

OpsRabbit

Enterprise Pricing & Commercialization Playbook

Hybrid product, deployment, run, and outcome-based model

Purpose: Preserve enterprise economics while changing how the value is packaged and sold.

Audience	Status	Primary Use
Internal	Draft for review	Socialize and test across enterprise opportunities

Core commercial thesis

Change the buying journey from a license discussion to a phased business-value conversation: Deploy → Run → Improve → Share Value.

1. Executive Overview

The pricing model evolves from a conventional software-plus-services structure into a hybrid commercial model. The underlying economics remain intact, but the customer sees a clearer path to value and a lower-risk way to adopt OpsRabbit.

Commercial evolution

Stage	Customer Buys	OpsRabbit Monetizes	Proof Point
Deploy & Enable	A production-ready deployment in the customer environment	One-time deployment and enablement fee	Integrations completed, platform operational, users enabled
Run & Scale	Predictable access to Operations Intelligence across an agreed scope	Annual platform and capability subscriptions	Incidents investigated, teams onboarded, workflows adopted
Improve	Measurable operational improvement	Renewal, capability expansion, and premium support	MTTR, escalations, repeat incidents, and effort saved
Share Value	Alignment around verified outcomes	Capped success fee	Agreed target achieved after stabilization

What does not change

- OpsRabbit remains a product, not a staff-augmentation offer.
- Subscription revenue remains the commercial foundation.
- Additional operational capabilities remain separately monetized.
- Deployment, customization, support, and success fees remain incremental.
- Success fees are upside, not a substitute for subscription revenue.

What changes

- Lead with the operational outcome and phased journey, not the rate card.
- Position deployment as business enablement, not technical setup alone.
- Quantify operational value before discussing price.
- Use historical workload baselines to define scope without discouraging usage.
- Introduce success fees only where attribution and measurement are credible.
- Shift the narrative from automation to operational autonomy.

2. Recommended Hybrid Pricing Model

Commercial Element	What It Includes	Working Price	Revenue Type	Example
--------------------	------------------	---------------	--------------	---------

Deploy & Enable	Discovery, deployment, standard integrations, configuration, agent grounding, admin and user enablement	\$25K-\$50K	One-time	Initial setup across agreed systems and teams
Platform Run	Multi-team platform, Core Operations Intelligence with one capability, customer-hosted deployment, customer LLM with consumption controls, standard upgrades and security patches, data-protection controls, and 8x5 support.	\$70K	ARR	Core production deployment
Capability Expansion	DevOps, CloudOps, Production Support, FinOps, or another operational capability	\$25K each	ARR	Add CloudOps after proving core incident intelligence
Premium Support	24x7 product support, priority escalation, and response commitments	\$30K	ARR	Mission-critical production deployment
Success Fee	Verified improvement against an agreed baseline	\$25K-\$75K capped	Variable	MTTR or escalation target achieved

Commercial formula: Platform + selected capabilities + deployment + optional premium support + capped success fee.

3. Enterprise Product Governance & Protection

Token / LLM Consumption Controls

OpsRabbit runs in the customer environment and uses the customer's LLM. To help customers manage AI consumption, the platform supports configurable controls such as usage limits, model selection, workflow-level guardrails, and monitoring of LLM consumption. The objective is predictable usage without compromising operational effectiveness.

Enhancements & Product Upgrades

The annual platform subscription includes standard OpsRabbit product enhancements and version upgrades released during the subscription term. Customer-specific enhancements or bespoke functionality may be separately scoped where required.

Security Patches

Standard security patches and critical product fixes are included as part of the annual subscription. OpsRabbit will provide updates in accordance with the supported release and deployment model.

Data Protection

OpsRabbit runs entirely within the customer environment. Customer data does not leave the customer-controlled infrastructure and is not used by OpsRabbit for model training or any secondary purpose. Data access, retention, identity, and security remain governed by the customer's policies.

This section may be updated as below

OpsRabbit runs within the customer environment. Operational data remains customer-controlled. For AI-assisted tasks, only the minimum relevant context, such as incident details, selected logs, alerts, configuration context, and runbook information, is sent to the customer-designated LLM endpoint. OpsRabbit does not externally retain or use customer data for training or secondary purposes.

4. Illustrative Customer Economics

These are planning examples, not rigid public tiers. The commercial structure should reflect the customer problem, deployment scope, and measurable value.

Scenario	Recurring Revenue	One-Time Fee	Success Fee	Year 1 Value
Initial Deployment	\$70K	\$25K-\$30K	Not initially	\$95K-\$100K
Expanded Deployment	\$95K-\$150K	\$30K-\$75K	Optional	\$125K-\$225K
Strategic Enterprise	\$200K-\$300K+	Custom	\$50K-\$75K capped	\$300K-\$500K+

Commercial guardrail

The success fee is upside. It must never replace the platform floor or create uncapped delivery risk.

5. How to Sell the Model

Step	Sales Focus	What the Team Establishes	Suggested Language
1. Problem	Where operations are breaking down	Incident delays, repeat issues, escalation burden, engineer effort, and service impact	Where is the current operating model creating the most friction or risk?
2. Baseline	Current operating model	Incident volume, MTTR, severity mix, team effort, and current cost	Before pricing, let us agree on the operational baseline.
3. Value Conversion	Financial value of improvement	Capacity released, downtime avoided, reduced escalation, and lower dependency	Let us quantify what even a modest improvement would be worth.
4. Position OpsRabbit	How the improvement is created	Evidence-led investigation, faster RCA, lower toil, and increasing autonomy	OpsRabbit works inside your environment and learns your operational context.
5. Commercial Journey	How the customer buys	Deploy, run, improve, expand, and optionally share value	The model is phased, predictable, and tied to measurable value.
6. Proof	How success is verified	Baseline, targets, measurement period, and expansion trigger	We jointly define success before the measurement period starts.

Lead with this

“OpsRabbit is deployed in your environment, learns your operational context, and helps your teams investigate and resolve incidents with increasing autonomy. We structure the engagement around deployment, predictable platform use, and measurable operational improvement.”

Avoid leading with this

- Per-seat or per-user pricing
- Pure per-incident pricing
- Token-based pricing, since the customer owns the LLM consumption
- A long list of technical SKUs before the business problem is clear
- A promise that fees are fully contingent on outcomes OpsRabbit cannot control

6. Value Conversion

Translate operational pain into financial value before discussing pricing.

- Engineer hours saved
- Reduced MTTR

- Lower escalation volume
- Fewer repeat incidents
- Reduced downtime exposure
- Lower outsourced support dependency
- More application coverage without proportional headcount growth

Operational baseline × expected improvement × unit economic value = addressable customer value

Illustrative example

Input	Assumption
Monthly incidents	1,000
Escalated to senior engineers	20%
Senior engineer effort per escalation	3 hours
Blended hourly cost	\$100

Current annual escalation effort: $1,000 \times 20\% \times 3 \times \$100 \times 12 = \$720K$

At 30% improvement: Annual value created = \$216K

Commercial conclusion: OpsRabbit should be priced as a fraction of the operational value it can release.

7. Success Fee Framework

Success fees should be used selectively. They are most credible when the baseline is reliable, the metric is measurable, and OpsRabbit has a meaningful causal role.

Metric	How to Measure	Important Caveat	Suitability
MTTR reduction	Compare similar incident classes against a historical baseline	Normalize for severity and incident mix	High
Repeat incident reduction	Track recurrence of known incident patterns	Requires stable taxonomy and tagging	High
Escalation reduction	Measure L1/L2 incidents escalated to senior teams	Staffing and policy changes may affect results	Medium-High
Investigation effort saved	Estimate engineer-hours avoided per incident	Requires an agreed method and sampling	Medium
Team-size reduction	Compare staffing cost before and after adoption	Influenced by broader workforce decisions	Low

Recommended mechanics

- Use a 6- to 12-month measurement period after stabilization.
- Agree the baseline, excluded events, severity classes, and source data in writing.

- Use one or two primary metrics, not a broad basket of loosely connected outcomes.
- Set a fixed maximum success fee.
- Use achievement bands, for example 10%, 20%, and 30% improvement.
- Exclude delays or failures caused by unavailable data, missing access, process gaps, or non-adoption.
- Make the success fee additive to subscription and support, not a credit against them.

Illustrative success fee bands

Verified Improvement	Illustrative Fee	Example
Below agreed threshold	\$0	No success fee; base subscription remains payable
Target achieved	\$25K-\$40K	Example: 15% MTTR reduction
Stretch target achieved	\$50K-\$75K	Example: 25%+ MTTR reduction plus repeat-incident improvement

8. Opportunity Qualification Checklist

Question	Why It Matters	Required Before
What operational problem is urgent now?	Prevents a generic AI conversation	Proposal
What is the incident volume and severity mix?	Defines scope, baseline, and success metrics	Solution scoping
What are current MTTR and escalation levels?	Supports value conversion	Commercial proposal
Which teams and environments are in scope?	Controls deployment complexity	SOW
Which systems must be integrated?	Determines deployment effort	SOW
Who owns access, data, and approvals?	Avoids delivery delays	Kickoff
What has already been built internally?	Strengthens build-versus-buy positioning	Executive discussion
Which outcomes can be measured reliably?	Determines suitability for success fees	Commercial approval

9. Commercial Approval Guardrails

- No uncapped success fee or unlimited liability tied to customer outcomes.
- No success fee without an agreed baseline and a customer-owned data source.
- No pure per-incident model that penalizes adoption or creates unpredictable bills.
- No commitment to incident reduction where better detection may initially increase recorded incidents.

- No standard implementation promise where integrations, environments, or workflows are materially complex.
- No ambiguity between product support and managed operations.
- No unlimited scope under the platform subscription.

Draft status: Internal review only. Not approved as a customer-facing rate card or contractual framework.