
WHITE PAPER

How to run a WMS, TMS or logistics ERP selection

Three passes, five dimensions, and the evidence to demand at each one.

For logistics service providers and industrial logistics operations.

IN ONE PAGE

A software selection tests five things, and only one of them is about features. Functional fit absorbs most of the evaluation time because it is the easiest thing to compare and the thing every vendor has prepared hardest for — but by the time three serious candidates are on the table, their functional coverage looks broadly similar. Technology, strategic alignment, the vendor as an organisation of people, and the way that vendor runs an implementation are what decide whether the choice still looks right in year three.

The order of work matters as much as the criteria. Set the bar first, derived from where the business intends to be in one, five and ten years, and score the systems and supplier you already have against it. Only then score the candidates, on the same scale. Only then ask for evidence — and choose a form of evidence that puts unfamiliar material in front of the vendor, because a scripted demonstration shows you only what was rehearsed.

What follows sets out the five dimensions, the questions worth asking in each, and the specific evidence to demand.

Most of the evaluation time in a software selection goes to the thing that discriminates least between the candidates.

A selection tests five things:

01	Functional fit	what the platform handles for your operation
02	Technology	how it is built, how it integrates, how it scales, how it ages
03	Strategic alignment	whether the vendor is going the same way you are
04	The vendor	who stands behind the platform, and for how long
05	Implementation method	how the vendor turns your operation into a working system

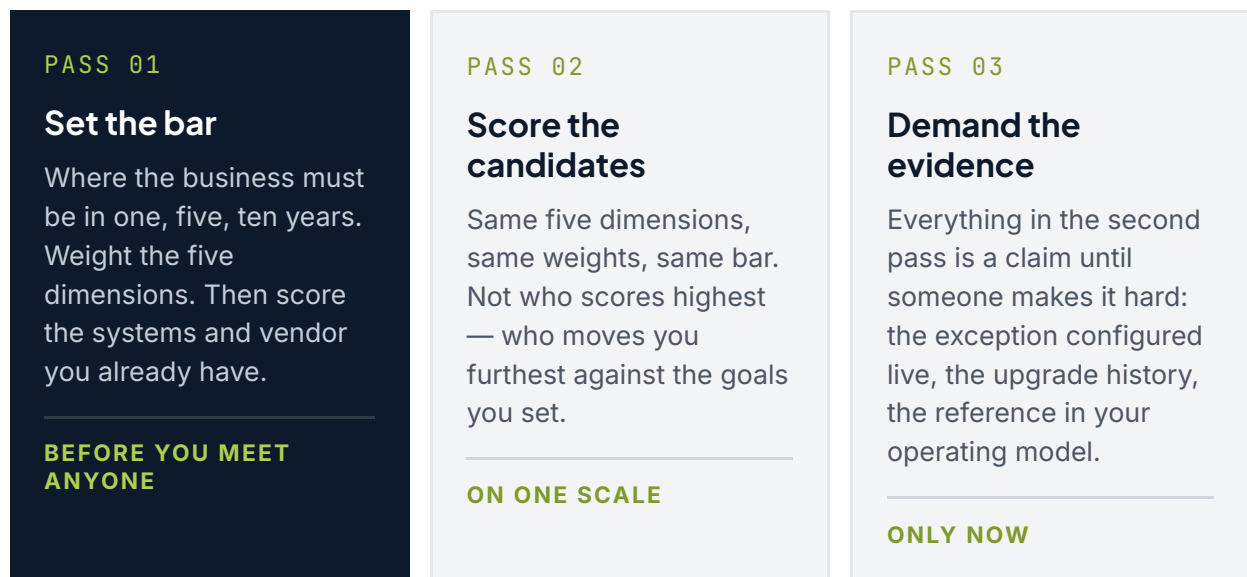
Only the first is about features — and it absorbs most of the room's attention, because the functional checklist is what every credible vendor has prepared hardest for, it is the easiest thing to put in a spreadsheet, and by the time you have three serious candidates their coverage looks broadly similar.

The other four decide whether you are still happy with this decision in year three.

1. Three passes, in this order

The most common process error is collecting evidence before deciding what matters. Demos get scheduled, the RFP goes out, references get called — and the criteria emerge afterwards, shaped by whatever the vendors happened to show.

Run it in three passes instead, and finish the first one before you meet anyone.



First pass — set the bar, then measure what you already have. Start from where you want the business to be next year, in five years, in ten. Everything else derives from that. What score would you need on each of the five dimensions to get there, and how much does each one weigh? Weights differ by role, which is why this is done explicitly and together: a CIO loads technology and implementation method, a COO loads functional fit, and both are right about their own column. Set the level that has to be cleared, then score your current systems and current vendor against it. Not a list of complaints — a measured distance between where you stand and what your own ambition requires.

Second pass — score the candidates on the same scale. Same five dimensions, same weights, same bar. The question is not who scores highest overall, but who moves you furthest against the goals you set in the first pass. A candidate who clears the bar on the two dimensions carrying most weight beats one who scores respectably on all five.

Third pass — demand the evidence. Only now. Everything in the second pass is a claim until someone makes it hard: the exception configured live, the upgrade history, the reference in your operating model, the requirement artifacts from a comparable implementation. That is what the right-hand column below is for.

The ordering is the whole point. A vendor can prepare for a demo — so decide what form the third pass takes, because that decides what you learn. A scripted demonstration shows you what was rehearsed. A workshop on your own processes, or a small proof of concept on your own data, shows you how the vendor behaves when the material is unfamiliar. Bring what they have not seen: the exception that never made it into the RFP, the flow that broke your last system, the question nobody asks. Functional evidence is the easy part to obtain. The other four dimensions only show themselves when something unexpected lands on the table.

WHAT IT TESTS	WHO USUALLY RAISES IT	THE EVIDENCE TO DEMAND
Functional fit what the platform handles for your operation	COO, operations lead	Have them configure your most awkward exception live, not describe it. Then ask who could have done that: their consultant, or your own team
Technology how it is built, how it integrates, how it scales, how it ages	CIO, CTO	The upgrade history of a heavily configured customer across three versions, and a security certification rather than a security slide
Strategic alignment whether the vendor is going the same way you are	CEO, CFO	The roadmap, and something they deliberately decided not to build — with the reasoning
The vendor who stands behind the platform, and for how long	CEO, CFO	A reference that operates like you do, spoken to without the vendor in the room, asked about the worst month of the project
Implementation method how the vendor turns your operation into a working system	COO, IT lead	The requirement structure and test scenarios from a comparable implementation, and the proposed split of configuration work for your case

Figure 1 — The five dimensions, the roles that raise them, and the evidence that settles them.

One question does not fit in a row, and I have put it at the end: what the vendor is doing with AI. It runs across all five.

2. Functional fit

Software that fits your operation at eighty percent and can be configured the rest of the way beats software that fits at ninety-five percent and cannot move.

The useful questions are not "does it have feature X":

- Where do our current systems slow the operation down, and where do they keep up?
- Which processes would we redesign if the software allowed it? Most operators have quietly shaped a process around a system limitation and stopped noticing.
- What do our customers come to us for that they could not get from the operator down the road — and how much of that sits in flows no standard system expects?

That last one determines how much configurability you need, which is a different question from how much functionality you need. It is routinely conflated with it. An operator running standard A-to-B transport, on processes that look like everyone else's, needs a system that does those processes well and little more. An operator that wins work precisely because it takes on the awkward handling, the exception flows and the requirements a customer could not place elsewhere needs a system that can hold all of that without a development project each time.

A second distinction explains a large share of failed selections: which kind of organisation the software was designed around. Some systems are built for shippers controlling their own transport — SAP TM, Oracle OTM — strong where transport is a cost line inside a wider enterprise estate. Others are built for operators who sell logistics: CargoWise leads internationally and is strongest in containerised forwarding, Descartes brings the widest network connectivity, Boltrics is strong in standardised warehousing. A third group sits between them and is served badly by both — the logistics arm of an industrial group, buying like a shipper and operating like a service provider.

A SHIPPER SYSTEM accounts for cost	AN OPERATOR PLATFORM has to know revenue, cost and margin per shipment — and invoice a different contract per customer
Invisible in a feature grid, unavoidable in year two.	

3. Technology

Start with what you already carry: where in the current stack is there maintenance debt that limits us — not what it cost to build, but what it costs to keep from breaking? Then four things sit under this heading, and the programming language

is not one of them.

Integration

A logistics platform has to exchange data with more outside systems than almost anything else you run: customers, carriers, subcontractors, agents, finance. Ask what the platform supports across the full range — file exchange, which is not obsolete and is still how a large part of the chain works, APIs, message queues, event streaming — and then ask the harder question about the tooling around it. When a new customer arrives with their own format, can your own people build and monitor that interface, or is every connection a vendor project with a price and a queue?

There is real money attached, though the published numbers come from a bigger world than most operators live in. [MuleSoft's 2025 Connectivity Benchmark](#), surveying 1,050 IT leaders in large enterprises, found development teams spending 39% of their time building custom integrations. A logistics operator of fifty to a few hundred people does not run hundreds of applications — it runs a handful of core systems and a tail of smaller ones around them. What travels from that research is the proportion, not the count: a serious share of whatever development capacity you have goes into keeping the seams between systems working. Every seam is a small permanent tax, and the selection is where you decide how many of them you are buying.

39%

of development time spent building custom integrations

<3 in 10

of those applications are integrated

Figure 2 — [MuleSoft 2025 Connectivity Benchmark Report](#), 10th annual edition, 1,050 IT leaders in large enterprises surveyed.

The data model

How the data is organised is an architectural question, not a reporting one, and it decides what you will be able to ask the system in three years. Whether an order, a handling and an invoice line point at the same underlying object or at three copies someone reconciles afterwards is the difference between a report you run and a report you rebuild.

Scale

Test the stack against your own plan rather than against a benchmark. If the intention is to grow substantially, to bring further business units, entities or countries onto the platform, can it carry that — and what has it already carried elsewhere?

Security

Increasingly the first thing your own customers will ask you about, which makes it your exposure and not only the vendor's. The question is not whether they take it seriously; everyone says yes. It is whether they can prove it: ISO 27001 certification, penetration testing by a third party, documented incident handling. Evidence, not assurance.

And what does "future-proof" mean for our operation? That one is usually answered with adjectives. Future-proof is not a property of software but a claim about how it behaves when your business changes — so test it against a change you can already see coming.

4. Strategic alignment

This one is not about where your business is going — you settled that in the first pass. It is about whether the vendor is going the same way.

A vendor's strategy is the long-term version of the other four dimensions: what is functionally deep today follows from where they invested over the last five years, and what will be true in five years follows from where they are investing now. The roadmap is the visible part. What they have decided not to build is the more informative part.

The uncomfortable thing here is that a vendor can be excellent and still be wrong for you.

Take one whose strategy rests on a deep configuration layer. They are asking you to invest more in the implementation, in exchange for a system your own people can reshape afterwards. If you have an IT team that once built your own systems and can carry that capability forward, the fit is strong and it pays back for a decade. If your processes sit close to the standard and you want the simplest, fastest implementation you can get — adapting to the software rather than the reverse — the same strategy costs you effort for a benefit you did not want. Both positions are defensible. They are not compatible with the same vendor.

Scope works the same way. Some vendors pursue vertical depth: a warehouse system that goes further into warehousing than anything else on the market. Others build one record across activities and partners and accept less depth in each. Neither is right in the abstract. The question is which of the two matches the way you intend to run the business over ten years — and if those two answers point in different directions, no amount of functional fit will fix it later.

5. The vendor

Ten years of selection, implementation, support and growth is a relationship between organisations of people, and the corporate material is designed to keep you from seeing that. Look through it.

Who is actually at the table, and who does the work

The people in the room during the selection are frequently not the people who will run your implementation or take your call in year four. Ask to meet the second group before signature. Whether that request is easy or awkward is itself informative.

Track record, in your context

Context beats count. A hundred references in a different operating model tell you less than three in yours — and if you are choosing between thousands of global references and a smaller number of companies that genuinely look like you, the second set is worth more, because you can call them and understand the answer.

Scale, in both directions

This cuts both ways and both risks are real. A five-person supplier may fit you perfectly and still leave you exposed: a thin bench, dependence on one or two individuals, nobody free when two customers need help in the same week. A global vendor gives you resources, continuity and an ecosystem of implementation partners, and gives you the experience of being one account among several thousand, in a support queue, on a roadmap you have no visible influence over. Decide in advance where on that range you want to sit. The practical version of the question: over ten years something will go badly wrong, and when it does, do we need to be able to phone a named person who can make a personal commitment about the fix — and at what vendor size does that stop being realistic?

VENDOR SCALE — BOTH RISKS ARE REAL

Five-person supplier

A thin bench. Dependence on one or two individuals. Nobody free when two customers need help in the same week.

Global vendor

Resources, continuity, an ecosystem of partners — and being one account among several thousand, on a roadmap you cannot influence.

Decide in advance where on that range you want to sit.

Whose interests stay aligned

Ownership structure, funding model and what happens to roadmap priorities after an acquisition are legitimate criteria, not impolite questions.

Whether you can disagree well

Every long implementation has a bad quarter — a missed date, a misread requirement, an invoice nobody expected. What decides the outcome is whether the two organisations can still reach a solution when the goodwill has thinned, and that is a question about culture rather than about contract. It is also testable: ask a reference to describe the worst month of their project and what the vendor did about it. A reference who cannot recall a difficult moment is either a very recent customer or a carefully chosen one.

Ask every vendor directly what their product does not do. One who cannot name a single boundary either has not thought about it, or plans to tell you later in the form of a change request.

6. Implementation method

This is the axis most often reduced to a Gantt chart, and the one where large implementations actually fail. Go-live is a date. What you are evaluating is whether the vendor is built to run a long, complex implementation, and that is a question about method: somebody has to understand your operation well enough to turn it into requirements that are structured, testable and complete, and then into a configuration that behaves the way the business really works. Ask how that translation is organised. Serious answers have a shape: work decomposed from milestones down through capabilities to individual features, each with test scenarios written from the business process rather than from

the software, each moving through the same phases — design, develop, configure, deliver — so that "done" means the same thing every time. Weak answers describe a methodology in adjectives and a plan in dates.

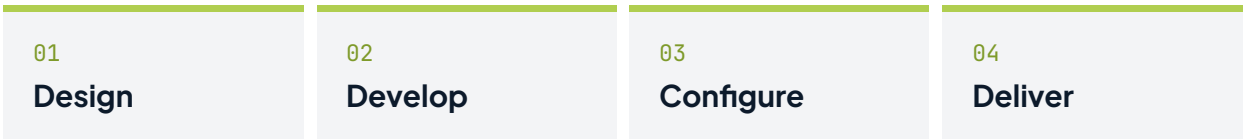


Figure 3 — Milestones decompose into capabilities and features; every feature moves through the same four phases, so "done" means the same thing every time.

Then the question that decides how the next two years feel: who configures what. There is no single right model. A customer with a strong internal team should be configuring a great deal themselves by the end; a customer without one should not be pushed into it, and should know what that costs. The split has to be designed against the capacity you actually have, written down before signature, and expected to move — mostly vendor at the start, progressively less.



That movement only happens if knowledge transfer is treated as a deliverable in both directions. The vendor has to learn your operation in detail, and your team has to learn the platform well enough to change it. Both usually happen by accident, if at all, and both show up in a single question to a reference: how much do you configure yourselves now, and how long did it take to get there?

And in logistics you cannot stop the trucks while you cut over, which makes the migration sequence a design problem rather than a project-plan detail. Adoption, margin visibility and reduced manual work are outcomes; define them before signature rather than negotiating them afterwards.

7. The question that runs across all five: AI

It is tempting to file this under technology. It does not sit there.

Strategically it is a question about rate of change: a vendor whose platform took three years to add a field will not absorb what is arriving now. Inside the vendor it changes how software gets built, tested and supported, and therefore what a change costs you in year three. In the implementation it touches how requirements, test scenarios and data mappings get produced. And functionally it is about what your own people get in the screens they use all day.

The gap between vendors experimenting and vendors shipping is wide right now, and it is not visible in a demo. What runs in production today, at a named customer, and what is roadmap? What did you stop doing manually in the last year? And what happens to our data — does it train anything, where does it run, who can see it? That last one is a procurement question as much as a technology question, and the answer is usually available in writing or not at all.

8. Everyone brings a different column, and that is the risk

DIMENSION	COO	CIO / CTO	CEO / CFO
Functional fit	■		
Technology		■	
Strategic alignment			■
The vendor			■
Implementation method	■	■	
AI	Belongs to everyone's column — and therefore to nobody's.		

Figure 4 — Where each role naturally enters the evaluation, and the question that falls between them.

A COO tends to enter at functional fit and implementation. A CIO or CTO at technology. A CEO or CFO at alignment and at the vendor. That is healthy. It is also where selections quietly go wrong: each person evaluates thoroughly within their own area and assumes the others are covered. AI is the clearest case — it belongs to everyone's column and therefore to nobody's.

The test of a sound selection is not that all five were examined, but that the answers were compared in the same room. The functional lead's enthusiasm and the CIO's reservations are data about the same decision.

Set the bar from where you want to be. Score what you already have against it. Then make everyone else prove it.



Karel Van den Berghe is founder and CEO of Globis Software, a platform for logistics service providers handling complex cargo flows. These are the questions we use on strategic software decisions ourselves, and the ones we expect to be asked. Disagreement welcome.