

Utility economics

Software construction and operation are becoming cheaper, but much of SaaS still prices scarcity, lock-in and feature withholding. This paper argues for complete products sold like dependable utilities: low-friction, fairly margined and valuable because they work, not because leaving hurts.

AUTHORED BY

Daniel Johnson & Gary McLean Hall

AI DRAFTING

Astra, Fable & the models of the moment

EVIDENCE CHECKED

15 September 2026. The date tells you when the cited material was reviewed, not when the future agreed to behave.

INVALIDATED BY

When cost-to-serve, market evidence or customer behaviour materially contradicts the model.

EXPIRY DATE

The reader's call. Before a low price is mistaken for a viable business plan.

The founders own the argument; AI helped draft the prose. The sources and limits are here so you can challenge both.

Contents

01. The race to the bottom has not arrived everywhere
 02. SaaS solved a real problem
 03. What public-company evidence shows
 04. Lock-in is not a moat worth celebrating
 05. The NordVPN illustration
 06. A simple economic model
 07. Complete products, not expansion traps
 08. AI changes construction and service cost
 09. Revenue share as aligned financing
 10. Anti-SaaS is an operating discipline
 11. Counterarguments
 12. What buyers should ask
 13. Conclusion
 14. References
-

The race to the bottom has not arrived everywhere

Digital goods can serve another customer at low marginal cost. Cloud infrastructure, open-source components, automation and now generative engineering continue to reduce the labour and capital required to create and operate many software products. Yet large parts of the SaaS market still price as if every feature were scarce.



The result is familiar: per-seat ladders, essential controls reserved for enterprise tiers, annual commitments, paid integrations, expensive data access and products designed around expansion revenue rather than the complete customer job.

Utility economics starts from a different premise. Charge what is needed to build, operate and improve a dependable product with a reasonable profit margin. Make the full ordinary job available. Let retention come from usefulness, accumulated configuration and trusted records—not the threat of losing capabilities or facing an extraction project.

This is not a prediction that every category will collapse to one price. It is an argument that the cost structure has changed faster than the dominant commercial model.

The thesis applies most strongly where a recognisable job can be completed repeatedly with limited human intervention. That condition matters. A service requiring bespoke negotiation, expert interpretation or substantial customer-specific operation may have low compute cost and still be expensive to provide. Calling it software does not make the surrounding labour disappear. A utility proposition must bound the ordinary job clearly enough that customers understand what is standard and the supplier can operate it sustainably.

Distribution is another constraint. Cheaper construction lowers one barrier to entry; it does not give a small supplier access to buyers or permission to bypass procurement. A viable utility needs a credible route to adoption, a manageable cost of onboarding and

sufficient trust for the consequence of the job. The argument is therefore about a changed opportunity set, not an automatic market correction driven by inexpensive code alone.

SaaS solved a real problem

Recurring software was a rational response to perpetual licences, painful upgrades and private infrastructure. A supplier could operate one evolving service; customers received updates without repeated installation programmes. Subscription revenue financed support and continuous improvement.

What recurring payment can fund

Continuing obligation

Security, operations, support, compliance and improvement.

Incomplete promise

A required part of the ordinary job withheld behind a higher tier.

The criticism is not “subscriptions are bad”. Operation has continuing costs. Security work, support, compliance and product improvement do not end at launch.

The problem begins when recurring value is replaced by recurring incompleteness. A basic tier is made deliberately inadequate. The right to use one's own data becomes a premium feature. A customer pays more because its team grew, even when supplier cost barely changed. Cancellation becomes a design failure rather than an accepted market signal.

Utility software may still be subscription software. The difference is what the payment buys and how power is distributed.

What public-company evidence shows

Public financial statements show that mature software businesses can carry high gross margins. The following are historical annual observations, not a claim about the latest quarter or directly comparable product economics:

Company and period	Observation	Primary evidence
HubSpot, year ended December 2025	Gross profit divided by revenue is approximately 83.8 per cent; reported GAAP operating margin is 0.2 per cent	2025 results and consolidated statements
Elastic, year ended April 2025	Subscription gross margin is approximately 79.6 per cent; services approximately 1.5 per cent, calculated from the separate revenue and cost lines	Fiscal 2025 consolidated statements
UiPath, year ended January 2025	Reported overall gross margin is 83 per cent, alongside an operating loss of 11 per cent of revenue	2025 Form 10-K, results as a percentage of revenue

The calculated percentages use $(\text{revenue} - \text{cost of revenue}) \div \text{revenue}$; HubSpot's reported gross-profit line gives the same numerator. Whole-number rounding produces the familiar 84, 80 and 2 per cent headlines, but the extra precision makes the comparison inspectable. HubSpot and UiPath also show why gross margin cannot be read as money the supplier simply keeps.

Read the accounting boundary

Gross profit

Revenue after the reported cost of revenue.

Not yet deducted

Other research, sales, administration and expenditure needed to reach net profit.

These figures do not prove overpricing. Gross profit must fund research, sales, administration and other expenditure not already included in cost of revenue; a high gross margin does not imply a high operating profit. Different products have different infrastructure and compliance costs. The figures establish only that delivery cost can be a relatively small share of subscription revenue in mature software businesses.

The relevant strategic question is what happens when construction, support and operation become cheaper for new entrants. High margins attract competition. AI-assisted delivery lowers the minimum efficient team for some products. Open components reduce reinvention. Commodity infrastructure makes small-scale operation affordable.

Our economic hypothesis is that lower entry and operating costs can put downward pressure on prices where buyers can compare and switch. That is conditional, not a forecast of convergence to cost. Distribution, differentiated value, procurement friction

and switching cost can protect prices even when a technically credible alternative exists.

Gross-margin figures require care because they sit above several economically important costs. A business may spend heavily acquiring customers, building new products or meeting obligations that are classified outside cost of revenue. Comparing percentages across companies without examining those definitions can produce a confident but shallow accusation. The filings establish reported accounting outcomes, not a universal estimate of the minimum sustainable price for the categories involved.

For a challenger, the useful question is which of those costs can be reduced without weakening the service. A narrower product may need less sales education; a partner channel may lower acquisition effort; deterministic operation may reduce support; and simpler packaging may remove administrative work. Each is a hypothesis to measure. If the new entrant merely reproduces the incumbent's acquisition and service burden at a lower price, a high incumbent gross margin does not rescue the challenger's economics.

Lock-in is not a moat worth celebrating

Some retention is earned. A product contains an organisation's configuration, workflows, history and learned practice. Moving would impose real change even if export were perfect. That accrued-data and configuration moat reflects value created together.



Other retention is imposed: proprietary formats, inaccessible exports, egress charges, contract timing and integrations designed to be difficult to replace. The UK Competition and Markets Authority's [cloud-services investigation](#), whose final decision was published in July 2025, examined technical and commercial barriers to switching and multi-cloud use. Cloud infrastructure differs from application SaaS, so this is an analogy about switching friction, not direct evidence that every application subscription has the same competitive effects.

A utility product should make exit possible. Provide usable exports, documented interfaces and clear retention. Let the product keep customers because the service remains better than the cost of change, not because change has been made artificially dangerous.

Usable exit is more demanding than a download button. Exported data needs intelligible structure, stable identifiers and enough context to interpret relationships and historical records. Configuration may matter as much as raw transactions: permissions, templates, rules and workflow settings can represent years of customer effort. A supplier should state which of these leave with the customer and which depend on its own product implementation.

This does not mean switching can be made effortless. A replacement must learn the domain, import the records and train users. Those are real transition costs even in an open system. The ethical and commercial distinction is whether the supplier adds unnecessary obstacles to those costs. Demonstrating an intelligible export and documented interfaces makes the retention claim testable: the customer stays because the service continues to earn its place, not because the supplier has made the alternative unknowable.

An export rehearsal makes that distinction concrete. Take a representative organisation, including historical corrections and awkward records. Export its data and documented configuration, then ask a technically competent reader who did not build the product to reconstruct a selected workflow history. Can they identify the actor, date, document version, relevant relationships and reason for a changed state? A directory full of files may satisfy a narrow download promise while failing that practical test.

The rehearsal is not a demand to reproduce proprietary source code or erase all transition effort. It is a test of whether the customer can understand and continue its own work. Where a calculated field depends on a supplier rule, the export contract should say whether the value, rule version or executable rule is included. "You own your data" has little operational meaning until those boundaries are clear.

The NordVPN illustration

[NordVPN's public pricing](#) illustrates a consumer-utility pattern: service bundles, longer commitments expressed as monthly-equivalent prices, and introductory offers with separate renewal terms. Its page states that subscriptions renew automatically at the then-applicable price unless cancelled, and that renewal may be more expensive than the introduction. The offer is a commercial example observed in September 2026, not a fixed price quoted by this paper.

Transfer the question, not the forecast

Consumer illustration

A substantial service sold with a utility-shaped price.

B2B evidence still needed

Acquisition, support, liability, usage and renewal under the actual service.

This is an illustration, not a direct template for regulated B2B software. Consumer VPN has different support, liability, acquisition and usage economics. The useful lesson is commercial shape: a technically substantial service can be sold to a wide market at a price that feels like an ordinary utility rather than a capital decision.

The model depends on scale and conversion. Lowering price tenfold does not automatically create one hundred times as many customers. Any such arithmetic is a scenario, not a forecast. Distribution, trust, market size and support load decide whether low pricing works.

Utility economics therefore begins with measured unit economics, not a slogan.

Introductory pricing also separates acquisition from continuing value. The first payment may attract a customer without demonstrating that the renewal price is sustainable or acceptable. A utility model should examine the whole relationship: acquisition cost, service received, renewal terms, cancellation and retained value. It should not borrow the attractive monthly-equivalent headline while ignoring commitment length or later pricing. For a new B2B product, the consumer example is therefore a prompt to test a commercial hypothesis, not evidence that a particular discount, price point or volume will transfer to a different market.

A simple economic model

Define annual fixed cost: product development, security, compliance, core operations and support capacity. Define marginal cost by useful unit: active organisation, filing, case, gigabyte, model inference or transaction. Add expected support and payment costs. Choose a profit margin that funds resilience and continued improvement.

BASE: 70 LEFT



MORE SUPPORT: 50 LEFT



Support eats contribution

With revenue fixed at 100 invented units, support rising from 10 to 30 cuts contribution from 70 to 50.

Grey / white / blue Other variable cost / support / contribution.

Invented sensitivity units · contribution before fixed cost, acquisition and tax.

Then model adoption, retention and concentration. Two hundred customers paying £1,000 annually and 20,000 paying £10 each both produce £200,000 in annual revenue. The second model creates one hundred times as many customer relationships, so even a small amount of human work per account can change its viability.

The right price is the lowest one that supports the whole service, its obligations and a rational return at a credible scale. “Cheap” without those constraints produces abandonment. “Premium” without corresponding cost or differentiation invites replacement.

Publish enough of the shape for customers to judge it: what is included, what causes price to change, how renewal works and how data leaves.

A useful model separates contribution from revenue. For each customer or unit, subtract payment, infrastructure, dependency and expected support costs from the price. The remaining contribution must cover fixed obligations and the required return. Then vary the assumptions that are least certain: adoption, renewal, heavy usage, support incidence and concentration. If viability depends on all of them being favourable simultaneously, the low price is a bet on an optimistic scenario rather than a demonstrated utility model.

Cash timing matters as well. An annual contract may fund service before some costs are incurred; monthly billing may leave less room for upfront onboarding. A revenue-share product may have substantial build cost before receipts exist. These are financing

differences, not reasons to disguise renewal terms. The commercial arrangement should make the obligations and timing explicit enough that both parties can distinguish a sustainable service from one temporarily subsidised by unpaid labour.

A sensitivity test without a sales forecast

Use an indexed model to expose the mechanism rather than imply a real product price. Suppose one annual customer contributes 100 revenue units. Payment and infrastructure consume 12 units, other variable services consume 8, and expected human support consumes 10. Contribution is therefore 70 units before fixed cost, acquisition and tax. If support rises to 30 units, contribution falls to 50: a twenty-unit cost change removes almost 29 per cent of the original contribution.

Constructed case	Revenue	Non-support variable cost	Support	Contribution
Base assumption	100	20	10	70
More recurring human help	100	20	30	50
Lower price, unchanged service burden	80	20	10	50
Lower price with operating improvement	80	20	5	55

These are invented units for sensitivity analysis, not Zero Drift prices, margins or observed support costs. If annual fixed obligations were 7,000 of the same units, the base case would require 100 retained customers to cover them; the high-support case would require 140. That calculation excludes acquisition, growth investment, tax and financing. It is deliberately a contribution break-even check, not a valuation or a profitability forecast.

The next question is empirical: how much human help occurs per retained customer, and why? An annual average can hide concentration. A few complex accounts might consume most of the support while the ordinary account is nearly self-serving. Segment by workflow, cohort and incident cause before deciding whether to improve onboarding, charge for genuinely different work or change the target market. Raising everyone's price can be the wrong response to one preventable integration failure; unlimited support can be the wrong promise for a service-intensive segment.

Complete products, not expansion traps

The core tier should complete the ordinary customer job. For payroll, that includes the workflows required to run payroll, not just store people. For governance, it includes the meeting cycle, permissions, records and publication required for compliant work. For filing, it includes submission and evidence, not a dashboard that points to a future integration.

A filing job needs its ending

Ordinary job

Prepare, validate, submit, confirm, correct and retain evidence.

Different work

Bespoke advice about a disputed filing can be a separate service.

Different prices can still be legitimate. Higher usage creates real cost. Additional products solve different jobs. Advanced assurance, bespoke integrations and priority support consume additional resources. A small organisation and a national operator may create different operational obligations.

The test is whether a tier charges for more work or withholds part of the job already promised. Utility economics rejects the latter.

Define the ordinary job through completed outcomes. In a constructed filing product, the job might include preparation, validation, submission, confirmation, correction and retention of evidence. Pricing preparation cheaply while charging separately for the only route to submission would make the headline tier incomplete. A genuinely different service, such as bespoke advice about a disputed filing, can remain outside that ordinary software boundary.

The boundary must also survive edge cases that belong to normal use. A rejected submission, changed account owner or recoverable integration failure is not automatically a premium feature because it requires more implementation effort. Conversely, unlimited volume and bespoke support need not be bundled at any price. The disciplined question is whether the additional charge corresponds to additional work, risk or scale, rather than withholding a capability required to finish the job already sold.

AI changes construction and service cost

Stanford's [2025 AI Index](#) documented a sharp reduction in the inference price associated with a particular benchmark capability, alongside improvements in smaller models. A benchmark-equivalent inference price is not the total cost of a dependable application:

verification, orchestration, data handling and human review still matter. Engineering agents may reduce mechanical implementation, testing, documentation and support-triage effort where the workflow supports them.

Where a cheaper model stops being a saving

Inference bill

One input into the cost of an accepted result.

Whole service

Add verification, orchestration, data handling and human repair.

The effect is not uniform. Complex domains still require expertise. Verification can consume a large share of the build. AI features themselves can create variable compute cost. Experienced developers may not become faster on familiar complex repositories simply by adding a coding assistant.

The durable change is option value. A small team can attempt narrower products, automate more operation and test more of the surface. Categories previously unable to support a bespoke complete product may become viable at utility prices.

Savings should reach the customer rather than only widening the supplier's margin.

Savings can be consumed before they reach a customer. Faster implementation may encourage a larger feature surface, which then increases support, testing and operating obligations. Generated code that is cheap to produce can still be expensive to understand and maintain. A utility builder needs to preserve the bounded product and spend some of the saved effort on making it reliable, observable and inexpensive to change.

The benefit should be tested over the service lifecycle. Track build and verification effort, accepted-change cost, recurring human toil and infrastructure use per useful unit. Distinguish a one-time construction gain from a lasting reduction in cost to serve. That distinction affects pricing: a low initial build bill does not justify a permanently low service price if continuing obligations remain high, while durable operating savings create a stronger basis for customer benefit.

Revenue share as aligned financing

Some domain products lack the capital for a conventional build but have a founder with distribution and expertise. A capped share of revenue can finance construction while aligning payment with product success.

Revenue share is financing

Domain founder

Market access, domain decisions and selling.

Product studio

Specification, construction and operation under explicit IP, cost and failure terms.

The arrangement works only when responsibilities are explicit. The domain founder owns market access, domain decisions and selling. The product studio owns specification, construction and operation. Intellectual property, data responsibility, costs, cap, tail and failure conditions must be agreed before work.

Revenue share is not free software. It is risk capital paid from future receipts. It belongs where distribution is credible and the resulting utility can reach sufficient volume. Used indiscriminately, it transfers speculative risk to the builder and obscures the true cost of support.

The agreement needs a decision model for ordinary disagreements. Who chooses a feature that might improve sales but raises support cost? Who funds a compliance change? What happens when distribution underperforms or one party stops supplying its promised work? A revenue percentage alone cannot answer those questions. The arrangement should state authority, reporting, cost treatment and the point at which either party can stop taking further risk.

Alignment is also bounded by the cap and tail. The builder needs a credible route to recover its investment and earn a return; the founder needs to understand the eventual economics and ownership of the product. The paper does not prescribe one contract shape. It argues that the risk allocation must be legible and attached to real responsibilities. Calling the arrangement a partnership does not remove the need to explain how it behaves under disappointment.

Anti-SaaS is an operating discipline

A credible alternative needs more than friendly pricing.

Keep the utility workable

Customer boundary

Usable export, documented integration and plain renewal terms.

Operating evidence

Support load, reliability and cost measured after launch.

- Build the complete ordinary job.
- Keep infrastructure and human toil low by design.

- Make pricing legible and tied to real cost or value units.
- Allow export and documented integration.
- Avoid artificial feature withholding.
- State renewal and introductory terms plainly.
- Preserve customer ownership of data and operating knowledge.
- Measure support load, reliability and cost after launch.

The product should feel boring in the best sense: available, understandable, affordable and easy to leave.

Counterarguments

High prices can finance difficult innovation and specialist support. Enterprise procurement, security review and liability are expensive. Per-seat pricing is sometimes a reasonable proxy for value and load. Bundling every capability can confuse small customers. Low headline pricing can itself become manipulative when renewal jumps or cancellation is difficult.

The thesis has a boundary

Utility-shaped opportunity

Repeatable software serving a class of customers at low marginal cost.

May stay expensive

Rare expert work, high liability and intensive human service.

Utility economics does not deny these facts. It demands that the commercial shape be justified by the service rather than inherited from a venture-growth playbook.

It also accepts that some software is not a utility. Rare expert systems, high-liability decisions and intensive services may remain expensive. The argument applies where repeatable software can serve a broad class of customers at low marginal cost.

A further objection is that a complete product can become hard to understand. Bundling the whole ordinary job does not require showing every control to every user at once. Product design can reveal complexity progressively while leaving the underlying capability available. Simplicity of experience and completeness of entitlement are different decisions; commercial packaging should not use the former as a convenient justification for withholding the latter.

Low pricing can also damage continuity if the supplier underestimates assurance or support. Buyers should therefore ask about sustainability as well as fairness. Can the operator fund maintenance, respond to incidents and preserve records if growth is slower

than expected? A cheap service that disappears and leaves customers with an expensive migration is not a successful utility. Reasonable margin and credible operation are part of the customer proposition, not concessions to it.

There is a serious counterargument from value pricing: a supplier that creates a large benefit need not price at marginal cost. That is correct. This paper proposes a commercial discipline, not a theorem that cost determines the morally permissible price. A utility builder chooses a reasonable return, broad adoption and durable customer value as its strategy. The strategy loses credibility if it describes every higher-priced competitor as exploitative without examining the work, risk and buyer value involved.

The proposition is also falsifiable. If qualified customers do not activate, if ordinary use repeatedly requires expert intervention, if acquisition cost cannot be recovered from retained contribution, or if portability still leaves the customer unable to interpret its record, the utility model has failed a substantive test. Lowering the price again does not answer those failures. The builder must improve the product and operation or revise the commercial hypothesis while preserving the job already promised.

What buyers should ask

Read beyond the headline price

Finish the job

Identify every required capability and whether the quoted tier includes it.

Leave the service

Inspect usable data and configuration export, ownership and renewal.

- Which capabilities are required to complete the job, and are they all included?
- What supplier cost or customer value causes the price to scale?
- What happens at renewal?
- Can data and configuration be exported in usable form?
- How much human administration is hidden behind the service?
- Who owns the code, data, domain model and operating knowledge?
- If AI lowers delivery cost, where does the customer see the benefit?

Conclusion

As production costs fall in some categories, suppliers have more room to compete on complete service and lower prices. Our proposition is that dependable products can earn retention through usefulness and accumulated value while charging a reasonable return. Whether that model wins in a particular market depends on distribution, trust and sustainable cost to serve.

References

- Nord Security, [NordVPN pricing](#), accessed September 2026.
- HubSpot, [2025 Form 10-K](#).
- HubSpot, [2025 results and consolidated statements](#).
- Elastic, [2025 annual report](#).
- Elastic, [Fiscal 2025 results and consolidated statements](#).
- UiPath, [2025 Form 10-K](#).
- Competition and Markets Authority, [Cloud services market investigation](#).
- Stanford HAI, [AI Index 2025: Technical Performance](#).