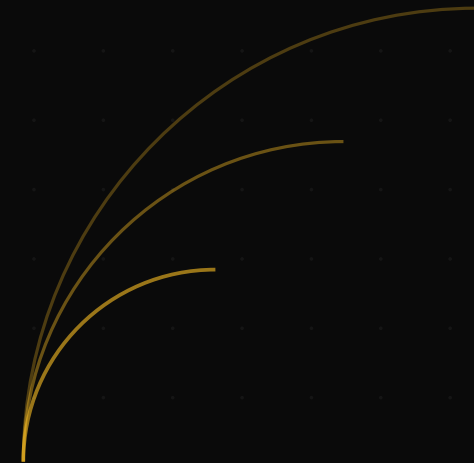


PROCUREMENT, IT, AND FINANCE LEADERS

How to evaluate a software vendor before you sign

A buyer's framework for evaluating a software vendor before signing: scope, hidden costs, lock-in, security, references, and the questions that expose risk.



Most software deals are lost before the contract is signed - not on price, but on scope.

WHAT'S INSIDE

7 steps in this guide

- 01 Write the scope before the vendor does
- 02 Find the costs the proposal leaves out
- 03 Price the lock-in before you're locked in
- 04 Separate the demo from the contract
- 05 Pressure-test security, data, and compliance
- 06 Take references that aren't on the vendor's list
- 07 Score proposals against your scope, not their pitch

01

Write the scope before the vendor does

The single highest-leverage move in any evaluation is to author your own scope first. When the vendor writes the scope, every ambiguity is resolved in their favour and becomes a future change order. When you write it, you set the requirements, the acceptance criteria, and the explicit exclusions.

A strong buyer-authored scope lists what must be delivered with testable acceptance criteria ("the integration syncs X records in under Y minutes with zero data loss"), names the exclusions you've seen vendors slip (data cleansing, environment setup, training depth), and defines the red lines you'll walk away over.

02

Find the costs the proposal leaves out

The quoted price is rarely the total cost of ownership. Across ERP, CRM, data, and custom-software deals, the same line items get excluded from fixed-fee proposals and resurface as billable work: data migration and cleansing, integration to your existing systems, additional environments, end-user training beyond a single session, and post-go-live hypercare.

For each exclusion, ask the vendor for a written dollar estimate now, while you still have leverage. "We'll scope that in a change order later" is the most expensive sentence in procurement.

- Data migration, cleansing, and legacy extraction tooling
- Integrations - each one needs a data contract, error handling, and UAT
- Non-production environments (dev, test, UAT)
- Training depth beyond train-the-trainer
- Post-go-live hypercare and named support
- Renewal uplifts and user-count true-ups

03**Price the lock-in before you're locked in**

Vendor lock-in is invisible at signing and only surfaces when you try to leave. Proprietary data formats, custom integrations, retrained staff, and egress fees all raise your exit cost. The defence is to make lock-in a number, not a feeling: project the switching cost at years 3, 5, and 7 and put it on the table during negotiation.

Ask explicitly: in what format is my data returned on exit, at what cost, and on what timeline? If the answer is vague, that's the lock-in.

04**Separate the demo from the contract**

What a vendor shows in a demo and what they commit to in the contract are two different documents. Demoware - features shown but never contractually committed - is one of the most common post-signature surprises. Keep a running list of every capability, SLA, and timeline claimed in the sales process, then check each one against the proposal. Anything demonstrated but not written down is a question, not a commitment.

05**Pressure-test security, data, and compliance**

Confirm where your data lives, how it's encrypted, who can access it, and what happens to it on exit. For regulated buyers, require the vendor to map their controls to your obligations (SOC 2, ISO 27001, HIPAA, GDPR) - not just claim certification. For AI vendors, this is its own discipline; see the AI-vendor guide.

06**Take references that aren't on the vendor's list**

The references a vendor provides are pre-screened to say good things. The signal is in the ones they don't offer: customers of a similar size, in a similar industry, who went live recently. Ask specifically about the gap between the sales promise and the implementation reality, and about anything that became a change order.

07

Score proposals against your scope, not their pitch

Lock your evaluation model - weighted criteria tied to your scope - before proposals arrive, so a slick presentation can't move the goalposts. Then score how much of your scope each proposal actually covers. Proposals that look cheaper usually cover less; a coverage score makes that visible and comparable.

PUT THIS TO WORK

Walk into every evaluation with the vendor's own playbook.

- A vendor-proof scope, issued before any proposal
- Gap analysis scoring every proposal against your scope
- A risk-weighted interrogation kit for the vendor meeting
- Scope-drift detection across proposal versions

Start free at benchsideai.com

Your first project is free. No card required.

