Data Product Ownership Playbook

Who owns what, how decisions get made, and how to keep a data product healthy after go-live. The part most programmes skip.

Bruce Dando
Data, AI & SAP Strategy
brucedando.com

The Ownership Problem

Building a data product is hard. Keeping it alive is harder. Most data products degrade not because of technical failure, but because nobody is clearly responsible for their ongoing care.

The pattern is predictable: a project team builds a beautifully governed dataset, ships it, and moves on. Six months later the schema is out of date, quality has drifted, and consumers have built workarounds. The product is technically still running, but nobody trusts it.

This playbook addresses the gap between "shipped" and "sustained." It is written for data leaders who want their data products to last beyond the project that created them.

> *"A data product without an owner is just a table that nobody maintains."*

What Ownership Actually Means

Ownership is not a title on an org chart. It is a set of ongoing accountabilities that someone has explicitly agreed to carry. A data product owner is responsible for:

- **Quality:** The data is accurate, complete, and refreshed on schedule. When it is not, the owner is the first to know and the first to act.
- **Relevance:** The product still serves its intended consumers. When business needs change, the owner adapts the product or retires it.
- **Communication:** Schema changes, outages, and deprecation plans are communicated proactively — not discovered by consumers in production.
- **Access:** The right people can use the data, and the wrong people cannot. Access reviews happen on a schedule, not ad hoc.

The Data Product RACI

Every data product involves multiple roles. The most common failure is assuming that one person does everything, or that "the data team" will handle it. Use this RACI to make responsibilities explicit at the start of every data product.

Activity	Domain Owner	Data Engineer	Data Steward	Consumer
Define business requirements	**A**	C	C	**R**
Set quality thresholds	**A**	**R**	C	I
Build and deploy pipeline	I	**R/A**	C	I
Monitor data quality	I	**R**	**A**	I
Approve schema changes	**A**	**R**	C	C
Manage access requests	**A**	I	**R**	I
Deprecation decisions	**R/A**	C	C	C
Consumer feedback	C	I	I	**R**

R = Responsible **A** = Accountable **C** = Consulted **I** = Informed

Where Ownership Should Sit

A common mistake is placing ownership inside the central data team. This creates a bottleneck and removes the people who understand the data best — the domain experts — from the accountability loop.

The most effective model is **federated ownership**: the domain team (Finance, Supply Chain, HR) owns the product, while the data platform team provides the infrastructure, tooling, and standards. The central team enables; the domain team owns.

Decision Framework

Data products generate decisions every week: Should we add a new field? Can we change the grain? Should we deprecate a product nobody uses? Without a framework, these decisions either stall (nobody wants to own the risk) or get made inconsistently.

Tier 1: Owner Decides Alone

- Adding optional fields that do not change existing structure
- Updating descriptions and metadata
- Fixing bugs in transformation logic
- Adjusting refresh schedules within existing SLAs

Tier 2: Owner + Consumers Agree

- Changing field names or data types (breaking changes)
- Modifying the grain of the dataset
- Changing the update frequency
- Adding or removing access for teams

Tier 3: Governance Board / Data Council

- Deprecating a product with active consumers
- Cross-domain products that span multiple owners
- Significant cost or compliance implications
- Exceptions to naming or governance standards

> *"The goal is not to create bureaucracy. It is to make sure the right people are in the room for the right decisions — and that everything else moves fast."*

Keeping a Data Product Healthy

A healthy data product is one that is accurate, used, and trusted. Here is a practical operating rhythm for maintaining product health after go-live.

Daily (Automated)

- Freshness check — has the product refreshed on schedule?
- Row count and null rate monitoring against thresholds
- Alerting on pipeline failures or source system outages

Weekly (Owner Review)

- Check quality dashboard for trends (not just pass/fail)
- Review any consumer support tickets or questions
- Clear backlog of minor metadata or documentation updates

Monthly (Stakeholder Sync)

- Usage metrics — is the product being consumed? By whom?
- Consumer satisfaction — any friction, missing fields, or format issues?
- Roadmap alignment — are upcoming business changes going to affect the product?

Quarterly (Health Assessment)

- Full quality audit against original specifications
- Access review — remove stale permissions, onboard new consumers
- Cost review — is the product still worth its compute and storage cost?
- Deprecation review — should any products be retired or consolidated?

When to Deprecate

Data products should not live forever. Deprecation is a healthy part of the lifecycle — it means your organisation is evolving. But it must be done thoughtfully to avoid breaking downstream consumers.

Signals That a Product Should Be Deprecated

- Usage has dropped to zero or near-zero for three consecutive months
- A newer product covers the same use case with better quality or grain
- The source system is being retired or replaced
- The cost of maintaining the product exceeds its demonstrated value

The Deprecation Process

- **Announce** — Notify all known consumers at least 60 days before shutdown. State the reason and the alternative (if one exists).
- **Freeze** — Stop accepting new consumers. Mark the product as "deprecated" in your catalogue.
- **Support** — Help existing consumers migrate to the replacement or alternative source.
- **Archive** — After the deadline, stop the pipeline but keep the metadata for audit purposes.
- **Remove** — After an additional grace period (30-90 days), delete the data if retention policies allow.

> *"Deprecating a data product is not failure. Keeping a product alive that nobody uses — that is failure."*

Putting It Into Practice

Ownership is not a document you write once. It is a set of habits your team builds over time. Start small: assign one owner to one product, agree on the RACI, set up the weekly rhythm, and iterate from there.

The companion guide, **Preparing Your First Data Product**, covers the build side: scoping, modelling, and publishing. Together, the two guides give your team a complete framework for shipping data products that last.

Want to go deeper?

I run workshops that help teams define their ownership model, build their RACI, and set up the operating rhythm for sustained data product health. Typically 2-3 hours with business and IT stakeholders together.

Get in touch: brucedando.com/contact