

Kurze betriebswirtschaftliche Analyse von Open Source Software

© 2006, Daniel Kühni, indato GmbH

Dieses Dokument untersteht der Creative Commons Attribution-ShareAlike 2.0 Germany Lizenz. Diese Lizenz kann entweder online unter <http://creativecommons.org/licenses/by-sa/2.0/de/> eingesehen werden oder aber per Brief bei Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA angefordert werden.

Inhaltsverzeichnis

0 Einleitung.....	1
1 Open Source Software.....	1
Spezielle Eigenschaften von Open Source Software.....	3
Kommerzielle Open Source Software.....	4
Geschäftsmodelle mit Open Source Software.....	5
Rechtliche Aspekte.....	6
Urheberrecht.....	6
Software-Lizenzen	6
Multi-licensing.....	8
Einsatz von Open Source Software in der Schweiz.....	9
Grenzen von Open Source als Geschäftsmodell.....	10
Fazit.....	10
2 Literaturverzeichnis / Quellenangaben.....	11

0 Einleitung

Diese Kurzanalyse ist ein Auszug aus der Abschlussarbeit von mir für das Nachdiplomstudium Betriebswirtschaft / Unternehmensführung am Managementzentrum der Hochschule für Technik und Informatik in Bern.

Die Analyse soll Open Source Software in einer kompakten Form aus betriebswirtschaftlicher Sicht beleuchten und als Grundlage für eigene Geschäftsmöglichkeiten und -modelle dienen können.

Kommentare und Anregungen sind jederzeit herzlich willkommen und können direkt an den Autor [<mailto:daniel.kuehni@indato.ch>] gerichtet werden.

1 Open Source Software

Das Phänomen Open Source Software (OSS) wurde im Rahmen von vielen Studien eingehend untersucht. Im deutschsprachigen Raum wurde vor allem die Entscheidung der Stadt München, ihre Verwaltung vollständig auf den Einsatz von OSS umzustellen¹, stark beachtet und führte zu einigen Publikationen, die sich mit dem Einsatz und den gesellschaftlichen Auswirkungen von OSS beschäftigen. Eine sehr umfassende Sammlung von Beiträgen enthält das Open Source Jahrbuch, das 2004 zum ersten Mal erschien [OSJahrbuch04] und dank der grossen Beachtung im 2005 in einer neuen Version aufgelegt wurde [OSJahrbuch05].

¹ Siehe Projekthomepage des Projekts LiMux [LiMuxHome05] für weiterführende Informationen.

Definition von Open Source Software (OSS)

Open Source Software, resp. die dazu verwendete Lizenz soll gemäss der allgemein akzeptierten Definition der Open Source Initiative [OSIOSSDefinition05] folgende Eigenschaften aufweisen:

1. Eine lizenzkostenfreie Weiterverbreitung (auch Verkauf) des Produkts muss erlaubt sein.
2. Der Quellcode (Source Code) des Produkts muss entweder mitgeliefert werden oder einfach zugreifbar sein.
3. Werke zweiter Hand (derived works) durch Modifikation des Ursprungswerks müssen erlaubt sein und die Verbreitung dieser muss unter denselben Bedingungen wie das Original möglich sein.
4. Der Urheber des Source Codes muss ersichtlich bleiben und Änderungen müssen als solche gekennzeichnet sein.
5. Die Diskriminierung von Personen oder Gruppen ist nicht erlaubt.
6. Die Einschränkung des Einsatzgebiets des Produkts ist nicht erlaubt.
7. Eine zusätzlich geforderte weiter einschränkende Lizenzierung ist nicht erlaubt.
8. Die Lizenz darf nicht produktspezifisch gefasst sein und muss dieselben Rechte erlauben, wenn das Produkt aus seiner ursprünglichen Zusammensetzung gelöst und in einer anderen Zusammensetzung vertrieben wird.
9. Die Lizenz darf keine Einschränkungen gegenüber anderer Software, die zusammen mit ihr vertrieben werden, verlangen.
10. Die Lizenz muss Technologie-neutral abgefasst sein.

Als Kurzdefinition kann sicher gesagt werden, dass man von Open Source Software sprechen kann, wenn der lesbare Quellcode (Source Code) gegen eine vernünftige Gebühr oder kostenlos erhältlich ist und Änderungen oder Erweiterungen unter gewissen Bedingungen zulässig sind, die in einer mitgelieferten Lizenz ersichtlich sein müssen.

Open Source Software ist nicht kostenlos

Wichtig im Zusammenhang mit Open Source Software ist der Aspekt der Freiheit, der häufig falsch interpretiert wird. OSS ist nur frei im Sinne von freier Meinungsäusserung, nicht im Sinne von kostenlos („free speech“ versus „free beer“). OSS kann sehr wohl kostenpflichtig sein, jedoch nie in Form von Lizenzgebühren, die für die Nutzung der Software zu entrichten wären. Hier liegt der Unterschied zur so genannten Free Software, die neben den Aspekten von OSS auch tatsächlich kostenlos angeboten werden muss.

Als Gegenspieler von Open Source Software gelten alle Unternehmen, die ihre Produkte nicht im Sinne der obigen Definition behandeln, insbesondere keinen Zugriff auf den Quellcode ermöglichen und häufig auch Schnittstellenbeschreibungen nicht veröffentlichen. Als Paradebeispiel und grosser Gegner von OSS sei hier stellvertretend Microsoft genannt.

Es ist selbstredend, dass zwischen OSS und den Closed Source Software-Anbietern, allen voran Microsoft, ein heftiger Schlagabtausch stattfindet, der zuweilen belustigende oder groteske Ausmasse annehmen kann. Legendär ist die Schlacht um die so genannten Halloween Memoranden [OSIHalloween05], in denen Microsoft ihre interne Strategie im Umgang mit OSS skizziert.

Studien sind sehr undurchsichtig und teilweise gekauft

David Wheeler hat eine bemerkenswerte Meta-Studie zum Thema Open Source Software verfasst [WheelerStudy05], die eindrücklich zeigt, zu welchen unterschiedlichen Ergebnissen Studien gelangen können, wenn die grundsätzlichen Annahmen nicht deutlich gemacht oder sogar verschwiegen werden. Seine Arbeit bietet eine Handhabe, um gekaufte Studien entsprechend kritisch einordnen zu können.

Spezielle Eigenschaften von Open Source Software

Open Source Software hat gewisse Eigenschaften, die charakteristisch sind. Eine Auswahl dieser Eigenschaften wird im folgenden kurz diskutiert. Dabei wird OSS als Oberbegriff für Freie und Open Source Software verstanden, somit als FOSS.

Kollaboratives Entwicklungsmodell

Die Entwicklung von FOSS verläuft grundlegend anders als diejenige von Closed Source Software. Eric Raymond hat dies in seinem berühmten Aufsatz „The Cathedral and the Bazaar“ [Cathedral00] im Jahr 1997 sehr treffend aufgezeigt. Die Analogie des Baus einer Kathedrale gegenüber einem Basar, in dem die Ideen und Vorschläge frei getauscht und wirtz zusammengearbeitet wird, sollen das Verhältnis der Entwicklung von Closed zu Open Source darstellen. Selbstverständlich ist dies überspitzt dargestellt, trifft aber den Kern sehr gut. Wichtig ist zu bemerken, dass entgegen der allgemeinen Erwartung FOSS Projekte sehr stark organisiert sind und - falls sie erfolgreich sein sollen - in diesen Aspekt sehr viel Aufwand investiert werden muss.

Offene und standardisierte Schnittstellen

Durch die Offenlegung des Quellcodes sind alle Schnittstellen automatisch offen gelegt. Da ein FOSS Projekt meistens nur eine Komponente darstellt, die im Zusammenspiel mit anderen Komponenten steht, werden sehr häufig bestehende Standards für die Schnittstellen herangezogen.

Keine Abhängigkeit von einem einzelnen Anbieter

Da bei FOSS keinerlei Nutzungsbedingungen zulässig sind, kann auch keine Abhängigkeit zu einem Anbieter damit erzeugt werden. Bei FOSS Projekten ist es üblich, dass derjenige Nutzer, der am meisten Interesse an der Weiterentwicklung hat und als guter Entwickler gilt, ein bestehendes Projekt zur Weiterentwicklung übernehmen kann.

Marketing und Vertrieb meist ausschliesslich über Internet

Das Internet spielt sowohl als Drehscheibe für die Entwicklung als auch als Vertriebs- und Marketingplattform eine zentrale Rolle. Sourceforge² bietet als grösste derartige Plattform über 100'000 FOSS Projekten eine Heimat.

Hohe Qualität durch aktive Mitarbeit

Diejenigen Produkte, die auf grosses Interesse stossen, werden von sehr vielen Entwicklern intensiv analysiert. Ein grosser Anteil der Benutzer liefert bei FOSS Projekten sehr detaillierte und fundierte Problembereichte an das Entwicklerteam ab, was bei der Fehleranalyse sehr wertvoll ist. Durch diese Effekte entsteht über die Zeit qualitativ sehr hochstehende Software.

Support nicht garantiert

Die Produktunterstützung ist bei FOSS nicht garantiert. Dieser Punkt wird bei den Gegnern von FOSS jeweils als zentralen Punkt herausgestrichen und wird auch von vielen Studien (z.B. [CTPOSSSchweiz03] und [FhGOSSStudie05]) als Problem-punkt dargestellt. Die Erfahrungen zeigen jedoch, dass der Support einer Community, d.h. einer Nutzer- und Entwicklergemeinschaft hinter einem Projekt, sehr engagiert sein und sehr rasch zu einer Problemlösung führen kann. Voraussetzung dafür ist natürlich, dass die Anfrage genügend spannend ist und die Community genügend gross. Unsere Erfahrungen zeigen auch, dass unsere Kunden mit dem Support von Closed Source Lieferanten häufig unzufrieden sind.

Weiterentwicklung häufig nach Lust-Prinzip

Ein weiterer häufig genannter Kritikpunkt gegenüber FOSS ist derjenige der ungeplanten Weiterentwicklung von Produkten. Da die Motivation als Entwickler eines FOSS Projekts primär ideell ist, wird dieser Entwickler seine Arbeitsleistung zuerst und vor allem in diejenigen Weiterentwicklungen stecken, die für ihn selber

„Linus' Law: Given enough eyeballs, all bugs are shallow.“
[Cathedral00]

FOSS Support ist sehr kontrovers

² www.sourceforge.net

am meisten Gewinn bringen, d.h. ein interessantes Problem darstellen. Software besteht allerdings nicht nur aus interessanten Teilen. Daher besteht eine potentielle Gefahr, dass FOSS Produkte nie wirklich Produktionsreife erlangen werden. Diese Vermutung lässt sich durch Praxisbeispiele zwar widerlegen, nicht jedoch in jedem Bereich gleich gut. So sind Infrastruktur-Produkte wie z.B. Betriebssysteme (Linux) oder Webserver (Apache) in sehr grosser Zahl in produktivem Einsatz und bewähren sich ständig. Im Bereich der Businesssoftware gehen einige Studien aber davon aus, dass die FOSS Produkte den entsprechenden kommerziellen Produkten noch nicht gewachsen sind.

Kommerzielle Open Source Software

Es gibt zunehmend Beispiele von funktionierenden Open Source Projekten, die von kommerziellen Firmen „übernommen“ und mittels einer Doppellizenzstrategie sowohl als OSS-Projekt weitergeführt als auch als kommerzielles Produkt mit weiterführender Funktionalität oder sonstigen Erweiterungen vertrieben werden. Als Beispiel sei hier das Projekt JasperReports [ZDNetBlogOS05] erwähnt, ein Produkt zur einfachen Erstellung aller Art von Berichten und Auswertungen, das in vielen anderen Produkten verwendet wird.

Lizenzprobleme beim Applikationsserver JBoss?

Probleme bei solchen Übernahmen stellen sich vor allem im Bereich des Urheberrechts und der Lizenzierung. Im Bereich des Urheberrechts muss sichergestellt werden können, dass alle Entwickler, die bisher Code zum Projekt beigetragen haben, ihr Urheberrecht abgetreten haben oder zumindest greifbar und willens wären, eine solche Abtretung vorzunehmen. Sofern dies nicht geschehen ist und eine entsprechende OSS-Lizenz verwendet wird, können tiefgreifende Probleme auftreten, wie das Beispiel eines der früheren Kernentwickler von JBoss deutlich aufzeigt [JBossLizenz05]³.

Eine solche Übernahme scheint also nur dann wirklich zu funktionieren, wenn der Kreis der aktiven Entwickler sehr klein ist und diese ein Interesse an der Übernahme haben.

Das grössere Problem besteht allerdings darin, dass die bereits etablierte Community hinter einem funktionierenden OSS-Projekt eine Übernahme goutiert und weiterhin ihre Freizeit in das Projekt zu investieren bereit ist. Für Entwickler besteht das Risiko, dass ihre Arbeit in der Zukunft von der Open Source Welt verschwinden würde, falls eine solche Übernahme durchgeführt wird.

Es gibt natürlich auch genügend Beispiele von OSS-Projekten, bei denen der kommerzielle Teil rein in den um das Produkt herum angesiedelten Dienstleistungen liegt, wie beispielsweise dem bereits erwähnten Applikationsserver JBoss. Dort sieht das Geschäftsmodell vor, dass die Hauptentwickler professionell an der Weiterentwicklung des Produkts arbeiten können, da sie bei einer Firma angestellt sind, die ein aktives Interesse am Produkt besitzt. Der Ertrag wird dabei meistens durch Schulung und Support erzielt. Marc Fleury, der Gründer der JBoss Inc., hat für dieses Geschäftsmodell den Begriff „Professional Open Source“ geprägt.

Professionalisierung von Open Source

Allgemein kann sicher festgestellt werden, dass heute Open Source viel von seinem ursprünglichen rebellischen und unkonventionellen Image verloren hat und zunehmend als Geschäftsmodell verstanden wird, bei dem letztlich auch Geld verdient werden darf. Der Begriff Professionalisierung im Sinne von Marc Fleury soll dabei aufzeigen, dass Open Source sozusagen erwachsen geworden ist und den Weg aus dem reinen Hobbybereich in echte Geschäftsmodelle gefunden hat.

³ Der frühere JBoss-Entwickler Rickard Öberg verlangt, dass diejenigen Teile von JBoss, an denen er die Urheberrechte besitzt - und das scheinen nicht wenige zu sein - neu unter der GPL lizenziert werden müssen. JBoss hatte bisher den Code unter der wesentlich weniger restriktiven LGPL veröffentlicht. Die LGPL erlaubt aber explizit jederzeit den Wechsel zur GPL durch den Urheber. Ob dieser Forderung tatsächlich Folge geleistet werden muss und welche Auswirkungen JBoss daraus erwachsen werden, ist noch völlig offen. Anmerkung: JBoss wurde zwar nicht eigentlich übernommen, jedoch durch die Gründung einer Firma JBoss Inc., an der auch Investoren beteiligt sind, kommerzialisiert.

**Oracle will JBoss
übernehmen**

Das Modell der Open Source Software wird offensichtlich sogar bereits so ernst genommen, dass grosse Software-Anbieter wie Oracle nun ihre Strategie dahingehend geändert haben, Open Source nicht mehr zu bekämpfen, sondern sich im grossen Stil einzuverleiben. Gemäss einem aktuellen Bericht in der Online-Ausgabe von BusinessWeek [OracleOSSBW06] soll Oracle bereit sein, 600 Millionen US-Dollar in die Akquisition von JBoss, Zend⁴ und Sleepycat⁵ zu investieren. Entsprechende Übernahmegespräche seien gemäss BusinessWeek im Gang.

Geschäftsmodelle mit Open Source Software

Raphael Leiteritz hat in seinem Aufsatz [Leiteritz04] aufgezeigt, welche prinzipiellen Geschäftsmodelle im Zusammenhang mit OSS überhaupt möglich sind.

- Produkt-Geschäftsmodelle
 - OSS-Distributor⁶
 - OSS-Applikationsanbieter⁷
 - OSS-Appliance-Hersteller⁸
- Dienstleistungs-Geschäftsmodell
- Mediator-Geschäftsmodell⁹

Leiteritz beschreibt die Modelle und vergleicht ihre Möglichkeiten anhand einer generellen Wertschöpfungskette für die Softwareindustrie.

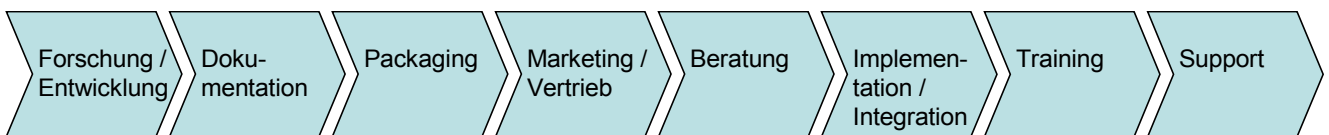


Abbildung 1: Wertschöpfungskette für Software

Jedes Geschäftsmodell deckt einen bestimmten Ausschnitt aus dieser Wertschöpfungskette ab, realisiert dabei ein unterschiedliches Gewinnmuster und besitzt unterschiedliche Möglichkeiten für Partnerstrukturen.

Leiteritz schliesst in seinem Aufsatz, dass nur das OSS-Appliance-Anbieter- und das Dienstleistungs-Geschäftsmodell wirklich überlebensfähig sind. Alle anderen Geschäftsmodelle müssen zwingend in Kombination mit sekundären Geschäftsmodellen - meistens im Dienstleistungsbereich - betrieben werden, um langfristig erfolgreich durchgehalten werden zu können.

Dadurch wird der Druck auf reine Dienstleister erhöht. Ausserdem sieht Leiteritz den Druck auf OSS-Dienstleister ebenfalls durch das Eindringen herkömmlicher IT-Anbieter (z.B. IBM) in diesen Bereich stark steigend, was aus heutiger Sicht sicher bestätigt werden kann.

⁴ Zend ist die Firma, die vor allem die Entwicklung der im Internet-Bereich sehr weit verbreiteten Programmiersprache PHP vorantreibt.
⁵ Sleepycat ist die Firma, die hinter der Open Source Datenbank Berkeley DB steht. Diese Datenbank wird von vielen Open Source Projekten eingesetzt.
⁶ Ein OSS-Distributor fasst OSS-Komponenten zu einem Paket zusammen und bietet dieses den Kunden an, meist in Verbindung mit eigenen Mechanismen für die Wartung und den Betrieb der Komponenten. Beispiele: Novell (SuSE), Red Hat, Mandrake.
⁷ Ein OSS-Applikationsanbieter kann ein bestehendes kommerziellen Produkt unter einer OSS Lizenz freigeben (Bsp. Netscape mit dem Mozilla Projekt oder IBM mit Eclipse), eine Neuentwicklung beginnen (meist mit Doppellizenzierung) oder ein bestehendes Projekt übernehmen und kommerzialisieren (Bsp.: JasperReports [ZDNetBlogOS05]).
⁸ Eine Appliance ist eine „Black-Box“, die meistens sowohl aus Hardware als auch aus Software-Komponenten besteht und einen bestimmten Zweck erfüllen kann (Bsp.: Firewalls, Settop Boxen).
⁹ Ein Mediator bringt unterschiedliche Interessengruppen über eine gemeinsame Plattform miteinander in Verbindung. Das vielleicht wichtigste Beispiel im OSS-Umfeld stellt sicher SourceForge dar, das beinahe alle weltweit verfügbaren OSS-Projekte hostet.

Rechtliche Aspekte

Die grundlegenden Aussagen zu den rechtlichen Aspekten von OSS stammen grösstenteils von einem Workshop des Vereins Openlaw (vgl. [OpenlawLOTS05]) an der LOTS 2005, einer Veranstaltung zur Förderung von Open Source an der Universität Bern.

Urheberrecht

Zentraler Aspekt aller rechtlichen Überlegungen im Bereich von Software ist das Urheberrecht des Autors. Die Open Source Bewegung wollte dieses Urheberrecht wieder dem Entwickler übergeben und diesem das alleinige Recht über die Bedingungen der Weitergabe der Entwicklungsergebnisse übertragen.

Bei den traditionellen Softwareprodukten muss das Urheberrecht von den Entwicklern an ihre Arbeitgeber abgetreten werden.

Software-Lizenzen

Software wird anders als materielle Güter nicht komplett dem Käufer übergeben, sondern jeweils unter gewissen Bedingungen zur Nutzung überlassen. Diese Nutzungsbedingungen sind in einer Software-Lizenz festgelegt. Der Käufer der Software muss diese Lizenz akzeptieren, damit er die erworbene Software auch tatsächlich nutzen darf. Bei Open Source Software sind diese Bedingungen gemäss der im Kapitel 1 vorgestellten Open Source Initiative [OSIOSSDefinition05] geregelt. Zentral ist das Recht auf Verbreitung und Weitergabe sowohl der Originalsoftware und daraus abgeleiteter, weiterentwickelter Software, rechtlich sogenannte Werke zweiter Hand.

Ein guter Überblick über Software-Lizenzen und ihre typischen Vertreter bietet die offene Wissensdatenbank des Projekts BerliOS [BerliOS05].

Copyleft anstatt Copyright

Als wichtigster Vertreter von OSS-Lizenzen betrachten wir kurz die vom GNU Projekt¹⁰ eingeführte Lizenz GPL (GNU General Public License), die aktuell in der seit 1991 gültigen Version 2 vorliegt. Darin wird ein wichtiges Prinzip vieler OSS-Lizenzen definiert, das sogenannte „Copyleft“. Dieses Prinzip soll aufzeigen, dass das Urheberrecht, das üblicherweise über ein Copyright abgesichert wird, bei OSS zwar nicht aufgehoben, aber dem Nutzer sozusagen überlassen¹¹ wird.

Copyleft ist ansteckend

Wichtig bei der GPL ist, dass dieses Copyleft-Prinzip einen virulenten Charakter besitzt, d.h. jede Software, die mit einer unter der GPL lizenzierten Software sozusagen in Berührung kommt, muss das Prinzip übernehmen. Konkret bedeutet dies, dass eine kommerzielle Software, die nur zusammen mit einer GPL-lizenzierten Software funktioniert, automatisch frei verfügbar gemacht werden muss. Dabei ist es rechtlich unerheblich, ob die GPL-lizenzierte Software mit der kommerziellen Software gemeinsam ausgeliefert wird oder nicht. Dieses für kommerzielle Interessen sehr problematische Verhalten der GPL wurde mit der Einführung der LGPL (GNU Lesser General Public License) dahingehend entschärft, dass nur Modifikationen an der LGPL-lizenzierten Software selber wiederum unter derselben Lizenz freigegeben werden müssen. Wird aber nur Code aus einer LGPL-lizenzierten Software in einer kommerziellen Software statisch oder dynamisch verlinkt, so ist dies nicht gefordert. Aus diesem Grund wird die LGPL häufig für Programmbibliotheken eingesetzt und hiess zu Beginn auch „Library General Public License“.

Aktuellerweise wird die GPL gerade einem intensiven Review unterzogen und soll in einer erneuerten Version 3¹² wesentlich freundlicher für die kommerzielle Nut-

¹⁰ <http://www.gnu.org/>

¹¹ Englisch Vergangenheitsform: „left“

¹² <http://gplv3.fsf.org/>

zung von GPL-lizenzierter Software sein. Durch diese Änderung wird die GPL wohl für viele Projekte wieder attraktiver, der Versionenwechsel muss aber bei bestehenden GPL-lizenzierten Softwarepaketen nicht automatisch vollzogen werden.

Andere OSS-Lizenzen wie beispielsweise die Apache Software License sind für einen kommerziellen Einsatz wesentlich freundlicher und gestatten die Verwendung von OSS-Komponenten innerhalb einer kommerziellen Software, ohne dass diese dadurch wiederum unter einer OSS-Lizenz freigegeben werden müsste. Bei diesen Lizenzen ist also das Copyleft-Prinzip nicht wirksam.

Fehlende Rechtspraxis

Die Rechtssicherheit im Umgang mit OSS im Allgemeinen und GPL-lizenzierter Software im Speziellen ist noch nicht restlos gegeben. Es gibt zurzeit weltweit noch kein wirksames Gerichtsurteil infolge eines Verstosses gegen eine OSS-Lizenz. Allerdings gibt es keine grundlegenden Zweifel an deren prinzipiellen Umsetzbarkeit. In einem Fall hat das Landgericht München jedoch die grundsätzliche Gültigkeit und Durchsetzbarkeit der GPL für Deutschland bestätigt (vgl. [HeiseGPL04]), es kam jedoch im Anschluss nicht zu einem eigentlichen Verfahren, da sich das Problem aussergerichtlich lösen liess.

Eine Übersicht über die zu Beginn des Jahres 2003 im OSS Projekt-Repository Sourceforge³ im Einsatz stehenden Projekte zeigt folgendes Bild (aus [BerliOS05]):

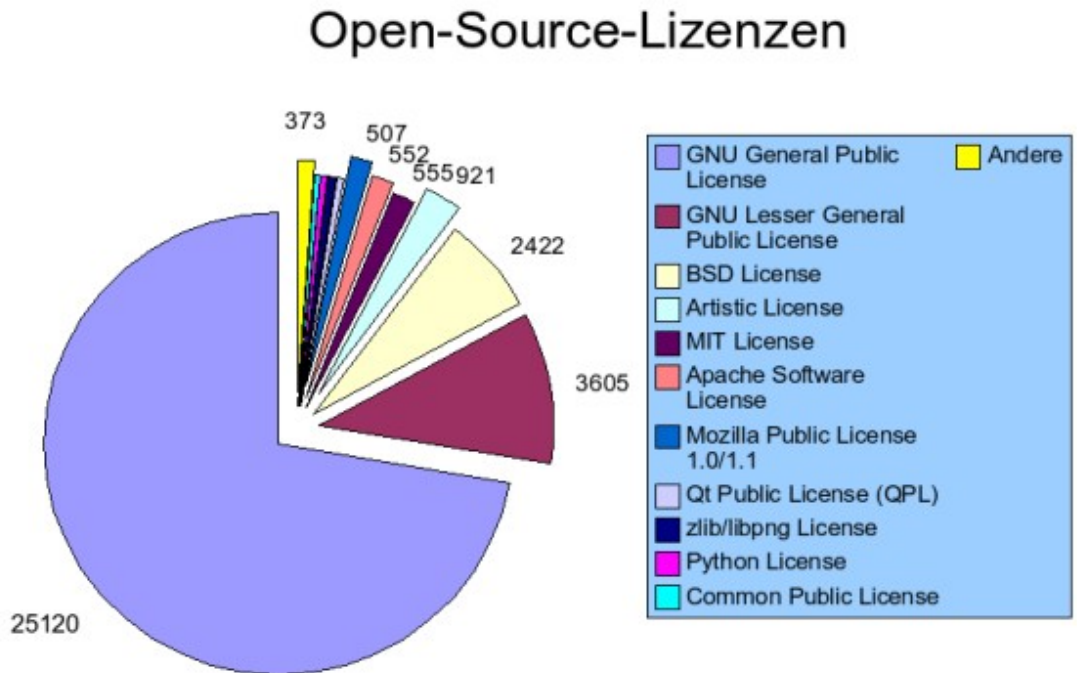


Abbildung 2: Übersicht der bei Sourceforge Anfangs 2003 verwendeten OSS Lizenzen

Damals waren bei Sourceforge erst ca. 50'000 Projekte vorhanden (aktuell sind es weit über 100'000 Projekte). Die mit deutlichem Abstand am meisten eingesetzte Lizenz war die GPL und ihre Schwesterlizenz, die LGPL.

Lizenzdschungel

Ein grosses Problem besteht zurzeit für den Nutzer von Open Source Software in der Vielfalt der aktuell eingesetzten Lizenzen. Als Beispiel soll hier eine sehr häufig im Einsatz stehende Kombination von OSS-Produkten betrachtet werden, ein so genanntes LAMP-System. LAMP steht dabei für Linux, Apache, MySQL und Perl, PHP resp. Python für die wesentlichen Vertreter im Applikationsstack. Im Beispiel gelangen folgende Produkte zum Einsatz:

- Das Betriebssystem Linux mit der GPL
- Der Webserver Apache mit der Apache License

- Die Datenbank MySQL mit entweder der MySQL Open Source License oder der MySQL Commercial License
- Die Skriptsprache PHP mit der PHP License oder Perl mit der GPL und der Artistic License

Das Beispiel zeigt das Problem eindrücklich auf. Ein Nutzer dieser Konfiguration müsste also mindestens vier Lizenzen prüfen, um sicherzugehen, dass er keine Bedingungen verletzt und dafür entsprechende Konsequenzen zu gewärtigen hätte. Der Wildwuchs der Lizenzen scheint beinahe ungebrochen. Die Liste der durch die OSI gemäss ihren Regeln zertifizierten OSS Lizenzen enthält aktuell fast 60 Einträge und sogar die OSI hat das Problem erkannt und spricht von einer „License Proliferation“, der sie durch entsprechende Empfehlungen entgegenzutreten möchte.

Jeder Autor hat sicher das Recht, seine Urheberrechte durch die entsprechend passende Lizenz schützen zu wollen. Der OSS-Bewegung als gesamtes erweist er jedoch einen Bärendienst, indem er zu den 60 bestehenden noch einmal eine neue Lizenz hinzufügt. Die Gefahr, potentielle Nutzer von OSS durch die schiere Anzahl und die Unübersichtlichkeit der Lizenzen vom Einsatz abzuschrecken, überwiegt den Nutzen bei weitem.

Glücklicherweise gibt es zurzeit Tendenzen, die diesen Dschungel ein bisschen lichten wollen. So hat ein Vizepräsident von HP auf der amerikanischen Linux-world in San Francisco kürzlich für den Einsatz von Standardlizenzen wie der GPL plädiert. Sun Microsystems hat seinerseits kürzlich im Zusammenhang mit seinem Engagement für OpenOffice.org auf ihre eigene OSS-Lizenz zugunsten der LGPL verzichtet, zugleich aber leider die Veröffentlichung des Quellcodes von Solaris wiederum unter einer neuen eigenen Lizenz durchgeführt.

Multi-licensing

Ein Urheber kann durchaus sein Produkt unter mehr als einer Lizenz verbreiten, typischerweise einer OSS- und einer kommerziellen Lizenz, was dem so genannten Dual-Licensing entspricht. Dabei muss der Lizenzgeber natürlich über das Urheberrecht des gesamten Werks verfügen. Dadurch wird einerseits gewährleistet, dass Weiterentwicklungen des OSS-Teils weiterverbreitet werden und andererseits kann der Lizenzgeber der kommerziellen Lizenz seine Investitionen in die kommerzielle Version damit abdecken und Gewinn erzielen.

Diese Modelle setzen natürlich voraus, dass die Community des OSS-Produkts dem Lizenzgeber der kommerziellen Version genügend Vertrauen entgegenbringt und von ihm auch entsprechende Gegenleistungen erwarten kann. So wird erwartet, dass der Gewinn des kommerziellen Produkts zumindest teilweise für die Weiterentwicklung der OSS-Version eingesetzt wird. Ausserdem wird eine gewisse Marketingleistung für die OSS-Version erwartet und die Pflege der Community in form von Konferenzen und Diskussionsforen muss ebenfalls intensiv und qualitativ hoch stehend erfolgen.

Als gutes Beispiel in dieser Hinsicht kann sicher IBM mit dem Projekt Eclipse herangezogen werden. Eclipse wurde von IBM im Jahr 2001 an die OSS-Community übergeben. IBM hatte bis dahin Entwicklungsleistungen im Umfang von 40 Mio. US-Dollar in Eclipse investiert und einige ihrer erfahrensten Entwickler weltweit am Projekt arbeiten und es weltweit vermarkten lassen. IBM verwendet Eclipse auch für seine kommerziellen Entwicklungsprodukte und spart dadurch eine Menge an internen Ressourcen ein, die die Entwicklungs- und vor allem die Test- und Dokumentationsarbeit der OSS-Community übernehmen müssten.

Einsatz von Open Source Software in der Schweiz

Das Beratungsunternehmen Cambridge Technology Partners hat 2003 eine Umfrage bei den IT-Entscheidern von 55 grossen Schweizer Unternehmen und Institutionen durchgeführt. Die Ergebnisse dieser Umfrage liegen in [CTPOSSSchweiz03] vor und liefern, obwohl die Umfrage bereits zwei Jahre zurückliegt, für diese Arbeit sehr wichtige Erkenntnisse.

Es wird sehr schnell ersichtlich, dass viele der befragten Entscheider OSS ein enormes Potential einräumen und dass sie überzeugt sind, damit einige ihrer brennendsten Herausforderungen lösen zu können. Sie nannten in ihren Antworten folgende Themen als die grössten Herausforderungen ihrer IT zurzeit:

- Kostendruck (85% der Nennungen)
- Sicherheit (79%)
- Ressourcenengpässe (58%)
- Abhängigkeit von Lieferanten (56%)
- Neue Produkte / Geschäftsfelder (48%)
- Geschäftsprozess Unterstützung (46%)
- Unterstützung neuer Kanäle (31%)
- Lieferanten Integration (31%)
- Fusionen / Reorganisationen (27%)

Interessant für einen allfälligen Einsatz von OSS ist vor allem die Konzentration der Aspekte Kostendruck, Sicherheit und Lieferantenunabhängigkeit in den ersten vier Nennungen. In diesen Bereichen sollte OSS klar Vorteile bringen. Bei den Ressourcenengpässen ist nicht klar herausgearbeitet, ob das Problem primär bei der Verfügbarkeit von entsprechendem Personal liegt oder fehlender Mittel diese zu entlöhen.

Fehlender Support stellt grösstes Risiko dar

Die klar grössten Defizite und Risiken für den Einsatz von OSS lagen gemäss der Studie zum Zeitpunkt ihrer Erstellung im fehlenden oder ungenügenden Support, im fehlenden Zugriff zu den notwendigen Skills und dem fehlenden Vertragspartner, der im Fall eines Scheiterns haftbar gemacht werden könnte.

Die Studie enthält ebenfalls eine Aufschlüsselung des zeitlich geplanten Einsatzes von OSS bei den befragten Firmen entlang des so genannten Applikationsstacks, d.h. entlang einer Achse beginnend beim Betriebssystem und aufsteigend bis in sehr spezifische betriebliche Einsatzgebiete wie beispielsweise CRM. Hierbei fällt auf, dass für OSS die klassischen Bereiche im tieferen Teil des Stacks reserviert zu sein scheinen, d.h. das Betriebssystem (Linux), der Internet Server (Apache HTTP Server), sowie Entwicklungswerkzeuge aller Art. Ebenfalls gut vertreten sind Gateways (wohl meist als Sicherheitskomponenten eingesetzt), Datenbanken und Applikationsserver.

Vertrauen in OSS für Business Software zurzeit noch gering

Je höher man dagegen im Applikationsstack aufsteigt, desto dünner wird die Luft für OSS. Einerseits liegen keine Pläne für den kurzfristigen Einsatz vor, andererseits traut man OSS hier wohl auch noch nicht richtig zu, entsprechend gute Angebote bereitzustellen. Diese Bereiche betreffen vor allem CMS, Office, Kollaborationswerkzeuge und CRM / ERP.

Gartner sieht in diesem Bereich jedoch in Zukunft eine deutliche Änderung, da gemäss [GartnerHotTopics05] im Jahr 2010 die globalen Top 2000 Organisationen bei einem Viertel ihrer Investitionen in Businesssoftware OSS berücksichtigen werden.

Grenzen von Open Source als Geschäftsmodell

Das OSS Entwicklungsmodell besitzt natürlich auch seine Grenzen. Es scheint Bereiche zu geben, in denen das Vertrauen der Kunden in Open Source Projekte nicht in genügendem Mass vorhanden ist und einer Grossfirma die gewünschte Kontinuität und Investitionsbereitschaft eher zugetraut wird.

Analogie Patentschutz in der Pharmaindustrie

Open Source verhindert durch seine Natur, im Code vorhandene Innovationen und Ideen unter Verschluss halten zu können. Analog der Pharmaindustrie kann man argumentieren, dass nur durch entsprechenden Patentschutz und damit Exklusivität der Nutzung Firmen mit ausreichenden Anreizen ausgestattet werden, damit sie überhaupt noch grosse Investitionen in Innovationen tätigen. Microsoft wird nicht müde, diesen Zusammenhang immer wieder zu betonen und die grosse Zahl der dadurch möglich gewordenen Innovationen aufzuführen.

Durch die Offenlegung der im Softwareprodukt enthaltenen Ideen und Techniken wird einem Konkurrenten der Einstieg natürlich entsprechend vereinfacht, vor allem dann, wenn die verwendete Lizenz keine Veröffentlichung der auf Basis der Open Source Software erstellten Erweiterungen mehr erzwingt. Dieser Nachteil muss durch entsprechende Massnahmen im Marketing und durch die aktive Mitarbeit einer breiten Community wettgemacht werden können.

Aktuelle Beispiele zeigen, dass dies auch bei sehr bekannten OSS-Projekten nicht immer der Fall zu sein scheint. So haben mit Nessus [HeiseNessus05] und Snort [HeiseSnort05] kurz nacheinander zwei prominente und weit verbreitete Projekte aus dem Sicherheitsbereich das Modell gewechselt und werden wohl in Zukunft nicht mehr vom Anbieter als quelloffene Software zur Verfügung gestellt werden.

Fazit

Open Source ist nicht frei im Sinne von kostenlos, sondern frei im Sinne der freien Meinungsäusserung.

Die Verwendung von OSS wird in Lizenzen geregelt, die sich an Standards halten müssen, damit sie von der Open Source Initiative anerkannt werden.

Open Source ist reif für den Einsatz, auch in geschäftskritischen Bereichen. Open Source wird auch bereits sehr breit eingesetzt.

Open Source bietet die Möglichkeit, mit entsprechenden Geschäftsmodellen auch kommerziellen Erfolg zu erzielen. Als Geschäftsmodelle mit Open Source sind nur Dienstleistungen und Appliances als reine Modelle wirklich tragbar. Mischformen mit Dienstleistungen sind jedoch immer möglich. Dadurch wird der Druck auf reine Dienstleister erhöht.

Der rechtliche Rahmen von Open Source ist durch die jeweils verwendete Lizenz gegeben. Diese Lizenzen sind rechtlich verbindlich, wenn auch die Rechtspraxis noch nicht im vollen Masse gegeben ist.

Solange eine Software nur intern verwendet wird und nicht in irgend einer Form weitergegeben wird, kann OSS beliebig eingesetzt werden.

Der als fehlend empfundene Support bremst das Vertrauen in OSS derzeit noch und verhindert in vielen Firmen den Einsatz in geschäftskritischen Bereichen.

2 Literaturverzeichnis / Quellenangaben

- [BerliOS05] Freie Wissensdatenbank OpenFacts von BerliOS. Thema Open Source Software-Lizenzen unter
<http://openfacts.berlios.de/index.phtml?title=Open-Source-Lizenzen>
- [Cathedral00] Raymond, Eric S.: The Cathedral and the Bazaar. Version 3.0
<http://www.catb.org/~esr/writings/cathedral-bazaar/>
- [CIOStudie03] Maas, Wolfgang: CIO-Querschnittsstudie „Die strategische Option Open Source Software“, Institut für Medien- und Kommunikationsmanagement, Universität St. Gallen, November 2003
<http://www.mcm.unisg.ch/im/studies>
- [CTPOSSSchweiz03] Cambridge Technology Partners: Open Source Software in der Schweiz (Umfrageresultate, Hintergründe, Erfahrungen), Oktober 2003.
swissopensourceurvey@ctp.com
- [FhGOSSStudie05] Renner, T.; Vetter, M.; Rex, S.; Kett, H.: Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit, Eine Studie der Fraunhofer-Gesellschaft. Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart, 2005.
- [FOSSDirectory] Verzeichnis der Free- und Open Source Software (= FOSS) Dienstleister und Anbieter in der Schweiz. Erstellt und betrieben durch /ch/open: Swiss Open Systems User Group.
<http://www.foss-directory.ch>
- [GartnerHotTopics05] Cearley, David W.; et. al.: Gartner's Positions on the Five Hottest IT Topics and Trends in 2005. Gartner Research, ID Nr: G00125868, Mai 2005.
- [GartnerPredicts04] Weiss, George J., et. al.: Predicts 2005: Open-Source Software Proliferates, Gartner Research, ID Nr: G00123850, November 2004
- [HeiseGPL04] Meldung vom 23.07.04 auf Heise Online, wonach das OSS Projekt netfilter/iptables vor dem Landgericht München eine einstweilige Verfügung gegen einen kommerziellen Anbieter erlangen konnte.
<http://www.heise.de/newsticker/meldung/49377>
- [HeiseNessus05] Meldung vom 7.10.05 auf Heise Online, dass die Entwicklerfirma des Sicherheits-Scanners Nessus die nächste Version nicht mehr unter der GPL veröffentlichen wird.
<http://www.heise.de/newsticker/meldung/64689>
- [HeiseSnort05] Meldung vom 6.10.05 auf Heise Online, wonach die Entwicklerfirma der Intrusion-Detection Software Snort von der im Securitybereich tätigen Firma Check Point übernommen wird.
<http://www.heise.de/security/news/meldung/64651>
- [JBossLizenz05] Mitteilung auf der Newsseite von entwickler.com über mögliche Lizenzprobleme bei JBoss, 06.09.2005.
<http://www.entwickler.com/itr/news/psecom,id,23934,nodeid,82.html>
- [Leiteritz04] Leiteritz, Raphael: Open Source-Geschäftsmodelle, in: [OSJahrbuch04]
- [LiMuxHome05] Homepage des Projekts LiMux - die IT-Evolution
<http://www.muenchen.de/linux>
- [OpenlawLOTS05] Widmer Mike J, Laux Christian: Handouts zum Workshop „Rechtliche Aspekte beim Umgang mit Open Source Software“ vom 17.02.2005, Universität Bern.
- [OracleOSSBW06] Lacey, S.: Oracle's Open-Source Shopping Spree. Beitrag auf BusinessWeek Online, 9. Februar 2006.
http://www.businessweek.com/technology/content/feb2006/tc20060209_810527.htm
- [OSIHalloween05] Raymond, Eric: The Halloween Documents.
 Online-Sammlung der als Halloween Memoranden bekannt gewordenen internen Microsoft Dokumente von Ende 1998.
<http://www.opensource.org/halloween/>

- [OSIOSSDefinition05] Open Source Initiative: The Open Source Definition, Version 1.9, 2005.
<http://www.opensource.org/docs/definition.php>
- [OSJahrbuch04] Gehring, Robert A.; Lutterbeck, Bernd (Hrsg.): Open Source Jahrbuch 2004, Zwischen Softwareentwicklung und Gesellschaftsmodell. Lehmanns Media, Berlin, 2005.
<http://www.think-ahead.org/>
- [OSJahrbuch05] Bärwolff, Matthias; Gehring, Robert A.; Lutterbeck, Bernd (Hrsg.): Open Source Jahrbuch 2005, Zwischen Softwareentwicklung und Gesellschaftsmodell. Lehmanns Media, Berlin, 2005.
<http://www.opensourcejahrbuch.de>
- [OSSMarktSchweiz05] Arpagaus, Jürg H.; Speck, Christian: Der Schweizer OSS-Markt. Eine Marktanalyse, 2004. Stiftung HfB Hochschule für Berufstätige, Zürich, 2005.
- [WheelerStudy05] Wheeler, David A.: Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! Revised Mai 2005.
http://www.dwheeler.com/oss_fs_why.html
- [ZDNetBlogOS05] Blog auf ZDNet zum Thema Akquisition von Open Source am Beispiel von JasperReports durch den Startup JasperSoft, 25.04.2005.
<http://blogs.zdnet.com/BTL/?p=1306>