



DATASTREAM OMS⁺

As seen on NAW MDM Distribution Intelligence

The Distributor Tech Dilemma: DIY vs. Off the Shelf

Distributors face a critical choice between building custom software and adopting off-the-shelf solutions, each with distinct costs, risks and benefits. Here's a guide for determining which path is right for your organization.

Author: Keith Fatula
Vice President of Solutions Engineering DataXstream

Every distributor chasing growth eventually reaches the same fork in the road: Should we build our own technology system, or should we buy one?

It's a tempting question, especially for companies with complex workflows and deep product knowledge. The idea of designing an IT solution from scratch feels like you're in control. It feels like strategy or even digital transformation. But far too often, DIY software becomes a detour that leads nowhere.

According to a McKinsey-Oxford study, 17% of large IT projects go so badly they "threaten the very existence of the company." These events have budget overruns of more than 200%.

On average, large IT projects overrun

budgets by 45%, miss deadlines by seven months and deliver 56% less value than predicted, according to the same study. For distributors, where margins are tight and service reliability is non-negotiable, a failed or under-performing IT project wastes resources while potentially crippling fulfillment schedules and eroding customer trust.

Yet many still believe their needs are too unique and specialized for off-the-shelf platforms. So, they build.

The Lure of Custom Projects

A midsize distributor reaches a growth ceiling. Orders rise, channels multiply, and the legacy platform groans with every quote. At first glance, the answer seems obvious: Build a bespoke commerce and order-management system tailored to your workflows.

But ambition doesn't always equal payoff. Older infrastructure is often a drag on digital transformation efforts. Custom technology builds appeal to distributors for many reasons: control, fit and sometimes internal pride. But they also carry hidden costs and long-term risks.

When does building make sense?

A custom build can pay off under specific conditions:

- **Defensible differentiation:** If the custom workflow gives a distributor an edge that packaged software can't replicate, then it is worth pursuing. For example, an electrical distributor that





- builds custom control panels or pre-assembled conduit kits for contractors might need specialized logic to manage thousands of components by voltage, phase or application while ensuring accurate substitutions and tracking for compliance.
- **Long investment horizon:** A custom platform should be viewed as a living asset that will span 10 years or more. It requires steady capital and a dedicated product team capable of adapting as demands change. Without that ownership mindset, technical debt will quickly accumulate.
- **Dedicated engineering talent:** A reliable custom build depends on engineers with real-world experience delivering complex software, not just modifying ERP code. They need to design for scalability as order volumes grow, maintain security and adapt quickly when business needs shift.
- **Regulatory uniqueness:** Some distributors operate under rules that mainstream platforms rarely address. For example, a medical device distributor may need FDA Part 11 audit trails for electronic records. These mandates require controls that are part of the code rather than bolt-on checklists. Even then, a segment-specific provider may have accounted for these needs in their out-of-the-box offering.

- **Flexible go-to-market timeline.** Some distributors can afford a longer development timeline. They don't need immediate ROI and are willing to wait as the platform is built and fine-tuned. That allows them to control every detail and tailor it to future needs. But without that kind of flexibility, custom development delays can turn into setbacks rather than strategic wins.

A single weak spot can derail a custom initiative, whether it leads to missed milestones or compromises the customer experience. What started as a strategic investment can quickly become a drain on time and resources.

Why Buy?

The term "off-the-shelf" often gets a bad rap, conjuring images of rigid, one-size-fits-all software. But today's solutions are a far cry from that. Many are built to be configured, extended and integrated without starting from scratch or sacrificing fit.

Choosing software to minimize pain in your organization can improve five key areas:

- **Time to value:** Vendors use a proven implementation playbook, so your team reaches go-live faster than if you start from a blank playbook.
- **Total cost of ownership:** Distributors must weigh subscription fees against the ongoing expense of in-house talent such as engineers and cybersecurity specialists, along with the infrastructure that supports automated builds and deployments.
- **Ongoing support:** Third-party software vendors patch security vulnerabilities and add new features, so your internal teams can focus on customer-facing innovation.
- **Integration:** Off-the-shelf platforms ship with tested connectors that let core systems share data automatically. Your ERP, CRM, pricing engine, and warehouse management software link through configuration screens rather than handwritten scripts. Because the connectors absorb version changes, an upgrade in one application no longer breaks the flow of orders or data.
- **Scalability:** Cloud-based platforms adjust to handle spikes in activity, such as peak sales periods or new customer onboarding. Distributors no longer need to plan for costly hardware upgrades or worry about slowdowns when order volume suddenly increases.

These advantages become even clearer when you examine how real distributors have navigated the build versus buy decision. One example comes from buildings materials distributor Pacific Coast Supply.

Pacific Coast Supply once planned to modernize its SAP order workflows with new custom in-house software. The project stalled under rising complexity, so leadership turned to an already packaged application embedded inside SAP instead of bolted on top. The positive results followed quickly:

- Customer service representatives

mastered the new interface in four days; training previously stretched up to four months.

- Unused inventory levels dropped 30% after better visibility exposed dormant stock.
- Gross margins inched up 1% during the first year, a meaningful lift.

More importantly, PCS avoided the snowball effect of unfinished features and undocumented customizations. Leadership eliminated more than half of its legacy code modifications, clearing the path for smoother future upgrades.

The company's CIO candidly summed up the lesson: Attempting to replicate a mature product "at a higher cost" made little sense once they found an off-the-shelf option that matched company requirements.

DIY vs. Buy: Choose with Clear Eyes

Whether you build or buy, the answer should hinge on strategy rather than ego. Start by mapping goals against the realities of time, talent and capital. Test every assumption:

- Can the organization quantify how custom code creates revenue beyond what pre-built configuration offers?
- Will internal teams commit to multiyear roadmap stewardship, including security patching and user-experience refinements?
- How soon must the new capability begin paying back through efficiency or sales lift?

Answering these honestly often reveals that the right packaged software already solves 80% of the problem. Why not let your people focus on the 20% that elevates service and relationships?

The market is moving too fast for vanity IT projects. Customers judge each interaction and purchase by your responsiveness. Leaders who move with urgency set the standard other distributors are expected to match. Today, that usually means standing on the shoulders of platforms purpose-built for the job, then investing your resources where human expertise still outperforms code.

Which path lets you spend more time delighting customers and less time debugging the engine that gets you there?

