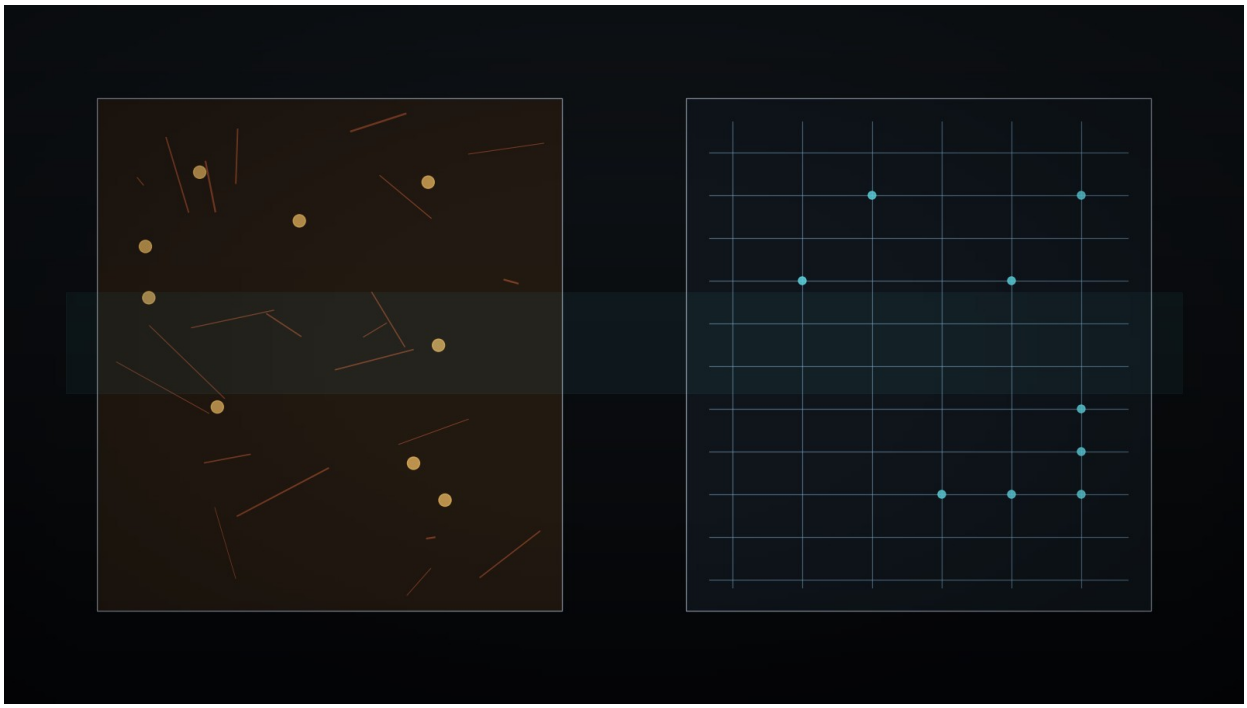
Custom Code and Legacy Complexity: What Happens to Your SAP Extensions in S/4HANA?

Every custom object built into an ECC landscape over the years deserves a documented decision — retain, redesign, replace, or retire — and it is the undocumented Z-code, not the documented kind, that quietly breaks migration budgets.



Every mature SAP ECC landscape contains custom code that nobody currently at the organization fully remembers commissioning. A pricing exception written for a customer relationship that ended years ago. A report built for a regulatory requirement that has since changed. An interface patched repeatedly to accommodate a partner system that was itself replaced. None of this code is inherently a problem — much of it, in fact, represents genuine, defensible business logic that reflects how the organization actually operates, distinct from SAP's generic standard processes. The problem is not that custom code exists. The problem is that most organizations cannot say, with confidence, which of it still matters.

An SAP S/4HANA migration forces that question to be answered, because custom code does not migrate automatically or safely by default. Every custom object needs a decision, and the decisions that get made by default — because no one had the information to make them deliberately — are where migration budgets and timelines most often break.

Why custom code is rated the hardest part of the migration

SAPinsider's 2025 benchmark research is direct about this: among all the tasks organizations rated for difficulty during their S/4HANA migration, adapting custom code scored 5.88 — the single highest-difficulty

rating in the survey, ahead of data migration, integration testing, and organizational change management. That is a meaningful data point, because it comes from organizations who have actually been through the process, not from vendors describing a hypothetical challenge.

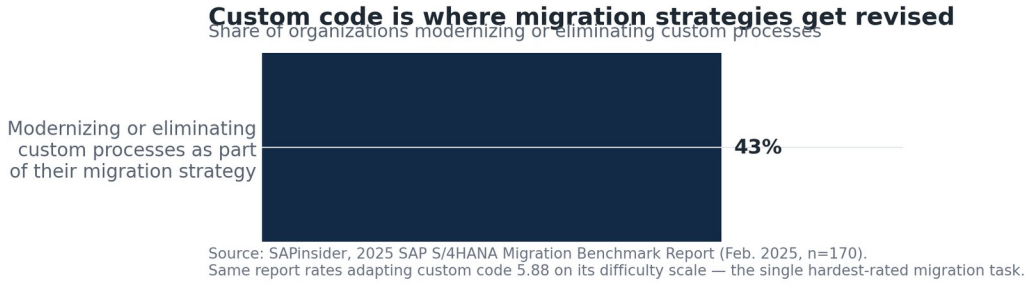


Figure 1. Custom-code modernization as part of migration strategy.

What this shows: Fewer than half of organizations are actively modernizing or eliminating custom processes as part of their migration — meaning for a majority, the custom-code decision is still being made implicitly rather than deliberately.

Part of what makes custom code difficult is architectural: S/4HANA's simplified data model and in-memory architecture mean that a meaningful share of ECC-era custom code — code written against database tables or logic structures that no longer exist in the same form — will not run unmodified after conversion, regardless of how well it performed for years in the legacy system. That is a technical reality of the platform shift, not a defect in the original code.

A four-way decision, applied deliberately

The organizations that navigate custom code well typically apply a consistent, deliberate decision to every custom object, rather than defaulting to "keep everything as it was" or "rebuild everything from scratch."

- Retain: the object reflects genuine, still-relevant business logic, is compatible (or can be made compatible) with S/4HANA's data model, and continues to earn its ongoing maintenance cost.
- Redesign: the underlying business need is real, but the object should be rebuilt to work correctly within S/4HANA's architecture, or reimplemented using newer, more maintainable tools such as SAP Fiori extensibility or the SAP Business Technology Platform rather than classical ABAP modification.
- Replace: the business need the custom object once addressed is now met natively by S/4HANA's standard functionality — a common outcome, since S/4HANA's simplified processes cover ground that required custom development in ECC.
- Retire: the object no longer serves a genuine business purpose, and it exists in the landscape purely as inherited complexity nobody has had reason to remove.

Making this decision well requires two things most organizations underestimate the need for: an accurate technical inventory of what custom code actually does — not what it was originally intended to do, which often diverges after years of undocumented patching — and business input on which of it still matters, since the technical team maintaining a custom object is rarely the same team that can judge whether the business logic it encodes is still relevant.

The cost of skipping the inventory

The predictable failure mode is treating custom-code assessment as a technical checklist completed quickly early in the project, rather than as a genuine cross-functional exercise. When that happens, one of two things tends to occur. Either too much custom code gets carried forward by default, because nobody made the case for retiring it, and the new system inherits the same complexity the migration was partly meant to reduce — reflected in SAPinsider's finding that only 43% of organizations are actively modernizing or eliminating custom processes, meaning a majority are not. Or, in the opposite failure mode, custom code gets stripped out aggressively to hit a timeline, and the organization discovers post-go-live that a "redundant" object was actually load-bearing for a process finance or operations depended on.

Both outcomes are avoidable with the same fix: treat the custom-code decision as a documented, business-reviewed exercise conducted early enough in the program that its findings can actually influence the choice between a brownfield, greenfield, or selective migration strategy — not as a cleanup task squeezed in after that strategic decision has already been made.

Next Step

Assessing which custom objects to retain, redesign, replace, or retire requires both an accurate technical inventory and informed business judgment — exactly the kind of evidence-based review Orpington Technologies' Full ERP Diagnostic Report is built to provide. From there, Orpington Technologies can either lead the custom-code remediation program directly through its Full ERP Implementation Partnership, or place certified specialists into an existing effort through ERP Staff Augmentation. Organizations facing a custom-code backlog are welcome to discuss which model fits their situation.

Sources

- SAPinsider, 2025 SAP S/4HANA Migration Benchmark Report — Detailed Findings, Feb. 2025 (n=170).