

CASE STUDY: CRM Overhaul

Title: *Rebuilding a Scalable CRM Spine by Cutting the Bloat*

Subtitle: How SortedOps redesigned a bloated CRM into a lean delivery system that actually supports ops.

The Problem

A national commercial cleaning company was relying on HubSpot, but their CRM had become a liability:

- 15,000+ contacts bloated with duplicates, internal users, and inactive leads
- Sales team creating contacts manually with zero structure or standards
- Confusing segmentation across teams
- CRM pricing scaled with *total contacts*, not actual value
- CRM data disconnected from their ERP (Odoo), creating double-entry and delivery drag

They weren't just overpaying, they were operating blind.

The SortedOps Approach

As an embedded systems partner, SortedOps stepped in to **re-architect the CRM layer**:

- Audited and de-duped 2,000+ records across sales, ops, cleaners, and internal roles
- Segmented active clients, live field staff, and high-value past clients
- Mapped ERP → CRM data structure to align with actual delivery workflows
- Migrated from HubSpot to a leaner, automation-friendly stack (ActiveCampaign)
- Rebuilt the contact structure to **reflect the actual business**, not legacy habits

Everything was scoped, built, and handed over without disrupting operations.

💡 The Outcome

- ▼ **CRM costs cut by 80%**—without breaking delivery
 - ⚡ **Faster workflows** across Sales and Ops
 - 🧠 Clear contact segmentation → cleaner automation & better reporting
 - 🛠️ A CRM designed for *how the business operates today*, not how it grew up
-

🤝 Why This Matters to Partners

This wasn't just a CRM migration. It was **a backend infrastructure fix** — delivered fast and clean, without dev bottlenecks or client confusion.

If you're an **agency, ERP team, or systems consultant** working with clients who have bloated tools, duplicate workflows, or delivery drag:

SortedOps plugs in behind the scenes to rebuild systems that scale.

→ **Want a systems audit before your client's CRM gets out of hand?**

[Book a Partner Discovery Call](#)

Audit → Sprint → Retainer. One clear backend lifecycle. Zero chaos.