The monetization blueprint.

Design

Northlane

Index

Why design matters

Why it matters

Once you know who you serve, how you want to win, and what outcomes you promise, the next step is to design your pricing system. This is where intent becomes reality: packaging, pricing model, value metric, and price levels.

The option space is known. There are only so many ways to package features, only so many pricing models. The art lies in configuring those options to fit your ICP, GTM motion, and commercial ambition i.e. the Align phase. The right design makes pricing feel natural, accelerates sales cycles, and creates clear paths for expansion. The wrong design feels forced, confuses customers, and leaves money on the table.

Founder prompt

- When a prospect lands on our pricing page or hears our pitch, does the structure make it easier to buy — or harder?

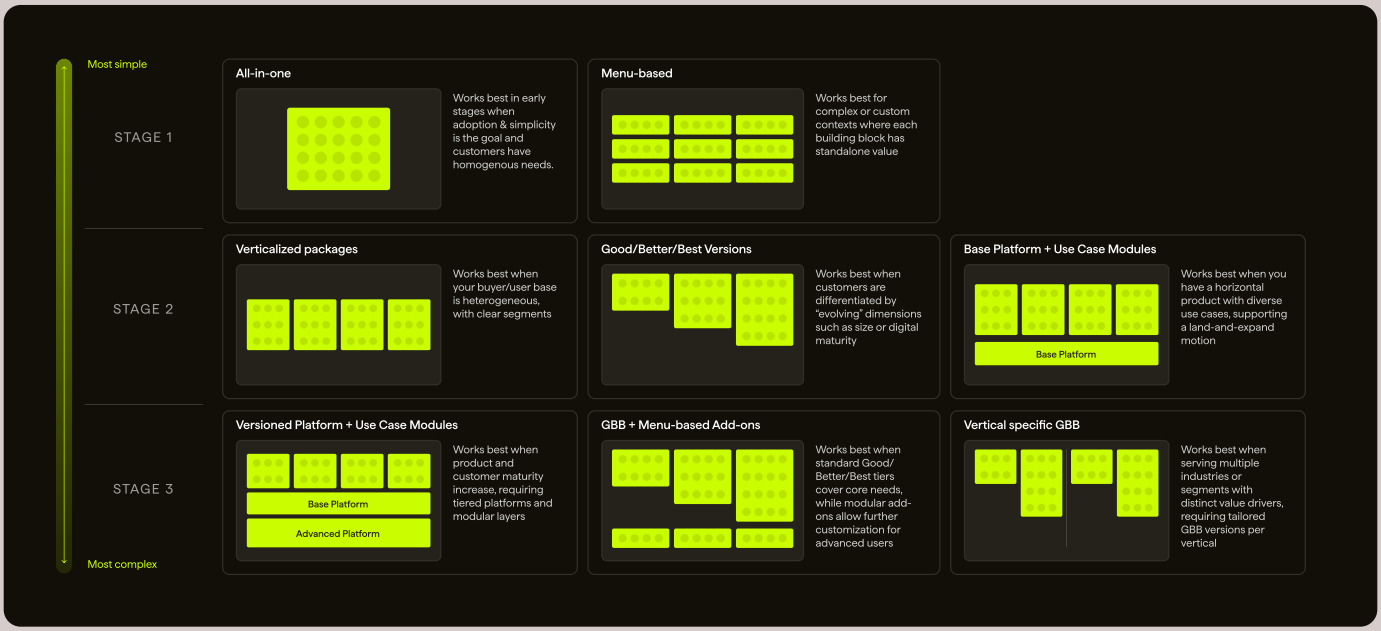
Packaging structure

Option space

Packaging is about how you bundle features and services into offerings.

- **Features** are elements you can toggle on or off in a binary way within your product.
- They differ from **metrics**, which often take a continuous form (e.g. number of seats, or volume of API calls).

The goal is for customers to recognize themselves in your packaging structure instantly. Speak their language, and sell the way they want to buy. If you can't, you've lost before you even start talking numbers.



Stage 1

Startups usually employ one or two packaging archetypes, with adoption as the primary commercial ambition:

- **All-in-one**: A single plan with everything included. Maximises simplicity, but leaves little room for upsell or segmentation → Works best in early stages when the product is focused and customers have homogeneous needs.
- **Build-your-own:** Menu-style, à la carte approach.High flexibility but also high complexity → Works best in complex/customised contexts (e.g. developer tools, services businesses, infrastructure), where each building block has standalone value.

Option space

Stage 2

As companies grow, we see an evolution to three other archetypes

- **Verticalized packages**: Different editions for distinct personas or industries. Enables targeting offerings to specific buyers and keeping value messaging relevant → Works well when the buyer/user base is heterogeneous, with clear use-case clusters.
- **Good / Better / Best**: The most familiar tiered model. Drives a natural upgrade path and creates price/value trade-offs. Premium packages should target segments with higher willingness to pay and demand for greater value → Works well when the customer base is differentiated along "evolving" dimensions such as size, digital maturity, or complexity.
- **Base platform + use-case modules**: A horizontal base package relevant for all customers, with optional add-on modules. Customers typically start with the base (and perhaps one module) and expand into more modules as their needs grow → Works well when you have a horizontal product with diverse use cases, supporting a land-and-expand motion.

Stage 3: Maturity

As companies mature, they often incorporate a combination of structures to capture the benefits of each. For example:

- **Versioned platform + use-case modules:** A tiered platform structure with modular layers. Customers typically start with a base platform and expand into advanced tiers and use-case modules as their needs and product maturity evolve → Works well when product and customer sophistication increase, requiring structured tiers and modular flexibility.
- **Good / Better / Best + menu-based add-ons:** A hybrid packaging structure that combines tiered editions with optional modular add-ons. Standard GBB tiers cover core needs, while add-ons allow deeper customization for advanced users → Works well when core segments share baseline needs, but power users require advanced functionality or flexibility.
- **Vertical-specific GBB:** Tailored versions of Good/Better/Best packages for distinct industries or customer segments. Each verticalized package reflects unique value drivers and use-case priorities. → Works well when serving multiple industries or personas with differentiated needs, requiring clear vertical messaging and positioning.

No single model is "best." The right choice is the one that aligns with your guidelines from Playbook 1 — ICP, motion, ambition, and value promise.

Founder prompt

- Do customers see themselves in our packages, or do we need to explain where they fit? Are we speaking their language?

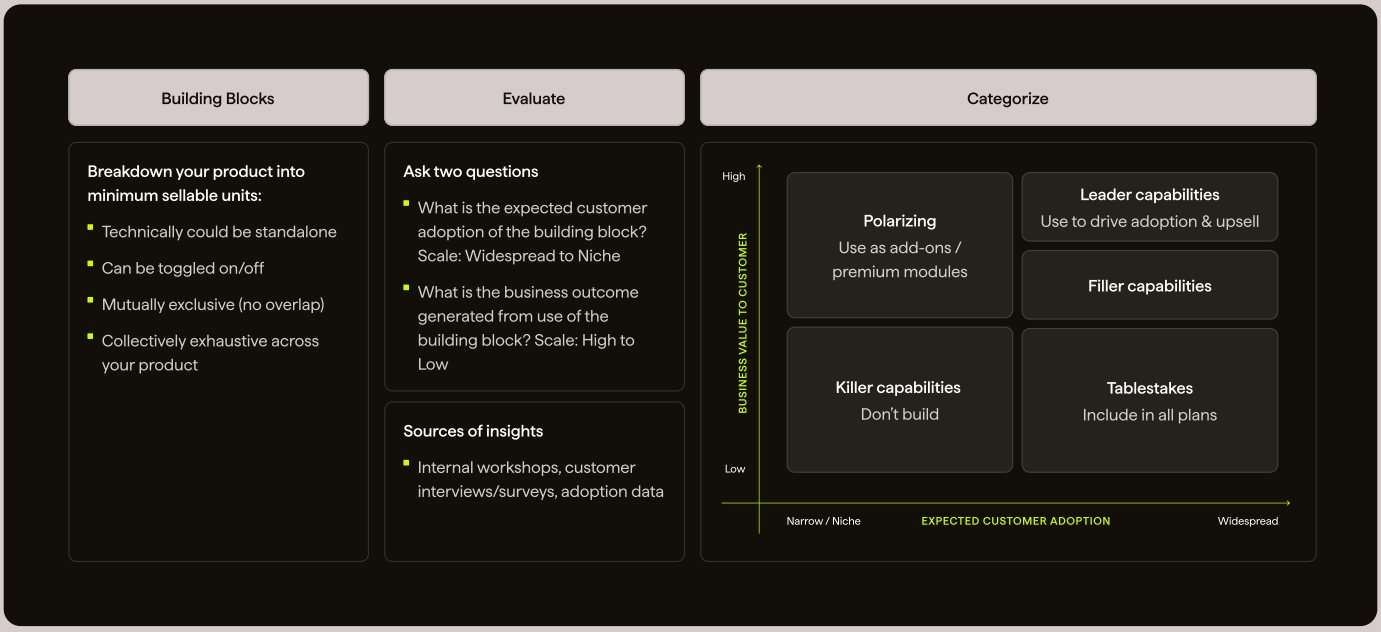
Feature placement

Role of building blocks

The next challenge is deciding which features go where. Strategic debates might include:

- **Goal**: Is the feature meant to drive acquisition, retention, or monetization?
- **Narrative fit**: Are segment needs homogeneous or heterogeneous? Do all customers need those features?
- **Spearheads**: Is there a single point of entry to our product? Or multiple?
- **Data**: What does usage data say? What features are most used? What do customers rank highest in willingness-to-pay surveys?
- **Stickiness**: Should we gate features that drive adoption and repeat use?

Some framing can help you structure the discussions. Your product, whether software, a service, or widgets, can be broken down into building blocks. Imagine these as Lego blocks that you can stack together to create your packages. Each of these blocks could technically be sold standalone, even though it may not be commercially astute to do so. categorizing these building blocks is the key driver towards determining your packaging structure.



Role of building blocks

For each building block, two questions can guide its role in your packaging:

- What is the expected customer adoption of the building block?

  Scale: Widespread (100%) to Niche (0%)

- What is the business outcome generated from the use of the building block?

  Scale: High [5] to Low [6]

Then you are ready to categorise:

- **Tablestakes** (High adoption, low business value): Necessary components that a customer would expect a vendor to include in any solution, but that don't directly drive business outcomes

  Implication → include in all packages by default

- **Leaders** (High adoption, high business value): Absolutely necessary, and customers would be willing to pay extra for it due to the business value to customer

  Implication → use strategically to drive upsell across packages, this is where willingness-to-pay is mostly extracted

- **Low relevance/value** (Low adoption, low business value): Consider removing these from your offering, and reducing investment

  Implication → why build & sell something of low relevance and value to your target segment?

- **Add-ons** (Low adoption, but high business value for those who adopt): Polarising building blocks that are not relevant for your whole target market

  Implication → monetise outside of core packages and monetise opportunistically:

- **Nice-to-have**: Essentially, anything that doesn't fit into the above 4 categories. Features that add some business value and are reasonably well adopted (>50%).

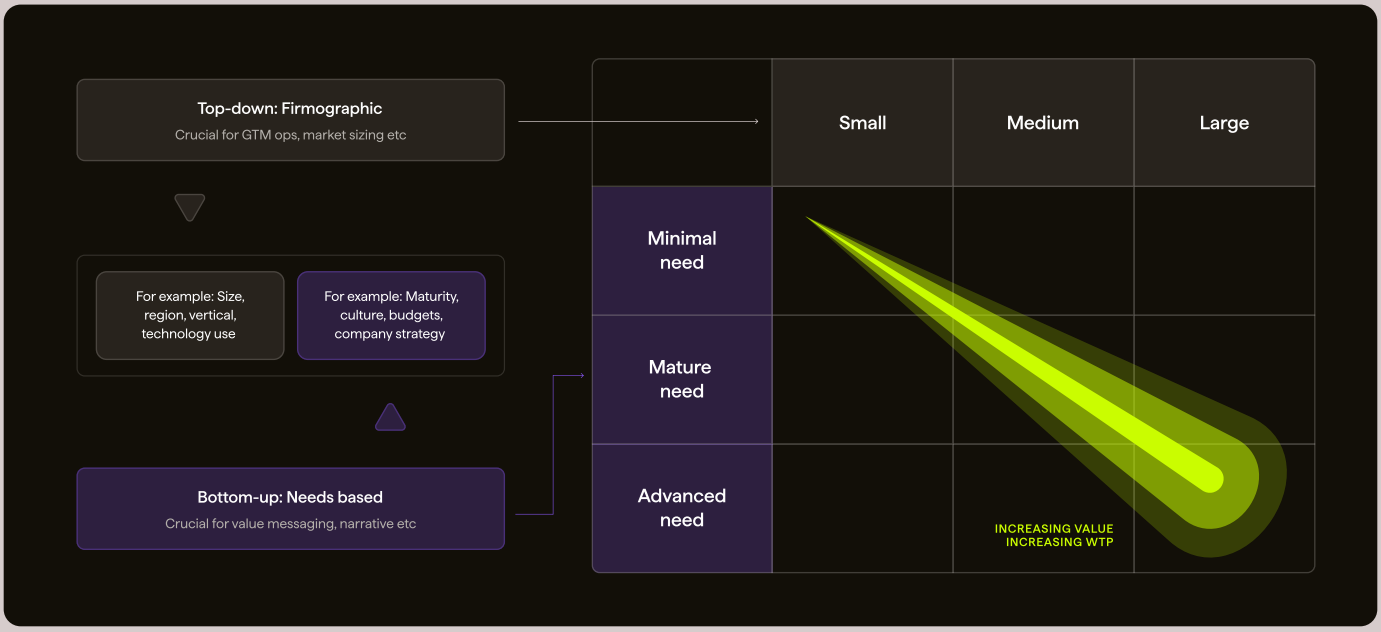
  Implication → offer as "fillers" within packages but don't focus your value messaging around them

Back to segmentation

As mentioned earlier, your segmentation should combine both top-down and bottom-up perspectives:

- **Top-down**: Based on firmographics such as size, region, vertical, and technology use. Core benefit is that GTM strategy and lead sourcing can be optimised with precise criteria.
- **Bottom-up**: Based on behavioural needs and jobs to be done.These needs often emerge from the building block exercise. If you see dispersion across customers (e.g. one person's "must-have" is another's "nice-to-have"), this indicates differentiated segment needs. Packaging structure should orient around these needs, supported by a value narrative tailored to the segmentation.

  Tip: Asking customers the "why" behind their adoption or value ratings is the key to unlocking bottom-up segmentation.



You must combine both segmentation dimensions to align your GTM strategy and packaging to both value and ability to pay.

Monetizing new features

New features

The same process applies whenever a new feature is designed.

- In the **launch phase**, there are a few options depending on commercial ambition:
  - Offer the feature as an add-on, while it gains targeted market feedback
  - Include it all in offers on a limited basis, to drive adoption
  - Include only in premium packages, to signal value from the outset
- After **initial validation**, decide how to incorporate it into the existing packaging structure using the same building block evaluation approach. If you choose to offer the feature within existing package(s), consider whether this warrants a price uplift.

Some framing can help you structure the discussions. Your product, whether software, a service, or widgets, can be broken down into building blocks. Imagine these as Lego blocks that you can stack together to create your packages. Each of these blocks could technically be sold standalone, even though it may not be commercially astute to do so. categorizing these building blocks is the key driver towards determining your packaging structure.

Founder prompt

- If this feature were to disappear tomorrow, which customer segment would stop buying our product? How much would they have been willing to pay to keep it?

Pricing model: The mechanics of monetization

Value metric

A strong value metric passes three tests:

- **Aligned to value** → Customers see it as a fair proxy for the outcomes you promise.
- **Scalable** → As customers grow and receive more value, their spend grows naturally.
- **Feasible** → Easy to measure, explain, and not gameable by customers.

	Shortlisted value proxies		
	Value proxy #1	Value proxy #2	Value proxy #3
Aligned to value Customers see it as a fair proxy for outcomes you are promising	—	✕	✓✓
Scalability As customers grow and receive more value, spend grows naturally.	—	—	✓✓
Feasibility Easy to measure, explain and is not gameable.	✓✓	✓✓	✕

Categories of Value Metrics

Metrics generally fall into the following categories:

- **Access metrics (seats, users, logins)**

  Simple and predictable.

  Risk: can limit adoption or undercut value (e.g. if AI reduces seat counts).

- **Usage metrics**

  Technical metrics (API calls, compute, storage) → Common in infrastructure where pricing must track variable costs. Harder to communicate value.

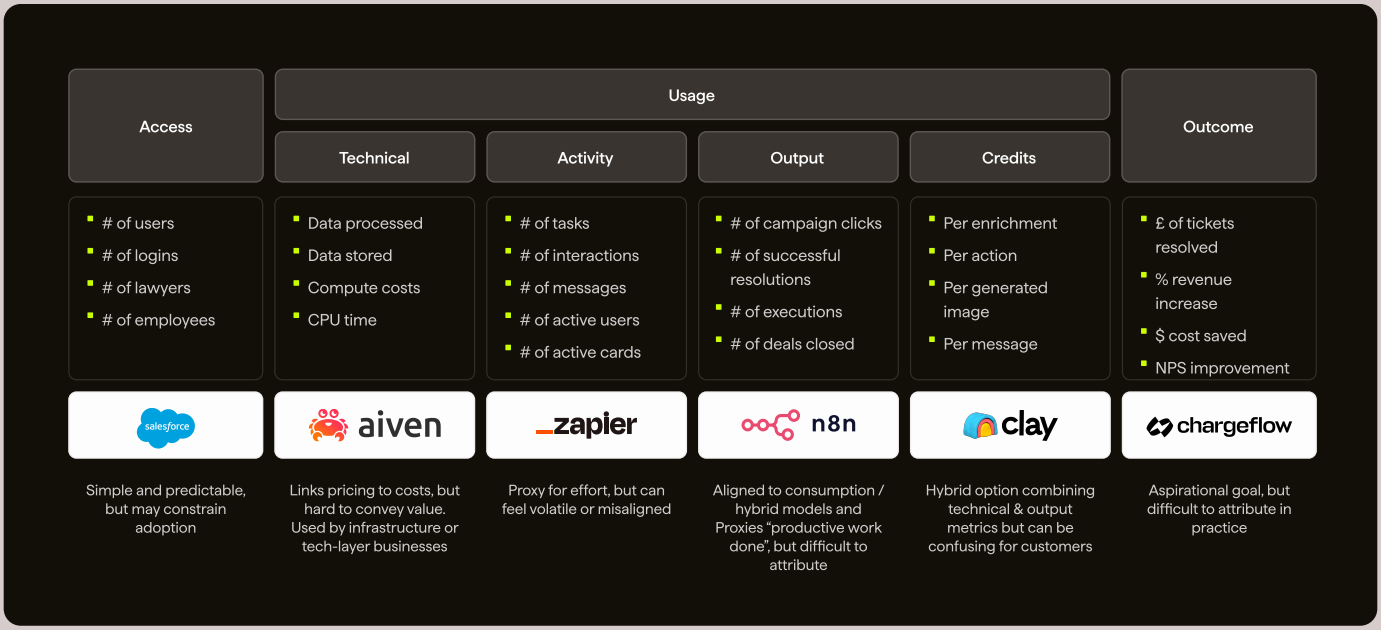
  Activity metrics (tasks, interactions, simulations) → Proxy for effort. Aligned to consumption but can feel volatile if not all effort is valuable.

  Output metrics (deals closed, successful resolutions) → Proxy for productive work. Better aligned to value, but attribution is harder.

  Credits: Allow you to combine technical, activity, or output metrics in a clean way. Risk: often difficult for customers to understand.

- **Outcome metrics (% savings, $ revenue uplift)**

  Ultimate alignment with value. Risk: attribution is complex, and the gap between product usage and business outcomes is often too large to measure precisely.



Evolution

Most companies start with a simple access or usage metric, then evolve toward a hybrid mix.

Some layer in secondary metrics to capture value differences across segments. Others use fencing metrics, where the metric acts as a cap or limit within a package rather than the primary driver of price.

Founder prompt

- If our product is wildly successful, will customers need more of what we charge for — or less?

The pricing model

There are only a few archetypes — the nuance lies in configuration.

- **Subscription (commit upfront)**: Customers pay for access in advance. Predictable and easy to budget. The classic SaaS model, usually paired with access metrics, but also possible with usage metrics via a drawdown model.
- **Consumption-based (pay retroactively)**: Customers pay only for actual consumption. Highly flexible, but bills can feel unpredictable. Carries cash flow risk for the vendor.
- **Hybrid**: A subscription with a minimum usage commitment, plus retroactive charges for overages. Now the most common model for scaling SaaS and AI. Balances predictability with fairness and shares commercial risk.
- **Outcome-based:** Customers pay based on results achieved. Strong alignment to

	Payment terms	Fees dependency	Volatility	Cashflow Risk	Industry fit	Buyer profile
Subscription Classic SaaS	Pay upfront	Fees based on forecast	Predictable	Customer bears cashflow risk	Traditional industries	Established companies
Hybrid New Status quo	Pay some upfront	Fees based on forecast, then actuals beyond forecast	Balanced	Balanced	Broad applicability	Mid-stage companies
Consumption En vogue	Pay in arrears	Fees based on actuals	Flexible	Vendor bears cashflow risk	Tech-savvy industries	Start-ups and scale-ups

Key Distinctions

The primary difference is when customers pay:

- **Upfront (subscription)** → predictability
- **After usage (consumption)** → flexibility

For AI products, where variable costs are real, hybrid models are winning. Some buyers prioritise predictability, others value flexibility — your pricing model must align with how your customers want to buy.

Usage limits, overage mechanics & allowances

Limit Options

The most overlooked aspect of pricing models is what happens when customers exceed their allowance or commitment for a specific usage metric.

- **Hard limits** → Common in free tiers, but also valued by enterprises to ensure spend predictability.
- **Hard limits with slowdown** → As the limit approaches, processing power is reduced so the threshold is never breached (common in technical AI products).
- **Hard limits with fair usage** → A hard limit is communicated, but with ~5% slack to reduce customer friction; beyond this, flex fees apply.
- **Soft limits** → Customers are notified when they exceed allocation, then billed flex fees (overages) on demand.

Overage Mechanics

When usage exceeds the limit, options include:

- **Auto-upgrade** → Move customers to the next tier after repeated periods over their allowance.
- **True-up** → Bill excess at renewal rather than mid-term.
- **True-forward** → Increase the customer's future commitment after overuse, waiving flex fees in exchange for the new commitment.
- **Top-ups** → Customers buy extra usage "packs" mid-term.
- **Rollovers** → Allow customers to carry forward unused credits from prior periods.

Best fits by model:

- Auto-upgrades → Work well in **PLG funnels.**
- True-ups & true-forwards → Common in **enterprise contracts.**
- Top-ups → Fit best in **self-serve models.**

Usage Allowances

For all consumption models, you must also define:

- **Allocation** → Is usage equal across users, pooled, or differentiated by user type?
- **Reset frequency** → Does the allowance reset daily, monthly, or annually?
- **Administration** → Who manages usage within the company, and what level of visibility should the admin console provide into usage and costs?

Founder prompt

- If a healthy account doubles usage tomorrow, will they celebrate — or will their CFO call us angry?

Overage Mechanics

When usage exceeds the limit, options include:

- **Auto-upgrade** → Move customers to the next tier after repeated periods over their allowance.
- **True-up** → Bill excess at renewal rather than mid-term.
- **True-forward** → Increase the customer's future commitment after overuse, waiving flex fees in exchange for the new commitment.
- **Top-ups** → Customers buy extra usage "packs" mid-term.
- **Rollovers** → Allow customers to carry forward unused credits from prior periods.

Best fits by model:

- Auto-upgrades → Work well in **PLG funnels.**
- True-ups & true-forwards → Common in **enterprise contracts.**
- Top-ups → Fit best in **self-serve models.**

Price levels

Setting price levels

Setting price levels is not about pulling numbers from thin air. Price points shape perception, segment customers, and anchor expectations.

- **Willingness-to-pay** → Test and learn through surveys, negotiations, pilots, and customer feedback.
- **Alternatives** → Acknowledge competitors, but don't obsess. Your pricing strategy should align with your business strategy, not theirs.
- **ROI** → You should be able to defend capturing 10–30% of the business value you create.
- **Tier deltas** → Ensure price jumps are meaningful, typically 2–3×.
- **Fences** → Define entitlements (users, projects, throughput) that trigger natural upgrades.
- **Volume discounts** → Reward scale while protecting margins.
- **Quid pro quo discounts** → Offer in exchange for longer commitments, upfront payments, etc.
- **Regional adjustments** → Adapt to local markets, but guard against arbitrage.

Use internal data, customer interviews, and quantitative research to inform levels. But don't wait for perfect data — testing in-market is often faster and more reliable.

Founder prompt

- Would a rational buyer clearly understand why one plan costs 3× another?

AI-specific considerations

STRATEGIC CONSIDERATIONS VS. SOFTWARE	IMPLICATIONS FOR PACKAGING & PRICING STRATEGY
Variable costs AI currently carries real compute/model COGS	Increased use of usage-based models that scale with variable costs, or hybrid models that give some overarching cost certainty
Usage volatility Workloads are notoriously "spiky"	Use of drawdown models that allow customer to use credits/usage at their own pace, with flex/on demand fees available for customers
Project use cases More AI use cases are experimental or project-based	Choice of selling recurring subscriptions (and tolerating higher churn) or accepting more "re-occurring" revenue, although not subscription based
Varied business outcomes AI products don't consistently deliver repeatable outcomes nor costs	Emergence of credits models that can communicate differentiated value / cost of each use case via "exchange rates"
Budgets AI agents aiming to replace workers could tap into headcount budgets (vs. software)	GTM targeting will need to incorporate CHROs and CFOs if the business outcomes are to include saving headcount
Higher value, lower prices AI that replaces human work will put pressure on seat-based revenue models	Less use of more # of seats as monetisation lever, although price per seat can grow based on increased value from AI
Vanity metrics Adoption without conversion requires even deeper pockets	Companies need to show they can convert from freemium to paid quicker, as freemium models require deeper pockets

Strategic Considerations

Agentic AI pricing is still in flux, and there's much to learn. What's clear already is that AI introduces challenges traditional SaaS models never had to face:

- **Variable Costs**

  Unlike SaaS, AI comes with real COGS in the form of compute and model usage. Moore's Law isn't bringing costs down yet — larger, more capable models often require more compute. Pricing must therefore protect gross margins, often via hybrid models that combine commitments with usage-based charges.

- **Usage Volatility**

  AI workloads are notoriously spiky. Pricing needs to balance flexibility with predictability, especially for enterprises. A common solution is capacity commitments with flexible overages to smooth the bumps.

Strategic Considerations

- **Project use-cases**

  Similarly, many AI use cases are experimental or project-based. The key question is whether to package as a recurring subscription (and tolerate higher churn) or design models around project metrics and one-off engagements?

- **Varied business outcomes**

  AI products don't consistently deliver a single, repeatable outcome. This variability makes them well-suited to credit-based models, where units of consumption can capture both fluctuating costs and uneven value delivery.

- **Budgets**

  The most transformative AI agents aim to replace workers, implying they should draw from headcount budgets rather than software budgets. The challenge lies in communicating and justifying that value in a way that finance leaders will accept.

- **Higher value, lower prices**

  If AI replaces human work, seat-based metrics quickly loses value. Some metrics can even create perverse incentives (e.g. better performance means customers pay less). Instead, tie pricing to tasks, throughput, or outcomes to align value with revenue.

- **Adoption vs. monetization**

  Beware of vanity metrics (e.g. sign-ups instead of conversions). Focusing on who truly converts improves both roadmap and support allocation. Fence too tightly, and adoption stalls; fence too loosely, and monetization collapses. The art is in striking the balance.

AI packaging considerations

There are important considerations for price models, which we will explore below. From a packaging perspective, however, not much has changed. The choice is essentially the same, and whether it is an AI or Software capability is largely irrelevant.

	Launch premium tiers	Add to all offers with price increase	Add to all offers without price increase	Offer as add-on to core offering	Standalone pricing strategy
Commercial ambition Adoption vs. Monetisation	Monetisation	Adoption & Monetisation	Adoption	Monetisation	Adoption / Monetisation
Target customer segment New vs. Existing	Existing	Existing	Existing	Existing	New
Capability use case New vs. Exists today	Exists today	Exists today	Exists today	New	New
Expected Business Value High vs. Mid vs. Low	High Value, Mid adoption (by those with higher WTP)	High Value, Wide adoption	Low Value, Wide adoption	High value, Low adoption	Any (except low value & niche)
Expected adoption Wide vs. Narrow					

Ask yourself the following questions:

- **Commercial ambition** → Are we prioritising adoption or monetization with this offer?
- **Target customer segment** → Is this aimed at new customers or existing ones?
- **Capability use case** → Does this capability serve a new use case or one that already exists today?
- **Expected business value** → Will customers perceive this as delivering high, mid, or low value?
- **Expected adoption** → Do we expect adoption to be wide across the base or narrow within a niche?

Your option space, depending on the answers:

1. Launch a premium tier with the new capability
2. Add new capability to all tiers, with no price increase
3. Add new capability to all tiers, with a price increase
4. Offer new capability as an add-on to existing offerings
5. Create a standalone monetization model for new capability

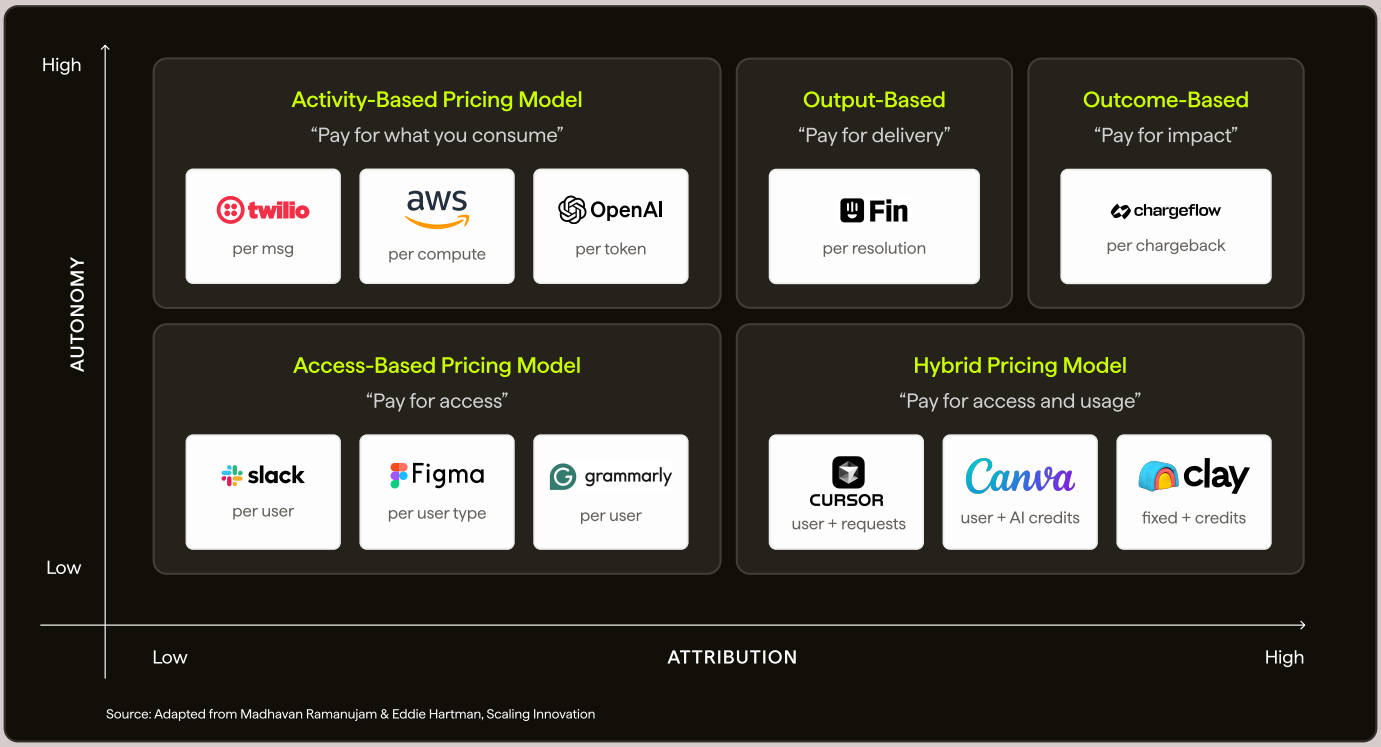
AI price model & value metric considerations

Price models in AI are still evolving as the technology matures — we're collectively in a learning phase, refining how to connect value delivery with monetization.

Two dimensions are particularly useful to understand today's landscape:

- **Autonomy:** How independently does the AI operate? Is it a co-pilot assisting humans to work more efficiently, or an autonomous agent capable of delivering results on its own?
- **Attribution:** How directly can the product's value be tied to measurable outcomes? Is pricing linked to access, usage, outputs, or impact?

Together, these axes help determine the most fitting pricing model and value metric for a given AI solution.



Source: Adapted from Madhavan Ramanujam & Eddie Hartman, Scaling Innovation

Pricing model archetypes

- **Access-Based Pricing Model — "Pay for access"**
  Common for co-pilot models where users benefit from AI assistance but retain control. Typically priced per seat or per user (e.g., Slack, Figma, Grammarly).
- **Hybrid Pricing Model — "Pay for access and usage"**
  Combines predictable access fees with a usage-based component — for instance, per AI request, per credit, or per generated output. This model fits tools that mix human input with AI automation (e.g., Cursor, Canva, Clay).

- **Activity-Based Pricing Model — “Pay for what you consume”**
  Used when AI operates autonomously and resource consumption directly drives value (e.g., Twilio, AWS, OpenAI).
- **Output-Based Pricing Model — “Pay for delivery”**
  Pricing reflects discrete, measurable outcomes such as completed resolutions or transactions (e.g., Fin).
- **Outcome-Based Pricing Model — “Pay for impact”**
  The most advanced form, aligning price with business impact — for example, per recovered chargeback or verified success metric (e.g., Chargeflow).

AI price model & value metric considerations

Both hybrid and credit-based models are gaining traction to balance flexibility with predictability, especially when results or usage vary.

Pure consumption models without a minimum commitment are becoming less common, as companies favour hybrid and drawdown structures that create stronger revenue certainty while maintaining flexibility. These are most prevalent at the infrastructure or LLM layers, where token or compute-based pricing directly aligns with COGS.

What “designed” looks like

Outcome

When pricing is well designed:

- Customers see themselves instantly in the packages. Speak their language.
- Sales can explain the model without slides. Simplicity sells.
- Expansion paths are obvious, fences feel natural, and usage limits feel fair.
- ARPU rises, discounting narrows, NRR climbs, and margins remain healthy.

At this stage, you’ve gone from strategy to structure. The next step is Playbook 3 — **Own** — where design is operationalized: communicating changes, migrating customers, enabling sales, and embedding pricing into your company DNA.

Ready to stop guessing? Start owning your pricing and grow with confidence.

See the impact at northlane.partners

info@northlane.partners

Northlane

Northlane