

BUYING OPEN SOURCE WITHOUT BUYING NEW LOCK-IN

An evaluation playbook for platforms, support, and sustainable operations

Published by it-governance.net, a BusinessFragen OG service
Austria · Germany · Switzerland

EXECUTIVE SUMMARY

Open-source licensing can improve transparency, portability, and negotiating power, but the license alone does not create sovereignty. Dependency can move from a proprietary product to a specialist integrator, a managed hosting environment, an undocumented customization, or an internal skills gap.

Procurement should therefore evaluate the complete operating model: product, community, implementation partner, hosting, security responsibilities, upgrade path, documentation, internal capability, and exit feasibility.

EIGHT EVALUATION DIMENSIONS

1. PRODUCT HEALTH

Assess release cadence, maintenance policy, security response, contributor diversity, roadmap transparency, backward compatibility, and the proportion of functionality reserved for commercial editions.

2. LICENSE AND RIGHTS

Verify the license for the exact components and versions in scope. Understand obligations for modifications, distribution, network use, trademarks, and third-party dependencies. Legal review should focus on the deployed solution, not only the project's headline license.

3. IMPLEMENTATION QUALITY

Require documented architecture, configuration, custom code, interfaces, deployment automation, testing, and decisions. Customizations should follow published extension mechanisms and remain separable where practical.

4. SUPPORT MODEL

Define response, restoration, escalation, security notification, and release support expectations. Distinguish community assistance from contracted accountability. Confirm who can provide credible second-source support.

5. HOSTING AND OPERATIONS

Evaluate deployment options, backup and recovery, observability, identity, privileged access, capacity, patching, and jurisdiction. Managed open source can still be operationally closed if customers cannot access their data, configuration, or deployment artifacts.

6. SECURITY RESPONSIBILITY

Create a responsibility matrix covering vulnerability monitoring, assessment, patch development, testing, deployment, incident handling, and evidence. Require software-component transparency proportionate to the risk.

7. TOTAL COST OF OWNERSHIP

Compare licenses or subscriptions, implementation, hosting, integration, operations, upgrades, security, training, internal staffing, and transition. Avoid comparisons that price commercial support but treat internal open-source operations as free.

8. EXIT READINESS

Require usable exports, documentation, source and build artifacts where applicable, credentials, configuration, transition support, and deletion evidence. Test one representative export or recovery process before final acceptance.

RFP QUESTIONS THAT CHANGE THE QUALITY OF THE DECISION

- Which components require separate commercial rights or subscriptions?
- Which functions cannot be operated without the proposed supplier?
- How are customizations upgraded and tested against new releases?
- What artifacts are handed over at each delivery milestone?
- Which alternative service providers can support the proposed stack?
- What is the demonstrated recovery time in the proposed environment?
- How are critical vulnerabilities handled outside normal release cycles?
- What are the customer's rights and practical steps at contract termination?

SCORING MODEL

Use weighted criteria linked to business risk. Product functionality should not dominate the score when operational continuity, data sensitivity, or public accountability are material. Separate mandatory evidence from scored quality. A persuasive presentation is not evidence of portability, maintainability, or recoverability.

CONTRACT AND ACCEPTANCE

Tie payment milestones to verifiable outcomes: documented interfaces, tested recovery, automated deployment, security evidence, complete handover, and accepted knowledge transfer. Define maintainability and transition as delivery requirements rather than optional end-of-contract activities.

THE DECISION PRINCIPLE

Choose the operating model, not merely the software. Open source creates value when rights, architecture, skills, support, and governance combine to preserve real options over the full service life.

© 2026 BusinessFragen OG · info@it-governance.net · it-governance.net