



IFLA  
2005  
OSLO

## World Library and Information Congress: 71th IFLA General Conference and Council

### "Libraries - A voyage of discovery"

August 14th - 18th 2005, Oslo, Norway

*Conference Programme:*

<http://www.ifla.org/IV/ifla71/Programme.htm>

Juni 14, 2005

**Code Number:**

**121-G**

**Meeting:**

**134 SI - Information Technology**

Open Source Software – Definition, Lizenzmodelle und organisatorische Konsequenzen (Einführung)

**Reinhard Altenhöner**

Die Deutsche Bibliothek,  
Frankfurt/Main, Germany

Abstract

*In zunehmendem Maße wird die internationale Softwareentwicklung von der Diskussion um das Zauberwort Open Source Software (OSS) bestimmt. Es liegt die Frage nahe, ob dieser Ansatz nicht auch für Bibliotheken relevant sein könnte. Dieser Beitrag beleuchtet im Sinne eines Einstiegs in das Thema daher zunächst generell: Wie entstand die Open Source-Bewegung, welche Vorläufer hatte sie? Welche Bedingungen knüpfen sich an die Verwendung von OSS, welche Ideen stecken hinter der Entwicklung dieses Begriffs? Wie werden bei OSS die Urheber- und Nutzungsrechte geregelt? Und schließlich: Wie sehen Erfolgsfaktoren für die Nutzung von OSS insbesondere in Bibliotheken aus und welche Ansätze gibt es bereits?*

1. Kaum ein Stichwort ist in den vergangenen Jahren so intensiv diskutiert worden wie Open Source Software (OSS). Spätestens als Linus Torvalds 1991 seine Idee von einem frei zugänglichen Betriebssystem ins Netz stellte, wandelte sich „Open Source“ bzw. die Idee der „Free Software“ von einem Hobby für Computerfreaks zu

einer ernstzunehmenden Alternative zur kommerziell orientierten proprietären „closed source“ Software. Dies zeigt sich nicht nur an den zunehmenden Verbreitungszahlen sondern auch an den wachsenden Marktanteilen, die Firmen, die sich auf die Betreuung und Anwendung von OSS-Programmen spezialisieren, erreichen können. Häufig taucht OSS in einem Zusammenhang mit Open Access auf und die Ausgangssituation ist durchaus vergleichbar. In beiden Fällen ist spürbar eine Veränderung im Gang, die darauf hinaus läuft, dass etablierte Vertriebswege für Software und für (elektronische) Dokumente durch neue Angebots- und Nutzungsformen in Frage gestellt und unter Umständen auch in ihrer Marktmacht beschnitten werden. Gleichzeitig gibt es aber auch wichtige Unterschiede, auf die zurückzukommen ist. Beiden Ansätzen gemeinsam ist aber eine enorme Dynamik, die mit der hohen Motivation der beteiligten Handelnden korrespondiert.

2. OSS findet insbesondere bei Versorgern, im Handel, aber auch in der Telekommunikationsbranche und bei der öffentlichen Hand immer weitere Verbreitung. In Deutschland ist der Einsatz von OSS mittlerweile auf Bundesebene ein erklärtermaßen für alle Anwendungen geltendes Prinzip, andersartige Entscheidungen zugunsten proprietärer Softwareprodukte geraten in einen gewissen Rechtfertigungsdruck. In Großbritannien gibt es das von JISC finanzierte OSS Watch, den Open Source Software Advisory Service.<sup>1</sup> Weltweit sollen es mittlerweile mehr als 800.000 Programmierer sein, die ihr Können in die Entwicklung von OSS einbringen. Rund 80.000 OSS-Projekte soll es weltweit geben und sie alle zeugen von der Kreativität und dem Qualitätsbewußtsein einer motivierten Gemeinschaft von Softwareentwickeln. Klangvoll Namen von OSS-Entwicklungen sind Apache, Geronimo, Emacs, Tomcat, Samba, Python, Eclipse, Gnome, Wikipedia und Mozilla Firefox, OpenOffice.org, mySQL, Firebird. Das dahinter zum Teil auch Firmen stecken, die der Entwicklergemeinde ihr Produkt als Source zur Verfügung stellen, ist dabei kein Schönheitsfehler, sondern Prinzip.

3. Eng verflochten mit der Begrifflichkeit OSS ist der Terminus „Freie Software“, in philogischer Perspektive ist er die ältere und nach wie vor gültige Bezeichnung, die allerdings durch den PR-Erfolg des Begriffs „Open Source“ seit 1998 etwas in den Hintergrund getreten ist. Als Reaktion auf die Kommerzialisierung der UNIX-Entwicklung entstand vor bereits mehr als 20 Jahren das GNU-Projekt („GNU is Not Unix“. Unter der Bezeichnung „Free Software“ galten folgende Leitbedingungen für die Weiternutzung und Weiterentwicklung<sup>2</sup>:

- a) Unbegrenzte Nutzung für den eigenen Zweck
- b) Studium der Funktionsweise für Anpassungen an eigene Bedürfnisse
- c) Weitergabe durch Kopie
- d) Verbesserung und deren Veröffentlichung

Kernanforderung ist die Verfügbarkeit des Quelltextes der Software in einer nachnutzbaren Form. 1985 wurde außerdem die Free Software Foundation (FSF)

---

<sup>1</sup> <http://www.oss-watch.ac.uk/>

<sup>2</sup> Nach Reiter, Bernhard E.: Wandel der IT: Mehr als 20 Jahre Freie Software. In: HMD, Heft 238, August 2008, S. 83 – 91. Zitiert nach der Web-Fassung, s. [http://intevation.de/~bernhard/publications/200408-hmd/200408-wandel\\_der\\_it\\_20j\\_fs.html](http://intevation.de/~bernhard/publications/200408-hmd/200408-wandel_der_it_20j_fs.html). Vgl. ferner auch O'Reilly & Associates: Open Source - kurz & gut. Nach der deutschen Übersetzung von Snoopy & Martin Müller. 1. Auflage April 1999 [http://www.oreilly.de/german/freebooks/os\\_tb](http://www.oreilly.de/german/freebooks/os_tb).

gegründet, die quasi als Vertriebsorganisation fungierte. Schon an den genannten Kriterien wird deutlich, dass im Vordergrund die Freiheit der Softwarenutzung steht, nicht deren Kostenlosigkeit. Die FSF fordert sogar dazu auf, Geld zu verlangen, allerdings nicht für die Software selbst, sondern die Dienstleistung um sie herum.

Unter dem Namen „Open Source“ versuchte ab 1998 die Open Source Initiative (OSI), mit freier Software größeres Interesse zu erzielen. Dazu wurden die „Debian Free Software Guidelines“ adaptiert, die unter technischen Gesichtspunkten erläutern, was „Freie Software“ ausmacht.

4. Was macht nun den Erfolg<sup>3</sup> der OSS aus, der Analysten zu der Aussage brachte, 2007 werde bereits jede 4. Firma OSS einsetzen.<sup>4</sup>

- Offenheit. Die in der Regel offen stattfindende Entwicklung entspricht der akademischen Tradition, Ergebnisse der (wissenschaftlichen) Arbeit unmittelbar auszutauschen, Forschungsdaten bereitzustellen usw.. Insofern schließt diese „Regel“ an den Erfahrungshorizont vieler Menschen an, nutzt eingeschlifene Kommunikationswege und Kooperationsverfahren.
- Flexibilität. Viele OSS-Projekte integrieren eine große Zahl mitarbeitender Entwickler mit sehr unterschiedlichen Schwerpunkten und Hintergründen. Dies erleichtert das Verständnis für spezielle Anforderungen und bietet gleichzeitig auch das Potential, erforderliche Anpassungen schnell durchzuführen.
- Schnelligkeit. Die Geschwindigkeit, mit der auf in der Software auftauchende Probleme, Fehler oder Sicherheitslücken reagiert wird, ist legendär. Eine große Gruppe von Personen, die ein Produkt zum Erfolg bringen will und die die Aufgaben unmittelbar angeht und neue Versionsstände testet, ist häufig wesentlich schneller und erfolgreicher als die proprietäre „Konkurrenz“.
- Motivation. OSS-Entwickler sind / erleben sich als Teil einer Gemeinschaft, die an einem kollaborativen Erfolg(sprodukt) arbeitet. Häufig sind es übrigens nicht nebenberufliche Hobbyentwickler, sondern professionelle Entwickler, die häufig im Hauptberuf OSS erstellen.<sup>5</sup> Auch durch die Grundregel bei öffentlich finanzierten Projekten die entstandene Software anderen kostenfrei zur Verfügung zu stellen (was nicht unbedingt OS oder Free Software bedeuten muss), fördert die Motivation.
- Standards. Das Aufsetzen auf häufig international verankerten (Internet)-Standards sichert in der Regel ein größeres Maß an Unabhängigkeit von einem einzelnen Lieferanten. Die Zugänglichkeit des Codes erlaubt es zumindest theoretisch, anstehende Aufgaben in andere Hände zu übergeben

---

<sup>3</sup> Vgl. hierzu auch den grundsätzlichen Aufsatz von Eric S. Raymond: The Cathedral and the Bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>, in dem er unter anthropologischen Gesichtspunkten die Bewegung analysiert.

<sup>4</sup> So das Marktforschungsinstitut Soreon Research, vgl. Kassensturz: Open-Source und proprietäre Software im Vergleich (2003), [http://www.soreon.de/site1/index.php/german/soreon\\_studien/software\\_hardware/kassensturz\\_open\\_source\\_und\\_propriet\\_re\\_software\\_im\\_vergleich](http://www.soreon.de/site1/index.php/german/soreon_studien/software_hardware/kassensturz_open_source_und_propriet_re_software_im_vergleich) und den Update 2004 unter [http://www.soreon.de/site1/index.php/german/soreon\\_studien/software\\_hardware/kassensturz\\_open\\_source\\_und\\_propriet\\_re\\_software\\_im\\_vergleich\\_update\\_2004\\_95\\_seiten\\_31\\_abbildungen\\_und\\_tabellen](http://www.soreon.de/site1/index.php/german/soreon_studien/software_hardware/kassensturz_open_source_und_propriet_re_software_im_vergleich_update_2004_95_seiten_31_abbildungen_und_tabellen).

<sup>5</sup> Nach einer Studie der Boston Consulting Group werden 40% der existierenden freien Software so entwickelt, s. Reiter, a.a.O..

– wenn gleich dies in der Praxis zumeist wenig realistisch ist. In einer erweiterten Perspektive gilt dies auch für die langfristige Les- und Nutzbarkeit der Software, weil der offene, auf Standards setzende Ansatz bei den Entwicklern von vornherein die erforderliche Nachhaltigkeit berücksichtigt.

5. Der Autor einer Software bestimmt über ihre Nutzung.<sup>6</sup> Selbst wenn alle genannten Nachnutzungen erlaubt sind, liegt eine Lizenz vor, die beachtet werden muss. Nur bei der „public domain“-Software wird ausdrücklich auf eine Urheberschaft verzichtet und es gelten nicht einmal die üblichen urheberrechtlichen Beschränkungen.<sup>7</sup> Auch in diesem Fall aber hilft die „Lizenz“, die rechtliche Situation rund um Nutzung und Vertrieb eines Softwareprojekts zu klären; sie hilft so, Unsicherheiten und Unklarheiten zu vermeiden.

Die Diskussion wird maßgeblich von den Debatten rund um die Definition der Free Software Foundation (FSF) und der der Open Source Initiative (OSI), die in der Open Source Definition (OSD) niedergelegt wurde, bestimmt.<sup>8</sup> Obwohl die jeweiligen Formulierungen eng beieinander liegen, legt die FSF in ihrem durchaus antikommerziellen Impetus strengere Maßstäbe an, als die OSD.<sup>9</sup> Das hat auch damit zu tun, dass die FSF und ihr Nestor Richard Stallman auch prinzipielle Überlegungen zu gesellschaftlichen Auswirkungen der Softwareentwicklung und ihrer Methoden einbeziehen, während die OSD technische und methodische Aspekte in den Vordergrund stellt, letztlich also deutlich pragmatischer mit dem Thema umgeht.<sup>10</sup> Kernpunkte der OSD sind:

- Freie Weiterverbreitung
- Zugänglichkeit des Quellcodes
- Veränderbarkeit des Codes und Weiternutzung in neuer Software
- Unverletzlichkeit des Originalcodes.<sup>11</sup>
- Keine Diskriminierung einzelner Personen und Gruppen
- Keine Einschränkungen für bestimmte Anwendungsbereiche (insbesondere Restriktionen zum kommerziellen Sektor)

---

<sup>6</sup> Vgl. hierzu auch St.Laurent, Andrew M.: Understanding Open Source and Free Software Licensing. Sebastopol, CA (O'Reilly): 2004. Vgl. Online-Version unter <http://www.oreilly.com/catalog/osfreesoft/book/>.

<sup>7</sup> Nicht näher betrachtet wird hier die haftungsrechtliche Seite der Lizenz, obwohl auch sie einen gewichtigen Gesichtspunkt darstellt und eine der ersten Lizenzen, die Berkeley Software Distribution (BSD) ganz besonders auf den Haftungsausschluß abhebt. In der Praxis haben entsprechende Haftungsregeln für proprietäre Software häufig nicht die gewünschte Wirkung; im Bereich der Freien Software dagegen entstehen oft sehr schnell Lösungen für bugs oder change requests, s.a. unter Anm. 4.

<sup>8</sup> <http://opensource.berlios.de/docs/definition.php>. Vgl. a. den Artikel Open Source Definition, [http://de.wikipedia.org/wiki/Open\\_Source\\_Definition](http://de.wikipedia.org/wiki/Open_Source_Definition).

<sup>9</sup> vgl. dazu die Diskussion in den einschlägigen Foren und auch in wikipedia zu open source.

<sup>10</sup> Ein gutes Beispiel ist sicher Linus Torvalds, der mit kommerziellen Verwertern der LINUX-Entwicklung völlig ungezwungen umgeht, solange Entwicklungsergebnisse auch wieder der Entwicklergemeinschaft zugute kommen. S. a. Interview mit L.T. von Hiro Yamagata unter dem Titel „Der Pragmatiker der freien Software“ in O'Reilly & Associates: Open Source - kurz & gut, a.a.O.

<sup>11</sup> Hier geht es vor allem um Korrekturen / Änderungen von Original-Software. Der Nutzer soll genau wissen, wer welchen Stand verantwortet bzw. welchen genau voraussetzt.

- Verbreitung der Lizenz (keine Weitergabe mit neuen Regeln!)
- Lizenz darf nicht nur für ein bestimmtes Produkt gelten (z.B. als Teil einer Softwaredistribution)
- Lizenz darf andere Software nicht beeinträchtigen (die z.B. auch auf dem Datenträger enthalten ist, Disclosure Agreements)

Die OSI hat versucht, auf diesen Grundsätzen basierend Lizenzen zu akkreditieren; die Akzeptanz dieses Modells ist aber insgesamt deutlich geringer ausgefallen, als seine Macher erhofft hatten. So ist der Verbreitungsgrad der GNU-Lizenz noch immer überragend.<sup>12</sup> Versucht man die verschiedenen Typen von Lizenzen für freie Software zu kategorisieren, kann man vier Gruppen bilden:<sup>13</sup>

a) GNU General Public License (Version 1: 1989) Kernpunkt ist, dass bei Weitergabe einer veränderten Software an Dritte diese wieder mit Quelltext unter GNU GPL zur Verfügung gestellt werden muss.

b) GNU Lesser Public License (GNU LGPL) stellt eine schwächere Ausgabe dar, weil die Software als Baustein für proprietäre Software genutzt werden kann, zwar als Baustein frei bleibt, nicht aber das entstandene Konstrukt.

c) X11 und andere ähnliche Lizenzen gewähren keinen Schutz. Die Weitergabe einer entwickelten Software kann ohne Quelltext und Freiheiten erfolgen, die Software kann als Grundstein für proprietäre Software verwandt werden.

d) eine weitere Kategorie bilden die „andersartigen“ Lizenzen, die zur GNU GPL inkompatibel sind: Ein Unternehmen gibt eine Software auch unter proprietärer Lizenz weiter, externe Entwickler erhalten dieses Recht aber nicht, sondern müssen es abgeben, wenn ihr Quelltext integriert werden soll.

6. OSS liefert in der konkreten Situation eine Grundlage, die angepasst werden kann/muss. Das würde bedeuten, dass nur große Einrichtungen, die über eine eigene Entwicklungsabteilung verfügen, erfolgreich OSS einsetzen können. Tatsächlich aber gibt es inzwischen eine breite Palette von Firmen (man vergleiche etwa die Linuxdistributionen), die Service und ergänzende Entwicklungen rund um die OSS anbieten. Erst in dieser Kombination wird der Einsatz von OSS zum Erfolg. Das haben auch große Firmen wie IBM oder Novell verstanden, die einerseits wichtige Softwareprodukte als Open Source zur Verfügung stellen, andererseits aber durchaus auch weiterhin kommerziell ausgerichtete proprietäre Software entwickeln, die sie als ergänzende Teilkomponenten zum „freien“ Grundprodukt verstehen. Zur Frage möglicher Kostenvorteile gibt es eine kontroverse Diskussion, bislang aber wenig konkrete Studien. Es zeichnet sich allerdings ab, dass - über einen längeren Zeitraum betrachtet - beim Einsatz von OSS – nach gegenüber proprietärer Software deutlich geringeren Einführungskosten - ein erhöhter Beratungs- und Anpassungsbedarf besteht, der auch zu vergleichsweise höheren Kosten führt. Auf einer mittleren Zeitspanne (oberhalb 3 Jahre) allerdings sinkt das Kostenniveau deutlich unter das für proprietäre Software.

<sup>12</sup> In Zahlen: GPL, LGPL und unmittelbar verwandte: über 50.000 Projekte, vgl. Sourceforge.net, [http://sourceforge.net/softwaremap/trove\\_list.php?form\\_cat=14](http://sourceforge.net/softwaremap/trove_list.php?form_cat=14)

<sup>13</sup> Vgl. Reiter, a.a.O.

7. Bibliotheken setzen an zahlreichen Stellen IT ein und sind damit potentielle Nutzer von OSS – Beispiele dafür gibt es genug. Aber auch spezielle OSS für Bibliotheken ist inzwischen entstanden, als bekannteste sicher Koha und Greenstone, die beide ihren Ausgangspunkt in Neuseeland haben. Weitere Beispiele werden in den Vorträgen dieser Session benannt und hier nicht weiter ausgeführt. Die Tatsache jedenfalls, dass es bereits eine eigene Web-Site zum Thema gibt, zeigt wie rasant sich das Thema entwickelt hat.<sup>14</sup>

8. Häufig wird inzwischen parallel zur “Open Source”-Bewegung „Open Access“ gesehen und diskutiert. Mittlerweile ist klar geworden, welches Potential in diesen Ansätzen steckt. Von besonderer Wichtigkeit ist dabei die Lizenzierung, d.h. die bewusste Zuweisung von (offenen) Nutzungsrechten zu einer Software oder zu einem Dokument; dabei hat die Frage der Veränderbarkeit eines Textes besonderes Gewicht. Erst diese klaren Regelungen schaffen die erforderliche Akzeptanz, die offensichtlich in der Softwareentwicklung schon weiter verbreitet ist, als im Kontext des wissenschaftlichen Publizierens. Auch für Bibliotheken ist die Bewegung wichtig; wenn gleich hier offenbar bislang eher der Gedanke des Open Access bekannt ist als der der OSS.

---

<sup>14</sup> <http://www.oss4lib.org>.