleanup: deduplicate with your rules, normalize the fields every other step depends on, and resolve your accounts to the provider. Only then does always on have something worth keeping fresh.

08

Analyze first: you can only match on what you already have

Enrichment is only as good as the identifiers you send. A record with a real website, a stored provider id and a clean link to its account matches reliably. A record with a company name typed three different ways does not. Yet almost nobody measures this before buying credits.

The fields that decide how well enrichment will work are specific:

- For accounts: the provider's company id, the website, the company name, and the address.
- For contacts: the link to an account, the provider's person id, the email, the LinkedIn URL, and the first and last name.

Filled is not the same as usable. A website field full of "n/a", "none", email addresses and company names looks complete and matches nothing. Readiness has to count the values that actually work, and report the junk separately.

The LiveCRM AI Readiness Scan measures exactly this. It shows how many of your records can be matched by id, how many will fall back to names, and how many cannot be checked at all. Then it tells you which runs, in which order, would move them up.

09

Then always on

A clean CRM starts decaying the day the cleanup ends. Always on is how it stays clean: the same rules, run continuously, in an order where every step makes the next one more reliable. The rest of this guide is how that works, starting with the step careless enrichment always skips.

10

Verify before you enrich

A bounced email is the loudest job change signal a CRM has, and careless enrichment treats it as a record to fill in. That is how a departed contact gets a fresh title and phone number written onto it, at a company they left.

Verification belongs before enrichment, not after it. Every contact's email is checked first, and the result decides what happens next:

- The email is valid: the contact goes on to enrichment as normal.
- The email bounces: nothing is enriched onto the contact. It goes to the job change workflow instead, which looks up where the person went and proposes what to do about it.

- The email bounces, but the provider still places the person at the same company: this is usually a changed email format, a new domain or a dead mailbox, not a departure. The contact is enriched, and the provider's current email replaces the dead one.

Never write enrichment onto a contact whose email has already told you something changed.

11

The ladder: every match makes the next one stronger

The order you enrich in matters as much as what you enrich. Each step should use the strongest identifiers a record has, and write back the ones that make the next step stronger. Done that way, a CRM gets easier to keep fresh every time it runs.

- 1 Accounts first. The provider matches each account on the best identifiers it has: the website if there is one, otherwise the name with address, city, state and phone. A match writes back the provider's company id and fills the website where it was blank.
- 2 Contacts next. A contact is matched on its email, or its name and its now-resolved account. A match writes back the provider's person id. The employer check can now compare company ids instead of company names.
- 3 Steady state. From then on, contacts are looked up by the person id, which stays with a person when they change jobs, and compared to their account by company id. Name matching only comes back for records that are new since the last run.

Records that never match are not asked again, unless something about them changes. That is where most wasted credits come from, and it is the easiest waste to stop.

12

Stage first, decide, then write once

Careless enrichment writes the moment the answer comes back. Every decision in this guide needs the opposite: somewhere to hold the provider's answer next to your CRM's data, compare them, decide, and only then write.

That place is a staging area. Everything the provider returns lands there first, beside the record as your CRM holds it. The matching, verification results, survivorship rules and routing all operate on the staged data. Your CRM is not touched until every step has made its decision.

Staging changes what a write is:

- Each record is written once, with every step's result on it, instead of being updated by five tools in five passes.
 - A field that someone changed in the CRM since it was staged is skipped and reported, not overwritten.
 - Anything that needs a human, such as creating a new account or contact, goes to a review queue complete, owner included.
 - One run's writes are one unit to reverse, so a bad batch can be undone.
-

13

Job change is a byproduct of enrichment

Enrichment and job change detection ask the provider the same question: where does this person work now? Careless setups ask it twice, through two tools, and keep opposite halves of the answer. One writes the new details onto the old record. The other notices the move, but only after the damage is done.

Done properly, it is one question and one decision:

- The provider matches the person, and the company it returns is the company on their account: enrich the record.
- The provider matches the person, and the company is different: write nothing to the record, and hand the answer to the job change workflow.
- The provider does not match the person: write nothing, and do not ask again.

The job change workflow already has everything it needs from that one answer: the new company, title, email and phone. It marks the old contact as departed, finds the new company in your CRM with your own matching rules, or proposes creating it, and proposes the contact at the new company. Then it backfills the account they left: it finds the people now in that role, or closest to it, and proposes them as new contacts, so the account is never left without coverage. Every one of those goes to review before anything is written.

14

Maintain: Always On and Subscription

Once the CRM is clean, keeping it that way should not be a project you run once a quarter. It should be a plan that runs on its own, and gets better every time it does. In LiveCRM that plan is Always On:

- 1 Account Enrichment resolves your accounts and fills in what is missing, under your update rules.

- 2 Verification checks every contact's email before anything is enriched.
- 3 Contact Enrichment enriches the contacts still at their company, and stages the ones who moved.
- 4 Job Change proposes what to do about each person who moved, including their replacements at the account they left.
- 5 Assignment routes every new account and contact under your own rules, with contacts inheriting their account's owner by default.

Always On starts where Define does. You define your TAM at two levels: the companies you could sell to, and the personas inside them you sell to, with your ideal customer profile as the core of both. That one definition is then used in two directions.

INWARD: A FILTER

Always On

It decides which records already in your CRM are worth keeping fresh. The list is no longer whatever looked incomplete; it is everything inside your TAM. Records outside it are not enriched at all. They become candidates to flag or purge, instead of a cost you pay every quarter.

OUTWARD: A SEARCH

Subscription

It finds the companies and people inside your TAM who are not in your CRM yet. It searches your TAM, checks every result against your CRM with your own matching rules so nothing is created twice, creates what is genuinely new, and routes it.

The two plans take turns on a connection, never running at once, so they can never create the same company twice. Defined once, the TAM cannot drift between what you keep fresh and what you add. Always On keeps your CRM true to your market. Subscription keeps it complete.

Your CRM stops being a snapshot of your market and becomes a live copy of it.

Each step is a LiveCRM tool you can also run on its own. Together they share one staging area, so each step sees what the steps before it found. When the last one finishes, the plan writes once. Direct updates are applied, anything that needs a human waits in the review queue, and every change is recorded with what it replaced.

15

Start with a readiness scan

Before you clean, enrich or buy another credit, find out what your CRM can actually be matched on and what it needs first. The AI Readiness Scan reads your CRM and shows you:

- how many accounts and contacts can be matched by id, by name only, or not at all;
- how many websites, emails and account links are usable rather than just filled;
- which cleanup runs to do first, and which runs, in which order, would then move your records up the ladder.

It takes minutes, it writes nothing, and it tells you whether your next enrichment will work before you pay for it.

Start your free trial and run your readiness scan today.

livecrm.ai/trial