

The Platform Discovery Checklist

The questions to answer before you build or buy an enterprise platform capability — scope, constraints, integration depth, and coverage — in the order a discovery works through them.

The most expensive platform decisions get made before anyone writes a line of code — or signs a contract. A discovery phase isn't overhead; it's risk reduction. This checklist is the order a good discovery works through, from framing the problem to the build-buy-partner call.

Work top to bottom. Don't skip to vendor demos — that's step five, not step one.

1 — Frame the problem

- What problem are we solving, **for whom**, and by when?
- What does “good” look like as a number we already track?
- What happens if we do nothing for another two quarters? If the answer is “not much,” stop here.
- Who owns this outcome after the project team leaves?

2 — Map the constraints

For every workflow the capability must support, mark whether a candidate solution handles it:

- **Natively** — works out of the box.
- **By configuration** — supported, needs setup.
- **By custom code** — possible, but it's now your code to maintain.
- **Not at all** — a hard blocker hiding inside a feature list.

The “custom code” and “not at all” rows are where timelines and budgets actually go.

3 — Test integration depth

Integration is rarely the demo's strong suit. For each system you must connect to, ask:

- Can we **read** the data we need?
- Can we **write** back — update records, not just display them?
- Can we **trigger** workflows and react to events?
- Can we surface data **by role and permission**, the way the enterprise actually works?

Read access is a screenshot. Write, trigger, and role-aware access is a product.

4 — Score requirements coverage

- Which requirements does each option cover today, without a roadmap promise?
- Which components are **nearly impossible to build yourself** — and therefore argue for buy?
- For everything you'd build, estimate the time honestly, then add the maintenance tail.
- Do you understand both your **short- and long-term** use cases? You can't compare cost or coverage fairly without both.

5 — Make the build / buy / partner call

These aren't mutually exclusive, and they're not permanent:

- **Buy** the non-differentiating parts — the things a vendor does better than you ever will.
- **Build** the part that is your differentiation, the thing the business is actually paying you for.
- **Partner** to validate a concept fast, then bring it in-house once it's proven.

Write the decision down with the reasoning. In eighteen months someone will ask why, and “it seemed right at the time” is not an answer.

A discovery done well makes the build boring — which is the goal. If you're at the start of one and the stakeholders already disagree about the problem, that's the moment a neutral facilitator pays for itself.