

3 Einführung in Open Source Software

Seit Mitte der 90er Jahre macht ein Phänomen von sich reden, dass unter dem Begriff „Open Source Software“ (OSS) bekannt ist. Zunächst misstrauisch beäugt hat sich OSS mittlerweile zu einem festen Bestandteil in IT-Organisationen entwickelt. Neben der ominösen „Community“ – damit wird gemeinhin die Gesamtheit aller an der Entwicklung von OSS beteiligten Personen und Unternehmen bezeichnet – sind auch einzelne große IT-Häuser wie Sun und IBM wesentliche Treiber diese Entwicklung. Das Betriebssystem Linux, der APACHE Webserver, die Browser Mozilla oder Firefox sowie die Office-Anwendung OpenOffice.org sind nur einige der bekanntesten Beispiele für erfolgreiche und in breitem Rahmen eingesetzte OSS.

Open Source bedeutet mehr als nur die Offenlegung des Source Codes einer Software. Vielmehr verbirgt sich hinter Open Source letztlich eine Philosophie. Diese resultiert in einer alternativen Sichtweise auf drei Teilaspekte der Ökonomik der Softwarebranche.

1. Softwareentwicklungsmodelle

Zu den Paradigmen der OS-Softwareentwicklung gehören: Partizipation, Kollaboration und Iteration. Partizipation meint in diesem Kontext die Teilhabe von Nutzern und Entwicklern am resultierenden Softwareprodukt ohne kommerziell geregelten Nutzungsausschluss. Mit Kollaboration wird die Teilnahme von Nutzern und Entwicklern am Softwareentwicklungsprozess bezeichnet. Die konsequent modulare und schrittweise Entwicklung von Software wird durch das Prinzip der Iteration festgehalten. Die Umsetzung dieser drei Paradigmen variiert stark von Projekt zu Projekt, ist aber letztlich allen OS-Entwicklungsprojekten mehr oder weniger gemein.

2. Lizenzmodelle

Open Source ist eine Software dann, wenn sie unter einer von der Open Source Initiative (OSI) anerkannten Lizenz vertrieben wird. Die OSI ist ein non-profit-Unternehmen, welches als Anwalt für OSS und OS-Projekte auftritt und sich als Vermittler von verschiedenen Strömungen innerhalb der OS-Bewegung versteht. Zu ihren zentralen Anliegen zählt die Pflege der Open Source Definition, einer Sammlung von Grundsätzen für die Anerkennung einer Softwarelizenz als „Open Source Initiative Approved License“ sowie die Durchführung der Zertifizierung von Open Source Lizenzen. Bis Juli 2008 durchliefen 72 Softwarelizenzen den Zertifizierungsprozess bei der OSI. Eine detaillierte Vorstellung der Wichtigsten hiervon bietet Kap. 3.2. Eine vollständige Übersicht über alle verfügbaren OS-Lizenzen ist auf [ifross] erhältlich.

3. Geschäftsmodelle und -strategien

Mit der OSS-Philosophie wird eine Reihe von Versprechen propagiert: transparente Prozesse und architektonische Offenheit sollen zu besser Qualität, höherer Verlässlichkeit, größerer Flexibilität, niedrigeren Kosten und einer Beendigung des „räuberischen Vendor-Lock-Ins“, also der Abhängigkeit von einem Softwarelieferanten, führen [OSI 2008]. Für Softwareanbieter, die ihre Produkte unter Open Source Bedingungen entwickeln und vertreiben, stellen diese Versprechen zentrale Verkaufsargumente dar. Sie nutzen OSS, um

Unternehmensziele zu erreichen, wie z.B. die Erschließung neuer Marktbereiche durch die Konfrontation bisheriger „Platzhirsche“ mit OSS, die Nutzung von Cross-Selling-Potenzialen mit Hilfe von Dual Licensing-Strategien oder die Verlagerung strategischer Geschäftsfelder durch die Freigabe von Software hin zu Service-orientierten Geschäftsmodellen.

Um ein Grundverständnis des Wesens von OSS, der dahinter stehenden Philosophie und der Gründe ihres Erfolges zu bekommen, bietet es sich an, einen detaillierteren Blick auf die Ursprünge und die Entwicklung des Phänomens OSS zu werfen. Im Anschluss an den geschichtlichen Exkurs bietet das vorliegende Kapitel in einem weiteren Abschnitt einen umfassenden Einblick in das sensible und teilweise undurchsichtige Thema der OSS-Lizenzmodelle und stellt diese einander und kommerziellen Lizenzmodellen gegenüber.

3.1 Historischer Abriss

OSS ist an sich kein neues Phänomen. Vielmehr liegt hinter OSS eine mindestens 20-jährige Geschichte, die darüber hinaus letztlich bis zu den Anfängen der Softwareentwicklung zurück reicht. Neu ist jedoch die Qualität der resultierenden Softwareprodukte, welche zum Teil ein Niveau erreicht haben, das den Vergleich mit kommerziell hergestellter Software erlaubt. Der Erfolg von Open Source (OS) trägt allerdings auch dazu bei, dass der (nicht geschützte) Begriff mittlerweile von Softwareanbietern zu Marketingzwecken missbraucht wird. Daneben ist in den letzten Jahren eine Fülle von OS-artigen Lizenzen entstanden, die äußerst verschiedene rechtliche Restriktionen und Konditionen aufweisen. Um ein differenziertes Verständnis der Thematik zu gewinnen, soll daher im Folgenden zunächst die Entwicklung und die heutige Situation der OS-Bewegung dargestellt werden.

Als der entscheidende Schritt hin zur Entwicklung dessen, was heute unter „Open Source“ verstanden wird, gilt im Allgemeinen die Usenet-Veröffentlichung von Richard Stallmans „Initial Announcement of the GNU Project“ im Jahre 1983 als Reaktion auf die Kommerzialisierungsbestrebungen von Softwareentwicklungen am Bostoner MIT. Darin beschreibt Stallman seine Beweggründe zum Start eines Projektes, mit dem Ziel, ein mit UNIX vergleichbares freies Softwaresystem zu entwickeln und lädt jedermann ein, sich am Projekt zu beteiligen. Seine Motivation hierfür führt er im berühmten GNU-Manifesto von 1985 an und – nach eigenen Worten – auf den Kant’schen Imperativ zurück: „I consider that the golden rule requires that if I like a program I must share it with other people who like it.“ [Stallman 1998]. Wenig später (1989) gründete er die Free Software Foundation (FSF), die bis heute zu den Meinungsführern in der OS-Community zählt und den Begriff „Free Software“ entscheidend geprägt hat – stets mit der Betonung auf „frei“ im Sinne von „Freiheit“ oder „freie Rede“.

Die vielleicht wichtigste Innovation dieser Zeit aus Sicht der Open Source Bewegung war die Entwicklung der General Public License (GPL). Diese neuartige Lizenz räumt den Nutzern einer GPL-lizenzierten Software das Recht (die Freiheit)

1. auf Zugang zum Quellcode,
2. die Software zu kopieren und weiterzugeben,
3. das Programm zu verändern,
4. das veränderte Programm (unter den selben Lizenzbedingungen!) weiter zu verbreiten

ein [Grassmuck 2002].

Von diesen „vier Freiheiten von Open Source Software“ ist wahrscheinlich die vierte die bemerkenswerteste. Sie stellt sicher, dass derartig lizenzierte, freie Software selbst und ihre Derivate immer frei bleiben. Dies ist das Grundprinzip des sog. „Copyleft“, das auch als „Viral Effect“ [Rosenberg 2000] von OSS bezeichnet wurde und auf das im Abschnitt Lizenzmodelle noch näher eingegangen wird.

Stallmans GNU-Projekt wurde ein voller Erfolg und zur Grundlage heutiger Linux-Betriebssysteme. Allerdings war das Vorgehen bei der Entwicklung noch geprägt von der herkömmlichen Methode einer hierarchisch geplanten und monolithisch umgesetzten Software. Dies änderte sich mit der Entwicklung von LINUX seit 1991 durch die Veröffentlichung des LINUX-Kernels von Linus Torvalds im Usenet. Immer mehr Entwickler beteiligten sich mit eigenen kleinen Projekten an LINUX oder anderen Softwarebausteinen, nutzten zum Austausch das im Entstehen befindliche Internet und

organisierten sich in Projektteams mit relativ flachen Hierarchien. Wie von Raymond 1998 in seinem richtungsweisenden Aufsatz „The Cathedral and the Bazaar“ beschrieben, ähnelt die Organisation der Open Source Community eher einem orientalischen Bazaar als der Organisation, die zum Bau einer Kathedrale notwendig ist und häufig in kommerziellen Softwareprojekten anzutreffen war bzw. immer noch ist.

Spätestens gegen Ende der 1990er Jahre erreichte dann die Verbreitung von Linux und seinen Derivaten vor allem auf dem Servermarkt eine kritische Masse. Dies führte zusammen mit der Verbreitung des ebenfalls nahezu vollständig auf Open Source Software basierenden Internet dazu, dass Open Source zu einem heute feststehenden (wenn auch häufig missbrauchten) Begriff avancierte und ganz Konzerne auf Basis von OSS aufgebaut wurden. Yahoo, Google, Amazon sind die Beispiel dafür, wie sich äußerst profitable Geschäftsmodelle praktisch vollständig mit Hilfe OS-basierter IT-Infrastruktur aufbauen lassen. Seit Anfang der 2000er Jahre beschränken sich erfolgreiche Open Source Projekte auch nicht mehr auf Basisinfrastruktur. Vielmehr verbreitet sich OSS zunehmen in allen Bereichen der IT-Landschaft und hält seit 2005 auch verstärkt Einzug in die Business Intelligence.

3.2 Lizenzmodelle

Wie weiter oben bereits erwähnt, wird eine Software dann als OSS bezeichnet, wenn sie durch eine OSI-autorisierte Lizenz geschützt ist. Die Anerkennung einer Lizenz durch die OSI ist dabei an die Bedingungen der Open Source Definition (OSD) geknüpft, von denen sich die Wichtigsten wie folgt zusammen fassen lassen:

- Jeder hat das Recht, die Software frei weiterzugeben.
- Das Programm muss den Quellcode beinhalten oder die Möglichkeit, den Quellcode zum Selbstkostenpreis zu bekommen (nicht-binäre Form).
- Veränderungen an der Software sind zulässig. Ihre Weitergabe unter den Lizenzbedingungen der Ausgangssoftware muss gestattet sein (Copyleft).
- Es erfolgt keine Einschränkungen auf bestimmte Nutzer oder bestimmte Verwendungsgebiete, d.h. die Lizenz darf niemanden benachteiligen.
- Die Lizenz schränkt die Weitergabe zusammen mit anderen Produkten, anderen Technologien und anderer Software nicht ein, d.h. sie darf z.B. nicht verlangen, dass alle anderen Programme, die auf demselben Medium weitergegeben werden, ebenfalls quelloffen sind.
- Die genannten Rechte dürfen nicht durch andere Lizenzen beschränkt werden.

Ein wesentliches und sowohl ideologisch als auch pragmatisch häufig diskutiertes Merkmal vieler OS-Lizenzen ist das sog. Copyleft. Beim Begriff des „Copyleft“ handelt es sich zunächst einmal um ein Wortspiel mit dem englischen Begriff „Copyright“. Das Wortspiel konstruiert einen Gegensatz zum englischen Begriff Copyright (wörtlich: „Kopierrecht“) durch vertauschen von „rechts“ (engl. „right“) und „links“ (engl. „left“) sowie durch die implizite Doppelbedeutung von „left“ im Sinne von „überlassen“ [Wikipedia 2008]. Durch das Copyleft wird das Recht zum Kopieren grundsätzlich überlassen, während es durch das Copyright grundsätzlich verboten ist. Allgemein gesprochen erzwingt das Copyleft die Freiheit von Weiterbearbeitungen und Fortentwicklungen eines freien Ur-Werkes, um dadurch dessen unfreie Vereinnahmung für immer zu verhindern und die unbeschränkte Verbreitung von Kopien und veränderten Versionen eines Werkes für immer sicherzustellen. Durch den Zwang zur Veröffentlichung von notwendigen Informationen für die Verwertung eines Werkes (z.B. Quelltext) schützt das Copyleft auch den freien Zugang.

Das Copyleft ist der entscheidende Diskriminator für Open Source Lizenzen. Der Großteil der Lizenzen lässt sich daher grob in zwei Gruppen teilen:

1. Lizenzen ohne Copyleft

Lizenzen ohne Copyleft zeichnen sich dadurch aus, dass sie dem Lizenznehmer alle Freiheiten einer Open Source Lizenz einräumen und für Veränderungen der Software keine Bedingungen hinsichtlich des zu verwendenden Lizenztyps enthalten. Damit kann der Lizenznehmer veränderte Versionen der Software unter beliebigen Lizenzbedingungen weiterverbreiten, also auch in proprietäre Software überführen.

Der wichtigste Vertreter der Open Source Lizenzen ohne Copyleft ist die Berkeley Software Distribution Lizenz (BSD). Die Pflichten eines Anwenders bei der

Weiterverbreitung einer unter der BSD stehenden Software bestehen lediglich darin, die BSD als Dokument beizufügen und den Namen des Urhebers nicht für Marketing-Zwecke zu gebrauchen. Die BSD wird von vielen Open Source Projekten gerade wegen ihrer liberalen Lizenzbedingungen geschätzt, da diese die Verwendung von BSD-lizenzierter Software im kommerziellen Rahmen praktisch ohne Einschränkungen erlauben.

2. Lizenzen mit Copyleft

Bei Lizenzen mit Copyleft wird der Lizenznehmer verpflichtet, von der ursprünglichen Software abgeleitete Werke ebenfalls nur unter den Bedingungen der Ursprungslizenz weiterzuverbreiten. In diesem Zusammenhang lassen sich Lizenzen mit strengem Copyleft von Lizenzen mit beschränktem Copyleft unterscheiden. Lizenzen mit beschränktem Copyleft ähneln den Lizenzen mit strengem Copyleft abgesehen davon, dass das Copyleft-Prinzip nur eingeschränkt gültig ist. Sofern Modifikationen der Software in separaten Dateien abgelegt sind, können diese Dateien auch unter bestimmten anderen, z.B. proprietären Lizenzbedingungen weiterverbreitet werden. Damit lässt sich die Kombination von Software unter verschiedenen Lizenztypen erleichtern.

Die beiden am weitesten verbreiteten Lizenzen mit Copyleft sind die General Public License (GPL) und die Lesser General Public License (LGPL), die beide mittlerweile in mehreren Versionen vorliegen.

- Die GPL gilt als die restriktive Open Source Lizenz in dem Sinne, dass bei der Kombination einer GPL-lizenzierten Software mit Quellcode aus einer beliebigen anderen – auch proprietären – Quelle, im Falle der Weiterverbreitung das gesamte Ergebnis unter GPL gestellt werden muss. Dies ist eine Folge des „Viral Effects“. Daneben enthält die GPL auch einen Haftungsausschluss, welcher allerdings nach deutschem Recht nicht zulässig ist. In der Praxis wird daher GPL-lizenzierte Software als Schenkung betrachtet. Damit haftet der Entwickler nur im Falle vorsätzlichen Verschweigens schwerwiegender Fehler. Allerdings versehen auch Hersteller proprietärer Software ihre Lizenzbestimmungen mit Einschränkungen der Gewährleistung [Maaß, Scherm 2005, S. 186].
- Die LGPL entspricht der GPL weitestgehend. Allein auf den Viral Effekt wird bei dieser Lizenz verzichtet. Dies ermöglicht es, LGPL-lizenzierte Software zusammen mit proprietärer Software zu verwenden, ohne dass das daraus hervorgehende Produkt wiederum unter die LGPL gestellt werden müsste.

Neben der grundsätzlichen Unterscheidung in OSS mit und ohne Copyleft sowie mit und ohne Viral Effekt lassen sich im Wesentlichen noch vier weitere Merkmale ausmachen, anhand derer sich die verschiedenen Lizenzbedingungen für Software unterscheiden. Im Einzelnen betrifft dies Regelungen hinsichtlich

- der Kosten der Nutzung,
- der Freiheit der Verbreitung,
- der Beschränkungen bei der Nutzung und
- der Zugänglichkeit des Quellcodes.

Anhand dieser Merkmale unterscheidet sich Open Source Software nicht nur von herkömmlicher proprietärer Software, sondern auch von anderen Verbreitungsformen wie Free- oder Shareware. Die folgende Tabelle stellt die verschiedenen Formen der Softwarelizenzierung anhand der sechs Merkmale gegenüber.

Tabelle 3-1: Traditionelle und quelloffene Softwarelizenzen im Vergleich²

	Kostenlose Nutzung	Freie Verbreitung	Uneingeschränkte Nutzung	Quellcode offen	Copyleft	Viral Effect
Proprietär	-	-	-	-	-	-
Shareware	Begrenzt	Ja	-	-	-	-
Freeware	Ja	Ja	Ja	-	-	-
BSD	Ja	Ja	Ja	Ja	-	-
LGPL	Ja	Ja	Ja	Ja	Ja	-
GPL	Ja	Ja	Ja	Ja	Ja	Ja

Der Großteil der OS-Projekte verwendet die GPL, LGPL oder BSD. Bei dem OSS-Hoster Sourceforge.net stehen über 90% (von 180.000 registrierten Projekten) unter einer dieser drei Lizenzen. Abbildung 3-1 zeigt die Verteilung der am häufigsten verwendeten OS-Lizenzen bei Sourceforge.net.

² Vgl. Maaß, C.; Scherm, E. (2005): Software-Lizenzierung; Perens, B. (1999): The Open Source Definition.

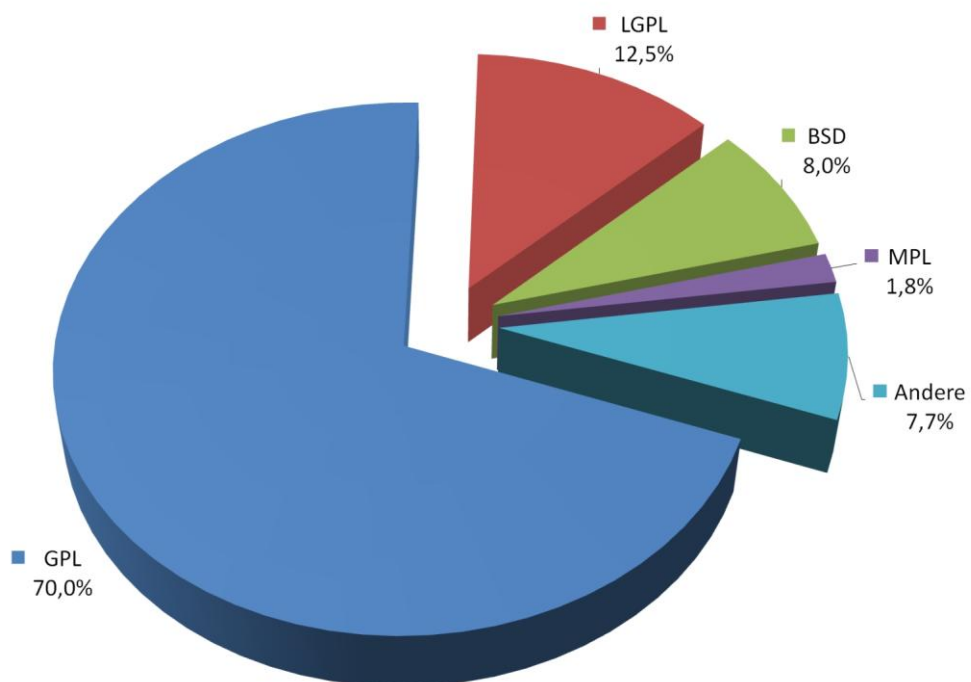


Abbildung 3-1: Verteilung der Lizenzen beim OS-Hoster Sourceforge.net³

Gerade sehr erfolgreiche und damit für den kommerziellen Einsatz zuvorderst interessante Projekte verwenden oftmals Derivate dieser Lizenzen. Daher ist vor einer Weiterentwicklung mit kommerziellem Interesse, also der Absicht zum Verkauf der entwickelten Software, immer eine gründliche Recherche der bestehenden Lizenzbedingungen notwendig. In der Regel lässt sich hier aber auf solide Informationsquellen wie [ifross] oder [OSI] zurückgreifen, welche die wichtigsten Merkmale auch bei ausgefallenen Lizenzen auflisten und bewerten. Für den unternehmensinternen Einsatz von Open Source Software bestehen hingegen prinzipiell keine Beschränkungen.

³ Vgl. Kleijn (2006): Open-Source-Lizenzen.

3.3 Literaturhinweise

- DiBona, C.; Ockman, S.; Stone, M. (Hrsg., 1999): Open Sources. Voices from the Open Source Revolution. London et al. 1999.
- Fitzgerald, Brian (2006): The Transformation of Open Source Software. In: MIS Quarterly, Jg. 30, H. 3, 2006, S. 587-598.
- Grassmuck, V. (2002): Freie Software. Zwischen Privat- und Eigentum. Bundeszentrale für politische Bildung. Bonn 2002.
- Hars, A. (2002): Open Source Software. Revolution auf dem Softwaremarkt? In: WISU das Wirtschaftsstudium, Jg. 31, 2002, S. 542-552.
- ifross (2008): Lizenz-Center. http://www.ifross.de/ifross_html/lizenzcenter.html, Abruf vom 24.06.2008.
- Jaeger, T. (2000): Zwang zur Freiheit. Freie Software als Ausweg aus dem Lizenzterror. In: CT - Magazin für Computertechnik, H. 8, 2000, S. 120.
- Jaeger, T.; Metzger, A. (2006): Open Source Software. Rechtliche Rahmenbedingungen der Freien Software. München 2006.
- Kleijn, A. (2006): Open-Source-Lizenzen. <http://www.heise.de/open/Open-Source-Lizenzen--/artikel/75786>, Abruf vom 24.06.2008
- Lutterbeck, B.; Bärwolff, M.; Gehring, R. A. (2008): Open Source Jahrbuch 2008. Zwischen freier Software und Gesellschaftsmodell. Berlin 2008.
- Maaß, C.; Scherm, E. (2005): Software-Lizenzierung. In: WISU das Wirtschaftsstudium, Jg. 34, H. 2, 2005, S. 185-188.
- Morner, M. (2003): Open Source Software. In: WISU das Wirtschaftsstudium, Jg. 32, H. 3, 2003, S. 318-321.
- Open Source Initiative (2008): The Approved Licenses. <http://www.opensource.org/licenses/index.html>, Abruf vom 23.06.2008.
- Perens, B. (1999): The Open Source Definition. In: DiBona, C.; Ockman, S.; Stone, M. (Hrsg.): Open Sources. Voices from the Open Source Revolution. London et al. 1999, S. 171-188.
- Raymond, E. (1998): The Cathedral and the Bazaar. <http://catb.org/~esr/writings/cathedral-bazaar/>, Abruf vom 24.06.2008.
- Rosenberg, D. (2000): Open Source. The Unauthorized White Papers. Chicago et al. 2000.
- Sauerburger, H. (Hrsg., 2004): Open-Source-Software. In: HMD-Praxis der Wirtschaftsinformatik, Heft 238, Heidelberg 2004.
- Schiffner, T. (2003): Open Source Software. Freie Software im deutschen Urheber- und Vertragsrecht. München 2003.
- Spath, Dieter (Hrsg.) (2006): Open Source Software. Strukturwandel oder Strohfeuer? Stuttgart 2006.
- Stallman, R. (2002): Why Software should be free. In: Gay, J. (Hrsg.): Free Software. Free Society: Selected Essays of Richard M. Stallmann, Free Software Foundation 2002, S. 119-132.

- Stallman, R. (1998): The GNU-Project. <http://www.gnu.org/gnu/the-gnu-project.html>, Abruf vom 23.06.2008.
- Stein, A. (2006): Die Open-Source-Bewegung. Entwicklung, Organisation, innere Ökonomie. Saarbrücken 2006.
- Torvalds, L.; Diamond, D. (2001): Just for Fun. Wie ein Freak die Computerwelt revolutioniert. München/Wien 2001.
- Wikipedia (2008): Copyleft. <http://de.wikipedia.org/wiki/Copyleft>, Abruf vom 24.06.2008.
- www.opensource.org (2008): The Open Source Initiative. Open Source Definition. <http://www.opensource.org/docs/definition.html>, Abruf vom 23.06.2008.