The Business Case for Outsourcing Game Operations

How studios and publishers can reduce operational risk, improve uptime, and scale 24/7 GameOps without building the full function internally

IN THIS WHITEPAPER

SECTION 01

Executive Summary

Running a live game 24/7 is an operating-model decision, not a tooling decision.

Studios and publishers have four options: stretch internal teams with expanded on-call rotations, build a full 24/7 GameOps function, use a generic managed service provider, or deploy a proven external GameOps layer. Each model carries a different cost structure, risk profile, staffing burden, and execution speed.

Some publishers should build the full function internally. Many studios — particularly those in early live service, pre-revenue scale, or mid-size — should not. The difference comes down to scale, predictability, operational maturity, and how much fixed cost the business can justify before demand is stable.

Zumidian provides a proven 24/7 GameOps layer that works inside the customer's existing tools, workflows, dashboards, alerts, and runbooks. The model is straightforward: qualify issues faster, act through approved procedures, validate recovery, and reduce the operational burden on internal teams — without replacing internal control.

> **Studios don't outsource GameOps because they don't care about operations. They outsource because the full 24/7 operating model is expensive, hard to staff, and inefficient to build before demand is predictable.**

Five questions that drive the decision

- What does true 24/7 coverage cost internally — including management, tooling, burnout, and low-utilization overnight shifts?
- Who acts after an alert fires? How many handoffs happen before the response begins?
- How fast can incidents be qualified, actioned, and verified at 3 a.m. without the original engineer?
- Can the team cover launch windows, major patches, and live events without pulling product engineers into operational firefighting?
- Which model gives the best balance of risk, cost, speed, control, and resilience at your current scale?

For live games, GameOps is a revenue-protection and operational-resilience function. The question is no longer whether operations matter. The question is which operating model is the right one.

SECTION 02

Live Games Create 24/7 Business Exposure

A live game does not fail only when the code breaks. It fails when players cannot play.

Online games depend on backend systems, game servers, matchmaking, authentication, payment services, databases, APIs, cloud providers, network routes, telemetry, deployment workflows, and live event processes. Any single weak point can become player-facing — instantly and publicly.

Operational failures affect more than infrastructure health

Business impact	Operational impact
Revenue and in-game transactions	Engineering productivity and roadmap velocity
Player trust and retention	Support volume and triage burden
Community sentiment and brand reputation	Launch confidence and publisher confidence

A backend issue can interrupt player sessions. A regional network problem can look like game instability. A failed deployment can turn a planned update into a public incident. Slow response can turn a contained technical problem into a business crisis.

> **Uptime is a business metric for live games. So are response time, recovery time, deployment stability, and operational visibility.**

Monitoring alone is not an operating model

The common assumption is that monitoring tools, dashboards, on-call rotations, and escalation channels are sufficient. They are necessary. They are not enough.

Alerts are not resolution. On-call is not the same as 24/7 operational coverage. A live game needs people and procedures that can quickly qualify issues, act within approved boundaries, validate recovery, and improve the model over time.

SECTION 03

Downtime Is Revenue Exposure

For a live game, the outage window is only part of the cost. The outage ends. The support backlog, lost trust, negative sentiment, missed transactions, and engineering disruption continue.

What downtime and instability produce

- Lost player sessions and missed in-game transactions
- Player frustration and elevated churn risk
- Higher support pressure and community backlash
- Engineering interruption and delayed product roadmap
- Reduced launch momentum and lower stakeholder confidence
- Reputational damage that outlasts the incident itself

The revenue exposure model

A simple framework for quantifying operational risk:

Incident frequency × Incident duration × Business impact per hour = Revenue exposure

The exact number varies by game, monetization model, region, platform, and player volume. The logic does not. Longer incidents create more exposure. Slow qualification creates more exposure. Escalation delays create more exposure. Poor operational visibility creates more exposure.

Faster response and verified recovery directly reduce the window where players, revenue, and internal teams are at risk.

The usual mistake

Most teams treat downtime as a detection problem. More dashboards, more alerts, more channels, more escalation paths. Better detection helps — but detection does not recover the service.

Reducing revenue exposure requires operational ownership: qualify the incident, assess severity and player impact, follow the approved runbook, act within defined boundaries, coordinate with the right teams, validate recovery, document the result, and improve the process afterward.

> **The value is not the alert. The value is what happens next.**

SECTION 04

The Hidden Cost of Building 24/7 GameOps Internally

Building 24/7 GameOps internally is a permanent operating model, not a one-time investment. It means covering nights, weekends, holidays, launches, patches, incidents, quiet periods, unexpected peaks, and post-release stabilisation — reliably, without depending on specific individuals.

The visible cost is staffing. The real cost is structural. Salaries and benefits are the starting point. The full cost stack includes:

- Recruiting, onboarding, and attrition replacement
- Management overhead and shift coordination
- Tooling, infrastructure, and documentation investment
- Training and operational process maturity
- Low-utilisation overnight coverage during quiet periods
- Key-person risk when senior operators are unavailable

The hidden cost often shows up as operational drag over time: burnout and retention pressure, engineering distraction from product work, uneven runbook quality, slow improvement cycles, and underperformance during high-risk windows.

Cost comparison

The table below compares typical fully-loaded cost profiles. Internal figures represent a common range once staffing, management, tooling, infrastructure, shift coordination, and overhead are included. Actual figures vary by team size, location, and operational maturity.

Model	Annualised cost profile	Operational tradeoff
Internal 24/7 GameOps	Often $600K–$900K+ per year (staffing, management, tooling, infrastructure, overhead)	Full internal control; high fixed cost; significant staffing and management burden
Generic NOC or MSP	Variable — typically structured around volume tiers	Limited game-specific context; escalation-heavy; weaker ownership of resolution
Zumidian's GameOps layer	**From $25K/month ($300K/year)**	Always there, 24/7 execution layer; existing-tool integration; no internal build-out required
Potential annual saving (internal vs. Zumidian)	**$300K–$600K+ before indirect savings**	Lower fixed burden; faster coverage; less internal operational drag

This is not an argument that every company should outsource. Some teams need internal operations at scale. It is an argument against assuming that internal build-out is automatically the safer or smarter answer.

> **The real question is whether the business needs to own the full 24/7 operating model internally or whether it needs reliable access to a proven operating layer that extends the team.**

SECTION 05

Monitoring Doesn't Resolve Incidents

Most live game teams already have monitoring: dashboards, alerts, tickets, chat channels, on-call paths. The gap almost always appears after the alert fires.

Who qualifies the signal? Who owns the response? Who understands player impact? Who knows the approved action? Who validates that the issue is actually resolved?

Escalation-heavy models lose time in the middle

> **Alert → Triage → Escalation → Waiting → Investigation → Response**

Every handoff adds delay. Every delay increases player impact. Every delay widens the revenue exposure window.

Zumidian's alert-to-resolution workflow

Step	What happens
Monitor	Continuous monitoring of infrastructure, services, deployments, player-impact indicators, and alerts across the live environment
Qualify	Assess severity, scope, affected services, operational context, and player impact to separate signal from noise
Act	Map to approved runbook; take action within defined access boundaries; coordinate with internal teams or escalate where required
Verify	Confirm recovery through dashboards, logs, metrics, and player-impact signals — not just infrastructure health alone
Improve	Document outcomes and refine alert thresholds, runbooks, escalation paths, and operational procedures after each incident

> **Better monitoring tells you something is wrong. Better operations gets it fixed. The gap is not always detection — it is qualified response, structured action, and verified recovery.**

SECTION 06

Launch Stability Depends on Operational Readiness

A launch is successful when the live environment holds up under real player demand. Going live is one milestone. The operational test starts when players arrive.

Launches, open betas, major updates, live events, and seasonal drops compress risk. They stress backend systems, matchmaking, authentication, game servers, databases, APIs, payment flows, telemetry, and internal coordination — all at the same time.

What changes during launch windows

- Player demand changes faster than models predict
- Backend load patterns become less predictable
- Support pressure rises sharply in the first hours
- Community sentiment forms quickly and publicly
- Engineers are pulled across multiple competing priorities
- Stakeholders need clear, accurate, timely updates
- Small operational gaps become immediately visible

A release checklist is not enough. Teams need live coverage, deployment monitoring, incident ownership, validated recovery procedures, operational dashboards, and post-release reporting — running concurrently.

Zumidian launch and release support

- Pre-release operational readiness review
- Deployment monitoring and live release validation
- Incident response throughout launch and high-traffic windows
- Post-release monitoring and stability tracking
- Rollback and recovery coordination where applicable
- Operational reporting for engineering, production, and leadership
- Runbook improvement and threshold refinement after launch events

> **Shipping the build is one milestone. Keeping the service stable when players arrive is the test.**

SECTION 07

The Zumidian Model

Zumidian gives studios and publishers a proven 24/7 GameOps layer without requiring them to replace their tools, rebuild their stack, or staff a full internal operations team.

> **The customer keeps control. Zumidian handles execution.**

How Zumidian fits into the existing operating model

Zumidian works inside the customer's existing environment — not alongside it as a separate system. That means:

- Your monitoring tools and alerting systems
- Your dashboards and operational visibility setup
- Your communication channels and escalation paths
- Your runbooks and approved action boundaries
- Your deployment calendars and release processes
- Your operational reports and stakeholder cadence

There is no parallel stack to manage. No handoff overhead. No context gap between internal teams and the GameOps layer.

Core services

Service	What it solves
Incident Management	24/7 incident qualification, runbook-driven response, coordination, resolution support, and recovery validation
Operational Analytics	Shared operational visibility across infrastructure, game services, deployments, and player-impact signals
Ping Monitoring	Global latency and packet-loss visibility to identify regional connectivity and player-experience issues early
Launch & Release Operations	Deployment validation, launch coverage, post-release monitoring, release risk reduction, and recovery readiness
White Label Operations	Out-of-hours and customer-facing operational support under the customer's brand where appropriate
Legacy Game Management	Operational continuity for mature revenue-generating titles without distracting internal teams from new development

Operating principles

- Works with existing tools and processes — no rip-and-replace
- Platform- and infrastructure-agnostic
- Operates within approved procedures and defined access boundaries
- Focused on resolution, not notification alone
- Reduces unnecessary escalation delays
- Provides continuous coverage without requiring internal shift builds
- Available on a flexible, month-to-month engagement model
-

> **The goal is operational control without operational burden.**

SECTION 08

Operating Model Comparison

The decision is not internal team versus outsourcing. The better question is how much of the 24/7 operating model needs to be owned internally, and at what cost, at your current stage.

Model	Best fit	Main risk	Zumidian?
Internal on-call only	Early-stage teams with limited live-service complexity	Fatigue, slow response, escalation delays, weak overnight coverage	Can complement or replace
Full internal 24/7 GameOps	Large publishers with steady operational demand and scale to justify fixed cost	High annual cost, hiring burden, management overhead, underutilisation during quiet periods	Can support or extend
Generic NOC or MSP	Teams needing basic monitoring and escalation coverage	Limited game-specific context; escalation-heavy; weaker ownership of resolution outcomes	Direct alternative
Proven GameOps layer (Zumidian)	Studios and publishers needing 24/7 coverage, incident response, analytics, launch support, and existing-tool integration	Requires clear runbooks, access boundaries, and operating alignment during onboarding	**This is Zumidian**

Zumidian is not a monitoring tool. It is not a staffing agency. It is not a generic outsourced help desk. It is a specialized GameOps operating layer, built for live games, that reduces operational burden without removing internal control.

SECTION 09

Is Your Current Model Exposed?

Use these questions to assess whether your current GameOps model is robust enough to handle live-service risk. If any of these expose gaps, the issue is probably not tooling — it is the operating model.

Coverage

- Do you have true 24/7 operational coverage — including nights, weekends, holidays, and launch windows?
- Are you relying on the same engineers for product delivery and overnight incident response?
- What happens when the person who best understands the system is unavailable?

Response

- Who acts after an alert fires?
- How many handoffs happen before response begins?
- Are runbooks executable at 3 a.m. without the original engineer present?
- Can recovery be validated without waiting on internal specialists?

Visibility

- Do engineering, production, LiveOps, and leadership have shared operational visibility?
- Can the team separate actionable incidents from alert noise?
- Are player-impact signals visible alongside infrastructure signals?

Launch and release readiness

- Is operational readiness formally part of every release plan?
- Are deployment windows actively monitored end-to-end?
- Are rollback and recovery paths documented and tested?
- Are post-release signals reviewed, reported, and acted upon?

Cost and sustainability

- What does 24/7 coverage cost once management, tooling, burnout, recruiting, and low-utilisation shifts are included?
- Are quiet periods overstaffed and high-risk windows understaffed?
- Is operational firefighting regularly pulling engineers away from roadmap work?
- Does the current model scale economically as player volume and service complexity grow?

SECTION 10

Next Step: Schedule a Game Operations Review

Zumidian can help evaluate your current live operations model across coverage, incident response, operational visibility, launch readiness, and cost structure.

A Game Operations Review identifies where your current model is exposed:

- Coverage gaps across nights, weekends, holidays, and launch windows
- Incident response bottlenecks and escalation delays
- Alert noise and signal-to-noise ratio issues
- Runbook maturity gaps and execution risks
- Deployment and launch risk
- Visibility issues across engineering, production, and leadership
- Staffing and cost structure pressure
- Opportunities to reduce operational burden without reducing operational control

There is no obligation and no sales pitch. The output is practical: an honest assessment of where the model is working, where it is exposed, and whether Zumidian is the right fit to address it.

Schedule a Game Operations Review

Or do your own Launch Readiness Assessment

Launch Readiness Assessment

Or compare your current model against a proven 24/7 GameOps layer

Cost Calculator