

Preparing Your First Data Product

A practical guide to scoping, modelling, governing and publishing a data product — written for data leaders and teams who have never shipped one.

Bruce Dando

Data, AI & SAP Strategy

brucedando.com

Why Data Products Matter

Most organisations have plenty of data. What they lack is a reliable way to package it so that other teams can find it, trust it, and use it without reverse-engineering someone else's pipeline. That is the problem a data product solves.

A data product is a governed, reusable, self-describing dataset that is owned by a specific team, built for a defined set of consumers, and published through a catalogue or marketplace. Think of it as the difference between a shared drive full of spreadsheets and a well-documented API.

What Changes When You Think in Products

- **Ownership becomes explicit.** Every dataset has a named owner accountable for quality, freshness, and relevance — not just whoever built the last report.
- **Consumers get a contract.** Schema, update frequency, grain, and known limitations are documented upfront — no more guessing.
- **Reuse replaces rebuilding.** Instead of five teams building five versions of "revenue by region," one product serves all five.
- **Governance scales.** Policies attach to the product, not to individual queries or extracts.

"If you can't describe who the consumer is and what question the data answers, you don't have a data product — you have a table."

Who This Guide Is For

This guide is for data leaders, product owners, architects, and analysts who are building their first data product — whether on SAP Datasphere, Databricks, Snowflake, or any modern data platform. The principles are platform-agnostic; the examples lean toward SAP because that is where the author has spent the most time.

Five Steps to Your First Data Product

Shipping a data product is not primarily a technical exercise. The hardest parts are scoping, naming, and getting agreement on ownership. The steps below are ordered by the sequence that works in practice, not by what feels most natural to an engineering team.

Step 1: Start With the Question, Not the Data

Every successful data product begins with a business question: "How much revenue did we generate by product line last quarter?" or "Which suppliers have the highest defect rate?"

Resist the urge to start by inventorying source tables. Instead, sit with the intended consumers — typically business analysts, planners, or operational managers — and document:

- The specific questions they need answered
- How often they need the data refreshed
- What decisions the data will inform
- Who else might benefit from the same dataset

If you cannot name at least one decision the data product will improve, stop and reframe. You are building infrastructure, not a product.

Step 2: Define the Scope and Grain

Grain is the single most important design decision. It determines what level of detail each row represents — one row per sales order line, per customer per month, per material per plant.

Get the grain wrong and every downstream metric will be wrong. Discuss these questions with your consumers:

- What is the finest level of detail needed? (transaction-level vs. aggregated)
- What time range should the product cover?
- What dimensions will users need to filter or group by?
- What measures (amounts, counts, quantities) are required?

Document the scope in a one-page brief before writing any code. Include what is explicitly out of scope — this prevents creep.

Step 3: Assign Ownership and Governance

A data product without an owner is just a table that nobody maintains. Ownership means a named person (not a team alias) who is accountable for:

- Data quality — accuracy, completeness, timeliness
- Schema changes — versioning and communicating breaking changes
- Access control — who can see it, who can use it in production
- Lifecycle — when to deprecate, archive, or replace

In practice, the owner is usually someone on the domain team (e.g., Finance, Supply Chain) with support from a data engineer. The owner does not need to write SQL — they need to care about the data.

Step 4: Build, Test, and Document

Now — and only now — you build. The implementation details depend on your platform, but the pattern is consistent:

- Connect to source systems and land the raw data
- Transform it into the agreed scope and grain
- Apply business logic (currency conversion, status mapping, hierarchy resolution)
- Add metadata: description, tags, update schedule, known caveats
- Write at least three validation checks (row counts, null rates, referential integrity)

Documentation is not optional. If a new analyst cannot understand the product from its metadata alone, it is not ready to publish.

Step 5: Publish and Iterate

Publishing means making the product discoverable and consumable. On platforms like SAP Datasphere, this means exposing it through the Data Marketplace. On others, it might mean registering it in a data catalogue or making it available via API.

- Announce the product to intended consumers with a short "what it does, where to find it, who to contact" note
- Set up monitoring: track usage, freshness, and error rates
- Schedule a 30-day check-in with consumers to gather feedback
- Plan the next iteration — a data product is never "done"

The first version should be deliberately minimal. Ship the 80% that covers the core use case, get feedback, then iterate.

Common Traps (and How to Avoid Them)

Building before scoping

The most expensive mistake is building a technically perfect dataset that nobody uses. Always validate demand before writing code. A 30-minute conversation with two or three consumers will save weeks.

Boiling the ocean

Your first data product should not attempt to unify all customer data across 14 source systems. Pick one domain, one use case, one consumer group. Prove the pattern, then expand.

Treating it as a one-off project

Data products need ongoing care — schema updates, quality monitoring, consumer support. If your organisation treats this as a "build and forget" exercise, quality will degrade within months.

Skipping the metadata

A data product without documentation is a burden, not an asset. Every field should have a description. Every product should state its update frequency, grain, and known limitations.

Ignoring the human side

Governance, naming conventions, and ownership agreements are harder than the technical build. Invest at least as much time in alignment conversations as you do in code.

Perfectionism

Ship a working product with five measures, not a planned product with fifty. The feedback you get from real users is worth more than any design document.

Data Product Readiness Checklist

Use this checklist before you publish. If you cannot tick every item, you are not ready to ship.

Category	Check
Scope	Business question is documented and validated with consumers
Scope	Grain is defined and agreed
Scope	Out-of-scope items are explicitly listed
Ownership	Named owner is assigned and has accepted accountability
Ownership	Escalation path for quality issues is defined
Build	Source-to-target mapping is documented
Build	At least 3 automated validation checks are in place
Build	Business logic (calculations, mappings) is documented
Metadata	Every field has a description
Metadata	Update frequency is stated
Metadata	Known limitations and caveats are documented
Publish	Product is registered in catalogue/marketplace
Publish	Consumer notification has been sent
Publish	30-day feedback check-in is scheduled

What Comes Next

You have scoped your first data product, assigned ownership, built and tested it, and published it for your consumers. That is a significant milestone — most organisations never get this far.

But a data product is a living thing. The next challenge is keeping it healthy: monitoring quality, managing schema changes, onboarding new consumers, and eventually deciding when to deprecate it. That is the subject of the companion guide, **Data Product Ownership Playbook**.

Want to go deeper?

I run a 2-hour workshop that takes your team from "we should do data products" to a concrete scope, ownership model, and 90-day delivery plan. Typically done with business and IT stakeholders together.

Get in touch: brucedando.com/contact

© 2026 Bruce Dando. All rights reserved. This guide is provided for personal and organisational use. Redistribution or resale is not permitted without written consent.