Internet-Ökonomie

“Open Source”

Stephan Uhlmann <su@su2.info>

31.08.2003

Inhaltsverzeichnis

1 Einführung

Diese Ausarbeitung zum Vortrag "Open Source" entstand im Rahmen der Veranstaltung "Internet-Ökonomie" an der Universität Potsdam. Es werden verschiedene wirtschaftliche Aspekte von Open Source Software beleuchtet. Als allgemeinen Lesestoff zu diesem Thema empfehle ich die Literatur, auf die im Anhang verwiesen wird.[1, 2, 3, 4, 5, 6]

Als kurze Begriffserklärung vorab einige Erläuterungen der wichtigsten Philosophien aus dem Open Source Umfeld:

- Open Source: Software, bei der der Quellcode frei verfügbar und veränderbar ist. Programme die darauf basieren, müssen jedoch nicht wiederum im Quellcode vorliegen, wenn sie veröffentlicht werden. Diese so genannten Derivate können also wieder Closed Source werden.[7]
- freie Software: Auch hier ist der Quellcode frei verfügbar und veränderbar. Im Gegensatz zu Open Source stellt jedoch eine entsprechende Lizenz sicher, dass bei der Veröffentlichung von Derivaten, der Quellcode wieder offen gelegt werden muss. "Frei" ist hier also nicht im Sinne von kostenlos, sondern in der Bedeutung von "Freiheit" gemeint. Freie Software ist eine Teilmenge von Open Source Software und ist im folgenden immer mit gemeint.[8]
- Shared Source: Software ist im Quellcode einsehbar, darf aber nicht verändert oder weiterverbreitet werden. Vorher muss ein so genanntes Non-Disclosure-Agreement unterschrieben werden. Dies ist weniger eine Philosophie, sondern ein Marketingbegriff aus der PR-Abteilung von Microsoft, der ihre Lizenz als Alternative zu Open Source erscheinen lassen soll.[9]
- Closed Source: Die Software ist nicht im Quellcode verfügbar. Daher kann das Programm auch nicht verändert werden. Sie wird auch "proprietäre Software" genannt.

2 Open Source Projekte vs. gewinnorientierte Unternehmen

2.1 Arbeitsweise & die Rolle des Internets

2.1.1 Open Source Projekte

Open Source Projekte sind entgegen oft verbreiteter Meinung nicht chaotisch organisiert sondern meist sogar hierarchisch strukturiert. Es gibt einen so genannten Maintainer, der das gesamte Projekt koordiniert und sich dafür verantwortlich fühlt. Meist ist das der Gründer des Projektes oder aber derjenige, der am meisten zum Projekt beiträgt. Bei komplexerer Software kann es für verschiedene Teilbereiche eigenständige Verantwortliche geben, so zum Beispiel zur Koordination der Übersetzungen oder der Dokumentation. Die Hierarchie insgesamt ist aber nur sehr flach. Wie z.B. beim KDE-Projekt muss es auch keinen übergeordneten Maintainer geben. Dann wird eine Person für einen befristeten Zeitraum bestimmt, die sich um die Abwicklung von Releases (Veröffentlichung von Meilensteinen) und die Einhaltung von selbst gesteckten Zeitplänen kümmert.

Das Internet ist von überlebenswichtiger Bedeutung für Open Source Projekte. Die Mitarbeiter sind global verteilt. Die Kommunikation untereinander (Email), der Vertrieb der Software (FTP-Server) und das Marketing (Website) funktionieren nur durch das Internet so direkt, global und kostengünstig wie es für diese Projekte notwendig ist.

Diskussionen werden auf technischer Ebene geführt. Es wird versucht immer die beste Lösung zu implementieren. Persönliche Anfeindungen und "Ego-Trips" sind verpönt und allgemein der beste Weg seine Reputation in der Gruppe zu verlieren. Man konzentriert sich auf die Überlegenheit des "Produktes" und muss sich nicht um davon ablenkende Ziele, wie z.B. time-to-market, Verdrängung der Konkurrenz oder reine Marketing-Gags wie sie in Unternehmen eine Rolle spielen, kümmern. Im Gegenzug kann ein Open Source Projekt an sich keine Garantien bzgl. von Release-Zeiten, Bugfixes oder Supportanfragen bieten. Hier gibt es jedoch inzwischen auch Unternehmen, die kommerziellen Support für Open Source Software anbieten.

Bei größeren Projekten wird oft ein Verein gegründet, der eine rechtliche Dachorganisation bildet (z.B. der KDE e.V.).

2.1.2 Unternehmen

Kommerzielle Unternehmen sind in der Regel von stark hierarchischen Strukturen geprägt. Mitarbeiter arbeiten meist lokal an einem Arbeitsplatz. Davon gibt es sicher auch Ausnahmen, wie z.B. "Klarälvdalens Datakonsult" oder "the Kompany" die sehr stark global verteilte Mitarbeiter haben. Nicht zufällig entstammen diese jedoch aus dem Umfeld von Programmierern freier Software.

Das Internet ist für viele Firmen wichtig, für die wenigsten jedoch überlebenswichtig. Dies hängt jedoch stark von der Branche ab. In der Regel können Unternehmen aber zur Not noch mit traditionellen Kommunikationsmitteln wie Telefon, Fax und Briefpost auskommen, was bei Open Source Projekten nicht möglich ist.

Die laufenden Kosten für die Gehälter der Mitarbeiter, Miete usw. zwingen zu wirtschaftlichem Arbeiten und der Notwendigkeit Gewinn zu erzielen. Primäre Ziele sind Gewinnmaximierung, neue Kunden zu gewinnen und diese zu behalten sowie strategische Ziele am Markt.

2.2 Motivation der Beteiligten

2.2.1 Entwickler von Open Source Software

Ein Open Source Projekt entsteht oft dadurch, dass eine Person mit der Software für sich selber ein konkretes Problem lösen möchte. Zusätzliche Quellen der Motivation, können der Wunsch etwas Neues zu lernen sein. Nicht zu vernachlässigen ist auch die Anerkennung, die einem widerfährt, wenn man merkt, dass eine Software von anderen Menschen auf der Welt benutzt und gebraucht wird. Diese Verbindung zwischen der Software und der eigenen Reputation spornt wiederum an, möglichst qualitativ gute Software zu veröffentlichen. Mitarbeiter an besonders großen und prestigeträchtigen Open Source Projekten können damit z.B. auch ihren Lebenslauf aufwerten. Viele Firmen erkennen diese Mitarbeit sehr wohlwollend an bzw. stellen gezielt Entwickler eines Projektes an. Exzellentes technisches Know-How, Teamfähigkeit, Managementqualitäten lassen sich auch am Erfolg eines Projektes ablesen. Für viele professionelle Softwareentwickler ist der Antrieb aber auch gerade die Auslebung eigener Kreativität, abseits der Einschränkungen bei der täglichen Arbeit.

2.2.2 Anwender von Open Source Software

Wichtigstes Argument für Open Source Software scheint meist das Fehlen von Lizenzkosten zu sein. Das ist sicher für viele ein wichtiger Aspekt, doch es gibt andere, für manche viel wichtigere Gründe Open Source Software einzusetzen. Der grosse Unterschied ist ja, dass der Quellcode einsehbar, modifzierbar und frei verteilbar ist.

Die im Programm implementierten Verfahren sind jederzeit auf ihre Korrektheit und Sicherheit überprüfbar. Sicherheitslöcher werden meist schnell erkannt und können im Notfall selber direkt im Quellcode behoben werden, auch wenn noch keine neue Version veröffentlicht wurde. Man ist dadurch nicht von der Freigabe neuer Versionen abhängig.

Als Beispiel, dass dies gar nicht so weit hergeholt ist, sei hier ein Fall aus meiner persönlichen Praxis erwähnt. Für ein Marktplatzsystem auf einem Webserver der Firma, für die ich arbeite, wird die Skriptsprache PHP eingesetzt. Dort wurde nun ein Sicherheitsloch bekannt, welches in einer neueren Version behoben wurde. Wir konnten diese neue Version jedoch wegen bestimmter Seiteneffekte nicht einsetzen. Da es für uns selber zu lange gedauert hätte, den Bugfix in die ältere Version einzupflegen, haben wir einen PHP-Entwickler engagiert, der das kurzerhand erledigt hatte. Mit proprietärer Software wäre das Problem nicht so leicht zu lösen gewesen.

Die Verfügbarkeit des Quellcodes sichert daher auch die Wartbarkeit der Software über das bestehen des Projektes hinaus. Dieser Investitionsschutz ist bei Closed Source Software nicht gegeben. Verschwindet z.B. das Unternehmen, dass die Software entwickelt hat, dann gibt es keine Möglichkeit die Software durch andere Programmierer weiter pflegen zu lassen.

Die Chance, dass im Programm für den Anwender unerwünschte Routinen versteckt sind, wie z.B. so genannte Backdoors oder Spyware, kann nahezu ausgeschlossen werden. Sowas wäre nur schwer bis gar nicht möglich versteckt zu halten. In besonders sicherheitskritischen Anwendungsgebieten (staatliche Einrichtungen) ist das sicher ein entscheidendes Kriterium.

Auch wenn nicht jeder den Quellcode nun auch verändern kann oder möchte, impliziert diese allgemeine Offenheit doch Vorteile, die proprietäre Software meist nicht bietet. Open Source Projekte bieten z.B. die Möglichkeit in direkten Kontakt mit den Entwicklern zu treten. Feedback zu geben und Fehler zu melden geben das Gefühl etwas "zurückzugeben" und so zum Projekt beizutragen.

2.2.3 Akteure in einem Unternehmen

Zur Vervollständigung des Bildes sei hier noch die typische Motivation für Arbeitnehmer, Geschäftsführer, Investoren usw., sich an einem Unternehmen zu beteiligen, erwähnt: primäres Ziel ist es Geld zu verdienen. Einige verfolgen aber sicher auch die eigene Selbstverwirklichung.

3 Open Source im Unternehmen

3.1 Alles anders?

Für Unternehmen als Anwender von Software spielt die Art der Software selten eine Rolle. Es kommt darauf an, dass eine bestimmte Aufgabe erledigt wird. Daher verändert die Entscheidung, Open Source Software einzusetzen oder zumindest in den Auswahlprozess mit einzubeziehen, normalerweise gar nichts. So erklärt sich z.B., dass im Bereich der Webserver-Software, wie sie als Basis für E-Commerce benötigt wird, der Apache einen Anteil von 63% erlangt hat[10]. Ebenso ist das freie Betriebssystem GNU/Linux im Backend, auf Servern, z.B. bei ISPs oder auch im Embedded-Bereich, weit verbreitet. Dort spielt hauptsächlich die Eignung der Software für die jeweilige Aufgabe eine Rolle.

Etwas anders sieht es bei Unternehmen aus, die selber Softwareentwicklung betreiben. Will man hier vom bisherigen Ansatz proprietärer Software abrücken, bedarf es anderer Geschäftsmodelle, diese möglichst gewinnbringend zu vermarkten. Insbesondere können Märkte, die bisher unprofitabel waren, einen speziellen Bedarf am Quellcode haben oder zum Prestigeaufbau geeignet sind, angegangen werden. Verschiedene Geschäftsmodelle werden im folgenden beschrieben.

3.2 Geschäftsmodelle

3.2.1 Software als Produkt

Auch Open Source Software lässt sich als Produkt vermarkten. Beispiele dafür sind die vielen kommerziellen Linux-Distributionen wie SuSE, RedHat oder Mandrake. Sie verpacken den Linux-Kernel als Betriebssystem, sowie etliche Open Source Software zu einem Komplettpaket. Dazu gibt es dann komfortable Installations-Software, Handbücher usw. Diese Distributoren arbeiten zwar meist nur kostendeckend, versuchen dann aber über zusätzliche Dienstleistungen Gewinn zu machen.

Ein weiteres interessantes Beispiel ist die Firma Trolltech. Diese verkauft ihre C++-Klassenbibliothek “Qt”, die auch als freie Software unter der GPL-Lizenz veröffentlicht wird. Unter anderem basiert die KDE-Desktopumgebung auf der Qt. Möchte nun eine Firma die Qt für ein proprietäres Produkt verwenden, so wäre dies nicht möglich, da die GPL-Lizenz das verbietet. Trolltech verkauft die Qt jedoch zusätzlich unter einer anderen, nicht-GPL-Lizenz, bei der der auf Qt basierende Quelltext nicht im Quellcode verfügbar gemacht werden muss. Als Autor und alleiniger Inhaber des Copyrights ist ihnen dieses “Dual-Licensing” erlaubt. Dies ermöglicht beides miteinander zu vereinen: für die einen freie Software, für die anderen proprietäre Software. Je nachdem was gewünscht wird.

3.2.2 kostenlose Lock-Produkte / Marktpositionierung

In weniger profitablen Märkten verteilen Unternehmen ihre Software oft als kostenlose Lockprodukte. Sie erhoffen sich, durch darauf aufbauende Geschäfte Gewinn zu machen. Oft wird auch versucht, sich in einem neuen Marktsegment erstmal möglichst schnell einen großen Marktanteil zu sichern. Software, die in diese Kategorien fällt, kann ohne Probleme zu Open Source gemacht werden.

Beispiele sind hier der Browser Netscape, jetzt Mozilla, der durch die Firma Netscape 1998 als Open Source freigegeben wurde, nachdem das langjährige Monopol durch den Microsoft Internet Explorer ins Wanken geriet und sich das Geschäft für Netscape nicht mehr rechnete. Ein weiteres Beispiel ist Sendmail, eine Software für Mailserver. Hier ist das Basisprodukt Open Source und bestimmte Zusätze können von der Firma Sendmail Inc., eine Gründung der Kern-Entwickler, erworben werden.

3.2.3 Dienstleistungen

Rund um Open Source lassen sich natürlich auch die typischen Dienstleistungen wie Support, Training und Consulting gewinnbringend vermarkten. In Deutschland gibt es viele solcher Firmen, wie z.B. LinuxCare, SuSE, Gonicus (ehem. ID-PRO), Bytec, innominate, um nur einige zu nennen.

3.2.4 Hardware-Zugaben

Hersteller von Hardware veröffentlichen oft ihre Treiber und andere Spezialsoftware als Open Source, da diese ohnehin nicht zum Kerngeschäft gehören. Firmen wie IBM oder Cobalt Networks (seit 2000 zu Sun), benutzen auch Open Source Software, um ihre Systeme mit einem vorinstallierten Betriebssystem auszuliefern.

3.2.5 Zubehör

Nicht zuletzt gibt es auch einen Markt an Zubehör wie Bücher, T-Shirts, Plüschtiere etc., die z.B. von der Firma O'Reilly oder Lehmanns Fachbuchhandlung erworben werden können.

3.3 Vorteile & Nachteile & Mythen

Open Source Software hat viele Vorteile aber auch ein paar Nachteile. Es existieren jedoch auch einige "Mythen", die immer noch Meinungen prägen.

Vorteile sind die Möglichkeit zur Einsicht in den Quellcode und eigene Modifikationen daran vorzunehmen. Durch diese Möglichkeit des "peer reviews" wird nicht zuletzt auch die Sicherheit der Software erhöht. Durch die Offenlegung und Verbreitung des Quellcodes kann eine Open Source Software nicht einfach so "verschwinden" und kann jederzeit neu angepasst werden, was einen gewissen Investitionsschutz darstellt. Man ist nicht abhängig von einem Anbieter. Es fallen keine Lizenzkosten an. Es gibt keine Beschränkungen des Einsatzgebietes der Software. Open Source Software ist meist stabiler, da die Entwickler keinem Zeitdruck zu Veröffentlichungen oder anderen marketing-strategischen Zwängen unterliegen. Die so genannte "Bananensoftware", wo eilig eine Beta-Version als fertiges Produkt verkauft wird, um dann erst beim Kunden über Hotfixes und Patches zu "reifen", findet man bei Open Source Software so gut wie nie.

Nachteilig ist oft, dass die Installation von Open Source Software,wenn sie nicht über einen Distributor erworben wurde, etwas komplizierter ist. Unternehmen brauchen daher eher besonders geschultes Personal in ihrer Systemadministration, was sich natürlich auf die so genannte TCO (Total Cost of Ownership) auswirkt.

Eines der Mythen ist, dass es zu Open Source Software keinen Support gibt. Neben den vielen Firmen, die professionellen Support als Dienstleistung anbieten, leistet jedes Open Source Projekt natürlich Hilfestellung zu ihrer Software. Man hat direkten Kontakt zum Entwickler und es gibt oft einen großen und kompetenten Nutzerkreis, die sich über das Internet organisiert, gegenseitig helfen. Als einziger Nachteil wäre hier vielleicht die fragmentierte Struktur des Supports zu nennen. Es gibt verschiedene Möglichkeiten, wie ein Projekt kommuniziert, z.B. Mailinglisten, IRC, Foren oder Bugtracker-Formulare auf der Website.

Ein weiterer Mythos ist die Annahme, dass Open Source Software völlig unkontrolliert und chaotisch entwickelt würde. Dies ist jedoch nicht der Fall. Der Maintainer eines Projektes, ist als “wohlwollender Diktator” der Koordinator des Projektes und fällt Entscheidungen zusammen mit den anderen Team-Mitgliedern. Sobald mehrere Entwickler an einem Projekt arbeiten, wird auch ein Versionskontrollsystem eingesetzt, dass die einzelnen Änderungen leicht nachvollziehbar macht. Durch den “peer review”, also das gegenseitige Begutachten und Testen der Änderungen der anderen, wird auch verhindert, dass jemand einfach irgendeinen Quatsch zum Quellcode hinzufügt.

Unternehmen, die Open Source Software einsetzen wollen, fragen oft nach Rechtsverbindlichkeiten. Diese sind zugegebener Maßen nicht vorhanden. Dafür sorgen schon die verschiedenen Open Source Lizenzen, die den Entwickler auch vor Haftung schützen. Gleiches gilt jedoch auch für proprietäre Software. Microsoft würde niemals Schadensersatz für die Umsatzeinbußen bezahlen, die einem E-Commerce-Unternehmen entstanden sind, weil jemand durch ein Sicherheitsloch im Internet Information Server in ihren Webserver eingebrochen hat.

4 Vergangenheit, Gegenwart, Zukunft

4.1 Der Ursprung und der große Hype

Der Ursprung von Open Source liegt in der Kultur der Kultur des Teilens. In den Anfangstagen der Computer war es üblich, Software untereinander auszutauschen und sich gegenseitig zu helfen. Sozusagen von Hacker für Hacker. (Der Begriff “Hacker” wird übrigens weitläufig missverstanden. Gemeint sind enthusiastische Computer-Freaks und nicht Leute, die in fremde Systeme einbrechen. Letzteres sind “Cracker”.) Mit der zunehmenden Kommerzialisierung der Softwareentwicklung ging diese Kultur verloren, was einige Leute, allen voran Richard Stallman, veranlasste 1984 das GNU Projekt und die Free Software Foundation zu gründen[11]. Mit zunehmendem Erfolg von Linux und freier Software allgemein während des dot-com-Hypes, entwickelte sich der Wunsch freie Software mehr zu kommerzialisieren. Das Ergebnis war die Gründung der Open Source Initiative (OSI) im Jahr 1998[12]. Diese erst prägte den etwas eingängigeren Begriff “Open Source”, ließ aber einige der eher philosophischen Ideale der Free Software Foundation und dem Begriff freier Software weg.

Größere Aufmerksamkeit, auch in den Mainstream-Medien, bekam Open Source im Jahr 1998 als Netscape ihren Browser als Open Source Software veröffentlichten[13]. Die so genannten “Halloween Dokumente”[14] aus dem gleichen Jahr, in denen Microsoft Linux als eine Bedrohung für ihr Geschäft identifiziert, blähten die Erwartungen weiter auf. Der Börsengang des Distributors RedHat, dessen Aktie am ersten Handelstag 1999 von $14 auf $5 schnellte und Anfang 2000 schon bei $260 lag, spiegeln auch hier die großen Erwartungen wider, die in Linux und Open Source gesetzt wurden. VA Linux hält seit 1999 den ungebrochenen

Rekord von fast 700% Wertzuwachs (von $30 auf $239) der Aktie am ersten Handelstag[15]. Dies waren jedoch die allgemein übersteigerten Erwartungen im IT-Sektor zu dieser Zeit.

4.2 Die Ernüchterung und der status quo

Viele Insolvenzen im IT-Sektor brachten dann die Ernüchterung und das Zerplatzen der dot-com-Seifenblase. VA Software (ehem. VA Linux) lag im April diesen Jahres bei einem Wert von $0.87 pro Aktie. Andere Unternehmen erholen sich langsam und sind dabei sich fest am Markt zu etablieren. RedHat zum Beispiel hatte im Jahr 2001 einen Umsatz von $78.9 Millionen und einen Verlust von $140.2 Millionen. Im Jahr 2002 waren es schon $90 Millionen Umsatz und nur noch $6.4 Millionen Verlust.

Open Source gerät auch zunehmend auf den Radar der Regierungen und der Politik. In Deutschland fördert das Bundesamt für Sicherheit in der Informationstechnik das Projekt "Ägypten" innerhalb ihres Projektes SPHINX zur elektronischen Signatur und Verschlüsselung in der öffentlichen Verwaltung. Ziel ist, dass hier auch Open Source Software eingesetzt werden kann und nicht einseitig auf Microsoft-Produkte gesetzt wird. Eine ähniche Zielsetzung im Bereich der Groupware-Lösungen hat die Förderung des Kroupware-Projektes durch die gleiche Behörde. "Open Source in der Verwaltung" ist ein im Moment stark diskutiertes Thema. Das Innenministerium hat mit IBM einen Kooperationsvertrag zur Förderung von Open Source Software abgeschlossen. Ziel ist es, diese in der Beschaffung überhaupt erstmal als Alternative zu berücksichtigen. Entscheidungen für Open Source Software, fanden in der Presse zum Teil viel Beachtung. So z.B. Anfang 2002 die Aktion "Bundestux"[16], bei der der Ältestenrat über die zukünftige Softwareausstattung im Bundestag zu entscheiden hatte oder die Entscheidung der Stadt München im Mai 2003, dass die ca. 14.000 Computer von Windows auf Linux umgestellt werden sollen[17].

Doch auch in der freien Wirtschaft setzen viele Unternehmen zunehmend auf Open Source. So wurde z.B. das neue Redaktionssystem der Zeitung "taz" komplett auf Open Source Komponenten umgestellt[18].

4.3 Chancen und Gefahren

Eine der Auswirkungen der weiteren Verbreitung von Open Source Software wird mehr Wettbewerb sein. Beim Ringen um die Entscheidung der zukünftige Softwareausstattung der Stadt München hatte sich überraschend Microsoft-Chef Steve Ballmer persönlich eingeschaltet[19]. Auf einmal konnte sogar ein um glatt sieben Millionen Euro günstigeres Angebot gemacht werden[20]. Zukünftig werden solcherlei Entscheidungen also zunehmend pragmatischer nach reinen Kosten/Nutzen-Abwägungen gefällt. Hier hat Open Source Software klare Vorteile.

Doch auch ein zunehmender Leidensdruck auf Seiten der Anwender wird einen Trend weg von monopolistischen Konzernen, bei denen Zwangsregistrierungen[21], Spyware[22] und Kontrollen durch Lizenzwächter[23] an der Tagesordnung sind, hin zu Open Source Software auslösen.

Zu wünschen wäre, dass Open Source Software durch den Wegfall der Lizenzkosten dazu beiträgt, die digitaleKluft zwischen Menschen die an der Informationsgesellschaft teilhaben können und denen die es sich nicht leisten können, kleiner wird. Ebenso ist es nun vielleicht auch mehr Unternehmen aus Entwicklungsländern möglich, in den IT-Markt einzutreten.

Aktuell bedroht wird Open Source durch die Bestrebungen Patente auf Software auch in Europa zuzulassen[24]. Kein Entwickler eines Open Source Projektes könnte sich die notwendigen Patentrecherchen leisten, um der Gefahr ein fremdes Patent zu verletzen aus dem Weg zu gehen. So bliebe nichts weiter übrig, als zu hoffen nicht verklagt zu werden.

Eine weitere Bedrohung in der Zukunft könnte die Einführung von TCPA[25] in Computern werden. Open Source Projekte würden sich die Kosten für eine Lizensierung ihrer Software nicht leisten können.

Literatur

[1] Frank Hecker, "Setting Up Shop"
http://www.hecker.org/writings/setting-up-shop.html

[2] Naomi Hoffman, "Open Source Software"
http://public.kitware.com/VTK/pdf/oss.pdf

[3] Eric Lee Green, "Economics of Open Source Software"
http://badtux.org/home/eric/editorial/economics.php

[4] Stefan Koch, "Produktivität in der Software-Entwicklung - Spezialfall Open Source: Open Source Software-Entwicklung"
http://wwwai.wu-wien.ac.at/~koch/lehre/oss-ws-01/oss_allg/

[5] Eric S.Raymond, "The Cathedral and the Bazaar"
http://www.catb.org/~esr/writings/cathedral-bazaar/

[6] Eric S.Raymond, "The Magic Couldron"
http://www.catb.org/~esr/writings/magic-cauldron/

[7] “The Open Source Definition”
http://www.opensource.org/docs/definition.php

[8] “The Free Software Definition”
http://www.fsf.org/philosophy/free-sw.html

[9] “Microsoft Shared Source”
http://www.microsoft.com/resources/sharedsource/default.mspx

[10] “Netcraft: August 2003 Web Server Survey”
http://news.netcraft.com/archives/2003/08/01/august_2003_web_server_survey.html

[11] “About the GNU Project”
http://www.fsf.org/gnu/gnu.html

[12] “History of the OSI”
http://www.opensource.org/docs/history.php

[13] “Netscape Press Release“
http://wp.netscape.com/newsref/pr/newsrelease558.html

[14] “The Halloween Documents”
http://www.opensource.org/halloween/

[15] “Start-ups und Linux (1) - Gründen mit dem Pinguin”
http://www.faz.net/s/Rub3C0B5C86BD3E4C2F8DE593FFA3707EEE/Doc~E225EC0634B5B43B284857516D5001820~ATpl~Ecommon~Scontent.html

[16] “Bundestux - Pinguine ins Amt”
http://www.bundestux.de/

[17] "Microsoft kann in München nicht "fensterln" "
http://www.heise.de/newsticker/data/anw-28.05.03-004/

[18] "Die drei von der Tankstelle"
http://www.taz.de/pt/2002/04/20/a0150.nf/text

[19] "Der Kunde ist auf jeden Fall Sieger"
http://www.sueddeutsche.de/wirtschaft/artikel/994/11983/

[20] "Microsoft gibt nicht auf"
http://www.sueddeutsche.de/muenchen/artikel/104/12092/

[21] "Ihr Recht auf Daten: Microsoft-Zwangsregistrierung"
http://www.pcwelt.de/ratgeber/windows/18925/

[22] "Ministerium: HP verletzt Datenschutz "
http://www.heise.de/newsticker/data/tig-10.03.03-000/

[23] "Drohbriefe an Software-Kunden"
http://www.heise.de/newsticker/data/cp-22.05.97-000/

[24] "FFII: Software Patents in Europe"
http://swpat.ffii.org/

[25] "TCPA / Palladium Frequently Asked Questions"
http://www.againsttcpa.com/tcpa-faq-en.html