Custom ERP or off the shelf: when building your own pays back

Buy packaged ERP for generic work and build only where the process is how you win. A decision table, module sequencing and master data rules for owners.

Hashir I. Qureshi, Co-founder and managing partner

November 22, 2026 · 10 min read



In brief

Should you build a custom ERP or buy one off the shelf? Buy packaged software for the functions every business runs the same way, such as accounting, payroll and tax. Build or extend only where the process is how you win, and only if you can own the specification and the master data. Here is a decision table, how to sequence modules, and why AI on bad master data is a not yet.

Contents

01 Why ERP projects disappoint: what the latest research says

02 The decision rule: buy the generic, build where you win

03 Where we say no to a custom build

04 How to scope a custom ERP before anyone writes code

05 How to sequence ERP modules so each one pays its way

06 How to govern an outside ERP build team

07 How to protect master data during an ERP project

08 Where AI helps in an ERP, and where it is not yet

09 How an ERP decision runs when the team works inside the business

10 Frequently asked questions

Should you build a custom ERP or buy one off the shelf? Buy packaged software for the functions every business runs the same way, such as accounting, payroll and tax, and build or extend only where the process is how your business wins. A custom build pays back when a package would force you to change the workflow that earns your margin, and when you can own the specification, the master data and the team that writes the code.

This article is for founders, owners and CEOs replacing spreadsheets or a system the business has outgrown.

01

Why ERP projects disappoint: what the latest research says

The evidence on ERP projects points at decisions, not code.

Gartner predicts that by 2027, more than 70% of recently implemented ERP initiatives will fail to fully meet their original business case goals, and that as many as 25% will fail catastrophically. Its ERP strategy lead repeated both figures at Gartner's Barcelona Symposium in November 2025, as The Register reported, naming customization too wedded to existing processes among the causes. Gartner's research on avoiding disappointing ERP initiatives ties the gap to technology-centric approaches that ignore the people who use the system.

Panorama's 2026 ERP Report, released in March 2026, found that more than a quarter of organizations exceeded their project budgets, with additional technology needs the leading cause. Panorama notes that organizations often discover fatal misfits late, then turn to extra technology, scope expansion and custom builds.

The lesson is not "never customize." The expensive custom work is the unplanned kind: built late to patch a package that never fit, or built to preserve habits nobody questioned. The build that pays back is decided early, for the few workflows that set you apart.

02

The decision rule: buy the generic, build where you win

Every business runs two kinds of process. Some are the same everywhere: the general ledger, payables, payroll, tax and basic purchasing. Accountants and regulators set the rules, and packaged software has encoded them for decades. Building your own adds cost and risk, and customers notice nothing.

Others are how the business wins: how you grade a used item, cost a made-to-order job or price a quote. If your margin depends on doing these differently, a package that forces the standard way either erases the edge or gets customized until every upgrade breaks something.

A third option is often right: extend. Run a packaged or open-source core for the generic functions and build only the modules that set you apart. Four questions settle most cases:

1. Would a customer notice, or pay for, the way you run this process?

2. Does your margin depend on it, through pricing, costing or yield?
3. Have you tested a package against your real transactions, not a vendor demo, and found a workflow it cannot run?
4. Can you own a specification and govern a build team for years, not months?

If the first three answers are no, buy. If all four are yes, build or extend the part that differs and keep everything else packaged.

Custom ERP or off the shelf: the decision table

Question	Points to packaged software	Points to a custom build or extension
Is the process how you win?	No. It is accounting, payroll, tax or standard purchasing	Yes. It is your pricing, costing, grading or fulfillment logic
What did the fit test show?	The package runs your real transactions with configuration	Core workflows fail, or adapting the package would take longer than building
What does your data look like?	Standard products and units	Item-level variation: condition, compatibility, bundles, custom specifications
Who carries the system after go-live?	The vendor handles upgrades, security and compliance	A named owner for the specification and an agreed maintenance arrangement
How fast must it be live?	Fast, with a narrow scope that protects management time	A phased build works because each module pays its way as it ships
Example from our case library	A womenswear brand on an off-the-shelf ERP	An electronics retailer and a stone manufacturer on custom modules

03

Where we say no to a custom build

We say no to custom builds where the process is generic. No business wins on its general ledger, payroll or tax module, and those rules change with regulation, so a packaged or proven open-source product should carry them. Even when our team designed a custom ERP for a natural stone manufacturer, it integrated proven open-source accounting rather than building accounting from scratch.

Sometimes the honest answer is that nothing should be built. A womenswear brand could not say how many units it held of any design, or at what quality. We recommended an off-the-shelf ERP rather than a custom build, and left advanced costing and complex routing out on purpose. In about five months the brand had its first accurate, graded inventory, live stock visibility between warehouse and storefront, and monthly profit and margin reporting for the first time.

We also say not yet to any build while the product and customer data underneath it is unreliable. A new system on bad master data is the old mess, running faster.

04

How to scope a custom ERP before anyone writes code

Scope decides cost, timeline and adoption. No build team can rescue a project scoped wrong.

- **Run the fit test with real work.** A pre-owned electronics retailer had chosen a packaged ERP pilot for its cost, and it stalled at the warehouse and operations end. Our team treated the stalled pilot as a diagnostic: it showed which workflows the package lacked and where adoption broke down, and that justified a custom build.
- **Write scope as workflows and decisions, not features.** For each module, name who uses it, which decision it supports and which records it creates. A feature list grows without limit.
- **Write down what is out of scope.** An ERP decision is a scoping decision before it is a software decision. Leaving features out on purpose protects management time.
- **Agree the outcomes first.** Name the numbers the system must move, such as hours of manual reconciliation or stock accuracy, and how each will be measured.

05

How to sequence ERP modules so each one pays its way

A big-bang launch bets everything on one cutover. Sequencing modules lets value arrive early and problems surface while they are small.

1. **Master data first.** At the electronics retailer, the product database for more than 20,000 items was redesigned in about 15 days and became the foundation for inventory, procurement and reporting.
2. **Then the module closest to the margin.** At the stone manufacturer, timestamps on every production stage fed time-driven activity-based costing, and a quotation module tested each quote for profit before it went out. The custom ERP gave the manufacturer true order costs within six months, and pricing and order selection built on that costing contributed an estimated 2 to 3 percentage points of the business's 10-point EBITDA expansion.

3. **Release in short cycles.** The retailer's modules shipped in two-week sprints, each in users' hands before the whole system was finished. The team delivered 22 custom ERP modules in 37 weeks against a 44-week plan, and manual reconciliation fell by an estimated 40 to 60%.
4. **Roll out one site or function at a time.** The manufacturer moved site by site with production running throughout. Orders were disrupted during cutover, and delivery recovered. Plan for that dip and tell staff and customers up front.

06

How to govern an outside ERP build team

Few owners want a permanent software department. The stone manufacturer's owners wanted a one-time investment and had no IT capability, so a specialist agency wrote the code while our team owned the specification and governed the build. The same rules apply whoever you hire:

- **Keep the specification on your side.** The person who decides what gets built should answer to you, not to the firm billing for the code.
- **Write user stories with the people who will use each module.** At the retailer, our team led the client's new IT team from user stories through development and rollout, and each department shaped the module it would use.
- **Measure working modules, not hours or percent complete.** Progress was judged by modules in users' hands, against the plan.
- **Own the code, the hosting accounts and the data.** Put it in the contract before the first sprint.
- **Design for whoever enters the data.** On the manufacturer's wet, dusty cutting floor, that meant bilingual, low-text screens, large touch targets, photo capture and rugged tablets.
- **Plan the handover before the build starts.** Decide who fixes faults and who changes the system. The manufacturer's system was handed over with periodic support and is still in use.

07

How to protect master data during an ERP project

Master data is the set of records every transaction points to: products, customers, suppliers, units, prices and bills of materials. When it is wrong, every report built on it is wrong, and nobody can see why.

- **Name one owner per data domain.** One person approves new products, another new suppliers. Shared ownership means no ownership.
- **Design the structure before you migrate.** For the retailer that meant condition, compatibility, bundles and accessories, item-level variation that standard product models in e-commerce rarely capture.
- **Count and clean before you trust the system.** The womenswear brand created an A/B/C grading standard, then counted and graded every unit against it.
- **Control the front door.** Limit who can create records, make key fields required and block duplicates at entry.
- **Write down the rules that live in people's heads.** At the manufacturer, the tacit knowledge of long-tenured managers was documented into standard procedures and a knowledge base.

08

Where AI helps in an ERP, and where it is not yet

AI is arriving in ERP whether you buy or build. Gartner predicted in August 2025 that 40% of business applications would include task-specific AI agents by the end of 2026, up from less than 5% in 2025. Packaged vendors are adding assistants and agents, and a custom build can call the same models.

Where AI genuinely helps today is narrow work on data that is already clean: reading supplier documents into draft records for a person to check, flagging duplicates in master data, and answering staff questions from written procedures.

AI on bad master data is a not yet. Gartner said in February 2025 that through 2026, organizations will abandon 60% of AI projects unsupported by AI-ready data, and that 63% of organizations either do not have or are unsure if they have the right data management practices for AI. A model reading duplicate products, missing costs and inconsistent units does not fix them. It produces confident answers on top of them. The womenswear brand's demand-planning model was built directly on its graded inventory. Any forecast, AI or not, needs that foundation.

Hold AI inside the ERP until each master data domain has a named owner, there is one record per product, customer and supplier, costs are known at the order level, the procedures the AI will follow are written down, and a person is assigned to review what it produces.

Once those hold, our AI and intelligent automation work starts from one narrow workflow with an owner and a target. Our guide to the five signals that mean AI is not yet the answer covers the wider test.

09

How an ERP decision runs when the team works inside the business

Armstrat is a Business Transformation Office (BTO™) that embeds forward-deployed engineers (FDEs) inside a client's business and focuses solely on delivering outcomes. ERP and systems programs sit within our business transformation work, alongside the operating model changes they depend on.

Most engagements begin with a diagnostic of two to four weeks. For an ERP decision, that is where the fit test happens, the build or buy line is drawn module by module, and outcomes and measures are agreed before work starts. A base fee covers delivery, a contingency is paid on results, and the team stays until the results hold. Ask Armstrat answers questions about fees and measurement, with sources.

10

Frequently asked questions

When does a custom ERP make sense?

A custom ERP makes sense when the process a package cannot run is the process your margin depends on, such as costing, grading or pricing, and a test with real transactions has confirmed the gap. You also need someone to own the specification and the data after go-live. Where the process is generic, such as accounting or payroll, buy packaged software.

Is a custom ERP more expensive than off-the-shelf software?

Usually more expensive up front, not always over the life of the system. Packaged software spreads development cost across many customers, but heavy customization adds cost at every upgrade. Panorama's 2026 ERP Report found that more than a quarter of organizations exceeded their ERP budgets, with additional technology needs the leading cause.

How long does it take to build a custom ERP?

It depends on scope far more than on technology. Built in modules and released in two-week cycles, a system delivers value before it is finished. In one Armstrat engagement, a pre-owned electronics retailer received 22 custom ERP modules in 37 weeks against a 44-week plan.

Can we combine packaged software with custom modules?

Yes, and it is often the right answer. Keep packaged or proven open-source software for generic functions like accounting, payroll and tax, and build only the modules that set you apart. In one Armstrat engagement, a stone manufacturer's custom ERP integrated open-source accounting, while costing, production tracking and quotation were built around how the business worked.

Should we add AI to our ERP now?

Only if the master data underneath is clean. On duplicate records, missing costs and inconsistent units, AI features produce confident wrong answers. Gartner said in February 2025 that through 2026, organizations will abandon 60% of AI projects unsupported by AI-ready data. Fix ownership and data first, then start with one narrow task and a person reviewing the output.

Who should own a custom ERP after it goes live?

The business should, even when an outside team writes the code. Name one owner for the specification, owners for each master data domain, and a maintenance arrangement agreed before the build starts. Keep the code, hosting accounts and data in your name. In one Armstrat engagement, a specialist agency wrote the code while Armstrat's embedded team owned the specification. The system is still in use.

Written by

Hashir I. Qureshi

Co-founder and managing partner, Armstrat

Hashir is Armstrat's managing partner. He runs senior client relationships, sets the firm's positioning and methodology, and decides which engagements Armstrat takes on, and which it turns down. Before founding Armstrat, he led restructuring across retail and supply chain businesses, owning the ERP, the cost structure and the operating cadence that runs across them.

About Armstrat

Armstrat is a Business Transformation Office (BTO™). A partner leads every engagement, and forward-deployed engineers (FDEs) work inside your business across five capabilities: business transformation; operations and supply chain; people and organization; growth, marketing and sales; and AI and intelligent automation.

Engagements are a base fee plus a contingency paid on results, not on documents. We use AI only where it earns its place, and we say not yet when a business is not ready for it.

Continue reading

- Business transformation
  armstrat.com/services/business-transformation
- AI and intelligent automation
  armstrat.com/services/ai-intelligent-automation
- How we work
  armstrat.com/about
- All case studies
  armstrat.com/work

Talk to a partner about where your business is stuck.

armstrat.com/contact hello@armstrat.com

Scan to contact us

Not sure where AI fits yet? The free AI Readiness Assessment takes about ten minutes:
armstrat.com/ai-readiness-assessment

© 2026 Armstrat. All rights reserved. This article is general information, not advice for your specific situation. The latest version is at armstrat.com/custom-erp-vs-off-the-shelf.