IntegraCI

The Internal Developer Platform Buyer's Guide

Build vs buy, a vendor-neutral evaluation checklist, and the red flags to watch, for teams choosing an IDP.

A practitioner guide · integraci.co/resources

Why this guide

At some point every engineering org hits the same wall: shipping software has become a patchwork of scripts, tickets, and tribal knowledge. Every team wires up CI, security scanning, deploys, and compliance evidence a little differently. Onboarding a new service takes days. Audits take weeks.

The fix has a name now: an internal developer platform (IDP). The harder questions are whether to build or buy one, and how to tell a real platform from a good demo. This guide gives you an honest framework for both. It is vendor-neutral, use it against every option, including us.

Part 1 — Build vs buy

What building really costs

Building an IDP in-house is rarely a side project. Teams that do it well treat it as a product with its own roadmap, and that means real, recurring cost:

- **A dedicated team.** Golden paths, self-service, policy gates, and audit trails do not maintain themselves. A durable platform needs a standing platform-engineering team, indefinitely.
- **Time to first value.** Wiring one templated pipeline is a sprint. A governed platform every team trusts, with security and compliance built in, is measured in quarters to years.
- **Opportunity cost.** Every engineer maintaining internal plumbing is one not working on the product your customers pay for.
- **The long tail.** The platform is never done. New tools, new compliance regimes, new cloud targets, and the AI tooling now entering every pipeline all become your team's problem to integrate and govern.

None of this means building is wrong. It means the honest price tag is a standing team and a multi-year commitment, not a one-time build.

What buying gets you

The reason to buy is not to save a few scripts. It is to start from a mature baseline instead of assembling one, the parts that are expensive to build and easy to get subtly wrong:

- **Golden paths** so a new service starts with CI, scanning, and deploy already wired.
- **Policy gates** that apply the same rules to every change, so security and compliance are the default, not a manual review.
- **A tamper-evident record** of what shipped, who approved it, and which checks ran, so audit evidence is a byproduct of delivery.
- **Governance for AI in the pipeline**, so the coding agents entering your SDLC run under the same gates and approvals as a human change.

The false choice: rip and replace

Many buy decisions stall on a wrong assumption, that buying a platform means throwing out the tools you already run and standardizing on someone else's stack. It does not have to work that way. A **control plane** sits on top of the tools you already use: it connects to your CI, registries, scanners, and clouds, then orchestrates, gates, and records across them. You keep your tools and gain the golden paths, the gates, and the audit trail on top.

> The real question is not "whose platform do we adopt," it is "how do we get governance and self-service across the tools we already run."

Part 2 — The evaluation checklist

Decide your criteria before the demos, then score every option against the same list. Score each 0 (no), 1 (partial), or 2 (yes). Weight the rows that are hard constraints for you.

Criterion	What to ask	Score
1. Tool-agnostic vs rip-and-replace	Does this connect to what we already run, or replace it? A control plane over your existing stack beats a forced migration.	0 · 1 · 2
2. Governance built in vs bolted on	Is a policy gate a first-class feature, or a webhook I have to wire and maintain?	0 · 1 · 2
3. Runs on your infrastructure	If we had to run this fully on-prem or air-gapped next year, could we?	0 · 1 · 2
4. A real audit trail	Is it tamper-evident, and is it produced automatically as a byproduct of every run?	0 · 1 · 2
5. Golden paths that template the hard parts	What does "create a new service" actually give me on day one, CI, scanning, deploy, and the gates?	0 · 1 · 2
6. AI governance	How does this govern AI-authored changes, and where does the AI run?	0 · 1 · 2
7. Pricing model	What makes the bill go up, and is that aligned with our growth rather than taxing every seat or run?	0 · 1 · 2

Part 3 — Red flags

- **"Certified compliant out of the box."** No platform certifies you. The honest claim is built-in controls and evidence. See through the wording.
- **Rip-and-replace disguised as "opinionated."** Opinions are fine. Forcing a full tool migration to adopt the platform is a cost they are hiding.

- **No self-host story.** If they cannot answer the on-prem question, assume the answer is no.
- **Audit as a manual project.** If evidence is not automatic, your audits will not get easier.
- **Pricing that scales with usage of the platform itself.** You will end up rationing the thing you just bought.

Where IntegraCI fits

To be consistent, here is where we sit on the same checklist. IntegraCI is a control plane that connects to the tools you already run rather than replacing them. Governance, policy gates, tenant isolation, and a tamper-evident audit trail are built in, not bolted on. It runs on your infrastructure, including air-gapped. Golden paths template CI, scanning, deploy, and the gates together. AI-authored changes run through the same controls, on your own model and infrastructure. Compliance ships as policy bundles mapped to the frameworks you answer to, described as controls and evidence, not a certification claim.

> Pressure-test your own case: scope it at integraci.co/tools/quote-estimator or see how it works at integraci.co/how-it-works.