BUYING GUIDE

Four ways to buy AI that does real work

Getting AI to carry recurring work in your business takes more than a model. Someone has to design the job, connect your systems, set what the AI may do on its own, check that work is finished, control the running cost and keep it working as things change. The four common ways to buy differ mainly in who carries those responsibilities.

The four options at a glance

Option	What you get	Who carries the work	Best when
1. Self-service agent platform	A product with agents, connectors and approvals that your team configures	Your team, with vendor support	You have people with the time and skill to design, build and run it
2. Platform plus an implementer	The same kind of product, set up by a partner or agency	The partner for the build; ongoing work per contract	You have chosen a platform and want help going live
3. Internal build	Your own engineering on models, frameworks and your systems	Your engineering and operations teams	AI is core to your product, or you need full control and have the team
4. Managed AI workforce, such as AI Workforce by Ampity	AI employees designed, built and operated around your roles, on your accounts	The provider, with your managers, under agreed coverage	You want defined responsibilities owned without building an AI team

What this guide helps you decide

- **What to compare:** who designs, builds, checks and maintains each responsibility.
- **What to test:** a pilot on real work, with acceptance agreed in advance.
- **True total cost:** every cost line, divided by the work you actually accept.
- **Fit, exit and support:** when each option suits, what you keep if you stop, and who answers when something breaks.

> **A note on fairness.** This guide is written by Ampity. Each option can be the right choice, and we say where ours is not the best fit. Whichever you pick, compare complete proposals for the same work.

What to compare

Feature lists rarely separate good options from each other. Most serious products and partners can show agents, connectors, approvals and dashboards. What differs is who does the work around them, before and after launch.

Responsibility	Self-service platform	Platform plus implementer	Internal build	Managed AI workforce
Role and process design	Your team	Implementer or your team, per contract	Your team	Provider, with you
Integrations with your systems	Your team, using platform connectors	Implementer	Your engineers	Provider builds connectors and checks as needed
Authority and approvals	You configure	Implementer configures	You build	Written in a role charter, then configured
Checking work is complete	Platform features you set up	As built	You build	Checks against source records, where access allows
Monitoring and incident recovery	Your team	Per contract	Your team	Provider, under agreed coverage
Changes and retesting	Your team	Per contract or change request	Your team	Provider, within agreed change scope
Cost reporting per result	You assemble it	Per contract	You build it	Role budgets and cost per accepted result
Accounts	Your platform account	Your accounts, per proposal	Your provider accounts	Your provider and tool accounts

What to test in a pilot

- Run real work, including your awkward exceptions, not a curated demo set.
- Agree acceptance before the pilot starts: what counts as done, checked against which records.
- Count failed, retried and corrected work, and the time your people spend supervising it.
- Test authority. Confirm the AI acts alone where agreed and stops where it should.
- Change something mid-pilot, such as a field, a rule or a priority. Watch how it is handled, and by whom.
- Ask to see the handover documentation you would receive if the engagement ended.

Use the same pilot scope, volumes and acceptance rules for every option you test.

True total cost

A plan price is one line of the cost. Compare options on the total operating cost of each result you accept, at your volumes, not on token or seat price alone.

> **Total operating cost per accepted result**
>
> Plans, subscriptions and platform fees + model, API and extra usage + implementation and custom software + ongoing management + supervision and rework
>
> divided by the results that meet your acceptance rule

Cost line	What to ask	Why it gets missed
Plans, subscriptions and platform fees	Which plans, seats or platform fees does the proposal assume?	Plans are compared without the usage limits they include
Model, API, calling and extra usage	What is included, how is the rest metered, and what happens at a limit?	Retries add usage, and some platforms meter runs that fail
Implementation and custom software	What is built, by whom, and which parts are fixed or variable?	It is often priced apart from the platform
Ongoing management	Who monitors, fixes and changes it, and at what coverage?	It is assumed to cost nothing after launch
Supervision and rework	How much review and correction time is expected during and after onboarding?	Your people's time is not on any invoice

How to model it

- Use your real monthly volumes and current provider plans with their published limits. Do not assume unlimited usage.
- Ask which steps run as software and which need model calls. That mix drives usage.
- Ask whether subscriptions you own can cover supported work, and which tasks need API access, paid extra usage or more infrastructure.
- Include a realistic share of work that needs rework or a decision by a person.
- Divide by work that meets your acceptance rule, not by work attempted.
- Ask every provider to fill in the same model, so proposals compare like for like.

Compare on one sheet

Monthly, at your volumes	Option A	Option B	Option C
Plans, subscriptions and platform fees			
Model, API and extra usage			
Implementation and custom software, over the term			
Ongoing management			
Supervision and rework (your time)			
Accepted results per month			
Total operating cost per accepted result			

Fit, exit and support

When each option fits

Self-service platform. A good fit when someone has the time and skill to design, build and run it, and the work matches the product's features. Less suitable when no one will own it after setup.

Platform plus implementer. A good fit when you have chosen a platform and want help going live. Check who maintains it after the build and how changes are handled.

Internal build. A good fit when AI is core to your product or you need full control, and you can staff engineering and operations for the long term.

Managed AI workforce. A good fit when you want defined responsibilities owned and run for you, on your accounts, without building an AI team. Less suitable when you want to own and run every component in-house from day one.

Exit questions for every provider

- Whose name are the AI provider, tool and platform accounts in?
- What do you keep if the engagement ends: role definitions, examples, configurations, code and documentation?
- How are ownership and licensing of custom components set out in the contract?
- What does a handover include, and how long does it take?

Support questions

- Which hours and channels does support cover, and what response can you expect?
- What counts as a change, and how are changes scoped and tested?
- Who is accountable when a check fails or an integration breaks?

Where Ampity fits

Ampity designs, builds and operates AI employees around the roles you need, on accounts you own, under agreed support coverage and change scope, with documented handover. Repeatable steps run as custom software, not repeated model calls. Suitable subscriptions in your name can cover supported work within their limits, and any API access, extra usage or infrastructure is scoped and budgeted upfront. Costs are reported per accepted result against each role's budget. Commercial terms, coverage and ownership of custom components are set out in a written, scoped proposal.

Start with a role design session: hello@ampity.io · ampity.io/company/contact

This guide describes common buying approaches in general terms, not any specific vendor's product.