

SALESFORCE-TO-HUBSPOT MIGRATION

Move the process. Not just the data.

A practical guide to custom objects, reliable data mapping, clean relationships, better processes, and confident adoption—with The Tech-Diplomacy Academy case study.

The Tech-Diplomacy Academy

7

pipelines rebuilt

CRM

contacts, companies, leads & deals

Web +

emails, segments & automation

EXECUTIVE SUMMARY

Moving from Salesforce to HubSpot is an opportunity to improve how the business works—not simply relocate its records. A sound migration maps standard and custom objects, cleans the data, preserves relationships, and rebuilds the processes around them. Training and ongoing ownership then protect that investment. This guide combines a documented RSM client engagement with a detailed framework for planning your own transition.

A successful migration leaves people with trustworthy data, clear processes, and the confidence to use the new system.

For business leaders, CRM owners, and operations teams. Inside: client proof; object design; field mapping; data cleanup; associations; cutover; training; and post-migration upkeep.

01 / THE CLIENT ENGAGEMENT

The Academy: a working foundation

The Tech-Diplomacy Academy's published case illustrates a migration that extended beyond records to the processes and assets needed to work in HubSpot. [1]

Discovery, cleanup, and a destination plan

RSM documented the Salesforce environment and future HubSpot operating model. The review covered record types, pipelines, emails, segments, and connected processes. Rather than copying every legacy decision, the work established what would move, what required mapping, and what needed to be built in the destination.

Core CRM data and seven rebuilt pipelines

The delivered scope included contacts, companies, leads, and deals, together with seven pipelines rebuilt in HubSpot. RSM reviewed the purpose and stages of each pipeline and connected the migrated data to the appropriate process, checking that records, relationships, and pipeline context remained meaningful.

Communications, web assets, and final validation

Existing emails and segments moved with the CRM work; new webpages, forms, and workflows supported the destination operation. Final review covered data, pipeline configuration, web and automation behavior, and reporting. The result was a HubSpot environment organized around the Academy's operating requirements.

"If you're looking for a partner who's responsive, knowledgeable, and genuinely invested in your success, RSM is easy to recommend."

K. Adkins

Testimonial published with the Academy case study. [1]

How to read the rest of this guide

Pages 3–9 present RSM's recommended migration approach. Custom-object examples, training, and upkeep recommendations are not assertions of additional Academy project scope.

02 / STANDARD & CUSTOM OBJECTS

Design the destination data model

A custom Salesforce object can represent something essential to the business: a service agreement, installation, membership, or another entity with its own records and relationships. Preserve that meaning in the destination design.

Map custom objects to custom objects when warranted

Inventory each Salesforce standard and custom object, its purpose, record volume, fields, owner, relationships, and dependent processes. Where a distinct business entity still needs its own lifecycle, map the Salesforce custom object to an appropriate HubSpot custom object. Define its properties, display fields, required values, associations, and any necessary pipeline before loading records. [2]

Do not flatten a real entity into a spare text field

A company may have several service agreements with different dates, statuses, and contacts. Storing them as a single company note loses structure and makes reporting or automation harder. Conversely, do not create a custom object merely because one existed in Salesforce: first check whether a HubSpot standard object meets the business need. Record the reason for each retain, redesign, consolidate, or retire decision.

Separate relationships from inherited behavior

Salesforce lookup and master-detail relationships carry different behavior; master-detail can affect access and deletion. [5] A HubSpot association is not proof that those rules migrated. Explicitly redesign permissions, calculations, rollups, validation, and automation where needed. Record types and Salesforce-specific logic also require a destination decision, not an assumption of automatic equivalence.

Confirm the platform can support the design

HubSpot custom objects require an eligible Enterprise subscription, and object/property limits depend on the account. [2] Verify the purchased capabilities, association limits, workflow and reporting needs, and user access before agreeing the design. Migration method and licensing should follow the required operation—not be discovered as gaps after the data move.

Design output

An approved object map: source entity, destination entity, business purpose, key fields, relationship rules, dependent processes, and an accountable owner.

03 / FIELD MAPPING & TRANSFORMATION

Make every mapped field explainable

Matching labels is not enough. A defensible mapping explains what a source value means, how it changes, where it lands, and how the team will know the result is correct.

Build a field-level mapping specification

For every in-scope field, document the source object and API field name, destination object and internal property name, data type, valid options, transformation rule, blank-value behavior, and validation test. Distinguish required business data from historical information that can be archived. Include ownership, dates, currencies, consent status, and fields used by workflows or reports.

Resolve formats and business meaning before import

Map picklist values to approved destination options; do not turn unexpected values into convenient defaults. Define date and time-zone handling, number precision, currency meaning, and whether a blank means unknown, not applicable, or intentionally cleared. A formula's exported value does not recreate its calculation. Decide whether each calculation must be rebuilt or retained as a dated historical snapshot. HubSpot import formatting must match the destination property types. [3]

Keep a durable record-identity crosswalk

Preserve the Salesforce source ID in a dedicated destination property and maintain a controlled crosswalk to the new HubSpot record ID. Use a compatible unique identifier for the selected import or API method. [3] A Salesforce ID is not a HubSpot Record ID. Retain source-to-survivor mappings when duplicates merge so later relationship loads and reconciliation still resolve correctly.

Test a difficult sample, then approve the rules

Include custom-object records, missing fields, duplicate candidates, inactive owners, unusual picklists, and records with several relationships. Review the transformed values with the business owner. Version the approved map; log later changes and rerun affected checks. This turns “the import worked” into evidence that the information still means what the team expects.

Illustrative mapping—not Academy project data

A Salesforce Service Agreement custom object maps to a HubSpot Service Agreement custom object. Its renewal date maps to a date property; its status maps through an approved option list. Its Account reference becomes a Company association resolved through IDs—not a pasted company name.

04 / DATA QUALITY BEFORE & AFTER THE MOVE

Clean data. Better habits.

Migrating poor-quality data quickly only makes the same problems available in a different system. Cleanup should be a planned workflow, with decisions the business can review and rules that prevent the defects returning.

Profile the data before deciding what to keep

Measure duplicates, missing values, invalid formats, inconsistent categories, stale records, missing owners, and broken references across standard and custom objects. Separate active data from history and approved exclusions. Age alone does not justify deletion: consider retention, reporting, and relationship dependencies.

Deduplicate with explicit survivor rules

Use reliable identifiers and agreed matching criteria; send ambiguous matches for human review. Shared names or domains alone may not establish identity. Decide which record and values survive and how ownership, history, and associations are retained. Keep an auditable merge map to redirect every merged record's links to the correct destination.

Normalize without inventing information

Standardize valid phone and address formats, country values, lifecycle definitions, and approved status options. Resolve inactive owners and distinguish missing values from genuine zero or false values. Preserve opt-outs and suppression rules; migration is not permission to contact someone. Quarantine unresolved cases in an exception register with an owner and disposition rather than silently dropping records or guessing values.

Put prevention into the new process

Identify the forms, integrations, imports, and manual habits that introduced errors. Assign a data owner, publish field definitions, control new property creation, and use appropriate validation and required-field rules. Check which rules apply to each entry route; test imports and integrations as well as manual entry. Re-profile after migration and compare results against agreed acceptance thresholds.

Evidence of a cleaner system

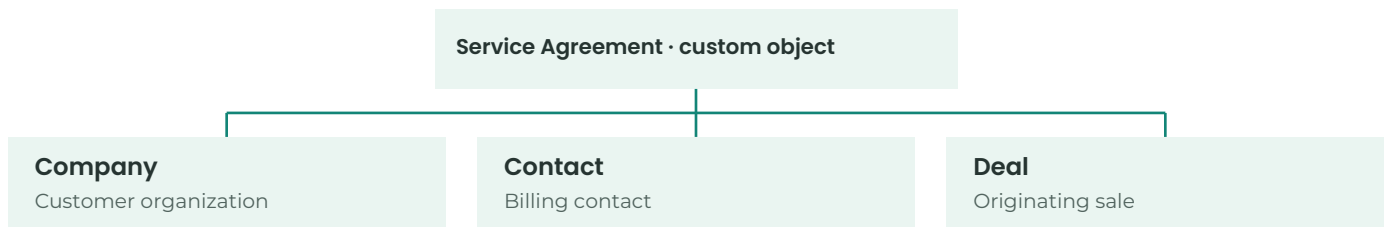
Track duplicate candidates, required-field completeness, unmapped values, missing associations, and unresolved exceptions. Show the baseline and final state using consistent definitions; every material discrepancy needs an explanation and an owner.

05 / ASSOCIATIONS & RELATIONSHIP INTEGRITY

Move the connections with the records

The same number of records in both systems does not prove a complete migration. Pull over all in-scope relationships between standard and custom objects, and verify that each link reaches the right record.

ILLUSTRATIVE RELATIONSHIPS — NOT CLIENT DATA



Inventory the relationship, not just the lookup column

Define each source reference, destination association, role or label, direction, and expected number of connections. Include one-to-many, many-to-many, parent/child, and custom-to-custom relationships. If a junction record stores dates, status, or other facts, assess whether it needs a destination object: an association label does not replace those facts. [4, 5]

Load the identities, then rebuild the links

Create destination schemas, association definitions, and required labels first. Resolve both ends through the source-to-HubSpot ID crosswalk, including merged-record survivors. Load associations after destination records exist, or use a supported multi-object import with tested identifiers. Labels must exist before import; verify their direction. [3, 4]

Reconcile each relationship type and test the result

Compare expected record pairs—not only association totals. Find missing parents, wrong-company links, duplicates, reversed roles, and unexpected connection counts per record. Inspect relationships in both directions and test dependent reports and workflows. Resolve launch-critical exceptions and document approved exclusions; unexplained gaps are not a successful migration.

Acceptance example

Every in-scope service agreement resolves to the intended company and required billing contact, with the correct role label. Exceptions are explicit, reviewed, and assigned—not hidden in a successful record import.

06 / BETTER PROCESSES & CONTROLLED CUTOVER

Improve the work—not just the software

Use the migration to simplify how information enters the CRM, how responsibility changes hands, and how teams recognize progress. Rebuilding the old complexity unchanged is not the only option.

Define the new process with the people who run it

For each pipeline or lifecycle, agree the entry criteria, stage meaning, required information, responsible role, next action, and handoff. Remove duplicate stages or fields only after their purpose is understood. Decide what belongs in automation and what requires judgment. Translate Salesforce validation, assignment, and workflow behavior into a deliberate HubSpot design; do not assume it comes across with the records.

Test complete journeys and the reports they feed

Use realistic scenarios: capture a contact, associate the right company, create or update a custom-object record, advance work, route a task, and verify the resulting report. Include rejected submissions, changed owners, and opt-out cases. Test with actual user roles, not only administrator access. Reconcile report definitions and filters; a different total can reflect a changed definition rather than lost data.

Rehearse the migration and control side effects

Run a test load in an appropriate isolated environment and review its exception log. Plan how automation, outbound communications, and connected systems are held or controlled during bulk loading. Define the freeze window and delta load for records and relationships changed after the initial extract. Assign one authoritative writer during transition to avoid conflicting updates or sync loops.

Make go-live a decision supported by evidence

Agree sign-off for record and association reconciliation, critical field values, end-to-end tests, reporting, user readiness, and open exceptions. Retain a source snapshot and a recovery plan that accounts for new work created after cutover. Do not retire Salesforce simply because a migration script finished. Confirm acceptance, necessary retention access, support ownership, and the retirement plan first.

A practical go-live gate

Business owners can complete their critical tasks with the correct access; required data and relationships reconcile; communications behave as intended; and the team knows where to get help.

07 / TRAINING & USER ADOPTION

Teach people how to do their work

Training is part of migration readiness, not an optional product tour after launch. People need to understand both where to click and why the new process works the way it does.

Train by role using the configured system

Show frontline users how to find and update records, navigate custom objects, maintain associations, log activity, and move work to the next stage. Show managers how to review pipeline health, exceptions, and reports. Give administrators separate instruction on access, properties, association labels, workflow maintenance, imports, and safe configuration changes. Use each team's actual responsibilities to define the curriculum.

Make relationships and data quality visible

Teach users when to create a new record versus update an existing one, how to select the correct associated company or contact, and what role labels mean. Explain the minimum information needed for a handoff and why free-text workarounds or duplicate fields undermine reports. Show how to flag a suspected duplicate or incorrect association without making an unreviewed merge or deletion.

Require practice—not just attendance

Use realistic, permission-appropriate exercises with safe training records: create an agreement, connect its company and contact, update its status, and confirm it appears in the correct view or report. Ask users to complete the task themselves and explain the next step. Capture errors and confusion as feedback about both the training and the configuration. A completed session is not proof of operational readiness.

Leave support people can reuse

Provide short role-based guides, recorded walkthroughs, a glossary of key fields and stages, and a clear escalation route. Schedule launch-period office hours and name internal champions who can help colleagues. Train a designated system owner and backup; include onboarding for future hires so knowledge does not disappear when one administrator or power user leaves.

Adoption evidence

Check whether users can complete their core tasks accurately, keep required fields and associations clean, and interpret the reports they rely on. Use recurring support questions to improve training and processes—not just count logins.

08 / POST-MIGRATION UPKEEP

Protect the system after go-live

A migration ends; CRM ownership continues. Agree a support period, responsible people, and an upkeep routine before handoff. The cadence below is a recommended starting point, not a fixed service commitment.

Stabilize first, then establish a routine

During the initial support period, review import or integration failures, routing issues, unexpected workflow behavior, broken associations, and user questions frequently. Assign severity, an owner, and a next action. Reconcile late-arriving changes rather than letting the new portal drift immediately after launch.

Review quality weekly and configuration regularly

Start with weekly checks for duplicates, missing required data, orphaned custom-object records, stale work, and sync failures. Review reporting definitions, workflow changes, property usage, and access on an agreed monthly or quarterly schedule. Investigate recurring defects at their source. Refresh training when the process changes and keep documentation current.

Assign ownership—and measure whether it is working

Name a business process owner, data steward, technical administrator, and backup contacts. Maintain a change log and a test-before-release process. Track unresolved exceptions, data completeness, association accuracy, process compliance, and recurring support issues against the launch baseline. Agree who handles ongoing support and improvements; do not leave upkeep as an assumed responsibility.

Plan the migration and the operation that follows.

Bring your object model, relationship map, data-quality concerns, and team workflows. RSM can help scope the data move, process improvements, training, and post-migration care as one coordinated plan.

Discuss your migration with RSM

Sources and scope: [1] documents the Academy engagement. [2–5] support platform guidance. The remaining recommendations and fictional examples are planning guidance, not additional Academy scope or guaranteed deliverables. Sources accessed 16 September 2026; confirm account capabilities during scoping.

[1. RSM: The Tech-Diplomacy Academy case study](#)

[2. HubSpot: Create and edit custom objects](#)

[3. HubSpot: Format import files](#)

[4. HubSpot: Create and use association labels](#)

[5. Salesforce Trailhead: Create Object Relationships](#)