EYEMAGINE | POINT OF VIEW

Build vs. Buy in 2026

Rethinking the SaaS default for workflow, orchestration, and intelligence systems.

A whitepaper for operators and executives evaluating the next generation of workflow, research, analysis, and dashboard tooling — and whether buying the category leader is still the obvious answer.

Published by EYEMAGINE | eyemagine.com

Executive Summary

For twenty years, the default answer to “should we build this ourselves or buy a SaaS platform?” has been buy. The logic was sound: SaaS vendors had specialized engineering teams, scale economics, and a relentless release cadence that no internal IT organization could match. Custom software was expensive, slow, brittle, and prone to abandonment.

That logic is no longer uniformly true — especially for one specific class of software.

Workflow, orchestration, research, and analysis tools — the category that includes work management platforms, research hubs, analyst workbenches, operations dashboards, and intake-to-delivery systems — have entered a period of genuine strategic re-evaluation. Three forces have converged:

- **AI-assisted development** has collapsed the cost and time to produce working software by an order of magnitude for well-scoped problems.
- **Composable architecture and commercial data APIs** mean the hardest parts of building — auth, storage, search, messaging, enrichment, geocoding, financial data, identity — are now commodity building blocks.
- **SaaS pricing and lock-in pressure** have pushed many enterprise buyers into contracts where the sticker cost is the smallest part of the total bill.

This whitepaper argues that the build-vs-buy decision for this specific class of software deserves a fresh look, that most buyers are working from a mental model that was accurate in 2018 and is now outdated, and that the right answer for many organizations is neither pure build nor pure buy — it is a deliberate composition of the two.

We offer a decision framework, three common archetypes, and a practical way to de-risk whichever path you choose.

1. Why the Old Framework Is Broken

The classic build-vs-buy framework weighs four variables: cost, time-to-value, feature completeness, and maintenance burden. On every dimension, SaaS has historically dominated custom development for anything that isn't a genuine competitive differentiator.

That framework assumed two things that are no longer safe assumptions.

Assumption one: software is expensive to build.

For a generation, "custom" meant a minimum six-figure engagement, a six-to-twelve month timeline, and ongoing maintenance costs that grew over time. AI-assisted development has not made software free to build — but it has fundamentally changed the economics for well-scoped problems. A competent developer working with modern AI tooling can now produce in days what used to take weeks, and in weeks what used to take months. For a focused workflow tool with clear inputs, clear outputs, and a defined user base, the cost curve has bent sharply downward.

> *The question is no longer "can we afford to build it?" The question is "do we have the clarity to specify it?"*

Assumption two: SaaS is cheap and low-risk.

Anyone who has renewed an enterprise SaaS contract in the last three years knows the sticker price tells only part of the story. Per-seat pricing scales unfavorably as teams grow. Implementation services often cost more than the first year of licenses. Required integrations, premium modules, sandbox environments, and API call limits are frequently priced separately. And the true cost of abandoning a platform you've configured, trained on, and built process around is rarely in the contract.

We are not arguing that SaaS is a bad choice. We are arguing that "buy" is no longer the obvious choice by default, and that the honest comparison looks very different than it did five years ago.

2. The Four Categories of Software

Not all software behaves the same way in a build-vs-buy analysis. Before running the numbers, it matters which category a system falls into.

Category	Examples	Default recommendation
Commodity infrastructure	Email, calendar, video conferencing, file storage, identity, payroll	Buy. Always. The scale economics are insurmountable.
Regulated systems of record	ERP, general ledger, HRIS, compliance platforms	Buy. The audit, certification, and regulatory burden makes building a bad bet.
Workflow & intelligence systems	Work management, research hubs, analyst workbenches, ops dashboards, intake systems	It depends — and the right answer has shifted. This is the category this whitepaper addresses.
Strategic differentiators	Proprietary pricing engines, scoring models, customer-facing experiences tied to your brand	Build. This is where your moat lives.

The third category is the one that has moved. Ten years ago, workflow and intelligence systems behaved like regulated systems of record — too complex and integration-heavy to build from scratch, so you bought a platform and made your process fit it. Today, many organizations discover that they have bought a powerful generic platform and then spent eighteen months and seven figures trying to force-fit a workflow that was specific to them all along.

3. The Hidden Costs Buyers Miss

Every SaaS evaluation starts with per-seat pricing. That is almost never the largest line item in the three-year total cost of ownership. The costs that catch buyers by surprise:

- **Implementation services.** For enterprise work management platforms, implementation partners frequently charge $150K to $600K for initial deployment. For larger rollouts, seven figures is not unusual.
- **Integration work.** Every connection to a system of record, data warehouse, identity provider, or custom application is billable. Vendors advertise “500+ integrations”; most are shallow and require middleware for anything non-trivial.
- **Internal administration.** Mature platforms require dedicated admins — often one to three full-time equivalents at enterprise scale — to maintain configurations, user permissions, custom fields, and workflow logic.
- **The underutilization tax.** Industry surveys consistently show that enterprise SaaS platforms are used at 30–50% of the capability buyers paid for. The features that drove the purchase decision are often the ones that never get turned on.
- **Change management.** Training, adoption, documentation, and the productivity dip during rollout. Typically budgeted at zero, actually costs 10–20% of implementation.
- **Switching costs.** Once your process is encoded in a platform’s custom fields, forms, and automations, leaving is a multi-month project in itself. This is the real source of SaaS pricing power at renewal.

A realistic three-year TCO for an enterprise work management platform at a mid-sized organization is frequently three to five times the sticker license cost. Any honest build-vs-buy comparison should use that number, not the per-seat quote.

4. What Has Actually Changed in 2026

Three shifts have combined to reopen a question most organizations considered settled.

AI-assisted development

The cost and time to produce software for a well-defined problem has dropped dramatically. This is not a promise about some future state — it is how competent teams are already working. A focused internal application that would have required a five-person team and nine months in 2021 can now be built by a small team in weeks, often with a single senior engineer directing AI-assisted workflows. The quality ceiling has risen alongside the productivity gain.

The implication is not "build everything." The implication is that the scoping bar for build has dropped. Problems that were previously too small to justify custom development now sit comfortably inside the build envelope.

Composable building blocks

The components that used to make building hard — authentication, storage, search, messaging, payments, document parsing, data enrichment, mapping, identity verification — are now mature, well-documented commodity services. Building a modern workflow application today is less about writing software from scratch and more about composing proven building blocks. This dramatically changes the risk profile of a custom build.

Commercial data APIs

For workflow and intelligence systems, the most valuable capability is usually not the UI — it is access to the right data at the right time. Commercial data providers now offer API access to company data, market data, financial filings, news, geographic data, identity data, property data, regulatory filings, and industry-specific datasets. A custom-built research or analysis tool can integrate exactly the data sources that matter for its specific use case, without paying for the forty sources it does not need.

> *In 2026, a well-built custom workflow tool can often have better data than the generic SaaS platform it is replacing — because it pulls from sources chosen for the actual job, not the vendor's partnership roadmap.*

5. The New Build Case — And Its Limits

If the above makes building sound uniformly attractive, it is worth being equally honest about where build still loses.

Where build now wins

- **Proprietary workflows.** If your process is specific, evolving, or a competitive advantage, forcing it into a generic platform's configuration layer is strategic malpractice.
- **Integration-heavy environments.** If you will need five or more custom integrations regardless, you are already building middleware. At that point, building the application layer on top is often cheaper than paying platform fees.
- **Data-rich, feature-light tools.** Research hubs, analyst dashboards, operational scoreboards. The value is in the data model and the insight surface, not in thirty features nobody uses.
- **Small, specific user populations.** Per-seat SaaS is punishing for specialized tools used by a small but critical team. Build economics are much more favorable below 50 users.
- **Situations where switching costs dominate the TCO.** If you expect your needs to evolve significantly over three years, building preserves optionality.

Where buy still wins

- **Standard, well-understood processes.** If you need generic project management the same way ten thousand other companies need it, the category leader is a fine answer.
- **Large user populations with basic needs.** At thousands of users with simple requirements, SaaS economics often beat custom.
- **Compliance-heavy environments.** If SOC 2, HIPAA, FedRAMP, or equivalent certifications are required and you don't have a security team, buying a certified platform is usually the right call.
- **Organizations without technical capacity.** Custom software you cannot maintain is a liability, not an asset. If you will not have engineering ownership in eighteen months, do not build.
- **Genuine urgency.** If you need something running in the next thirty days, a configured SaaS platform is almost always faster than any build — even an AI-accelerated one.

6. A Decision Framework

Before signing a SaaS contract or commissioning a custom build, answer these seven questions honestly. The answers point clearly toward one path or the other in most real-world cases.

- **1. How specific is the workflow?** Standard → buy. Proprietary or evolving → build favors.
- **2. How many users, and how does that scale?** Under 50 favors build. Thousands favor buy.
- **3. How many integrations are required?** Zero to two → buy works. Five or more → build deserves serious consideration.
- **4. Where does the value live — features or data?** Features → buy. Data and insight → build favors.
- **5. Do we have or will we retain technical ownership?** If no, buy. Abandoned custom software is the worst outcome.
- **6. What is the realistic three-year TCO on both paths?** Include implementation, integration, admin, change management, and switching costs. Not just licenses.
- **7. How much will the requirements change?** Stable → buy. Evolving → build preserves optionality.

Answer all seven and the right path is rarely ambiguous. When the answers split evenly, the right call is usually a hybrid: buy the commodity layer, build the differentiated layer, connect them deliberately.

7. Three Archetypes

Most situations we see fit one of three patterns.

Archetype A: The over-featured platform

The organization buys a category leader because it was the safe choice. Eighteen months in, they are using perhaps a third of the capability, paying a premium for the rest, and their workflow has been reshaped around the platform's assumptions rather than their own. Renewal is approaching and the quoted increase is substantial.

Recommendation: Conduct an honest utilization audit. If you are using less than half the platform, run a competitive evaluation against both a lighter SaaS alternative and a scoped custom build. The goal is not necessarily to switch — it is to restore negotiating leverage and make an informed renewal decision.

Archetype B: The integration spiral

The organization bought a platform expecting out-of-the-box integrations to solve their data problem. Two years later they have built a custom middleware layer, hired an integration specialist, and are paying for both the platform and the people who make it work. The platform has become the least valuable part of the stack they built to support it.

Recommendation: You are already building. The question is whether the platform in the middle is still earning its keep. Often, replacing it with a purpose-built application on top of the integration layer you have already built is faster and cheaper than the next renewal cycle.

Archetype C: The differentiated workflow

The organization has a process that is genuinely specific — it is how they compete, or how they deliver something competitors cannot. They are trying to force that process into a generic platform and losing the specificity in the configuration.

Recommendation: Build. The value of the differentiator is greater than the cost of owning custom software. Buy commodity infrastructure around it — auth, storage, notifications, reporting primitives — and focus the custom investment on the part that actually matters.

8. De-risking Either Path

Both paths carry risk. Both can be managed.

If you buy

- **Run a real evaluation, not a vendor tour.** Build a scoring rubric against your actual requirements before the first demo.
- **Negotiate as if you will leave.** Contract terms around price protection, data portability, and exit assistance have teeth only if you negotiate them upfront.
- **Budget honestly.** Use three-year TCO including implementation, integration, admin, and change management — not the per-seat sticker.
- **Pilot before committing.** A paid three-month pilot against your real workflow is worth more than a six-week RFP.

If you build

- **Start with one workflow, not ten.** Ship a narrow version that solves one problem completely before expanding scope.
- **Buy everything that is not differentiated.** Auth, storage, search, monitoring, and infrastructure should be commodity services, never custom-built.
- **Ensure technical ownership continuity.** Either retain engineering capacity internally or contract for ongoing ownership — never hand-off-and-pray.
- **Write the requirements before you write the code.** AI-assisted development has collapsed the cost of building. It has not collapsed the cost of building the wrong thing.
- **Plan for iteration, not perfection.** Custom software's real advantage is that it can evolve with you. Budget and staff for that evolution from day one.

9. How EYEMAGINE Thinks About This

We are an AI-led digital transformation agency. We build custom software for clients, and we also help clients buy, implement, and integrate SaaS platforms. We have no structural incentive to prefer one answer over the other — which is exactly why our clients ask us.

Our position is simple: the build-vs-buy decision deserves the same rigor as any other major capital allocation. The default to buy, established when custom software was genuinely expensive and risky, is no longer automatically correct for workflow, orchestration, research, and analysis systems. The honest comparison in 2026 looks different than it did in 2021, and the organizations making the best decisions are the ones that update their mental model before they sign.

The right outcome is almost never pure build or pure buy. It is a deliberate composition: buy the commodity layer, build the differentiated layer, integrate them purposefully, and own the data that makes the whole stack intelligent.

> *The question is not "build or buy." The question is "what should we own, what should we rent, and where does the real value live?"*

Talk to us

EYEMAGINE helps operators and executives make clear, defensible decisions about workflow, orchestration, and intelligence systems — and then executes on whichever path the analysis points to. If you are evaluating a major SaaS purchase, renewing an enterprise contract, or considering a custom build, we can help you bring rigor to the decision and confidence to the recommendation.

Visit eyemagine.com or contact your EYEMAGINE advisor to start a conversation.