How we operate

CAPABILITIES & ENGAGEMENT MODEL

data-engineering-grade RevOps. We run your HubSpot like a warehouse-backed system, not a spreadsheet.

The operating philosophy

Most RevOps practices are built on trust. They trust that the lifecycle stage a rep set last quarter still means what it meant, that a dashboard built three integrations ago still reflects current logic, that the workflow no one touched in eight months still does what its name says.

> We do not operate on trust. We operate on inspection.

Every property, stage definition, association rule, and workflow dependency is treated as a claim that must be verified against the data. When a definition lives in a single governed location, every downstream report inherits the same answer. When a workflow is versioned, you can trace exactly what changed, when, and what broke. The result is a HubSpot portal that behaves like a source of truth, not a filing system that happens to have dashboards on top.

How an engagement flows

STAGE 1

Audit

Every engagement starts with an inspection of the data layer. We do not start by building; we start by finding out whether the foundation can support what you want to build on it. The audit covers property hygiene, lifecycle stage integrity, association completeness, deduplication state, and workflow dependencies. The output is a defect register: each finding ranked by severity with a remediation estimate attached.

STAGE 2

Architecture roadmap

Before we write a single workflow or migration script, we produce an architecture decision. It defines the object model, association rules, the lifecycle stage taxonomy with one agreed definition per stage, and the canonical property set that carries your reporting load. It is a working reference every subsequent build decision is checked against.

STAGE 3

Build

Build executes the roadmap in a defined sequence: data cleanup and dedup first, then property rationalization, then lifecycle restoration, then automations, then the reporting layer. Changes are documented as they happen. Every workflow gets a clear name, a stated purpose, a write map of the properties it modifies, and its trigger. You own that documentation.

STAGE 4

Embedded retainer

Build ends. Governance does not. The embedded retainer puts a senior operator inside your stack each month to work the backlog, extend the architecture as the business changes, and run a quarterly review of properties, stages, and workflows. The same operator who understands your data model stays on it. No ramp-up, no translation loss.

Working cadence & communication

We operate **async-first.** Synchronous calls are reserved for decisions that actually require them: scope changes, architecture reviews, handoffs.

The default rhythm is a **weekly written update** at the start of each week: what was completed, what is in progress, what is blocked, and what input we need before the next update. Written to be read in three minutes.

For anything that benefits from a walkthrough, we record a **Loom.** A recorded walkthrough beats a meeting: pause it, rewind it, share it with a new hire, refer to it six months later.

Metric definitions live in a **single governed document.** If a definition changes, the document changes first, and every report that references it is updated. The same metric never means two things in two dashboards.

What makes Capstan Works different

Most HubSpot agencies are HubSpot-only. Their ceiling is what the interface exposes; when the data problem needs a warehouse, they stop. We are built at the intersection of modern data-stack engineering and HubSpot operations.

We can run a **dbt model** against your Snowflake or BigQuery warehouse, push the clean output back to HubSpot via a **reverse-ETL layer,** and have it reflected in your reports: all governed, versioned, and auditable. The number in your dashboard traces back to inspectable logic in a version-controlled data model.

And the person who designed the architecture is the person doing the build. No handoff to a junior executor after the sales call.

How we work

- Definitions before dashboards.
- The operating cadence is non-negotiable.
- Baselines before optimizing.
- Alerts and automation replace manual checking.
- Revenue leaks quantified in dollars.
- Nothing ships without a measurable output.

Deliverables by phase

PHASE 1

Audit

- RevOps charter (scope, owners, success criteria)
- Full CRM property audit, each property classified Required / Recommended / Deprecated
- Integration health map across every connected system
- Field-completeness baseline (target 95%+)
- Duplicate-rate assessment (target under 5%)
- RevOps maturity scorecard
- Revenue-leak scorecard, every gap quantified in dollars

PHASE 2

Architecture

- Written lifecycle-stage definitions with explicit exit criteria
- One-sentence MQL / SQL / SAL definitions plus handoff SLAs
- Lead-routing rules
- Lead-scoring model (Fit x Engagement) with a plain-language score-explanation doc for reps
- Reporting architecture with governed metric definitions
- Change-control and data-governance plan

PHASE 3

Build

- Automated dedup / merge plus email validation
- Deal-hygiene guardrails (required fields, no blank next step, stale-deal alerts)
- SLA-breach, churn-risk, renewal, and expansion automations
- Persona dashboards (CEO, CRO, VP Marketing, VP Sales, VP CS, and per-rep)
- Core reports: MQL-to-Won waterfall, multi-model attribution, pipeline coverage, forecast delta
- Warehouse integration: ETL + dbt + reverse-ETL writing scores back into HubSpot

Phase 4 · Embedded retainer cadence

WEEKLY

- Pipeline snapshot
- Forecast delta
- Deal-movement
- Speed-to-lead
- Zero unassigned leads

MONTHLY

- Waterfall + stage conversions
- Attribution summary
- Rep productivity
- Data-quality score
- Revenue-leak scorecard
- Experiment review

QUARTERLY

- QBR decks per GTM function
- Win/loss deep-dive
- NRR walkthrough
- Tech-stack consolidation
- Capacity / territory review