CRM data migration: A practical process overview

CRM data migration is the process of moving data, workflows, and assets from one CRM to another. It matters because your CRM is the operational backbone of your revenue team, and when the data inside it is wrong, every process built on top of it breaks too.

I've seen more CRM migrations than I can count, and the ones that fail almost always fail the same way: the team underestimated scope, skipped data cleansing, or rushed to go-live without a validated rollback plan. The ones that succeed treat migration as a structured business change, not a bulk data transfer.

This guide covers the full CRM data migration process from planning through hypercare.

Table of Contents

- What is CRM data migration?
- CRM Data Migration Plan
- CRM Data Migration Data Cleansing
- CRM Data Migration Field Mapping
- CRM Data Migration Sequencing
- CRM Data Migration Historical Data
- CRM Migration, Integrations, and Security
- CRM Data Migration Validation
- CRM Data Migration Tools
- CRM Migration Checklist
- CRM Data Migration Go Live and Hypercare

- Frequently Asked Questions About CRM Data Migration
- Final Thoughts

What is CRM data migration?

CRM data migration is the process of moving records, relationships, history, permissions, and connected workflows from one CRM to another. That definition matters because the phrase “moving data” undersells what is actually involved.

CRM data migration requires more than a simple CSV import. A true migration covers:

- **Entities:** Contacts, companies, deals, tickets, custom objects
- **Relationships:** The links between companies and contacts, contacts and deals, deals and activities
- **History:** Emails, calls, notes, tasks, meetings, and attachments
- **Permissions:** User roles, team structures, property-level access
- **Dependencies:** Workflows, sequences, integrations, and reports built on top of that data

Each layer adds complexity. A contact record is connected to a company, associated with open deals, threaded into email history, and tied to automation sequences. Breaking one of those relationships creates orphaned records, broken pipelines, or gaps in reporting on day one.

Migration is also distinct from integration. An integration keeps two systems in sync on an ongoing basis. Migration is a one-time (or phased) movement of structured data, intending to make the new CRM the single source of truth. You may run both, but they are different workstreams with different owners.

Think of CRM data migration as a phased business change, not a technical event, much like the strategic approach required for

revenue performance management.

CRM data migration has goals (what does a successful migration look like?), constraints (what is the freeze window? what is the rollback trigger?), and success criteria (what record counts, accuracy rates, and user validation tests must pass before go-live?). Every migration decision flows from those three inputs.

This guide walks through the full end-to-end process: plan → cleanse → map → sequence → test → migrate → validate → go-live → hypercare.

CRM Data Migration Plan

The migration plan is the document your entire team works from. It defines who owns what, the timeline, how decisions are made, and what happens when something breaks. I have found that teams who invest two to three weeks in planning save months of cleanup on the back end.

Roles and RACI

Every CRM migration needs clear ownership across four functions:

- **Migration lead (RevOps or CRM admin):** A revenue operations role responsible for owning the plan, sequencing, and validation sign-off, is essential to understanding what is revenue operations and its critical role in data migration success.
- **Data owner (operations or IT):** Owns cleansing decisions, deduplication rules, and survivorship logic
- **Business stakeholders (sales, marketing, service leadership):** Approve scope decisions, especially what gets archived versus migrated
- **Technical owner (developer or SI partner):** Executes API-

based migration, builds field transformation scripts, and runs reconciliation jobs

Assign a RACI (Responsible, Accountable, Consulted, Informed) matrix for every major phase. The most common failure point is ambiguity around who approves a go/no-go decision. Define that before you start.

Phase Map

A well-structured CRM migration runs through eight phases:

1. **Assess:** Audit current data, map object inventory, document dependencies
2. **Cleanse:** Deduplicate, normalize, remove stale records, establish golden records
3. **Map:** Align source fields to destination fields, handle gaps and transformations
4. **Test:** Run a sandbox migration with a representative data sample
5. **Migrate:** Execute phased migration by object type and priority
6. **Validate:** Record counts, spot checks, and user acceptance tests
7. **Cutover:** Freeze source system, execute final delta migration, go live
8. **Hypercare:** Monitor errors, support users, resolve edge cases (typically 2–4 weeks)

Sandbox Usage

Always run your first migration in a sandbox environment, not production. A sandbox lets you test field mapping, surface transformation errors, and validate relationship integrity before any real data is touched. HubSpot’s sandbox environments are purpose-built for this use case, allowing you to mirror your production portal and iterate without risk.

Pro tip: Run your sandbox migration at least twice. The first run reveals gaps in field mapping. The second run, after you've fixed those gaps, is the one you use for your validation baseline.

Risk Register and Change Management

Document your known risks before the migration begins. Common risk items include:

- **Data quality worse than expected → mitigation:** Extended cleansing window with clear exit criteria
- **Integration failure at cutover → mitigation:** Smoke tests pre-cutover, rollback trigger defined
- **User adoption gaps → mitigation:** Training sessions scheduled before go-live, not after
- **Source system data loss → mitigation:** Full export and backup before any migration begins

Change management is the underrated half of CRM migration, and it directly impacts your ability to achieve sales optimization across the organization. Your users need to know what is changing, when, and why before they log into a new system for the first time.

A communication plan with milestone updates (kickoff, sandbox complete, go-live window, hypercare end) keeps stakeholders aligned and reduces day-one friction.

HubSpot's Smart CRM was designed to simplify this transition. Its unified data mode reduces the complexity of remapping relationships compared to fragmented legacy systems.

CRM Data Migration Data Cleansing

Data cleansing should happen before full CRM migration, not during or after. This is the rule I emphasize most strongly with every team I work with. Dirty or duplicate data increases

the risk of bad records being carried into the new CRM, and fixing data quality in the new system is significantly harder than in the old one.

Data Audit

Start with a complete data audit. For each object type (contacts, companies, deals, tickets), document:

- Total record count
- Percentage of records missing key fields (email, company name, deal amount)
- Duplicate rate (identified by exact email match, fuzzy name match, or domain-level deduplication)
- Stale records (no activity in 18–24 months, or clearly invalid data)
- Inconsistent picklist values (e.g., “New York,” “NY,” “New York, NY” all in the same field)

This audit produces your data quality baseline. You’ll use it to set cleansing targets, prioritize effort, and measure progress.

Deduplication and Normalization

Deduplication is a time-consuming and crucial part of data cleansing. Define your matching rules before you start. I’ve found that an exact email match is the safest starting point for contacts. From there, you can layer in fuzzy matching on name + company, or domain-level deduplication for company records.

Normalization means establishing and enforcing data standards across the dataset: phone number formats, country codes, picklist values, and lifecycle stage definitions. Document your standards in a data dictionary. These standards clean your legacy data and become the governance rules for your new CRM.

Golden Records and Survivorship Rules

When two duplicate records get merged, survivorship rules define which field values survive. For example, if two contact records have different phone numbers, keep the most recently updated one. If both have email addresses, merge them into a primary-secondary structure.

Document your survivorship rules before deduplication begins. Undocumented rules lead to inconsistent decisions across thousands of records, creating new data quality problems at scale.

Pro tip: HubSpot Data Hub includes native deduplication workflows and data quality automation tools that can enforce survivorship rules at scale without manual review of every record pair. Use it during the cleansing phase to build rules you'll carry into production.

CRM Data Migration Field Mapping

Field mapping aligns source CRM fields with destination CRM fields. It sounds straightforward. In practice, it is where most migration projects hit their first significant bottleneck because no two CRMs use the same data model.

Building Your Field Inventory

Before you can map anything, you need a complete inventory of your source system's objects and properties. For each object type, document:

- All fields, including custom fields and legacy fields that have not been used in years
- Field type (text, number, date, picklist, lookup, multi-select)
- Picklist values and whether they match destination

equivalents
- Lookup relationships (e.g., deal owner → user record)
- Whether each field is actively used or can be retired

Build this inventory in a mapping spreadsheet with columns for: source field name, source field type, source picklist values (if applicable), destination field name, destination field type, destination picklist values, transformation required (yes/no), and migration status.

Handling Mapping Conflicts and Gaps

Three types of mapping conflicts come up in nearly every migration:

- **Type mismatches:** Source has a text field, destination requires a picklist. You need to normalize values before migration.
- **Field gaps:** Source has a field that doesn't exist in the destination. Decide whether to create a custom property, map to the closest available field, or archive the data.
- **Naming conflicts:** Source uses "Account Owner," destination uses "Contact Owner."

Relationship mapping is a separate, and equally critical, workstream. Relationship mapping preserves links between companies, contacts, deals, and activities. If you migrate contacts before companies, the company association has nowhere to point. If you migrate deals before contacts, the deal owner association breaks.

Pro tip: HubSpot's CRM import tool supports field mapping at upload, so you can map source columns to destination properties in the UI before committing the import. Use this during sandbox testing to validate your mapping logic before running production migration.

CRM Data Migration Sequencing

What order should you migrate objects?

Migration sequencing helps prevent orphaned records. This is one of the most technically important decisions in the entire process, though it is often overlooked.

The core rule: migrate parent objects before child objects. Companies or accounts are often migrated before contacts and deals because they depend on them. Here’s the standard recommended sequence:

1. Users (required to assign ownership to all downstream records)
2. Companies / Accounts
3. Contacts (associated with companies)
4. Deals / Opportunities (associated with contacts and companies)
5. Tickets / Cases (contacts and companies)
6. Custom objects (after any parent objects they reference)
7. Activities: notes, calls, emails, tasks, meetings (associated with contacts, companies, deals)
8. 8. Attachments and documents (after all associated records exist)

Deviating from this sequence creates orphaned records, which are records with broken associations because the parent they reference doesn’t exist yet. Orphaned records are painful to fix retroactively and introduce data integrity risk that compounds over time.

Pro tip: Run a post-migration association audit after each batch. Check for null company_id on contacts, null associated_contact on deals, and null owner on any object. Catching orphaned records by object type makes remediation dramatically faster.

CRM Data Migration Historical Data

Should you migrate every historical activity?

No. And trying to do so is one of the most common reasons migrations run over time and budget.

Historical activities and attachments should be evaluated against four criteria before including them in scope:

- **Legal:** Does your industry require activity retention (HIPAA, GDPR, SOC 2, financial regulations)? If so, how long, and in what form?
- **Operational:** Do your sales or service teams actively reference historical activities when working accounts? If activities are more than 24 months old and rarely accessed, the migration value is low.
- **Analytics:** Are historical activities used in reports, attribution models, or forecasting? If so, they need to migrate. If not, archiving is lower-risk and lower-cost.
- **Storage:** Large attachment libraries (proposals, contracts, call recordings) can significantly increase migration time and cost. Evaluate whether these belong in the CRM or in a dedicated document management system.

My recommendation for most migrations is to migrate 12–18 months of activity history into the new CRM. Archive everything older into a read-only data store (a separate cloud storage bucket, a legacy CRM in read-only mode, or a data warehouse). Document the archiving decision and communicate it to stakeholders before go-live.

For email history specifically, most modern CRMs, including HubSpot, support inbox connection at the user level, which means future emails are logged automatically. Historical email import is often the highest-effort, lowest-ROI item on the migration list.

CRM Migration, Integrations, and Security

Integrations are the silent dependency that breaks the most migrations. I have seen go-lives derailed at the last hour because a marketing automation sync was still pointing at the old CRM, or because a Zapier workflow was writing duplicate records into production.

Integration Inventory

Before cutover, you need a complete integration inventory of all revenue operations tools connected to your current CRM, documenting each tool's data flows and endpoint requirements:

- Integration name and tool
- Owner (the person responsible for reconfiguring it post-migration)
- What data it reads and writes
- Endpoint changes required (new CRM API, new field names, new object structure)
- Credential update requirements (OAuth tokens, API keys, webhooks)
- Throttling or rate limit considerations at cutover
- Smoke test procedure to verify it works post-migration

Integrations need an inventory, owner assignment, and smoke tests before cutover. Smoke tests should run in your sandbox environment before production cutover, using real field names and sample records.

Pro tip: HubSpot Data Hub's data sync keeps connected systems aligned during and after migration. For tools with a native HubSpot integration in the HubSpot Marketplace, the reconfiguration is often a simple reconnection with no custom API work required.

Permissions Remapping

Permissions remapping should match real user roles and access needs in the new CRM, reflecting the distinct responsibilities between marketing and operations. Permissions remapping is an opportunity to rationalize your security model rather than just replicate it.

For each user group, document: what objects they need to see, what properties they should be able to edit, what records they should own versus view-only, and whether their access should be team-scoped or global. Then map those requirements to the new CRM's permission sets, and test them with real users before go-live.

Build security testing into your validation checklist: log in as a rep, a manager, and a read-only user, and verify that each can see exactly what they should and nothing more.

CRM Data Migration Validation

Validation is the last checkpoint before go-live and the phase most teams underinvest in. 'The data looks about right' is not a validation standard. Validation includes record counts, sampled spot checks, automated comparisons, and user acceptance testing — all four, not just one.

Validation Framework

- **Record count reconciliation:** Total records in source versus total records in destination, by object type. Any discrepancy greater than 0.1% requires investigation before go-live.
- **Sampled spot checks:** Pull a random sample of 50–100 records per object type and manually compare field-by-field against the source. This surfaces transformation errors that aggregate counts will not catch.

- **Automated comparisons:** For large datasets, build a script that compares source and destination records by unique ID and flags mismatches. This is especially important for high-volume objects, such as activities.
- **User acceptance testing (UAT):** Have 3–5 real users from different teams (sales rep, sales manager, marketing ops, service rep) log into the new CRM and validate their day-one workflows. Their sign-off is your go/no-go gate.

How do you plan rollback safely?

Rollback planning requires backups, trigger conditions, time windows, and communication paths, and it must be planned before the migration starts, not after something breaks.

Define each of the following before go-live:

- **Backup:** When was the last full export from the source CRM? Confirm it exists and is accessible.
- **Trigger conditions:** What failure scenarios trigger a rollback? (e.g., >5% record count discrepancy, critical workflow failures, user authentication failures)
- **Time window:** How long after go-live can you execute a rollback? (Typically, 24–72 hours before data is written in the new system makes a clean rollback impossible)
- **Communication path:** Who makes the rollback decision? Who notifies users? Who coordinates with IT and vendors?

Pro tip: Keep the source CRM in read-only mode for at least two weeks post-go-live. This gives you a clean reference point for any validation questions and a recovery path if edge cases emerge.

CRM Data Migration Tools

The right CRM data migration tool depends on your data volume,

technical resources, timeline, and the complexity of your field mapping and transformation logic. Here's how I would think about the decision:

When should you use a CRM data migration tool?

Use a dedicated CRM data migration tool when:

- Your dataset exceeds 50,000 records across multiple object types
- You have complex relationship structures (e.g., many-to-many associations, custom objects)
- You need bidirectional field transformation logic (not just field renaming)
- Your source system has an API-accessible data export, but doesn't support native CSV export of all objects
- You need a repeatable, auditable migration process with a rollback capability

For simpler migrations — clean data, standard objects, under 25,000 records — HubSpot's native import tool handles contacts, companies, deals, and tickets via CSV with in-UI field mapping. This is the fastest path to production for smaller teams.

Migration Tool Options

Here are the primary tool categories and representative options:

Native Import (HubSpot):

- Best for: small-to-mid migrations, standard objects, teams without developer resources
- Handles: contacts, companies, deals, tickets, custom objects (via CSV)
- Limitations: no relationship migration via CSV for

complex associations; no delta migration support

iPaaS / Data Sync (HubSpot Data Hub):

- Best for: keeping source and destination systems in sync during a phased migration; post-migration integration management
- Handles: two-way field sync, custom field mapping, data formatting rules
- G2 Rating: 4.4/5; users highlight bidirectional sync and HubSpot-native workflow triggers as standout features

FYI: iPaaS/Data Sync tools like HubSpot Operations Hub serve as a revenue operations platform, keeping source and destination systems in sync during phased migrations and post-migration integration management.

Dedicated Migration Tools (Trujay, Migrate.io, Data2CRM):

- Best for: large migrations, complex object structures, non-technical teams who need a managed migration path
- Handles: most major CRM-to-CRM migration paths with pre-built field maps
- Limitations: variable support quality; validate that your specific source-to-destination path is well-supported before committing

Custom API Migration (Developer-built):

- Best for: enterprise migrations with custom objects, complex transformation logic, or proprietary source systems
- Handles: any object, any field, any transformation, with full control over sequencing and validation
- Limitations: requires developer resources; higher upfront cost; maintenance burden if source or destination APIs change

Pro tip: Whatever tool you use, run it in your sandbox first. Every tool has quirks, such as rate limits, edge cases in

handling associations, and encoding issues with special characters. Discover those in the sandbox, not in production.

CRM Migration Checklist

What is the best way to track progress?

Track CRM migration progress by phase, with explicit completion criteria for each item before moving to the next. Here's the checklist I use:

PHASE 1: ASSESS

- Complete object and field inventory for all source CRM data
- Document all integration dependencies (tools, API connections, webhooks)
- Document all user roles and permission structures
- Run data quality audit (duplicate rate, completeness rate, stale record volume)
- Define success criteria and go-live acceptance thresholds
- Assign RACI for all migration phases

PHASE 2: CLEANSE

- Establish data standards and normalization rules
- Execute deduplication (with documented survivorship rules)
- Remove or archive records below the retention threshold
- Normalize picklist values across all affected fields
- Document cleansing outcomes (before/after record counts)

PHASE 3: MAP

- Complete field mapping spreadsheet (all objects)
- Identify and resolve mapping conflicts and gaps
- Define field transformation logic

- Map all relationship/association types
- Map all permission groups and user roles to destination equivalents

PHASE 4: TEST (SANDBOX)

- Execute sandbox migration following the defined object sequence
- Record count reconciliation by object
- Run sampled spot checks (50–100 records per object)
- Validate association integrity (no orphaned records)
- Test integration smoke tests in the sandbox
- Conduct internal UAT review

PHASE 5: PRODUCTION MIGRATION

- Confirm source CRM backup is complete and accessible
- Define rollback trigger conditions and time window
- Execute production migration in the defined object sequence
- Monitor for errors in real time

PHASE 6: VALIDATE

- Record count reconciliation (source versus destination) – all objects
- Sampled spot checks across all object types
- Automated field-level comparison for high-volume objects
- User acceptance testing sign-off from each stakeholder group
- Integration smoke tests in production
- Security testing (log in as each user role)

PHASE 7: CUTOVER

- Set the source CRM to read-only
- Execute delta migration (records created/updated since initial migration)
- Confirm the delta record counts reconcile
- Distribute new CRM access credentials to all users

- Send user-facing go-live communication

PHASE 8: HYPERCARE

- Assign hypercare support owner (RevOps lead or CRM admin)
- Create an error log for day-one issues
- Schedule daily standups for the first week post-go-live
- Define hypercare end criteria
- Document lessons learned for future migrations

CRM Data Migration Go Live and Hypercare

Go-live is not the end of a CRM migration. It is the beginning of a 2–4 week stabilization period called hypercare, and treating it as such is the difference between a smooth transition and a chaotic first month.

Go-Live Day

On go-live day, three things need to happen in sequence:

- 1. Source CRM freeze: set the source system to read-only. No new records should be created there from this point forward.
- 2. Delta migration: capture and migrate any records created or updated in the source CRM during the migration window. This is the gap between your initial migration and the freeze point.
- 3. User access: distribute new CRM credentials, confirm logins, and verify that each user can access their records and workflows.

The delta migration is where many teams cut corners and where data loss most often occurs. Even a 48-hour migration window can generate hundreds of new records in an active sales

environment. Build delta migration into your go-live runbook, not as an afterthought.

Hypercare

Hypercare follows CRM go-live as a structured support period during which your migration team actively monitors for errors, responds to user issues, and ensures revops automation workflows are functioning as designed.

Best practices for hypercare:

- Assign a dedicated hypercare owner; this should be your CRM admin or RevOps lead, not a help desk ticket queue
- Create a shared error log where users can flag issues with enough context to reproduce and fix them
- Run daily standups for the first week (15 minutes: what broke, what was fixed, what's still open)
- Keep the source CRM in read-only mode until hypercare closes
- Define explicit hypercare end criteria: X days without critical errors, Y% of users validated their workflows

Pro tip: HubSpot's Sales Hub and Service Hub include activity feeds, deal pipeline views, and ticket queues, making it easier for users to self-audit their data after the migration. Point your hypercare team to these views on day one, because they are faster than custom reports for surfacing missing records.

A well-executed hypercare period typically runs 2 weeks for small migrations and 4 weeks for enterprise migrations. The goal is to catch edge cases that only surface in real-world use and fix them before they become permanent data quality problems.

Frequently Asked Questions About CRM Data Migration

How long does a CRM data migration typically take?

Timeline varies significantly by scope. A small migration (under 25,000 records, standard objects, limited integrations) can be completed in 4–6 weeks.

A mid-market migration (50,000–500,000 records, multiple object types, 5+ integrations) typically takes 2–4 months. Enterprise migrations – complex custom objects, large datasets, many integrated systems – can run 4–9 months. The cleansing phase is usually the longest, regardless of record volume. Budget conservatively.

How much should a CRM data migration cost?

Cost depends on whether you are self-managing, using a migration tool, or engaging a systems integrator. Self-managed migrations using native import tools primarily incur internal labor costs (50–200+ hours for a mid-market migration). Dedicated migration tools like Trujay or Data2CRM typically cost $500–$5,000, depending on record volume and complexity.

A full-service SI engagement for an enterprise migration can range from $20,000 to $ 150,000 or more. The highest hidden cost is always data cleansing, so budget at least 30–40% of the total project effort for it.

Can you migrate attachments and email histories?

Yes, with important caveats. Attachments (files, proposals,

contracts) can be migrated if they are accessible via the source CRM's API or export, but large attachment libraries add significant time and storage cost. Email history migration depends on how emails were logged in the source system; BCC-logged emails are generally easier to migrate than inbox-synced threads.

In most cases, I recommend migrating 12–18 months of email history and archiving the rest, rather than attempting a full historical email migration.

What happens to automation and workflows during migration?

Automation and workflows do not migrate automatically and must be rebuilt in the new CRM. This is a separate workstream from data migration and should be staffed accordingly. Before go-live, document every active workflow in the source CRM: trigger, conditions, actions, and owner.

Rebuild and test in the destination CRM sandbox. Deactivate source workflows at the same moment you activate destination workflows — not before, or you'll have a gap during which automations won't run.

What is the difference between migration and integration?

Migration is a one-time (or phased) movement of data from one system to another, to establish a new system of record. Integration is an ongoing, bidirectional sync between two systems that are both in active use.

Migration replaces the source system. Integration connects two systems that continue to coexist. Some projects involve both: you migrate your CRM data to HubSpot, then set up a Data Hub data sync to keep HubSpot connected to your ERP or billing system on an ongoing basis.

Final Thoughts

A well-executed CRM data migration gives your team a clean foundation to grow from. A poorly executed one creates a data debt that compounds for years. The difference, in my experience, is rarely about technology, but about planning, sequencing, and disciplined validation.

The process outlined in this guide works across CRM platforms and team sizes. The core principles do not change: cleanse before you migrate, sequence parents before children, validate before you go live, and support your users through hypercare.

HubSpot’s Smart CRM and Data Hub were designed to make this process more reliable and easier to maintain. Whether you are migrating from Salesforce, a legacy system, or a spreadsheet-based setup, they provide your team with a data model, quality automation, and an integration layer to migrate with confidence and maintain clean data afterward.

Source