ManagerWISSEN

ANZEIGE

Ausgabe Januar 1/2026

Ad Special im Harvard Business manager
www.manager-wissen.com

OPEN SOURCE SOFTWARE

# Open Source als strategischer Imperativ für Innovation, Souveränität und nachhaltigen Unternehmenserfolg

*Von Marcel Scholze*

**Open Source ist nicht ein reines IT-Thema, sondern ein Wirtschaftsfaktor von strategischer Bedeutung. In einer Welt geopolitischer Spannungen, wachsender Cloud-Abhängigkeiten und rasanter technologischer Umbrüche wird digitale Souveränität zur ökonomischen Notwendigkeit. Unternehmen, die Open Source aktiv steuern, sichern sich Innovationskraft, Effizienz sowie Resilienz und senken zugleich ihre Kosten und Risiken. Entscheidend ist jedoch die Einstellung: Nur wer Open Source nicht als technisches Werkzeug, sondern als strategisches Vehikel versteht, kann dessen Potenziale realisieren. Es geht um mehr als Code: Es geht um Kontrolle, Wettbewerbsfähigkeit und Zukunftsfähigkeit im digitalen Zeitalter.**

### *Warum Unternehmen jetzt eine Open-Source-Strategie brauchen*

Nahezu jedes Unternehmen nutzt Open Source Software (OSS), häufig fragmentiert und ohne zentrale Steuerung. Diese unkoordinierte Nutzung verursacht Sicherheits- und Compliance-Risiken, ungesteuerte Abhängigkeiten und verpasste Innovationen und Marktpositionierung.

Eine gezielte Open-Source-Strategie schafft Transparenz und Steuerbarkeit auf Managementebene. Sie definiert klare Ziele, Verantwortlichkeiten und Ressourcen, legt Prozesse für Lizenz- und Sicherheitsprüfungen, Contribution und Vendor-Management fest und steuert den Einsatz in Entwicklung, Produkten und Betrieb. Messbare KPIs erlauben eine kontinuierliche Bewertung des Erfolgs und sichern die langfristige Wertschöpfung.

### *Was ist Open Source Software?*

OSS unterscheidet sich von proprietärer Software dadurch, dass ihr Quellcode offen einsehbar ist. Die Software kann an die individuellen Bedürfnisse angepasst, weiterentwickelt, in den Technologie-Stack integriert und weitergegeben werden. Die Grundlage bildet das Open-Source-Lizenzmodell, das die Freiheiten der Nutzung, Veränderung und Weitergabe einräumt. Im Gegensatz dazu verfolgt proprietäre Software einen anderen Ansatz: Der Quellcode bleibt geschlossen und die Nutzung, Updates, der Support sowie die Weiterentwicklung werden durch den Anbieter i. d. R. mittels entgeltlicher Lizenzen und Verträge geregelt.

Unternehmen ohne solche Strukturen riskieren nicht nur operative Ineffizienzen, sondern auch rechtliche und regulatorische Konsequenzen – etwa durch Anforderungen aus dem Cyber Resilience Act (CRA) oder dem Digital Operational Resilience Act (DORA). Wer hingegen konsequent steuert, kann Kosten senken, Entwicklungszyklen verkürzen, Risiken minimieren und digitale Souveränität gewinnen. Praxistipp: Starten Sie pragmatisch mit einem Audit der aktuellen Nutzung, schließen Sie bestehende Governance-Lücken und setzen Sie priorisiert Maßnahmen um, die kurzfristig Sicherheit gewährleisten und langfristig nachhaltige Wertschöpfung sichern.

### *Einordnung zu wichtigen Themen rund um Open Source Software*

OSS bietet Unternehmen klare Vorteile in den folgenden zentralen Handlungsfeldern:

**1. Security:**
OSS wird von einer globalen Community kontinuierlich geprüft, Schwachstellen werden schnell identifiziert und geschlossen. Etwaige Sicherheitsbedenken gegenüber OSS lassen sich nicht durch die Bevorzugung proprietärer Lösungen mindern, da auch diese umfangreich OSS enthalten; zudem erschwert die fehlende Transparenz über die eingesetzten Open-Source-Komponenten als auch über den proprietären Code insgesamt die eigenständige Beurteilung der Sicherheit des Gesamtprodukts. Mit OSS können Unternehmen selbst prüfen, Schwachstellen beheben und die Software verbessern.

**2. Lizenzkomplexität:**
Open-Source-Lizenzen sind klar definiert und juristisch etabliert. In professionellen Compliance-Prozessen lassen sie sich automatisiert steuern, während proprietäre Verträge oft komplexe Sonderklauseln, Lock-ins und langfristige Bindungen enthalten.

**3. Supportverfügbarkeit:**
Für geschäftskritische OSS können Unternehmen weltweit individualisierbare Support-, Wartung- und SLA-Optionen von unterschiedlichen Anbietern beschaffen. Proprietäre Lösungen binden Support häufig exklusiv an den Hersteller, inklusive Preis- und Roadmap-Abhängigkeiten.

### *Was erwartet Sie in diesem Special?*

Open Source Software ist seit Jahren ein unverzichtbarer Bestandteil der IT-Landschaft, über 95 Prozent aller Software enthalten OSS-Komponenten. OSS ist Innovationstreiber, Kostensenker und Baustein digitaler Souveränität – dennoch bestehen in manchen Unternehmen Vorbehalte zu Sicherheit, Haftung, Wartung und Support. Dieses Special zeigt praxisnah, wie Führungskräfte strategische Rahmenbedingungen für den vorteilhaften und sicheren Einsatz von Open Source Software schaffen können.

***Inhaltsübersicht:***



**Impressum**

**Verlag:** Reif Verlag GmbH · Alfred-Jost-Straße 11 · 69124 Heidelberg
Peter Reif · peter.reif@reifverlag.de · www.manager-wissen.com

**Redaktion:** Christian Deutsch · info@deutsch-werkstatt.de
Regina Gödde, E-Mail: regina.goedde@reifverlag.de
Layout: metropolmedia, 69245 Bammental · Druck: ColorDruck Solutions, 69181 Leimen

**4. Stabilität:**
Globale Infrastrukturen – Internet, Cloud, Automotive, Telekommunikation – laufen seit Jahren auf OSS mit hochgereiftem, unter Dauerlast optimiertem Code. OSS liefert kontinuierlich „Proof-of-Production" in geschäftskritischen Umgebungen.
**5. Usability:**
Moderne Open-Source-Produkte setzen auf konsequentes Produktdesign, Nutzerfeedback in Echtzeit und schnelle Iteration – schlechte Usability überlebt in diesem Modell schlicht nicht. Proprietäre Software ist oft an Roadmaps gebunden, die sich an Vertriebszyklen orientieren, nicht am tatsächlichen Nutzererlebnis.
**6. Transparenz:**
OSS macht Architektur, Abhängigkeiten und Schwachstellen sichtbar - Entscheider:innen können nachvollziehen, welche Komponenten im Kern der Wertschöpfung eingesetzt werden. Dies ist ein zentraler Faktor für regulatorische Vorgaben wie DORA und CRA, welche ein zentrales Software-Inventar mittels Software Bill of Materials (SBOM) erfordern. Proprietäre Black Boxes erschweren solche Nachvollziehbarkeit, Open Source ermöglicht sie „by Design".

### *Die strategische Rolle von Open Source Software für den Unternehmenserfolg*

Unternehmen können die mit OSS verbundenen Compliance- und Sicherheitsrisiken durch professionelle Praktiken minimieren. Gleichzeitig lassen sich die Vorteile von OSS strategisch als konkrete Chancen für den Gesamterfolg nutzen.

Der strategische Einsatz von OSS kann einen positiven Einfluss auf verschiedene Geschäftsbereiche und Prozesse haben. Dazu zählen unter anderem eine gesteigerte Innovationskraft in der Anwendungsentwicklung, souveräne (interne) Lösungen und Prozesse, eine beschleunigte Entwicklung digitaler Produkte für den Markt sowie die Stärkung der Wettbewerbsposition und die Entwicklung neuer Geschäftsmodelle durch und mit Open Source. Es gibt vielfältige Möglichkeiten und Potenziale, Open Source Software strategisch für den Erfolg eines Unternehmens einzusetzen. Die Nicht-Ausschöpfung solcher Chancen kann im internationalen Wettbewerb zu einem signifikanten Nachteil führen.

> „Der bewusste Einsatz und das aktive Management von Open Source Software sollten von Unternehmen als wichtiges Instrument verstanden werden, das strategisch auf den Unternehmenserfolg einzahlt."

### *Digitale Souveränität sichern – Vendor-Lock-ins reduzieren*

Offener Quellcode und transparente Abhängigkeiten verschaffen Unternehmen Kontrolle über ihre IT-Infrastruktur und senken das Risiko von Vendor-Lock-ins. Im Falle kritischer Abhängigkeiten von proprietären Vendoren sind OSS-Lösungen echte Alternativen und stärken die Verhandlungsposition gegenüber Herstellern.

Open Source trägt nicht nur zur Reduzierung von Lizenzkosten bei, sondern insbesondere von strukturellen Lock-in-Kosten. Wer über Kenntnisse und Kontrolle des Codes verfügt, kann Preise, Service-Level und Weiterentwicklung aus einer Position der Stärke heraus verhandeln. Proprietäre Modelle erzeugen durch Migrationshürden und Named-User-Lizenzen stille Fixkosten, die stetig steigen können.

Darüber hinaus gewährleistet der Einsatz von OSS die Sicherheit und Flexibilität in Anpassung, Wartung und Weiterentwicklung. Unternehmen können ihre IT-Landschaft langfristig an strategischen Zielen ausrichten – unabhängig von geopolitischen Interessen, Exportrestriktionen oder der Produktpolitik eines Herstellers. Das gezielte Opensourcen eigener Produkte kann ein strategischer Schritt sein, um Kundinnen und Kunden mehr Souveränität zu ermöglichen und damit aktuelle Marktbedürfnisse aktiv zu adressieren.

### *Innovation beschleunigen durch offene Kooperation*

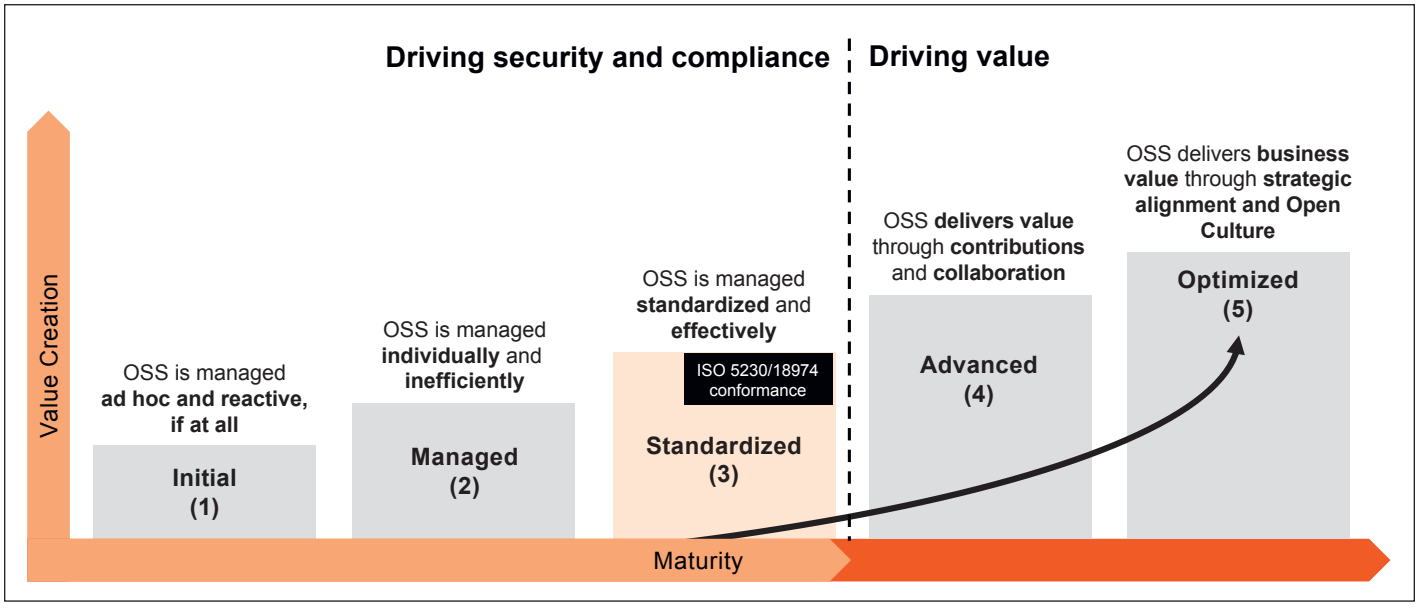
Über die reine Nutzung hinaus entfaltet insbesondere das aktive Engagement im OSS-Ökosystem weitere Vorteile und sollte daher auch Bestandteil der übergeordneten OSS-Strategie sein. Unternehmen profitieren von dem Know-how globaler Communities, beispielsweise durch die schnelle Verfügbarkeit von Patches oder durch Feedback zu selbst veröffentlichten Projekten.

Gleichzeitig stärkt die aktive Beteiligung am Ökosystem die Wahrnehmung des Unternehmens am Markt als verantwortungsbewusster und innovativer Akteur. Entwicklerinnen und Entwickler schätzen die Möglichkeit, sichtbar zu kollaborieren – beispielsweise über GitHub. Dies macht OSS zu einem wichtigen Faktor im Wettbewerb um digitale Talente. Open Source ist somit nicht nur ein Innovationstreiber, sondern auch ein wichtiges Merkmal moderner Arbeitgeber.

### *Kosten senken, Ressourcen schonen*

Der strategische Einsatz von OSS bietet Unternehmen zahlreiche Vorteile, die über die Einsparung von Lizenzkosten hinausgehen. Die offene Verfügbarkeit von Komponenten ermöglicht eine Reduzierung des Aufwands für Entwicklung, Integration und Wartung. Support und Betrieb können flexibel extern bezogen oder gezielt intern aufgebaut werden.

OSS ermöglicht die Bereitstellung maßgeschneiderter Lösungen, ohne dass kostspielige, etwaig überdimensionierte Zusatzmodule oder verpflich-




*PwC OSS Maturity Modell: Wertschöpfung durch professionalisiertes OSS-Management.* *Quelle: PwC*

tende Upgrades erforderlich sind. Investitionen fließen gezielt in den eigenen Kompetenzaufbau statt in Lizenzmodelle. Zudem unterstützt Inner Sourcing – die offene Zusammenarbeit innerhalb des Unternehmens – die Innovationsfähigkeit und trägt durch Ressourceneffizienz auch zur Erreichung von Nachhaltigkeitszielen bei.

Insgesamt zahlt eine OSS-Strategie auf den ROI einer Organisation ein, indem die Time-to-Market von Produkten erheblich gesenkt wird und gleichzeitig das Risiko geschäftskritischer Compliance oder Security-Vorfälle minimiert wird. So können beispielsweise Lizenzverstöße oder Sicherheitslücken dramatische Auswirkungen auf die Business Continuity und den Gesamterfolg haben, wenn Produktionslinien ausfallen, Produkte zurückgerufen werden müssen bzw. deren Vertrieb gänzlich eingestellt werden muss.

### *Resilienz stärken durch Transparenz und Software Bill of Materials (SBOM)*

Der Einsatz von offenem Quellcode gewährleistet Transparenz in Bezug auf Abhängigkeiten und Risiken. Unternehmen können Sicherheitslücken, Lizenz- und Konzentrationsrisiken frühzeitig erkennen und aktiv steuern. Die Erstellung sogenannter Software Bill of Materials (SBOM) erleichtert zudem die Reaktion auf neue Schwachstellen und die Erfüllung regulatorischer Anforderungen – etwa aus EU CRA und DORA.

Standardisierte SBOMs und Zertifizierungen nach internationalen Normen wie ISO/IEC 5230 und ISO/IEC 18974 beschleunigen Due-Diligence-Prozesse (z. B. im M&A), senken Integrationsrisiken und ermöglichen eine risikoorientierte Beschaffung – insbesondere in regulierten Branchen und KRITIS-Umgebungen.

### *Exkurs: ISO-Standards zum OSS- Management als Basis für regulatorische Compliance*

Im Kontext zunehmend komplexer Lieferketten, wachsender Cyber-Bedrohungen und verschärfter regulatorischer Vorgaben wie EU CRA und EU DORA ist es unerlässlich, dass OSS Governance, Security und Lizenz-Compliance nachweisbar in operative Prozesse integriert sind.

Die ISO/IEC 18974 für OSS-Security-Management und ISO/IEC 5230 für Lizenz-Compliance-Management bilden hierfür eine wichtige Grundlage, sowohl für die unternehmensweite Governance als auch die Zusammenarbeit mit Dienstleistern. Der Fokus beider Normen liegt auf der Etablierung angemessener Governance-Strukturen, um ein durchgängiges Management von Open Source Software während des gesamten Lebenszyklus zu gewährleisten. Dies umfasst unter anderem dokumentierte Richtlinien und Prozesse von Open Source Software mit klar definierten Rollen und Verantwortlichkeiten.

Eine Zertifizierung durch Dritte stellt einen wichtigen Aspekt dar, da sie als Nachweis der Konformität Vertrauen in der Software-Lieferkette schafft.

### *OSS-Governance-Reifegradmodell*

Das PwC OSS-Reifegradmodell beschreibt fünf Stufen der Entwicklung von Open-Source-Managementsystemen in Unternehmen. Es zeigt, wie OSS von einem operativen Werkzeug zu einem strategischen Werttreiber wird:

**1. Initial / Ad-hoc:**
OSS wird unkoordiniert eingesetzt, Prozesse fehlen weitgehend. Compliance- und Sicherheitsrisiken sind hoch, Transparenz über Abhängigkeiten gering.

**2. Managed / Repeatable:**
Erste Governance-Strukturen existieren, Nutzung und Risiken werden teilweise dokumentiert. Prozesse sind noch nicht vollständig standardisiert oder ISO-konform.

**3. Standardized:**
Prozesse sind nach ISO/IEC 5230 und 18974 aufgebaut und erfüllen regulatorische Anforderungen wie EU CRA und DORA. Unternehmen können Risiken systematisch kontrollieren und Compliance sicherstellen.

**4. Advanced:**
OSS wird strategisch eingesetzt, Unternehmen beteiligen sich aktiv am Open-Source-Ökosystem. Beiträge aus der Community werden genutzt, eigene Projekte veröffentlicht, Innovation und Talentbindung gestärkt.

**5. Optimized:**
Offenheit ist fest in der Unternehmenskultur verankert. OSS prägt Prozesse, Produkte und Geschäftsmodelle. Unternehmen erzielen maximale Effekte in Innovation, Resilienz, Kostenkontrolle und Wettbewerbsfähigkeit.

Je höher der Reifegrad, desto größer der messbare Beitrag von OSS zum Unternehmenserfolg. Wer sein OSS-Management kontinuierlich weiterentwickelt, transformiert Open Source von einem operativen Tool zu einem strategischen Hebel für Innovation, digitale Souveränität, Risikoreduktion und nachhaltige Wertschöpfung – intern wie entlang der gesamten Wertschöpfungskette.

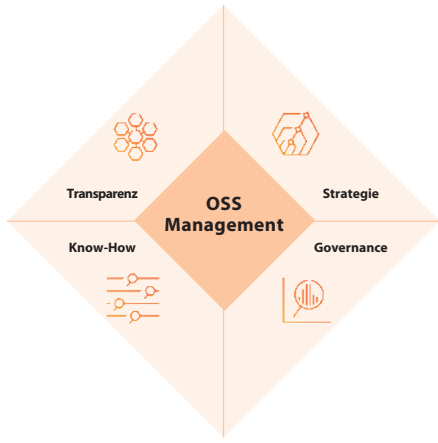
### *Entscheider:innen müssen jetzt aktiv werden!*

Professionelles Open-Source-Management ist längst kein „Nice-to-have" mehr, sondern eine entscheidende strategische Wahl. OSS ist heute ein integraler Bestandteil moderner IT-Infrastrukturen und damit ein strategischer Erfolgsfaktor sowie ein Instrument, um Innovationskraft, Resilienz und digitale Souveränität gezielt zu stärken.

Entscheiderinnen und Entscheider müssen jetzt die notwendigen Strukturen schaffen, um im internationalen Wettbewerb nicht den Anschluss zu verlieren. Im Zentrum steht die strategische Nutzung der Open-Source-Potenziale im Einklang mit der Unternehmensstrategie. Dabei sollte besonderer Fokus auf eine solide Governance nach anerkannten ISO-Standards gelegt werden.

### *Vier Schritte sind dabei entscheidend:*

**1. Transparenz schaffen:**
Identifizieren Sie den bisherigen Open-Source-Footprint Ihrer Organisation, kritische Abhängigkeiten, Sicherheits- und Compliance-Risiken. Nur auf dieser Basis lassen sich die Potenziale von Open Source gezielt und kontrolliert realisieren.




*Vier wesentliche Schritte zum professionellen OSS-Management.* *Quelle: PwC*

**2. OSS-Strategie entwickeln:**
Richten Sie Ihre Open-Source-Strategie an den Unternehmenszielen aus und prüfen Sie, ob Ihr Geschäftsmodell angepasst werden muss. Differenzieren Sie dabei den Mehrwert von OSS z. B. in Ihrer eigenen Produkt- und Serviceentwicklung, in der Enterprise-IT, bei Mergers & Acquisitions, bei der Teilhabe im OSS-Ökosystem und in Bezug auf die eigene Souveränität sowie Souveränität Ihres Angebots an Ihre Kunden.

**3. Governance und Tooling aufbauen:**
Etablieren Sie ein integriertes Open-Source-Management mit klaren Richtlinien, Prozessen, Rollen und Verantwortlichkeiten – über IT, Recht, Einkauf, Produktentwicklung und Management hinweg. Eine durchgängige Toolchain ist essenziell, um OSS entlang des gesamten Lebenszyklus effektiv zu steuern.

**4. Kompetenzen und Community stärken:**
Der Aufbau von Open Source Know-how ist ein klarer Wettbewerbsvorteil. Führungskräfte profitieren durch besseres Risikomanagement und höhere Innovationsfähigkeit, während Entwicklungsteams die Besonderheiten von OSS verstehen und gezielt im Ökosystem mitwirken. So entsteht ein nachhaltiger Mehrwert im Einklang mit den Unternehmenszielen.

### *Fazit*

Wer Open Source strategisch steuert, verwandelt Risiken in Innovation, digitale Souveränität und Unabhängigkeit, stärkt Resilienz und Wettbewerbsfähigkeit – und macht OSS so zum entscheidenden Hebel nachhaltiger Wertschöpfung.

**Der Autor**

***Marcel Scholze,*** *Director für Open Source, Digitale Souveränität und IT-Sourcing bei PwC Deutschland.*

*Mehr hier :*

# Compliance im Open-Source-Umfeld mit Community-Spirit sichern

*Von Larissa lütke Zutelgte und Stefan Thanheiser*

**Open Source – no rules? Think again! (Pro)aktives Management von Open Source Software (OSS) ist entscheidend für Sicherheit und Compliance. Scope und Sizing des Compliance-Systems sollten sich am spezifischen OSS-Risiko ausrichten: Denn aktives Engagement in der OSS-Supply-Chain stellt andere Anforderungen als rein hausinterne Nutzung. Unsere Erfahrung zeigt: Transparenz entsteht nicht durch Kontrolle, sondern durch Zusammenarbeit, gute Tools und engagierte Menschen. Wer Open Source aktiv managt, braucht kein starres Regelwerk – sondern Klarheit, Kompetenz und kontinuierliches Housekeeping.**

### *Open Source - große Chancen verlangen Verantwortung*

Open Source Software (OSS) steckt heute in jeder Software. Damit rücken bei Unternehmen Fragen zu Architektur, Security und Lizenzierung ebenso in den Fokus wie beim Management proprietärer Software.

Regulatorische Vorgaben wie MaRisk, DORA und der Cyber Resilience Act erhöhen zusätzlich den Druck, sich professionell mit OSS-Compliance auseinanderzusetzen. Denn auch die Nutzung unterliegt rechtsgültigen Verträgen mit den Urhebern, die Verpflichtungen für den Lizenznehmer beinhalten. Dieser muss nachweisen können, dass er die Software vertragskonform und unter Einhaltung aller Verpflichtungen nutzt.

Die Herkunft, Lizenzkompatibilität und gegebenenfalls vorhandene Security-Schwachstellen von Fremd- und Eigensoftware müssen geprüft werden.

Je nach Position eines Unternehmens in der Software Supply Chain werden Informationen von Zulieferern benötigt („upstream"), um diese beispielsweise bei Services gegenüber Kunden ausweisen zu können („downstream").

### *OSS-Management: Einfach und komplex zugleich*

Ein Vorteil von OSS im Hinblick auf die Lizenz-Compliance besteht darin, dass ganze Lizenzen - etwa Apache-2.0 oder MIT – vorab geprüft und für bestimmte Einsatzszenarien freigegeben werden können. Diese (Nicht-)Freigaben gelten dann für viele Komponenten gleichzeitig, sodass nicht jede einzelne erneut bewertet werden muss. Komplexitätssteigernd wirken Dependencies und transitive Dependencies: Eine OSS integriert eine andere OSS, welche wiederum selbst auf weiteren OSS-Komponenten basiert. Durch diese mehrstufige Verschachtelung können komplexe Software- und Lizenz-Stacks entstehen. OSS-Communities achten zwar gewöhnlich auf Lizenzkompatibilität, doch dies ist nicht in allen Fällen garantiert.

### *Unser Weg zur „OSS-Schwarmintelligenz"*

Atruvia hat die internationale OpenChain-Spezifikation (ISO/IEC 5230) als Blaupause genutzt, um Rollen, Prozesse und Verantwortlichkeiten für den Umgang mit Open Source Software klar zu definieren. Statt sofort ein umfassendes Regelwerk einzuführen, setzte das Unternehmen bewusst auf organisches Wachstum: Strukturen sollten sich entwickeln, reifen und breite Akzeptanz finden.

Der Startpunkt war eine Grassroots-Initiative: Engagierte Kolleg*innen aus IT-Architektur, Software-Asset-Management, IT-Security und der CI/CD-Pipeline schlossen sich zusammen, um den Einsatz von OSS gemeinsam zu steuern. Unterstützt durch Legal und Risikomanagement entstand daraus das Open Source Governance Board (OSGB) – heute die zentrale Instanz für ein unternehmensweites, transparentes OSS-Management.

Das OSGB versteht sich als Steward für Open Source: Es fördert den aktiven Einsatz von OSS, schafft Orientierung und berät, um rechts-, sicherheits- und lizenzkonforme Nutzung abzusichern. Nach dem Prinzip „Trust but Verify" werden die Leitplanken kontinuierlich fortgeschrieben – abgestimmt über alle Funktionsbereiche hinweg. So entsteht ein belastbares Gleichgewicht zwischen Innovationsförderung und Risikomitigation – ein Governance-Ansatz, der Open Source strategisch nutzbar macht, ohne seine Agilität zu verlieren.

### *Von der Theorie zum Werkzeug*

Die bei Atruvia etablierten Strukturen und Prozesse bildeten die Grundlage für den nächsten Schritt: die Einführung eines Open-Source-Scanners, der während des Build-Prozesses Lizenzen und Sicherheitsrichtlinien automatisch überwacht. So wurde ein kontinuierliches Compliance- und Security-Monitoring geschaffen, das Risiken frühzeitig erkennt und die Entwicklungsprozesse absichert.

Gerade für größere Unternehmen, die Software entwickeln, ist diese technische Unterstützung unverzichtbar. Die schiere Menge an Open-Source-Komponenten – bei Atruvia sind es mehrere Zehntausend – lässt sich ohne Automatisierung weder effizient noch zuverlässig managen.

Bei der Auswahl geeigneter Tools kommt es auf die Art, Tiefe und Intensität der OSS-Nutzung an. Der Markt bietet eine große Bandbreite an Lösungen – von proprietären Scannern bis zu leistungsfähigen Open-Source-Tools wie FOSSology oder OWASP Dependency Check. Je nach Compliance-Schwerpunkt können zusätzliche Funktionen integriert werden, etwa zur Lizenz- und Schwachstellenanalyse, zum Code-Vergleich oder für Self-Service- und Legal-Workflows.

Nach eingehender Bewertung entschied sich Atruvia – trotz der starken Verbundenheit mit Open Source – bewusst für eine umfassende proprietäre Scanner-Suite. Ausschlaggebend war dabei die Möglichkeit, sämtliche Compliance-Anforderungen zentral und skalierbar abzubilden.

### *Fazit*

Open Source ist aus moderner Softwareentwicklung nicht wegzudenken – bringt aber lizenzrechtliche, sicherheitstechnische und organisatorische Herausforderungen mit sich.

Bei Atruvia begegnen wir dieser Komplexität mit einem strukturiertem OSS-Governance-System auf Basis der OpenChain-Spezifikation, das bereichsübergreifende Zusammenarbeit fördert und technische Automatisierung etwa durch Build-begleitende Scanner etabliert.

### *Unsere Handlungsempfehlungen*

Für einen nutzenorientierten und sicheren Einsatz von OSS empfehlen wir folgende Maßnahmen:

**1. Analyse, Orient, Decide**
Umfang und Tiefe der OSS-Nutzung bestimmen das Anforderungsprofil für Aufbau und Durchführung eines OSS-Managements.

**2. From Grassroots to Governance**
Erfolgreiches OSS-Management ist abhängig von Mitarbeitern aller Organisationseinheiten. Enablement OSS-begeisterter Menschen ist der Hebel für breite Akzeptanz.

**3. Keep it simple**
Keine seitenlangen Anträge. Wenige, klare Regeln, kurze Lernangebote, konkrete Beispiele und interne Sprechstunden wirken besser als Verbote.

**4. Compliance as Code**
Build-begleitende Compliance-Automation ersetzt manuelle Prüfungen. Compliance-Tools und OSS-Scanner skalieren und automatisieren das Compliance-Handling.

**Kurz gesagt:** Wer Open Source strategisch einsetzt, verbindet Innovation, Effizienz und Compliance. Ein durchdachtes Governance-Modell macht OSS zu einem echten Wettbewerbsvorteil – statt zu einem Risiko.

**Die Autoren**

***Larissa lütke Zutelgte*** *und* ***Stefan Thanheiser,*** *Software Asset Manager bei Atruvia AG, fokussieren sich auf lizenzsicheren Einsatz von Open Source Software. Beide schätzen die funktionsübergreifende Zusammenarbeit im Atruvia Open Source Governance Board.*

# Das Open Source Program Office als Strategievorteil

*Von Sebastian Wolf*

**SAP und Open Source – diese Kombination mag zunächst überraschen. Viele sehen in dem deutschen Software-Riesen eher einen traditionellen Software-Anbieter als einen Vorreiter im Bereich Open Source. Doch dieser Eindruck täuscht: SAP begreift Open Source seit vielen Jahren als strategisches Thema und betreibt seit 2018 ein eigenes Open Source Program Office (OSPO). Der folgende Bericht zeigt, wie ein OSPO erfolgreich implementiert werden kann und welchen konkreten Wertbeitrag es leisten kann.**

### *Die Bedeutung von Open Source für SAP*

SAPs Engagement im Bereich Open Source reicht weit zurück: Bereits 1998 wurde unser Hauptprodukt SAP R/3 auf Linux portiert. Seitdem hat SAP ihren Open-Source-Fußabdruck stetig vergrößert. Open Source ist im Unternehmen unverzichtbar, denn heute nutzen fast alle SAP-Produkte Open-Source-Software oder hängen davon ab. Weiterhin entwickeln wir selbst Hunderte eigener Open-Source-Projekte, darunter prominente Beispiele wie SapMachine (eine OpenJDK-Distribution) oder OpenUI5 (ein JavaScript-UI-Framework).

Sowohl die Nutzung als auch die aktive Beteiligung an Open-Source-Projekten folgen dabei klaren unternehmerischen Zielen: Durch die Nutzung von Open Source Software (OSS) können wir die Entwicklungskosten deutlich senken und unsere Entwicklungsgeschwindigkeit erhöhen. Wir nutzen eigene Beiträge beispielsweise dazu, unsere Plattform und Anwendungen für Kunden- und Partnerentwicklungen leichter zugänglich zu machen. Der Nutzen ist messbar: Im Rahmen der Corona-Warn-App ergab eine Befragung einen Return on Investment von 200 Prozent – bestätigt durch eine Untersuchung der Weltbank[1].

Gleichzeitig bringt der Einsatz von Open Source auch einige Herausforderungen mit sich. Hierzu zählen vor allem die rechtlichen Rahmenbedingungen, etwa im Bereich Lizenz-Compliance, aber auch die Sicherheit sowie verschiedene operative Aspekte wie Mitarbeitertrainings oder die Gestaltung unternehmensinterner Open-Source-Prozesse.

### *Das OSPO als zentrale Stabsstelle für Open Source*

Bis 2018 wurden Open-Source-Themen bei SAP dezentral in verschiedenen Abteilungen betreut, was schwierig zu koordinieren war. Mit der Gründung des Open Source Program Office[2] als Stabsstelle beim Chief Technology Officer (CTO) bündelte SAP alle Aktivitäten in einer zentralen Einheit.

### *Erfolge des OSPO bei SAP*

Die Investition in ein zentrales OSPO hat sich vielfach bewährt. Drei Beispiele verdeutlichen den konkreten Wertbeitrag:

**1. Unternehmensweite Open Source Policy und Training**

In Zusammenarbeit mit weiteren Abteilungen entwickelte das OSPO eine einheitliche Open-Source-Policy, die unternehmensweit gültige Standards und Verfahren definierte. Begleitet wurde diese durch ein virtuelles Trainingsprogramm, das für alle Entwicklungsrollen verpflichtend ist. Dadurch konnte nicht nur die Rechtssicherheit erhöht, entsprechende Risiken beim Einsatz von Open Source minimiert sondern auch das Open-Source-Bewusstsein im Unternehmen geschärft werden.

*Zentrale Aufgaben des SAP Open Source Program Office (OSPO)*

- Entwicklung und Durchsetzung unternehmensweiter Open-Source-Richtlinien
- Koordination aller Open-Source-Aktivitäten zwischen Geschäftsbereichen
- Aufbau und Vermittlung von Expertise sowie Best Practices
- Sicherstellung von Lizenz-Compliance und Risikomanagement
- Förderung einer offenen, kollaborativen Unternehmenskultur

**Ergebnis:** Transparenz, Sicherheit und Innovationskraft im Umgang mit Open Source.

**2. Corona Warn App-Projekt**

Im Jahr 2020 wurde SAP gemeinsam mit der Deutschen Telekom beauftragt, die Corona Warn App vollständig als Open-Source-Lösung zu entwickeln. Dank der etablierten Strukturen des OSPO konnten alle erforderlichen Teilprojekte innerhalb kürzester Zeit aufgesetzt und die entsprechenden Richtlinien umgesetzt werden. Dies war ein entscheidender Faktor für die erfolgreiche und transparente Umsetzung.

**3. IPCEI-CIS und NeoNephos-Foundation**

Im Rahmen des EU-Projekts IPCEI-CIS[3] zur Schaffung einer Sovereign Cloud-Infrastruktur koordinierte das OSPO nicht nur die Veröffentlichung verschiedener Komponenten als Open Source, sondern trieb auch die Gründung der NeoNephos-Foundation[4] unter dem Dach der Linux Foundation Europe voran. Unser OSPO konnte so über Unternehmensgrenzen hinweg strategische Partnerschaften fördern und zur Gestaltung technologischer Standards beitragen.

### *Erfolgsfaktoren für OSPOs*

Die Erfahrungen bei SAP zeigen: Ein OSPO trägt nur dann zum Unternehmenserfolg bei, wenn es fest in der Unternehmensstrategie verankert ist und mit entsprechenden Befugnissen ausgestattet ist. Kritische Aufgaben wie das Erstellen und Umsetzen von Richtlinien, das Lizenzmanagement und die Betreuung der Community sollten von einem zentral verantwortlichen Team übernommen werden. Wenn diese Funktionen auf verschiedene Bereiche verteilt sind, entstehen unnötige Risiken und Ineffizienzen.

Ein OSPO sollte sich stets als Service-Abteilung verstehen, welche die Entwicklungsteams unterstützt. Richtlinien, Werkzeuge und Prozesse sollten immer auf ein Minimum reduziert und regelmäßig auf ihre Notwendigkeit und Zielorientierung hin überprüft werden. Nur so wird das OSPO als vertrauenswürdiger Ansprechpartner wahrgenommen.

Der Austausch mit Open-Source-Verantwortlichen anderer Unternehmen ist elementar wichtig, sei es zum Erfahrungsaustausch oder um sich gegenseitig zu unterstützen. Open-Source-Experten verschiedener Unternehmen organisieren und vernetzen sich in der TODO Group[5] oder der OSPO Alliance. Ihre Ressourcen und Hilfestellungen[6] können bei der Gründung eines eigenen OSPO unterstützen.

*Open Source bei SAP – Take-Aways auf einen Blick*

- Offenheit steigert Innovation, Effizienz und Wettbewerbsfähigkeit.
- Das OSPO bündelt Richtlinien, Wissen und Verantwortung.
- Klare Governance beschleunigt Entwicklungsprozesse.
- Zusammenarbeit mit der Community erhöht Qualität und Geschwindigkeit.
- Kulturwandel und Trainings sichern nachhaltige Open-Source-Kompetenz.

### *Fazit*

Das Beispiel SAP zeigt: Ein OSPO ist weit mehr als eine organisatorische Einheit – es ist ein strategischer Enabler für Innovation, Zusammenarbeit und digitale Souveränität. Unternehmen, die Open Source gezielt steuern, schaffen nicht nur mehr Transparenz und Sicherheit, sondern stärken auch ihre technologische Unabhängigkeit und Innovationskraft.

Quellenangaben:
1) Leveraging Open Source as a Public Institution – New analysis reveals significant returns on investment in open source technologies
2) SAP Open Source
3) Cloud – Competition Policy – European Commission
4) NeoNephos Foundation
5) TODO Group // Talk openly, develop openly
6) How to create an open source program office | TODO Group // Talk openly, develop openly

**Der Autor**

***Sebastian Wolf*** *beschäftigt sich seit über 25 Jahren mit Open Source und arbeitet als Principal Software Architect im SAP-OSPO. Hierbei begleitet er seit 2020 den Aufbau und die Weiterentwicklung von Open Source im Unternehmen.*

# Vom Nutzer zum Gestalter: Community-Engagement stärkt Erfolg

*Von Katrin Kahle*

**Open Source ist kein Einkaufsgegenstand, sondern kollaborative Wertschöpfung. Wer Open Source Software (OSS) strategisch einsetzt, sollte die Community-Beteiligung fest einplanen, denn sie erhöht Qualität und Sicherheit, verkürzt Entwicklungs- und Zertifizierungszyklen, stärkt technologische Souveränität und reduziert Abhängigkeiten. Entscheider erfahren in diesem Beitrag, wie sie die Unterschiede zu Closed Source bei der Nutzung von Open Source adressieren können - Governance, Beteiligung, Planungssicherheit und Kommunikation mit dem Maintainer sind hierbei entscheidende Faktoren.**

Open Source ist heute strategische und kritische Infrastruktur. Der größte Wert entsteht nicht durch Konsum, sondern durch Beteiligung. Wer sich in Communities einbringt, rückt näher an Roadmaps, erkennt Risiken früher, beschleunigt Sicherheitspatches und gewinnt Gestaltungsmacht über eine Basistechnologie, von der Produkte und Geschäftsmodelle abhängig sind.

### *Vom Produktnutzer zum Technologiepartner*

Viele Unternehmen behandeln Open Source wie ein Produkt: Sie erwarten Stabilität, Support und SLAs. Open Source funktioniert jedoch als Beziehungssystem. Die Effekte wachsen mit der Nähe zum Projekt: strukturierte Bugreports, reproduzierbare Testfälle, Dokumentation, Reviews, gezieltes Co-Funding oder eigene Beiträge. So wird aus dem „Einkauf von Code" ein Investment in Resilienz und Einfluss. Dieser Aktivitätswechsel verbessert die Steuerbarkeit: Anstatt Änderungen extern zu „erbitten", sind Sie Teil eines gestaltbaren Prozesses.

### *Was bei Open Source anders geplant werden muss als bei Closed Source*

1. **Bewertung der Kritikalität:**
   Je bedeutsamer die jeweilige OSS für den Unternehmenszweck ist, desto stärker sollte Community-Beteiligung bzw. Contribution eingeplant werden.
2. **Analyse der Governance Struktur:**
   Jedes OSS-Projekt ist anders – analysieren Sie Strukturen, Rollen und Beteiligungsmöglichkeiten.
3. **Governance & Verantwortung:**
   Definieren Sie in Ihrer Organisationsstruktur frühzeitig Rollen und Policies: Wer entscheidet über Komponentenwahl? Wer darf Beiträge freigeben? Wie ist der Security-Prozess organisiert (Meldewege, Patch-Fenster, Backports)? Eine OSPO-ähnliche Funktion – im Konzern als Office, in kleineren Firmen als klar benannte Rolle wie beispielsweise Steward – verzahnt Nutzung, Beiträge, Compliance und Community-Beziehungen für das entsprechende OSS-Projekt.
4. **Transparenz & Planbarkeit:**
   Fragen Sie bei kritischen Abhängigkeiten nach Release-Fenstern, geplanten Abkündigungen (Deprecation) und Reaktionszeiten. Synchronisieren Sie Ihre Produktroadmaps mit den Release-Rhythmen des Projekts. Planbarkeit entsteht nicht durch „Wunschtermine", sondern durch regelmäßig abgestimmte Roadmaps.
5. **Upstream-Nähe statt Dauer-Fork:**
   Je weiter sich ein eigenes Projekt vom ursprünglichen Open-Source-Projekt entfernt, desto höher werden Aufwand und Kosten für Integration, Wartung und Sicherheit. Wer nah am Original bleibt, bleibt kompatibel, spart sich aufwendige Prüfungen und vermeidet veraltete Änderungen. **Praxistipp:** regelmäßig mit dem Originalprojekt synchronisieren, eigene Änderungen frühzeitig zurückgeben („upstreamen"), lokale Anpassungen möglichst klein und überschaubar halten und aktiv an Schnittstellen und Weiterentwicklung mitwirken.
6. **Sicherheitsmodell:**
   Durch die Nähe zur Community und Kenntnis der Prozesse kann auch eine Bewertung der Sicherheit vorgenommen werden und anhand eigener Anforderungen in den eigenen Entwicklungsprozess eingebracht werden.

### *Offenheit vs. Exklusivität*

Häufige Sorge: „Wenn wir etwas beitragen, sehen Wettbewerber unsere Ideen." Die Lösung ist eine klare Entscheidungsmaxime.

- **Gemeinsame Infrastruktur** (Stabilität, Portierungen, Sicherheits-Fixes, generische Schnittstellen) wird offengehalten und upstream gebracht. Das senkt Redundanzen, teilt Lasten und erhöht Qualität (non-differentiating Software).
- **Differenzierende Logik** (Domänenregeln, spezielle Workflows, Integrationen, UX) bleibt unternehmensintern. Wo Signaling-Risiken bestehen (z. B. Markteinführung), kann zeitlich verzögertes Upstreamen sinnvoll sein: erst produktiv nutzen, dann den generischen Anteil veröffentlichen. So verbinden Sie Wettbewerbsfähigkeit mit Wartbarkeit.

### *Maintainer – Treuhänder statt Gatekeeper*

In industriellen Kontexten sind Maintainer Stabilitätsanker: Sie verantworten Integrität, Qualität, Security-Prozesse und Planbarkeit der Releases. Zugleich senken sie Hürden zur Mitarbeit (Dokumentation, Review-Fenster, Kommunikation) und moderieren Interessen zwischen Wettbewerbern.

### *Praxistipps für Entscheider*

**Strategie:** Klären Sie das Zielbild: Effizienz, Innovationsgeschwindigkeit, Souveränität – oder alles drei. Prüfen Sie Projektgesundheit (Aktivität, Reaktionszeiten, Review-Dichte), Lizenz-Eignung und die Rolle des Maintainers. Planen Sie Beiträge von Beginn an ein – nicht erst, wenn es brennt.

**Beschaffung:** Sie „kaufen" keine Lizenz, sondern entwickeln Beziehung und Verlässlichkeit: Governance, Releasezyklen, Security-Kommunikation, Prozesse und Tools. Berücksichtigen sie Upstream-Policy, Reaktions-/Patch-Fenster, Metriken und Eskalationspfade.

**Einführung und Pflege:** Halten Sie Upstream-Nähe, vermeiden Sie Langzeit-Forks. Benennen Sie eine interne Steward-Rolle, die die Verbindung zum Projekt hält und Feature-Wünsche konsolidiert.

**Community-Eintritt:** Der Eintritt muss kein formaler Akt und an Geld gebunden sein: strukturierte Bugreports, reproduzierbare Tests, Doku-Beiträge, Teilnahme an Usergroup-Meetings etc. – das alles macht Sie zum Community-Mitglied. Verhalten Sie sich wie ein Partner: Kapazitäten respektieren, Zusagen einhalten, transparent kommunizieren.

**Wirkung messen:** Typische Management-Metriken: Time-to-Fix, Merge-Quote, Wiederverwendungsgrad, Anteil ersetzter Eigen-Patches, Security-Reaktionszeit, Recruiting-Effekte. Sichtbare Beiträge zahlen auf Marke und Arbeitgeberattraktivität ein.

### *Best-Practices für das Management*

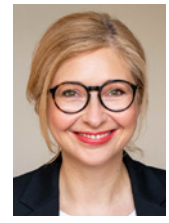
Basierend auf unserer Erfahrung als Maintainer in einem kleinen, industriell geprägten Ökosystem:

- Nähe zahlt sich aus:
  Unternehmen, die testen, melden und beitragen, verkürzen eigene Zyklen, reduzieren Wartungskosten und verbessern Auditierbarkeit.
- Gemeinsam stark:
  Co-Funding von generischen Funktionen verteilt Aufwand und reduziert spätere „Fork-Schulden".
- Exklusivität bleibt möglich:
  Differenzierung entsteht oberhalb des OSS-Kerns: in Integrationen, Domänenlogik, Services und Zertifizierungen.

### *Fazit*

Open Source entfaltet sein volles Potenzial, wenn Unternehmen vom Konsum- in den Gestaltungsmodus wechseln. Planen Sie Community-Beteiligung als festen Bestandteil Ihrer Technologie- und Beschaffungsstrategie ein – mit klarem Governance-Rahmen, OSPO-Denken und professioneller Zusammenarbeit mit den Menschen hinter dem OSS-Projekt. So wird Offenheit zur Souveränität: mehr Kontrolle, mehr Resilienz, mehr Innovationskraft.

**Die Autorin**



***Katrin Kahle** ist Head of Product bei Kernkonzept, Maintainer des sicheren Open-Source-Betriebssystems L4Re. Sie verbindet die Interessen von Kunden und des OSS-Projektes in L4Re-Produkten– zugelassen bis VS-GEHEIM und zertifiziert (CC EAL4+, ASIL B).*

# Welchen Wertbeitrag Open-Source-KI-Modelle liefern können

*Von Roger Meier und Oliver Fendt*

**Offene KI-Modelle verändern die Rahmenbedingungen: Unternehmen können leistungsfähige Sprachmodelle lokal betreiben, Kosten senken und sensible Daten sicher verarbeiten. Doch was macht KI-Anwendungen wirklich erfolgreich? Reichen offene Modelle allein oder braucht es neue Strategien, um Wirtschaftlichkeit, Datensouveränität und Innovationskraft gleichzeitig zu sichern? Siemens hat sich diesem Praxistest gestellt.**

Als Meta 2023 sein Sprachmodell LLaMA unter einer offenen Lizenz veröffentlichte, setzte das eine Welle an Innovationen in der KI-Community in Gang. Kurz darauf entwickelte der Informatiker Georgi Gerganov mit llama.cpp eine Inferenzmaschine, die selbst auf einem Standardarbeitsplatzrechner mit Grafikbeschleuniger beeindruckende Leistung zeigte.

Diese Entwicklungen markierten einen Wendepunkt – auch für uns. Nach erfolgreichen lokalen Tests und intensiven Diskussionen im Team entschieden wir, das Thema konsequent weiterzuverfolgen und Sprachmodelle über unsere Plattform code.siemens.com für andere Entwickler zugänglich zu machen.

### Offenheit braucht klare Regeln

Die Open-Source-Initiative (OSI) definiert Open-Source-AI als Systeme, deren Aufbau und Komponenten vollständig zugänglich und veränderbar sind – Grundlage für Kontrolle, Transparenz und technologische Souveränität[1]. Doch Offenheit allein genügt nicht: Verantwortungsvolle KI erfordert klare Governance, interdisziplinäre Zusammenarbeit und die Einhaltung des europäischen AI-Acts, um Sicherheit und Regelkonformität zu gewährleisten.

### Open Source und Standardisierung nehmen Fahrt auf

Nicht nur die Zahl offener Sprachmodelle wächst rasant – auch die dafür benötigte Software ist frei verfügbar und entwickelt sich dynamisch weiter. Ein Beispiel ist vLLM, eine an der UC Berkeley entwickelte Inferenzmaschine, die heute als zentraler Baustein zahlreicher KI-Plattformen gilt. Das Projekt wird aktiv von Universitäten sowie Unternehmen aus der KI- und Chipbranche unterstützt und steht stellvertretend für die zunehmende Professionalisierung des Open-Source-Ökosystems.

Dank OpenAI-kompatibler Schnittstellen lassen sich Sprachmodelle inzwischen nahezu nahtlos austauschen – intern wie extern. Ein vorgeschalteter Proxy sorgt dafür, dass Nutzer diesen Wechsel gar nicht bemerken. Das steigert die Flexibilität, senkt Integrationsaufwände und ermöglicht eine echte freie Anbieterwahl.

Auch auf Protokollebene schreitet die Standardisierung voran: Mit dem Model Context Protocol (MCP), das Anthropic im November 2024 veröffentlichte, können zusätzliche Werkzeuge und Datenquellen einfach in Anwendungen eingebunden werden. Der Ansatz findet zunehmend Unterstützung in der Branche und könnte sich langfristig zu einem verbindenden Standard entwickeln – ähnlich wie einst APIs im Cloud Computing.

1) https://opensource.org/ai/open-source-ai-definition

### Praxistipps für den erfolgreichen Einsatz von Open-Source-KI:

- **Datenhoheit sichern:** Modelle lokal betreiben, um sensible Daten zu schützen.
  *Benefit:* Maximale Kontrolle und Compliance.
- **Pilotprojekte starten:** Kleine Tests durchführen, bevor KI skaliert wird.
  *Benefit:* Risiken minimieren, Potenziale früh erkennen.
- **Flexibilität einplanen:** Standardisierte Schnittstellen und Inferenztools nutzen.
  *Benefit:* Anbieterwechsel und Integration leicht möglich.
- **Kontextqualität prüfen:** Sind die eingesetzten Daten qualitativ hochwertig, relevant und aktuell?
  *Benefit:* Verlässliche und robuste Ergebnisse.
- **Modellauswahl hinterfragen:** Kann das Modell den Kontext korrekt verarbeiten?
  *Benefit:* Höhere Präzision und Anwendbarkeit.
- **Performance bewerten:** Welche Antwortlatenz ist akzeptabel? Würde ein kleineres Modell effizienter arbeiten?
  *Benefit:* Effiziente Nutzung von Ressourcen und bessere User Experience.
- **Validierung sicherstellen:** Lässt sich die Modellqualität objektiv überprüfen, z. B. über Benchmarks, oder basiert die Einschätzung nur auf subjektivem Eindruck?
  *Benefit:* Vertrauen in Ergebnisse und langfristige Skalierbarkeit.
- **Ergebnisse messbar machen:** KPIs für Genauigkeit, Latenz und Validierbarkeit definieren.
  *Benefit:* Objektive Bewertung und Nachvollziehbarkeit.
- **Abteilungen vernetzen:** IT, Entwicklung, Compliance, Cybersecurity und Fachbereiche einbinden.
  *Benefit:* Ganzheitliche Governance und Risikominimierung.
- **Souveränität ausbauen:** Internes Finetuning durchführen, Abhängigkeit von Drittanbietern reduzieren.
  *Benefit:* Kontrolle über Technologie und strategische Freiheit
- **Innovation und Nachhaltigkeit verbinden:** Ressourcen effizient nutzen und ökologische Verantwortung berücksichtigen.
  *Benefit:* Langfristige Wettbewerbsfähigkeit und positives Unternehmensimage.

### Vorteile von Open-Source-KI

Open-Source-KI bietet Unternehmen weit mehr als Kostenvorteile: Sie schafft Unabhängigkeit von Herstellern, sichert Datenhoheit und ermöglicht volle Transparenz und Kontrolle über Modelle und Finetuning-Prozesse. Sensible Informationen können intern verarbeitet werden, während interne Anpassungen und Erweiterungen die Innovationskraft stärken.

Doch KI ist kein Allheilmittel. Sie kann Wissen zugänglich machen, Prozesse beschleunigen und Ideen fördern – ersetzt aber keine menschliche Expertise. Besonders in der Softwareentwicklung unterstützt sie bei Dokumentation, Tests und Code-Generierung, erfordert jedoch weiterhin fachkundige Validierung und Qualitätssicherung.

### Nachhaltige KI und Ausblick

Open-Source-KI ist nicht nur ein technologischer, sondern auch ein nachhaltiger Weg. Wir betreiben unsere Modelle in einem der modernsten Rechenzentren weltweit mit Solarenergie und Seewasserkühlung und erweitern die Infrastruktur kontinuierlich. So verbinden wir Innovation mit ökologischer Verantwortung.

Langfristig sind offene KI-Systeme ein unverzichtbares Element für Forschung, wirtschaftliche Souveränität und nachhaltige Wettbewerbsfähigkeit. Wie Open-Source-Software einst das Fundament des Internets legte, wird Open-Source-KI zur Basis der nächsten technologischen Entwicklungsstufe. Entscheidend ist, die Technologie bewusst und verantwortungsvoll einzusetzen, und zwar nicht, um Menschen zu ersetzen, sondern um neue Formen von Wertschöpfung, Transparenz und Innovation zu ermöglichen.

Details zu unserer Umgebung sind in unserem Blogartikel verfügbar: *https://blog.siemens.com/2025/10/our-sovereign-ai-journey-building-a-self-contained-sustainable-and-cost-effective-llm-platform/*

**Die Autoren**


© siemens.com

***Roger Meier** ist Distinguished Engineer und bei Siemens verantwortlich für die Entwicklerplattform code.siemens.com. Er fokussiert sich auf Entwicklerproduktivität und -zufriedenheit und bringt mehr als 21 Jahre Erfahrung in Forschung, Entwicklung und IT mit.*

***Oliver Fendt** ist Senior Manager Open Source bei Siemens und hat zahlreiche Projekte initiiert und unterstützt. Er ist Mitglied des Governing Boards von OpenChain sowie stellvertretender Vorsitzender des Open Source-Arbeitskreises im Bitkom.*

# Open Source bei M&A: Chancen erkennen, Risiken managen

*Von Thomas Urband*

**Open Source Software (OSS) ist längst kein Nischenphänomen mehr. In nahezu jedem Unternehmen spielt sie eine Rolle. Daher ist die Bewertung von Open-Source-Komponenten ein Pflichtbestandteil der Due Diligence bei M&A-Transaktionen. Unzureichende Compliance, kritische Lizenzbedingungen, unklare IP-Rechte oder fehlende Dokumentation können den Wert des Assets maßgeblich beeinflussen. Für Entscheider gilt: Wer die Chancen von Open Source kennt und Risiken früh identifiziert, schafft Mehrwert und vermeidet teure Überraschungen.**

### *Open Source als Deal-Faktor*

Aus Käuferperspektive stehen zwei Fragen im Fokus: Welche Rechte gehen am Technologie-Stack über und welche Verpflichtungen und Risiken erwerbe ich mit? Unklare Lizenzlagen, fehlende Dokumentation oder Copyleft-Verstöße können zu Nachlizenzierungen, Offenlegungspflichten oder Unterlassungsansprüchen führen und den Wert der Software-Assets sowie die Integrationsstrategie beeinträchtigen. Aus Verkäufersicht erhöht ein lückenlos dokumentierter und lizenzkonformer OSS-Einsatz die Glaubwürdigkeit der technischen Darstellung und reduziert das Risiko von Preisabschlägen.

### *Die typischen Risikofelder*

OSS ist in modernen Softwarearchitekturen unerlässlich, auch in Hardware eingebettete Komponenten sind betroffen. Die Nutzung bringt spezifische Risiken mit sich. Um diese zu erkennen und zu steuern, ist die Berücksichtigung typischer Risikofelder essenziell.

**1. Lizenzen & Copyleft-Exposition**

Nicht alle OSS-Lizenzen sind gleich: Permissive Lizenzen wie MIT, BSD oder Apache erzeugen meist überschaubare Pflichten, während Copyleft-Lizenzen wie GPL oder AGPL bei statischer oder bestimmter dynamischer Kopplung Offenlegungspflichten auslösen können. Der Copyleft-Effekt bedeutet, dass Veränderungen eines Werks unter derselben Lizenz veröffentlicht werden müssen. Im Worst Case müssten dadurch proprietäre, wertbildende Software-Assets offengelegt werden. Auch Beschränkungen der kommerziellen Nutzung sind zu beachten, insbesondere Creative-Commons-Lizenzen untersagen oder beschränken diese häufig.

**2. Dokumentation & SBOMs**

Oft ist die Dokumentation der eingesetzten Open-Source-Komponenten unvollständig. SBOMs (Software Bill of Materials) existieren nicht oder sind lückenhaft. Dadurch sind Lizenz- und Sicherheitsrisiken schwer einschätzbar und realisieren sich häufig in der Integrationsphase.

**3. IP-Clearance & Contributor Rights**

Zu prüfen ist die Herkunft des Codes: Liegen CLAs (Contributor License Agreements) oder anderweitige, rechtlich belastbare Rechteübertragungen vor und sind diese dokumentiert – insbesondere bei externen Partnern und Community-Contributions? Fehlt eine lückenlose Rechtekette, bestehen Unterlassungs- oder Nachlizenzierungsrisiken, die zentrale Module und Funktionen gefährden können?

### *OSS-Stolperfallen im Deal vermeiden*

- Strukturierte Vorbereitung erhöht die Transparenz und vermeidet Überraschungen
- Lizenzkonformität prüfen: Copyleft-Risiken und kommerzielle Einschränkungen erkennen
- IP-Rechte klären: Herkunft des Codes nachweisen und lückenlose Rechtekette durch CLAs sowie Arbeits-, Freelancer- und Software-Entwicklungsverträge absichern
- Vertragliche Schutzmechanismen gestalten: Garantien und Freistellungen vereinbaren
- Prüfung und Mitigierung der Risiken im Schulterschluss mit technischer und kommerzieller Due Diligence

**4. Governance & Prozessreife**

Gibt es ein OSPO (Open Source Program Office) oder klare Richtlinien, die Einsatz und Freigabe von OSS, Lizenzprüfungen, Schwachstellenmanagement, Dokumentation und Community-Engagement steuern? Ausgereifte Prozesse sind ein Indikator für niedrige Deal-Risiken und ermöglichen risikoorientierte Entscheidungen.

### *OSS-Due-Diligence: Key Factors*

Eine wirkungsvolle Prüfung ist zielgerichtet und risikoorientiert. Sie verbindet forensische Code-Analyse mit rechtlicher Bewertung und kommerzieller Einordnung. Der Scope orientiert sich an der Relevanz für das Zielproduktportfolio und die maßgeblichen Werttreiber.

**1. Datenvalidierung als Grundlage**

Am Anfang steht die vollständige Identifikation und technische Validierung der OSS-Komponenten: Toolgestützte Scans (inklusive Lizenzen, Snippets, Abhängigkeiten) werden mit Entwickler-Interviews, Repository-Reviews und SBOM-Checks kombiniert.

**2. Rechtliche Analyse**

Identifizierte Komponenten werden auf Lizenzen, Kompatibilität, Offenlegungspflichten und Einschränkungen geprüft. Von zentraler Bedeutung ist, wer an welchen Software-Assets welche Rechte hält. Proprietäre Software wird gemeinsam mit den OSS-Komponenten betrachtet. Ausgehend von Beschreibungen der wertbildenden Assets und Interviews wird geprüft, ob die Rechte exklusiv beim Unternehmen liegen und kommerziell verwertet werden können. Es wird geprüft, wer an der Entwicklung beteiligt war und wie Rechte vertraglich gesichert sind (etwa in Arbeits-, Freelancer- oder Software-Entwicklungsverträgen).

**3. Kommerzielle Einordnung**

Die kommerzielle Einordnung antizipiert Effekte auf Kaufpreis, Deal-Struktur, Integrationsaufwand und zukünftigen Revenue und quantifiziert sie idealerweise.

### *Verkäufer: Den Deal sichern durch Offenheit und Vorbereitung*

Wer sich früh mit Open Source auseinandersetzt, verbessert seine Verkaufschancen. Ein Gesundheitscheck der eingesetzten OSS und der Risikofelder – idealerweise 6 bis 9 Monate vor dem geplanten Verkauf – erhöht die Transparenz, reduziert Risiken und steigert die Deal-Readiness. Die Bereitstellung gut aufbereiteter SBOMs, Lizenzhinweise, Architekturdokumentationen sowie Informationen zu Prozessen signalisiert den Käufern Verlässlichkeit. Das führt zu transparenten Datenräumen, beschleunigten Due-Diligence-Prozessen, geringeren unerwarteten Risiken und einer stärkeren Verhandlungsposition.

### *Käufer: OSS-Due-Diligence systematisch angehen*

Käufer sollten Open Source strukturiert prüfen und kritische Themen mit klaren Eskalationskriterien priorisieren. Red Flags sind Copyleft-Exposition, Einschränkungen der kommerziellen Nutzbarkeit des Kern-IP und fehlende Rechteketten. Bei relevanten Findings sind belastbare Mitigationsmaßnahmen – vor Signing/Closing oder im Nachgang – sowie passende Garantien und gegebenenfalls Freistellungen erforderlich. Die Mitigation sollte eng mit der technischen Due Diligence erfolgen, um realisierbare Lösungen zu finden (z. B. Austausch kritischer Bibliotheken, Anpassung des Linking-Modells, Lizenzwechsel, kommerzielle Subscriptions). Die kommerzielle Bewertung der Findings ist entscheidend, um Auswirkungen auf den Asset-Wert und den Kaufpreis zu prüfen.

### *Fazit*

Der bewusste Umgang mit Open Source ist eine zentrale Erfolgsgröße für solide Bewertungen, sichere Transaktionen und reibungslose Integration. Open Source ausschließlich defensiv zu betrachten, verschenkt Potenzial. Eine professionelle OSS-Due-Diligence bietet Rechtssicherheit und schafft Mehrwert: Sie liefert belastbare Entscheidungs- und Handlungsoptionen. Sind Copyleft-Exposition und Rechte an den Software-Assets geklärt, SBOMs vollständig und Governance-Modelle vorhanden, steigt die Deal-Readiness. Werden Defizite identifiziert, können sie mitigiert und so ein nachhaltiger Wertbeitrag erzielt werden.

**Der Autor**

***Thomas Urband,*** *Senior Manager bei PwC Legal AG, ist Rechtsanwalt und Experte für OSS-Compliance, IP und Licensing.*