

Intelligent SAP BW Migration

Move the meaning first and turn
months of migration into weeks.



**A paper on the end-to-end journey from
legacy BW to governed, AI-ready data products.**

Read how One Data executes a top-down, semantic migration that turns the most complex SAP BW cloud migrations into a structured, accelerated and trust-centered project.

Why Migrations Fail and How to Avoid It

A Summary



Traditional SAP BW migrations recreate technical debt.

They move thousands of objects line by line into a new environment, carrying dormant assets, undocumented logic and inconsistent definitions at full cost. Studies by Horváth¹ and ISG² independently confirm it: over 60 percent of SAP transformations exceed both budget and schedule, and only 15% finish on time. The root cause is not technical execution but a knowledge gap: teams start moving data before they understand what it means, why it exists, or whether it is still needed.

One Data takes the opposite approach.

Instead of translating objects one by one, the platform reconstructs business meaning first: it extracts metadata from the legacy BW landscape, maps lineage at object and field level, and surfaces the business logic buried in ABAP routines. That semantic layer lets teams see which assets are still active, which are redundant, and which can be retired, before a single byte is migrated. Only what still delivers business value moves forward, packaged as governed data products with explicit ownership, lineage and quality contracts.

^{*1} Horváth, "Business Transformation Unlocked", 2025

^{*2} ISG, "State of SAP Migration", 2026

In a live SAP BW-to-Snowflake migration, this approach compressed months of manual analysis and re-coding into weeks, with up to 80 percent less effort and cost, zero coordination bottlenecks, and a target environment that was measurably cleaner than the system it replaced.

More importantly, the governed data products that resulted are platform-independent and reusable. The exact foundation reporting, cross-entity consolidation and AI initiatives need to be trustworthy.

This paper walks through the end-to-end journey of an SAP BW migration in One Data: from metadata extraction through scope reduction, target design, automated code translation and validated go-live, showing at each stage how understanding the landscape first turns an open-ended risk into a scoped, predictable project.



15%

of SAP migrations
finish on time and
on budget

60%+

of SAP transformations
exceed budget
and schedule

~80%

less migration
effort and cost
with One Data

1

The Urgent Need for Migration



SAP has drawn a clear line under classic SAP Business Warehouse.

Every new capability currently flows into SAP BW/4HANA, SAP Datasphere and the SAP Business Data Cloud, not into classic BW. Mainstream maintenance for BW 7.5 ends on 31 December 2027¹, and optional extended maintenance at a premium on the maintenance fee, runs only to the end of 2030. Staying past that line means no new capability, growing security and compliance exposure once support stops, the phase-out of Excel-based BEx front-ends, and a rising bill to keep an aging system alive while the people who understand it become harder to find.

Knowing the deadline is easy, but meeting it on time is not.

A 2025 Horváth study² found that over 60% of SAP transformation projects exceed both budget and schedule, only less than 10% finish on time, projects run 30% longer than planned on average, and nearly two-thirds report significant quality deficits after go-live. ISG's 2026 study³ a year later reaches almost the same place from an independent sample group, still with close to 60% both over budget and behind schedule. Both point to the same root cause, and it is not technical execution: the problems are set in the planning phase of the migration project, where complexity is underestimated, scope expands unexpectedly, and governance across multiple vendors is weak.

*1 <https://www.sap.com/documents/2019/02/1ca4c940-3a7d-0010-87a3-c30de2ffd8ff.html>

*2 <https://www.horvath-partners.com/en/press/detail/study-shows-sap-s-4hana-transformations-rarely-go-as-planned-60-percent-exceed-budget-and-schedule-two-thirds-dissatisfied-with-result-quality>

*3 <https://isg-one.com/advisory/sap/2026-state-of-sap-migrations-report>

1

The deeper reason is that a BW migration is not just a data move but a re-engineering of logic.

SAP's modern platforms are SQL-first, so every ABAP start routine, end routine and field-level rule needs to be understood, matched to the business question it answers, and rebuilt.

A typical BW environment, which consists of 10 to 20 years of organic growth, thousands of InfoProviders, DataStore Objects and MultiProviders, much of it undocumented and built by people who have since left, makes that an intimidating exercise. The honest conclusion is that the SAP BW migration challenge is not primarily technical; it is a knowledge problem: what does the system do, why, and does it still need to?

That reframing is also an opportunity to modernize.

A migration on this scale only happens once a decade, and it is the rare moment to retire two decades of accumulated debt, standardize fragmented definitions and arrive in the cloud with data the business can truly trust. The alternative is twice as expensive: paying to re-engineer and host objects nobody uses and then managing the analytics and AI that depend on clean, well-governed data. ISG's research¹ is explicit that the value organizations expect from AI after a migration depends on precisely that foundation. This is a point this paper returns to at go-live.

^{*1} <https://isg-one.com/advisory/sap/2026-state-of-sap-migrations-report>

2

The One Data Approach in Brief

One Data inverts the traditional migration sequence.

Rather than translating objects one by one and hoping the result adds up, the platform first builds a unified semantic model of the entire legacy landscape from metadata alone. It provides a map that shows lineage traced from the query layer through CompositeProviders down to the underlying DataStore Objects, at both object and field level. Only then does anything move.

The vehicle for that move is the governed data product, a business-aligned asset that carries its own definition, the logic that produces it, its full lineage, its ownership, and the quality contracts that keep it reliable.

Because these products are defined from metadata before raw data is migrated, the organization knows exactly what to preserve, what to consolidate and what to retire before it pays to move a single byte.

Making metrics traceable and definitions explicit also forces inconsistencies to surface during modeling rather than after go-live, so trust becomes structural rather than aspirational.

	Conventional bottom-up migration	One Data semantic, top-down
Starting point	Inventory technical objects, convert one by one	Build a semantic knowledge layer from metadata first
Scope	Move everything, including dormant assets	Retire, consolidate and migrate only what serves the business
ABAP logic	Manual re-engineering, syntax-driven	AI-agent translates captured business intent to native SQL
Trust & quality	Source inconsistencies migrate unchanged	Data contracts gate every product before go-live
Cloud cost	Full cost + recurring hosting of unused objects	Reduced scope lowers migration and run cost
When issues surface	After go-live as rework and overruns	During modeling, before any data moves

Illustrative comparison based on One Data methodology and independent findings on migration overruns.

Move the meaning first. The data will follow.

The companion paper [Beyond Code Translation: A Semantic Approach to SAP BW Cloud Migrations ↗](#) details that methodology.

The rest of this paper shows how One Data operationalizes it, end to end.

3

The End-to-End Journey

Turning the semantic approach into a working migration over a large BW landscape calls for a platform built for the task. One Data delivers full transparency, governed data products and a validated target design before the migration itself begins. In practice, the work proceeds through five stages.

1

Extract & Load
Metadata

2

Optimize & Reduce
Scope

3

Re-design &
Translate

4

Pull & Run
Code

5

Validate &
Iterate

Stage 1:

Extract & Load Metadata

One Data connects to the legacy SAP BW environment and extracts its metadata automatically: system objects, ABAP source code, transformation logic, documentation and structural information. What matters here is the depth of extraction.

Lineage is captured at both object and field level – from the query layer, through CompositeProviders, into the underlying DataStore Objects – and the mapping rules, formulas and ABAP routines acting on each field are recorded together with the objects they belong to.

Logic and context stay together, which is precisely what a manual inventory easily loses. Imported into the platform and transformed into the platform's standardized metadata model, the result is a more detailed picture than the source system exposes on its own.

This step acts as the foundation for every decision that follows.

Stage 2:

Optimize & Reduce Migration Scope

With metadata extracted and standardized, the platform can now identify relationships across the entire system. The collected metadata is assembled into a semantically rich model built on graph-based relationships, semantic indexing and contextual retrieval. It shows an interconnected view of the whole landscape: which objects exist, how they relate, what logic they contain and what business purpose they serve.

One Data's Data Map then makes the relevance of every object visible. In every BW environment mapped, a substantial share turns out to be dormant: InfoProviders nobody queries, DataStore Objects orphaned by a report retired years ago, MultiProviders superseded and never removed. A conventional migration moves them anyway and pays for them multiple times: to re-engineer their logic, to test them, and in ongoing cloud hosting costs. Retiring dormant assets before the move is the cheapest saving in the entire migration project, and it is only possible if you can see which objects they are.



One Data

Stage 3:

Re-design & Translate

Dormant objects are identified and the migration scope is reduced to what the business uses, so now the design phase can begin on solid ground. Architects and business stakeholders design the target by deciding for each object to either preserve, consolidate or retire it.

One Data provides a recommendation grounded in its understanding of the landscape, so teams start from an informed position rather than a blank sheet. A guided migration assistant within the platform drives this phase, mapping the source system into One Data and generating data product drafts that business users refine.

Within the Use Case Builder, teams choose which objects to migrate, set the target system and build the flow toward clearly scoped data products. All data products include defined ownership, quality standards, data and governance contracts and business context.

Not every object needs to become a fully governed product. The platform represents intermediate stages in form of a bronze, silver and gold layering, so that objects needed for the migration but not destined to be products still have a defined place in the design. Nothing has moved yet.

Translation follows directly from that blueprint. AI agents turn each specification into executable, target-native code, and the ABAP-to-SQL re-engineering that consumes months of manual effort is compressed into a fraction of it. Because the agent works from captured business intent rather than ABAP syntax alone, what it produces is clean, optimized SQL or target-platform-native code.

Architects keep full control throughout, supervising and adjusting the output: the automation removes the labour, but not the control.

Stage 4:

Pull & Run Translation Code

Now that the target design is settled and every transformation specified, the focus shifts from what to build to how to deliver it safely. Generated code is never pushed into a target system unchecked. It passes through a version control system that acts as both a review layer and a security layer, and the translation agent works iteratively, searching for the best result before it is delivered.

The generated pipelines and artefacts are pushed to Git for engineers to pull, review and run on demand. Therefore, a team that wants a fully automated path has it, while a team that requires human sign-off before anything runs in production works from the same generated code.

Once the code is pulled and run, the pipelines are executed in the chosen environment. Because the target design and the data products were settled upstream, this stage executes a known plan; it is not where surprises are discovered.

Stage 5:

Validate & Iterate Results

With pipelines executed in the target environment, the final gate is verification against the standards set during design. Every pipeline that runs is validated against the data contracts defined during design.

The governance contracts established earlier – quality rules, schema requirements, service-level commitments – gate each data product before it goes live. A product that fails its contract does not pass but returns for adjustment. Because those checks were specified first, validation is objective and repeatable.

The target receives only what was planned, scoped and validated, and the loop is iterative: results are validated, refined and re-run until each product meets its standard, so the environment that goes live is measurably cleaner and more trustworthy than the one it replaces.

Built to stay independent

The business logic is captured in platform-independent terms which makes the choice of destination a deployment decision rather than a re-modeling exercise.

One Data generates target-native artefacts for the platforms teams move, like dbt-ready models and macros for Snowflake, lakehouse-native pipelines for Databricks, and governed data products aligned to SAP's own data-product direction for SAP Business Data Cloud, and Datasphere.

If the strategy changes later, the same governed products extend to a new environment without being rebuilt from scratch. This is the difference between a migration that locks you in and one that keeps your options open.

Example in Production:

SAP BW to Snowflake in weeks

A leading life science and technology company faced a labyrinth of inter-connected queries, DataStore Objects and composite providers powered by dynamic ABAP.

Treating each migration component as a governed data product, **One Data extracted all metadata from BW and ERP** ↗, exposed complete data lineage in the Use Case Builder, and let the team build selectively – even enriching products with sources from Azure Synapse.

Automation packages translated business-language requirements into dbt-ready code, complete with column descriptions, quality requirements and transformation logic, pushed directly to GitLab.

Work that would traditionally have taken months of manual translation and verification was accomplished in weeks, with full transparency at every step. And the company did not only migrate, but it modernized.

It arrived in Snowflake with a foundation for ongoing innovation.

4

After Go-Live: Lifecycle Management

An intelligent migration produces assets that outlast the migration itself and change how the organization works with its data from that point on.

The semantic model becomes the organization's living documentation of its data environment – transparent, and continuously reconciled against reality rather than frozen in a slide deck at go-live.

Alongside it, lifecycle management gives an ongoing view of which data products exist, who owns them, how they connect and whether they still match their contracts. This layer does not end at go-live; it governs the landscape as it evolves.

The added value lies in reuse. Because each data product is a self-contained building block, any team can discover it, understand it and re-deploy it into a new context without reconstructing the logic from scratch.

A metric defined and contracted for the migration becomes the same trusted metric a finance analyst, a supply-chain planner or a downstream data scientist consumes, and – because the logic is platform-independent – the same product can run on SAP Business Data Cloud today and another environment tomorrow.

The semantic foundation that de-risked the move becomes the basis for what comes next: reliable reporting, cross-entity consolidation, and initiatives such as process mining that are only as trustworthy as the data feeding them.

This is also what keeps the environment clean for AI rather than merely making it ready once. AI models and agents are only as reliable as the data they consume.

If that data carries undocumented assumptions or invisible quality issues, the outputs are untrustworthy.

ISG's research¹ is blunt on this point: the payoff from AI after an SAP migration depends on well-governed data and on integrating SAP with non-SAP sources.

And SAP's own direction, with the Business Data Cloud and the Databricks partnership², assumes exactly that foundation. AI readiness, understood properly, is not a milestone reached at go-live but a property the governance layer sustains.

*1 <https://isg-one.com/articles/index-insider---ai-s-payoff-after-an-sap-migration-will-depend-on-more-than-just-erp?searchTerms=sap+migration>

*2 <https://news.sap.com/2025/02/sap-databricks-open-bold-new-era-data-ai/>

5

Conclusion

Move the meaning first.
The data will follow.

The deadline is fixed and the industry's record is poor, but the two are connected. Migrations overrun because they are treated as a code-translation problem, moving two decades of debt into the cloud at full cost and finding the surprises only after go-live. Changing the sequence changes the economics.

Understanding the landscape first lets teams retire dormant objects before they incur re-engineering, testing and hosting cost. AI-assisted ABAP-to-SQL translation compresses months of work into weeks to up to 80% less effort and cost. Settling scope, ownership and acceptance criteria in the design phase closes the governance gaps behind most overruns and shortens the expensive period of running legacy and cloud in parallel.

And because the output is a set of platform-independent, governed data products, the organization is left with reusable, AI-ready assets rather than a one-time deliverable. The return is not only a faster migration. It is a foundation that keeps paying back.

The real migration is not from one system to another. It is from fragmented, undocumented data to data that is understood, governed and trusted – the foundation your KPIs, your reports and your AI depend on.

Your next steps to reach your migration deadline on time:

1. Map your BW landscape

from metadata to see which objects are still used, before you scope anything.

2. Define target data products

with explicit contracts and decide what to preserve, consolidate and retire.

3. Automate ABAP-to-SQL translation

from captured business intent, with architects keeping review control.

4. Validate against contracts

iterate, and govern the products through their lifecycle beyond go-live.

See it on your own landscape!

Experience One Data [here ↗](#)
and request a free X-Ray vision
of your data [here ↗](#)