



Strategien und Werkzeuge
für den Transfer von
Forschungsergebnissen
im Open-Science-Kontext

www.OpenTransfer.eu

ENTSCHEIDUNGSHILFE OPEN BUSINESS MODELS

Dr. Ivonne Bieber, Dr. Tobias Engert, Dr. Kathrin Göbel, Peter Häfner,
Dr. Thomas M. Jahn, Dr. Vladimir Voroshnin, Dr. Björn Wolf



Gefördert durch:



Inhalt

Vorbemerkungen.....	3
Nutzenversprechen und vermarktbar Ressourcen.....	3
Akteure.....	6
Geschäftsmodelle.....	8
1 Dienstleistungen.....	9
Variante: Forschungs- und Entwicklungsprojekte / Auftragsforschung.....	11
2 Verkauf fertiger Software oder Hardware	13
3 Duale Lizenzierung.....	15
4 Gemeinwohlfinanzierung.....	18
5 Mitgliedschaft / Community.....	21
6 Open Core / Freemium.....	23
7 Verkauf von Zubehör oder Merchandise.....	26

Vorbemerkungen

Open-Source-Software (OSS) und -Hardware (OSH) werden häufig mit dem Gedanken an kostenfreie Nutzung verbunden – und das ist auch ein großer Teil ihres Erfolgs. Für Hobby-Entwickler:innen, die in ihrer Freizeit an Projekten arbeiten, ist das vollkommen passend. Doch viele der heute besonders innovativen und technisch anspruchsvollen Open-Source-Projekte entstehen in Forschungseinrichtungen oder Unternehmen. Dort fallen nicht nur Entwicklungs- und Testkosten an, sondern auch langfristige Aufwendungen für Wartung, Support und Weiterentwicklung.

Damit solche Projekte nachhaltig weiterbestehen können, benötigen sie stabile Einnahmequellen – ohne dabei den offenen Charakter und die Prinzipien von Open Source zu gefährden. Die nachfolgenden Geschäftsmodelle geben Ideen, wie Organisationen, insbesondere aus Forschung und Industrie, finanzielle Ressourcen generieren können, während sie die Grundideen von Open Source bewahren.

Nutzenversprechen und vermarktbare Ressourcen

Jedes Geschäftsmodell braucht etwas zu Vermarktendes, was die jeweiligen Nutzer als wertvoll, erstrebenswert und damit kaufwürdig erachten. Für hier zu betrachtende Open-Source-Projekte muss dies zum einen ein **Nutzenversprechen** beinhalten (auch Wertversprechen bzw. Value Proposition genannt): die OSS/OSH muss aus Perspektive des Nutzers bzw. des Käufers einen handfesten, nachhaltigen Wert besitzen, damit es lohnt, sie zu erlangen und zu nutzen. Zum anderen muss für ein funktionierendes Geschäftsmodell eine **vermarktbare Ressource** (ein Asset) vorhanden sein. Etwas, was sich abgrenzen, spezifizieren und verkaufen lässt. Das kann zum Beispiel ein physisches Produkt, ein spezifischer Service für einen definierten Zeitraum oder die Arbeitszeit einer Person sein.

Das Nutzenversprechen einer OSS/OSH liegt allem Weiteren zugrunde. Ohne Nutzen keine Nutzer, aber auch: ohne mehr bzw. mindestens gleich viel Nutzen wie konkurrierende (auch kommerzielle) Lösungen keine bzw. nur wenige Nutzer. Dabei muss zumindest der **Kernnutzen** realisiert werden: die OSS/OSH erfüllt auf technischer Ebene das, was sie soll – so fehlerfrei, einfach nutzbar und schnell wie möglich.

Um interessanter und wertvoller als kommerzielle Alternativen zu sein, benötigt es aber noch **Zusatznutzen**, z.B. einen oder mehrere der folgenden:

- **Geringere Kosten:** Open-Source-Software/-Hardware ist nicht automatisch „günstiger“, nur weil keine Lizenzkosten anfallen: Für Nutzer entstehen zusätzlich Kosten etwa für Installation bzw. Fertigung, Betrieb, Schulung oder Zertifizierung. Erst wenn diese **Gesamtkosten der Nutzung** („Total Cost of Ownership“) bei vergleichbarem Nutzerwert unter denen einer kommerziellen Alternative liegen, ist OSS/OSH tatsächlich kostengünstiger.
- **Kein Lock-in-Effekt:** mit jeder Software-/Hardware-Lösung legen Nutzer sich auf einen Lösungsweg fest. Eine Änderung ist mit, teils recht hohem, Aufwand für Suche nach Alternativen, Testen, Anpassen, Schulung usw. verbunden. Bei einer kommerziellen Lösung ist man nach der Festlegung auf eine Lösung oft von einem quasi monopolistischen Anbieter abhängig. Wenn dieser sich entscheidet, die Lösung nicht mehr oder nur noch sehr teuer anzubieten, bekommt man unverschuldet ein Problem. Open Source reduziert diesen Lock-in-Effekt, weil Know-how und Grundlagen offen sind und Nutzer bei Bedarf zu anderen Dienstleistern bzw. Anbietern wechseln und sie die Lösung auf derselben offenen Basis weiterführen können.
- **Modularität und Robustheit:** größere Open-Source-Projekte werden von Communities entwickelt. Um die Beiträge großer Communities sinnvoll integrieren zu können, sind OSS/OSH häufig modular aufgebaut. Dieser modulare Aufbau macht die Gesamtlösung in vielen Fällen robuster und einfacher zu warten. Damit bekommen Nutzer stabilere Lösungen und sind nicht auf Monopolisten für Service und Support angewiesen.
- **Transparenz und Nachvollziehbarkeit:** nicht nur bei sicherheitsrelevanten Anwendungen ist Transparenz ein Wert an sich: der Nutzer kann (entsprechendes technisches Verständnis vorausgesetzt) jederzeit nachvollziehen, was eine OSS/OSH genau tut. Damit kann die Eignung für bestimmte Anwendungen viel besser und genauer sichergestellt werden als bei einer kommerziellen „Black Box“. Open-Source-Lösungen können viel einfacher auf individuelle Anforderungen angepasst werden, weil alle zugehörigen Informationen verfügbar sind. Bei kommerziellen Alternativen ist das meist nicht der Fall. Oft verhindern Lizenzbedingungen oder sonstige Regelungen auch Anpassungen oder Veränderungen.
- **Community-Ressourcen:** die zu großen Open-Source-Projekten gehörenden Communities vereinen teils recht große, diverse und flexible Ressourcen für Entwicklung und Tests. Damit kann ein Open-Source-Projekt in vielen Fällen schneller auf notwendige Änderungen reagieren und in verschiedenen Umgebungen schneller getestet werden als bei einem kommerziellen Anbieter mit limitierten Entwicklerkapazitäten. Kleine Projekte haben diesen Vorteil leider nicht.

- **Immaterieller Nutzen:** neben dem direkten Nutzwert können OSS/OSH auch durch „weiche“ oder strategische Effekte ökonomisch relevant sein: etwa durch das gute Gefühl, am Richtigen mitzuwirken, durch die Förderung von Potenzialen in Communities (z. B. Wissenschaft) und durch größere strategische Unabhängigkeit. Zudem kann kritische Infrastruktur resilienter werden, wenn Entwicklung und Wissen breit verteilt sind.

Was sind nun aber die konkret vermarktbaren Ressourcen, die in einem Geschäftsmodell verkauft werden können?

- **Know-how und Informationen:** Unterstützen eigene Mitarbeitende bei Weiterentwicklung, Anpassung und Implementierung, mit oft über die Dokumentation hinausgehendem Erfahrungswissen.
- **Personal mit Know-how:** Entwickler:innen oder Support-mitarbeiter:innen, die das Projekt anpassen, erweitern oder betreuen.
- **Fertige Produkte:** Werden auf Open-Source-Basis gefertigt und können direkt eingesetzt werden.
- **Zugänge:** Zugriff auf Ressourcen, Entscheidungen und Netzwerke hilft, Einfluss auf zukünftige Entwicklungen zu nehmen.
- **Vertrauen:** durch z.B. Image, Marken, Zertifizierungen, getestetes Zubehör und Material etc. hilft, die Bereitstellung von Produkten zu erleichtern und fördert deren Verkauf an Kunden
- **Besondere Rechte:** Duale Lizenzen, Early-Access-Versionen oder Sondervereinbarungen, die über die Standard-Open-Source-Lizenz hinausgehen. Sie erlauben Dinge, die anderen nicht möglich bzw. erlaubt sind: z.B. Regeln allgemeiner Open-Source-Lizenzen (wie z.B. Weiterentwicklungen auch Open Source stellen zu müssen) nicht beachten zu müssen, etwas früher zu haben oder etwas Besseres zu bekommen als normale Nutzer mit Open-Lizenzen.

Akteure

In den folgenden Geschäftsmodellen stehen vereinfachend die Entwickler von OSS/OSH bzw. deren Organisation (Creator) auf der einen und die Nutzer aus Wirtschaft, Verwaltung und Forschung auf der anderen Seite. Das heißt aber nicht, dass diese beiden die einzigen Akteure in einem erfolgreichen Geschäftsmodell sein müssen. Vielmehr können in jedem der folgenden Geschäftsmodelle Teile von weiteren Akteuren übernommen werden. Manchmal sind weitere Akteure zwingend nötig, weil die Entwickler und deren Organisation bestimmte Aktivitäten nicht tun wollen, können oder dürfen.

Gründe für zusätzliche Akteur:innen können folgende sein:

- **Trennung vom Träger der OSS/OSH:** Die Entwickler:innen bzw. ihre Organisation können oder wollen bestimmte Geschäftsmodelle nicht umsetzen. Dies ist häufig der Fall, wenn z.B. Forschungseinrichtungen diese Geschäfte nicht als ihrem Kernauftrag zugehörig betrachten und die ggf. damit zusammenhängenden wirtschaftlichen Risiken nicht tragen wollen oder dürfen (z.B. aus rechtlichen Gründen).
- **Finanzierung:** Manche Organisationen können eine notwendige Finanzierung schwieriger oder gar nicht erlangen: Forschungseinrichtungen dürfen meist keine Kredite oder ähnliches aufnehmen, Investoren verlangen bestimmte Rechte im Gegenzug für eine Finanzierung, die ein etabliertes Unternehmen oder eine öffentliche Einrichtung nicht einräumen können oder dürfen.
- **Organisatorische Grenzen:** Große Organisation unterliegen häufig komplexen Regeln wie Tarifverträgen oder dem öffentlichen Haushaltsrecht, die bestimmte organisatorische Maßnahmen sehr aufwendig oder unmöglich machen.

In vielen dieser Fälle können andere Organisationen diese Aufgaben zumindest teilweise übernehmen. Szenarien wären die folgenden:

- **Gründung einer neuen Organisation (Spin-off/Spin-out):** Die wesentlichen Aufgaben des Geschäftsmodells werden in einer neuen Organisation angesiedelt. Das befreit von verschiedenen Zwängen gerade öffentlicher Einrichtungen. Hier muss allerdings auf die Eigentumsverhältnisse geachtet werden, weil öffentliche Einrichtungen bei signifikantem Miteigentum manche dieser Zwänge „vererben“. Besondere Aufmerksamkeit verdient hier die Frage wie sichergestellt wird, dass neue Organisation und Open-Source-

Entwickler auch langfristig ähnliche Strategien verfolgen und sich nicht auseinanderentwickeln.

- **Nutzung einer spezialisierten Tochterorganisation:** Größere Organisationen v.a. aus dem öffentlichen Bereich haben heute spezialisierte Tochterorganisationen, die wirtschaftliche Tätigkeiten wie z.B. Technologietransfer, Auftragsforschung und ähnliches abwickeln. Diese können meist recht einfach mitgenutzt werden. Zu klären ist vorab, wieviel der zu vermeidenden Probleme wirklich gelöst werden, weil manche dieser Tochterorganisationen aus unterschiedlichen Gründen ähnlichen Zwängen unterliegen wie die Mutterorganisationen.
- **Erledigung durch Partnerunternehmen:** Bestimmte Aufgaben können an ein oder auch mehrere unabhängige und eher lose, partnerschaftlich mit den Entwicklern verbundene Unternehmen ausgelagert werden. Hier ist die Trennung am einfachsten, allerdings auch das Risiko, dass Entwickler und Partnerunternehmen sich „auseinanderleben“. Deshalb kann es sinnvoll sein, von vornherein mit mehr als einem Partnerunternehmen zu planen.

Verknüpfung zwischen Entwickler:innen und zusätzlichen Akteur:innen

Wichtig ist bei diesen drei Alternativen zu klären, in welcher Form das Open-Source-Projekt von dem/den Geschäftsmodell(en) profitiert. Das kann auf verschiedenem Weg geschehen:

- Die Entwickler bzw. deren Organisation und der Betreiber des Geschäftsmodells schließen eine Art Lizenzvereinbarung, die dafür sorgt, dass ein Teil der Erlöse bzw. Gewinne aus dem Geschäftsmodell an die Entwickler weitergereicht wird (Erlösbeteiligung).
- Die Betreiber des Geschäftsmodells übernehmen Kosten des Open-Source-Projektes (komplett oder nur einen Teil) und finanzieren diese mit Erlösen aus dem Geschäftsmodell. Das können z.B. Entwicklungs- oder Supportpersonal sein, die beim Betreiber angestellt werden. Vorstellbar ist auch die Anschaffung von ggf. notwendigem Equipment, welches dann z.B. über Leih- oder Nutzungsverträge den Entwicklern zur Verfügung gestellt wird.
- Die Entwickler bzw. deren Organisation profitieren materiell nicht direkt davon, allerdings fördert das Geschäftsmodell die Nutzbarkeit, die Zuverlässigkeit oder die Verfügbarkeit von OSS/OSH und nutzt dem Open-Source-Projekt indirekt.

Geschäftsmodelle

Auf Basis von möglichen Nutzenversprechen, vermarktbaren Ressourcen und möglichen Konstellationen von Akteuren lassen sich unterschiedliche Geschäftsmodelle entwickeln, die unterschiedliche Kombinationen dieser Ressourcen nutzen, um Einnahmen zu generieren.

Im Folgenden werden sieben grundsätzliche Geschäftsmodelle vorgestellt. Diese sieben sind keine abschließende Auflistung, sondern sollen vielmehr Anregung sein, über Möglichkeiten zum Einnahmen erzielen nachzudenken. In anderen Quellen oder Leitfäden werden Geschäftsmodelle teils anders strukturiert oder sehr viel detaillierter untergliedert. Gerade für Open-Source-Software gibt es Darstellungen mit deutlich mehr Geschäftsmodellen. Das ist kein Widerspruch zu dieser Auflistung, lediglich eine andere Darstellungsform.

Die Geschäftsmodelle schließen sich meist nicht gegenseitig aus, es kann in vielen Fällen sinnvoll sein, mehrere parallel zu implementieren. Die einzelnen Geschäftsmodelle sind der Übersicht halber alle gleich strukturiert:



Basics

Grundlagen zum Geschäftsmodell

Relevant für Open-Source-Software (OSS) und Open-Source-Hardware (OSH).

Das Geschäftsmodell wird aus Sicht der Entwickler der OSS bzw. OSH geschildert, mit besonderem Fokus darauf, dass die Entwickler in oder im Umfeld einer Forschungseinrichtung angesiedelt sind. .



Nutzen

Ein Geschäftsmodell kann nur funktionieren, wenn etwas mit klarem Nutzen für diejenigen angeboten wird, die dafür zahlen.



Eignung

Nicht jedes Geschäftsmodell ist für alles gleich gut geeignet. Das kann für Software anders sein als für Hardware. Es kann auch Unterschiede geben, wenn die Forschungseinrichtung selbst, eine Tochter- bzw. eine der Forschungseinrichtung nahestehende Gesellschaft oder ein wirtschaftlich unabhängiges Unternehmen (Startup) diese anbietet.



Beispiele

Hier werden einzelne Beispiele erklärt, bevorzugt bekannte Beispiele.



zu beachten

Einzelne der Geschäftsmodelle können verschiedene Probleme oder Besonderheiten aufweisen, z.B. in der Produkthaftung.

1 Dienstleistungen



Basics

Der Verkauf von Dienstleistungen ist das sicher mit Abstand meistgenutzte Geschäftsmodell im Open-Source-Umfeld. Das Kernprinzip ist einfach: Die Software oder Hardware bleibt frei und offen, die Wertschöpfung entsteht durch die Mühe der Anpassung. Hier wird konkret die Ressource »Know-how und Informationen« sowie »Personal mit Know-how« monetarisiert. Es funktioniert für sehr viele, auch kleinere OSS- oder OSH-Projekte.



Nutzen

Die Nutzer:innen von OSS / OSH benötigen einiges an Wissen, um zu einem fertig einsetzbaren Produkt zu gelangen. Unterschiedlichste Services helfen dem Nutzer, ein fertig einsetzbares Produkt zu erhalten und für seine Zwecke erfolgreich und effizient einzusetzen.

Implementierung: Kompilierung, Installation, Integration in die Zielumgebung bzw. Vorbereitung der Fertigung und Beschaffung von Bauteilen.

Anwendersupport: Entwicklung von Erweiterungen oder Schnittstellen, Schulungen, Anpassungen an spezifische Anwendungsbedingungen, Interpretation von Ergebnissen.

Qualitätssicherung: Prüfung auf Norm-Konformität, Einhaltung von Leistungsparametern und Zertifizierungen.



Eignung

Dienstleistungen können von allen möglichen Akteuren angeboten werden – der Forschungseinrichtung selbst, einem Unternehmen im Umfeld der Forschungseinrichtung wie auch einem unabhängigen Unternehmen / Startup.

Zu berücksichtigen ist allerdings, dass Dienstleistungen schlechter skalieren als ein physisches Produktgeschäft: für ein Mehr an Umsatz muss meist ein ähnliches Mehr an personellem Aufwand betrieben werden. Der zu verkaufende Wert ist das Know-how und die Expertise der beteiligten Personen, welches viel schwieriger zu vervielfältigen ist als ein physisches Produkt in Serienfertigung.



Beispiele

TYP03: Ein weit verbreitetes Content Management System (CMS) für Webseiten. Die Software ist kostenlos, Partnerunternehmen bieten Installation, Programmierung und Wartung als Dienstleistung an.

RedHat / SUSE: Anbieter von Linux-Distributionen für Server und Unternehmen. Die Software ist kostenlos, Support-Verträge, Schulungen und garantierte Updates kosten Geld.

White Rabbit (CERN): Ein Open-Source-Hardwaresystem für präzise Netzwerk-kommunikation. Das CERN bietet zusätzlich Schulungen, Tests und Implementierungs-beratung für Industriekunden an.

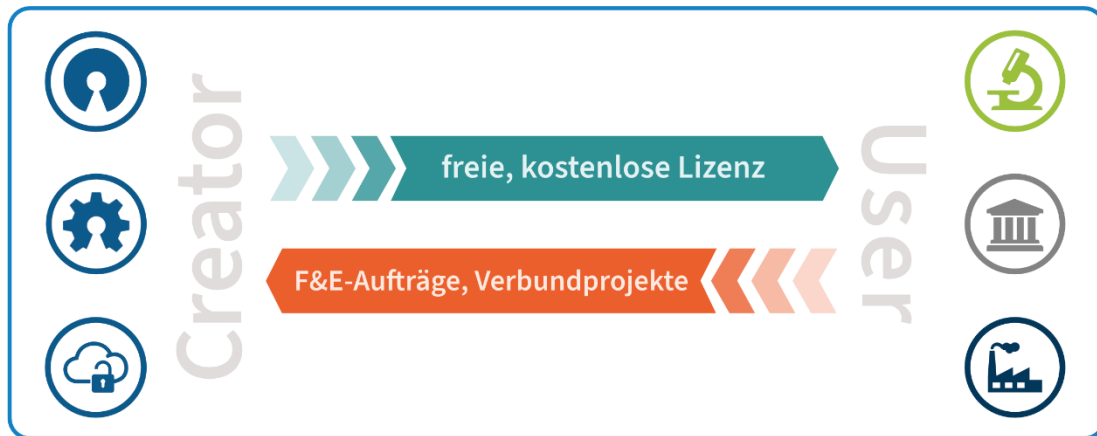


zu beachten

Dienstleistungen lassen sich gut an die Bedürfnisse und Kapazitäten der jeweiligen Organisation anpassen.

Durch die ab ca. 2027 geltende, neu geregelte Produkthaftung können solche Dienstleistungen aber dazu führen, dass die OSS/OSH nicht mehr als „außerhalb einer Geschäftstätigkeit erstellt“ gilt und damit eher als Produkt eingestuft wird, wodurch das Haftungsrisiko für Entwickler steigt.

Variante: Forschungs- und Entwicklungsprojekte / Auftragsforschung



Dieses Geschäftsmodell ist eine Variante des vorherigen (Dienstleistungen). Da F&E-Projekte und Auftragsforschung für Forschungseinrichtungen besondere Bedeutung haben und besondere Anforderungen aufweisen, sei es hier als Variante separat behandelt.



Basics

Dieses Geschäftsmodell verbindet die Bereitstellung von OSS / OSH mit dem eher klassischen Geschäft einer Forschungseinrichtung: der Auftragsforschung. Dabei forscht und entwickelt die Forschungseinrichtung im Auftrag, in Abstimmung oder gemeinsam mit Unternehmen an Themen im Zusammenhang mit OSS / OSH. Das können Weiterentwicklungen der zugrundeliegenden Technologie oder Anpassungen an bestimmte Anforderungen der Nutzer sein. Dafür wird die Forschungseinrichtung direkt vom Auftraggeber beauftragt oder wird im Rahmen eines öffentlich geförderten Verbundprojektes von einem öffentlichen Geldgeber bezahlt.

Dieses Geschäftsmodell ist eine Variante des Dienstleistungsgeschäftsmodells. Dabei hängt es von der generellen Lizenz der jeweiligen OSS / OSH ab, ob die Ergebnisse sofort, mit Zeitverzug bzw. anderen Einschränkungen oder gar nicht unter eine Open-Source-Lizenz gestellt werden.



Nutzen

Das Grundprinzip ist, dass ein Unternehmen notwendige Weiter- oder Anpassungsentwicklungen einer OSS / OSH nicht selbst vornimmt, sondern

damit diejenigen betraut, die selbige entwickelt und deshalb meist eine viel tiefere Erfahrung damit haben.

Eignung

Dieses Geschäftsmodell bietet sich vor allem für Forschungseinrichtungen an, weil F&E-Projekte und Forschungsaufträge tägliches Geschäft für diese sind und aus ihrer Perspektive weniger Nachteile aufweisen als andere Dienstleistungen oder Fertigungsaufträge. Teilweise werden F&E-Projekte und Forschungsaufträge aber auch über den Forschungseinrichtungen gehörende oder nahestehende Unternehmen abgewickelt (siehe oben Abschnitt Akteure).

Beispiele

White Rabbit ist eine OSH des CERN. Für Hersteller von White Rabbit Produkten übernimmt das CERN Weiter- oder Anpassungsentwicklungen durch die Gruppe der White Rabbit Entwickler. Den Kernbereich der Technologie betreffende Entwicklungen werden dabei sofort unter die gleiche Open-Lizenz gestellt wie White Rabbit (CERN-OHL-W). Entwicklungen für die Geräteperipherie können proprietär lizenziert werden.

Das **Open Compute Project** (OCP) wurde von Facebook initiiert und entwickelt Server-, Storage- und Netzwerk-Hardware als Open Source. Unternehmen wie Intel, Microsoft und Cisco beteiligen sich aktiv und bringen ihre F&E-Ergebnisse ein.

zu beachten

Abhängig von der gewählten Open-Lizenz müssen hier ggf. Standardverträge der Einrichtung bezüglich der Rechte an Entwicklungen (IP) angepasst werden.

2 Verkauf fertiger Software oder Hardware



Basics

Dieses Modell fokussiert auf den Verkauf des fertigen Ergebnisses. Bei Open-Source-Hardware (OSH) bedeutet dies die physische Fertigung, Verpackung und den Vertrieb der Hardware. Bei Open-Source-Software (OSS) umfasst es oft das Hosting der Software als Service (SaaS) oder die Bereitstellung eines vollständig getesteten und installierbaren Softwarepakets. Hier wird primär die Ressource »Fertige Produkte« monetarisiert.



Nutzen

Der wesentliche Nutzen ist, eine OSS/OSH ohne zusätzlichen Aufwand nutzen zu können, wie es auch bei kommerziellen Produkten die Regel ist. Bei OSS kann das neben der fertigen Software auch regelmäßige Updates enthalten. Für OSH entfällt der Bedarf für Materialbeschaffung, Anlagen und der für die Fertigung notwendigen Kenntnisse.



Eignung

Der Aufwand unterscheidet sich stark zwischen OSS und OSH: Software beansprucht relativ wenig Ressourcen. Die Herstellung fertiger, zuverlässiger OSH-Produkte in größerer Stückzahl ist viel aufwendiger. Das ist nicht die Kernkompetenz von Forschungseinrichtungen, auch die ggf. dafür notwendigen Investitionen für Anlagen können sie schlecht rechtfertigen. In bestimmten Fällen, wenn z.B. für die Fertigung der Zugriff auf spezielle, teure Maschinen notwendig ist, können aber forschungsnahe Unternehmen eingebunden werden.

Für private Unternehmen ist dieses Geschäftsmodell attraktiv, weil die Produktion fertiger Produkte gut skaliert: mit relativ wenig zusätzlichem Aufwand kann ein signifikant höherer Umsatz erzielt werden.



Beispiele

Arduino: Ein Mikrocontroller-Board für Elektronik-Projekte. Physische Boards werden verkauft, die Markenlizenzierung finanziert die Stiftung.

Opentrons: Ein Hersteller von Lab-Robotern für Pipettier-Aufgaben. Hardware und Teile der Software sind Open Source, spätere Generationen nutzen teilweise proprietäre Komponenten.

Nextcloud: Die Cloud-Speicher- und Kollaborationslösung kann selbst gehostet (siehe auch »Open Core«) oder aber auch als SaaS (Cloud-Dienst) von Anbietern oder dem Hersteller gemietet werden.

LABmaker: Berliner Startup, das ausgewählte OSH-Lösungen v.a. für den Laboreinsatz auf Nachfrage fertigt. Dabei wird nicht massengefertigt, sondern abhängig vom Auftrag mit Einzelfertigungstechnologien wie 3D-Druck.

Safran und weitere Anbieter fertigen Geräte auf Grundlage der White Rabbit OSH des CERN. Durch eigene Entwicklungsarbeit oder gemeinsam mit dem CERN werden diese Geräte auf spezielle Einsatzbedingungen angepasst. Dieses Geschäft ist so attraktiv, dass der erste White-Rabbit-Anbieter Seven, ein spanisches Startup, mittlerweile durch den Technologiekonzern Safran aufgekauft wurde.



zu beachten

Bei Bereitstellung fertiger Software- oder Hardwareprodukte erfüllt man die Produkteigenschaft der neuen EU-Produkthaftungsrichtlinie. Durch den Verkauf dieser Produkte ist auch das Kriterium „im Rahmen einer gewerblichen Tätigkeit“ erfüllt und der Anbieter haftet voll im Rahmen der Produkthaftung.

3 Duale Lizenzierung



Basics

Im Kern der dualen Lizenzierung steht, dass OSS / OSH nicht allen Nutzergruppen oder für alle Nutzungsarten gleichermaßen zur Verfügung steht. Bestimmte Nutzergruppen (z.B. Privatpersonen, gemeinnützige Akteure, Bildungs- oder Forschungseinrichtungen) erhalten die OSS / OSH unter einer Open-Source-Lizenz. Andere Akteure, z.B. gewerbliche Unternehmen, müssen kommerzielle Lizenzen erwerben.

Dies kann verstärkt werden durch die Nutzung einer sog. Copyleft Lizenz, die dazu zwingt, abgeleitete Werke unter die gleiche Lizenz zu stellen (und damit kommerzielle Nutzungen unattraktiv macht) – parallel werden kommerzielle Lizenzen angeboten.

Eine Variante der dualen Lizenzierung ist das sog. Time-Release Open Source. Dabei werden Technologien erst nach einer gewissen Zeit oder bei Vorliegen neuerer Versionen als Open Source verfügbar gemacht.



Nutzen

Das Modell ermöglicht unterschiedliche Bedingungen für unterschiedliche Nutzergruppen. Damit kann eine große Verbreitung durch klassische Open-Source-Nutzung für eine Vielzahl von Nutzern mit kommerziellen Lizenzerlösen von ausgewählten Nutzergruppen verbunden werden.

Freies Wachstum: Für die breite Öffentlichkeit bleibt das Projekt kostenlos und attraktiv, was die Weiterentwicklung, die Bildung einer Community und das Feedback stärkt.

Kommerzielle Verbindlichkeit: Unternehmen erhalten sofort Zugriff auf die aktuellsten Versionen und können sie kommerziell einsetzen. Sie erhalten vertraglich geregelte Leistungen und ggf. auch Haftungsoptionen, die bei reinen Open-Source-Lizenzen fehlen.

Bei Time-Release OS bekommen zahlende Nutzer aktuellere Versionen mit zusätzlichen oder verbesserten Funktionalitäten oder höherer Stabilität, Komfort oder Zuverlässigkeit.



Eignung

Duale Lizenzen sind bei Forschungseinrichtungen wie auch kommerziellen Anbietern möglich und verbreitet. Allerdings erwarten Erwerber kommerzieller Lizenzen dann auch kommerzielle Bedingungen hinsichtlich Qualität, Service und Haftung, was für öffentliche Einrichtungen oft schwer zu erfüllen ist.

Time-Release Open Source hingegen wird nahezu ausschließlich von kommerziellen Unternehmen genutzt: Durch die zeitlich beschränkte Exklusivität sichern sie sich Marktvorteile und Erlöse, später veröffentlichen sie die Technologien als Open Source.



Beispiele

MySQL: Ein weit verbreitetes Datenbanksystem. Es ist sowohl unter einer freien GPL-Lizenz als auch unter einer kommerziellen Lizenz verfügbar, um proprietäre Nutzung zu ermöglichen. Mittlerweile hat dies dazu geführt, dass zwei unterschiedliche Softwarepakete entstanden sind, bei denen die kommerzielle Variante als qualitativ hochwertiger gilt.

Pine64: Ein Hardware-Hersteller für Smartphones und Einplatinencomputer. Neue Designs sind zunächst exklusiv für zahlende Kunden (Time-Release), bevor sie als Open Source veröffentlicht werden.

Open Compute Project (Meta): Ein Projekt für Data-Center-Infrastruktur. Designs werden erst nach einer gewissen Zeit der internen Nutzung veröffentlicht, um Marktvorteile zu sichern.



zu beachten

Duale Lizenzierung ist anfällig für unlauteres Verhalten. Nutzer, die eine kommerzielle Lizenz erwerben müssten, könnten sich als Mitglieder der

anderen Nutzergruppe ausgeben. Daher ist zu überlegen, ob und welche Kontrollmechanismen es geben kann.

Time Release Open Source wird von vielen in der Open-Source-Community eher als Marketinggag angesehen, weil aktuelle Versionen kommerziell sind und nur „alte“ Technologie geöffnet wird.

4 Gemeinwohlfinanzierung



Basics

Die ersten Entwicklungs- und Verbreitungsschritte von OSS/OSH entstehen häufig aus einem Unternehmensvorhaben, einem Forschungsprojekt oder privater Neugier. Mit wachsender Komplexität sowie steigenden Nutzer- und Entwicklerzahlen nehmen Koordinations-, Kommunikations- und Supportaufwände zu. Oft erreicht das Projekt dann einen Punkt, an dem dieser Aufwand in der bisherigen Form nicht mehr vollständig tragbar ist.

Hier können der Staat, Stiftungen oder auch spendende Unternehmen sowie Privatpersonen einspringen und durch einmalige bzw. regelmäßige Zahlungen ohne eine direkte Gegenleistung diesen Aufwand decken helfen. Typische Formen dafür sind:

- Spenden
- Sponsoring
- Crowdfunding
- öffentliche Finanzierung (Grund- oder Projektfinanzierung)



Nutzen

Für die Nutzer entsteht kein unmittelbarer Mehrwert; sie profitieren „nur“ indirekt von einer kontinuierlich weiterentwickelten OSS/OSH-Lösung. Die Geldgeber erhalten hingegen:

Politische Vorteile: Vermeidung von Marktmonopolen, Erhalt von Technologie Kompetenzen, Stärkung von F&E Kapazitäten

Altruistische bzw. PR-Effekte: Imageaufwertung, Sichtbarkeit

Stabile Technologiebasis: Bei gemeinsamer Unternehmensfinanzierung entsteht eine gemeinsam genutzte, unabhängige Infrastruktur.

Früher Zugang: Im Crowdfunding-Fall können Unterstützer einen frühen, bevorzugten Zugang zu einem noch knappen Produkt erhalten.

Eignung

Das Modell eignet sich je nach Organisationsform in unterschiedlichem Maße: Forschungseinrichtungen erhalten gewöhnlich leichter öffentliche Mittel als Unternehmen (u. a. wegen Wettbewerbs- und Beihilferecht), und gemeinnützige/öffentliche Träger erhalten eher Spenden sowie teils steuerliche Vorteile.

Der Aufwand für die Akquise ist jedoch hoch, weil viele Akteure um Aufmerksamkeit und Förder- bzw. Spendengeld konkurrieren. Für kleine und mittlere OSS/OSH-Projekte bleibt der finanzielle Erfolg daher oft hinter den Erwartungen zurück.

Beispiele

Wikipedia: Die weltweit bekannte Online-Enzyklopädie. Der Betrieb wird durch Spenden finanziert, um die Unabhängigkeit von werbenden Unternehmen zu sichern.

Mozilla (Firefox): Ein Open-Source-Webbrowser. Wird von einer gemeinnützigen Stiftung getragen, die sich durch Partnerschaften mit Technologiekonzernen finanziert.

LibreOffice: Eine Office-Suite als Alternative zu kommerziellen Produkten. Wird von einer Stiftung unterstützt, die sich fast ausschließlich durch Spenden und Fördergelder finanziert.

Signal: Open-Source-Messenger mit Ende-zu-Ende-Verschlüsselung. Die Entwicklung wird von der gemeinnützigen Signal Foundation getragen, die ihr Budget über wiederkehrende Spenden- und Crow-Funding-Kampagnen aufbringt. So bleibt der Dienst werbefrei und unabhängig von kommerziellen Interessen.

Pine64: Hersteller von Einplatinen-Computern, Smartphones und Laptops. Das Unternehmen finanziert die ersten Produktionsrunden über Crow-Funding-Plattformen (Kickstarter, Indiegogo) und bietet

Unterstützern frühzeitigen Zugang zu den Geräten sowie exklusive Editionen. Die Firma ist als kommerzielles Unternehmen organisiert.

Purism: Produzent der „Librem“-Reihe (Smartphones, Tablets, Laptops) mit starkem Fokus auf Datenschutz. Die Entwicklung und Fertigung wird durch mehrere Crow-Funding-Kampagnen sowie private Investoren finanziert. Unterstützer erhalten Vorab-Lieferungen und besondere „Founder“-Pakete; auch hier handelt es sich um ein kommerzielles Unternehmen.



zu beachten

Mit diesem Einnahmemodell fällt man unter die (neue) EU-Produkthaftungsrichtlinie: auch eine Stiftung, die Geld in Form von Spenden annimmt, handelt gewerblich.

5 Mitgliedschaft / Community



Basics

Ähnlich wie die Gemeinwohlfinanzierung basiert die Mitgliedschaft auf einem indirekten Nutzenversprechen. Ein Mitglied unterstützt allgemein die weitere Entwicklung und Unterhaltung durch Geldzahlungen. Im Gegenzug können ihm bestimmte Rechte eingeräumt werden. Die Mitgliedschaft kann formalrechtlich in einem Verein, aber auch ohne eine spezielle Rechtsperson lediglich durch vertragliche Vereinbarung erfolgen.



Nutzen

Der Nutzen kann sowohl mittelbar als auch unmittelbar sein:

Direkter Zugang: Mitglieder erhalten eine bestimmte Anzahl an Dienstleistungen (Support, Entwicklung) und/oder Zugriff auf bessere Varianten der OSS/OSH.

Rabatte: Unternehmen bieten teilweise Rabatte auf Ersatzteile, Zubehör oder Schulungen an.

Mitsprache: Ein mittelbarer Nutzen ist die Mitbestimmung darüber, wie und in welche Richtung die weitere Entwicklung der OSS/OSH erfolgt.

Community-Status: Mitglieder profitieren von einem Netzwerk gleichgesinnter Organisationen und einem verbesserten Image.



Eignung

Das Mitgliedschaftsmodell ist eher für Forschungseinrichtungen oder ihnen nahestehende, eher gemeinnützig orientierte Organisationen geeignet. Die zu erzielenden Erlöse sind in den meisten Fällen überschaubar.

und dienen eher der Deckung der notwendigen Kosten für Services oder das Communitymanagement.



Beispiele

White Rabbit Collaboration (CERN): Ein Netzwerk für Hardware-Entwickler. Mitglieder erhalten bevorzugten Support und führen gemeinsame F&E-Projekte durch. Mittelbar ist auch eine Mitsprache über die weitere Entwicklungsrichtung enthalten, allerdings nicht als formales Recht.

RISC-V: Eine offene Prozessorarchitektur. Die Entwickler-Community nutzt Mitgliedschaften, um Einfluss auf die Entwicklung zu nehmen und F&E-Projekte zu finanzieren.

Open Targets: Eine pharmazeutische Forschungsplattform. Pharmaunternehmen zahlen Mitgliedsbeiträge, um ein Forschungsprogramm zu finanzieren und zeitlichen Vorsprung bei Ergebnissen zu erhalten.



zu beachten

Das Image eines Open-Source-Projektes kann beschädigt werden, wenn der Eindruck entsteht, zahlende Mitglieder würden die Entwicklung in eine bestimmte Richtung lenken. Eine Mitgliedschaft ist zudem nicht ausreichend, um größere Erlöse zu erzielen.

6 Open Core / Freemium



Basics

Um einen Open-Source-Kern werden proprietäre Peripherieelemente angeboten, teils ist der Kern ohne Peripherie gar nicht nutzbar. Besonders im Softwarebereich sind Basisfunktionen kostenlos, zusätzliche Komponenten kosten Geld (Freemium). Eine weiteres Koppelgeschäft ist, wenn der (möglichst bekannte) Name eines OSS/OSH-Projektes markenrechtlich geschützt und nur gegen Bezahlung lizenziert wird.

Hier wird primär die Ressource »Exklusive Rechte« (Zusatzfunktionen) sowie »Vertrauen« (Marke) monetarisiert.



Nutzen

Der Nutzen für die Anwender sind technische oder organisatorische Zusatzfunktionalitäten, die den Einsatz einer OSS / OSH komfortabler oder effizienter gestalten. Teilweise werden diese Peripherieelemente auch zwangsweise dazu verkauft im Sinne eines Lock-in-Effektes: der Kern ist ohne Peripherie für bestimmte Anwendungen sonst gar nicht nutzbar. Nutzer umgehen das teilweise, indem sie notwendige Peripherie selbst entwickeln.

Zusatzfunktionalitäten: Premium-Features machen den Einsatz von OSS/OSH komfortabler oder effizienter (z. B. Enterprise-Features, Admin-Tools).

Markenwert: Nutzer profitieren vom guten Ruf einer etablierten OSS/OSH-Marke und signalisieren damit Qualität und Seriosität. Die

lizenzierte Marke schafft Vertrauen bei Endkunden, da eine eindeutige Herkunft und Unterstützung gewährleistet ist.

Eignung

Das Open-Core-Modell ist vorwiegend für kommerzielle Akteure geeignet, da es eine klare Trennung zwischen Community- und Enterprise-Version erfordert. Die Markenlizenzierung hingegen kann auch durch Forschungseinrichtungen effektiv genutzt werden, um Missbrauch des eigenen Namens zu verhindern und gleichzeitig Einnahmen zu generieren.

Beispiele

Android (Google): Ein mobiles Betriebssystem. Der Kern (AOSP) ist Open Source, aber viele Zusatzfeatures und die Markenlizenzierung für Hersteller sind kostenpflichtig. Außerdem schließt Google Koppelgeschäfte ab, ohne die Hersteller Android nicht mit der von den Endnutzern gewünschten Funktionalität anbieten können.

Nextcloud: Eine Cloud-Speicher- und Kollaborationslösung. Die Basisversion ist kostenlos, spezifische Enterprise-Module (z. B. für Dokumentenbearbeitung) erfordern eine Lizenz.

WordPress: Die Basis-Software ist kostenlos und Open Source. Der Markenname ist geschützt, und Hosting/Enterprise-Services kosten Geld.

Arduino: Bietet Lösungen mit mehr Komfort wie z. B. Gerätediagramme nur gegen Geld an. Über die Lizenzierung der Marke Arduino und weiterer Marken fließt Geld zurück an die die Weiterentwicklung unterstützende Arduino-Stiftung.

Firefox / Thunderbird, RedHat / Fedora, WordPress: Die OSS sind grundsätzlich kostenlos verfügbar. Sobald aber der jeweilige Markenname in der Vermarktung genutzt werden soll, ist dafür in unterschiedlicher Form Geld zu bezahlen.

zu beachten

Kostenlose und kostenpflichtige Elemente müssen sorgfältig ausbalanciert werden, um weiterhin den Charakter und Nutzen einer Open-Source-Lösung bieten zu können – nicht nur den Nutzern, sondern ggf. auch einer Entwicklungscommunity. Die kommerziellen Elemente

wiederum müssen so attraktiv sein, dass sie ausreichend Einnahmen generieren können.

Open Core wird von vielen in der Open-Source-Community als kritisch angesehen, weil von manchen Anbietern kommerzielle Elemente sehr dominant gestaltet werden und der Open-Source-Charakter unterzugehen droht.

Die Markenlizenzierung durch Forschungseinrichtungen erfordert ein professionelles Lizenzmanagement mit enger Überwachung der Nutzung der jeweiligen Marken.

7 Verkauf von Zubehör oder Merchandise



Basics

Parallel zu einer OSS bzw. OSH werden begleitend Produkte verkauft wie Merchandise-Artikel oder Zubehör aller Art. Dies reicht von physischen Bauteilen (wie zur OSS passende Elektronikkomponenten oder Sensoren, Komponenten für den Bau oder Verbrauchsmaterial für den Betrieb einer OSH) bis hin zu markenbezogenen Artikeln.

Hier werden primär die Ressource »Vertrauen« (geprüfte Qualität) sowie »Zugang« (Community-Zugehörigkeit) monetarisiert.

Nutzen

Als Nutzen für die Käufer kommen drei Kategorien infrage:

Komfort: Für die OSS / OSH notwendiges Zubehör wie Bauteile, Zusatzkomponenten oder zugehörige Verbrauchsmaterialien werden aus einer Hand (dem OSS/OSH-Anbieter) gekauft, was den Bezug dieser sowieso notwendigen Dinge vereinfacht (Bezug aus einer Quelle, idealerweise geprüfte Qualität oder Kompatibilität).

Unterstützung: Durch den Kauf dieses Zubehörs oder Merchandise-Artikel über den OSS / OSH Anbieter wird dieser mittelbar finanziell unterstützt – anders als beim Kauf des Zubehörs über unbeteiligte Dritte.

Signaling: Mit Merchandise kann der Käufer seine Zugehörigkeit zu einer Community oder die finanzielle Unterstützung dieser in seinem Umfeld demonstrieren.



Eignung

Dieses Geschäftsmodell kommt eher für kommerzielle Anbieter in Frage, da mit dem Handel von Zubehör die normalen kommerziellen Risiken verbunden sind (Vorfinanzierung von Produktion, Haftungsrisiken, einbrechender Abverkauf etc.). Lediglich Merchandise ist auch im Umfeld von Forschungseinrichtungen gebräuchlich.

Während der Verkauf von Zubehör auch bei kleineren Projekten funktioniert, ist das Merchandising nur bei wirklich großen und bekannten Projekten wirtschaftlich lohnend.



Beispiele

Raspberry Pi: Ein Einplatinencomputer für Bildung und Hobby-Projekte. Der Hersteller verkauft offizielle Boards sowie Zubehör wie Gehäuse, Netzteile und Speichermedien.

Prusa Research: Die Hardware des Herstellers von 3D-Druckern war bis 2023 Open Source. Prusa verkauft neben den Druckern auch Ersatzteile, Filamente und Verbrauchsmaterial.

Linux-Ökosystem: Nutzt das Maskottchen (Tux) für Merchandise (T-Shirts, Sticker). Dient primär der Markenbindung und Community-Bindung, wobei die Nutzung des Symbols durch Lizenzvereinbarungen geregelt ist, um die Markenintegrität zu wahren.



zu beachten

Der Handel mit Zubehör und Merchandise muss wie ein gewerbliches Geschäft angelegt werden, inkl. einem geeigneten rechtlichen Rahmen (Struktur, aber auch AGB und sonstige Verträge), einer Finanzierung (Vorfinanzierung von Handelsware), einer geeigneten Vertriebsstruktur und passender Logistik (Lagerung, Versand, etc.).

Impressum

Datum der Veröffentlichung: September 2026

Version 1.0

Herausgeber

Technologietransfer, GSI Helmholtzzentrum für Schwerionenforschung GmbH

www.gsi.de/innovation

OpenTransfer

Im Rahmen von OpenTransfer wurden Strategien und Geschäftsmodelle für den Transfer im Kontext von Open Science entwickelt, Werkzeuge für die konkrete Unterstützung von Transferprozessen geschaffen sowie Organisationsstrukturen für OpenTransfer in den Forschungseinrichtungen eingeführt.

Alle Informationen, Guidelines und Werkzeuge finden Sie im Wiki:

www.OpenTransfer.eu

Das OpenTransfer-Team

Dr. Kathrin Göbel, Albrecht Haag, Dr. Tobias Engert

GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt

Dr. Thomas Jahn, Dr. Ivonne Bieber

IPHT Leibniz-Institut für Photonische Technologien e.V., Jena

Dr. Vladimir Voroshnin, Dr. Björn Wolf

HZDR Helmholtz-Zentrum Dresden-Rossendorf e.V., Dresden

Peter Häfner

INNOcentric GmbH, Leipzig



Haftungsausschluss

Dieses Werk gibt allgemeine Empfehlungen. Es ersetzt keine Rechts- oder Fachberatung. Für konkrete Projekte sind ggf. zusätzliche Prüfungen nötig.

Copyright

© 2026 OpenTransfer

Lizenz

 Dieses Werk ist lizenziert unter einer CC BY SA 4.0 International Lizenz:
CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0/deed.de>

Sie dürfen dieses Werk vervielfältigen, verbreiten, öffentlich zugänglich machen und bearbeiten, auch kommerziell, sofern Sie den Herausgeber nennen, auf die Lizenz verlinken und Änderungen kennzeichnen. Ausgenommen sind die Logos der herausgebenden Organisationen; sie dürfen in bearbeiteten oder gekürzten Fassungen nicht weiterverwendet werden.

Das zugrundeliegende Verbundvorhaben »OpenTransfer – Strategien für den Transfer von Forschungsergebnissen im Open-Science-Kontext« wurde mit Mitteln des Bundesministeriums für Forschung, Technologie und Raumfahrt (BMFTR) unter Förderkennzeichen 03IO2204 gefördert. Die Verantwortung für den Inhalt liegt bei den Autor:innen.

Gefördert durch:

