

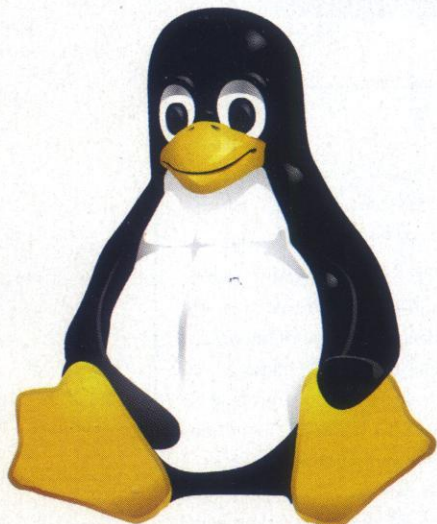
# Open Source: Frei, Die Lizenzen der offenen Innovation

Von Antoine Authier

**Der Aufschwung des Internets und die rasante Entwicklung der Kommunikationsmedien und -techniken haben eine ungekannte Verbreitung von Erkenntnissen, Projekten und Arbeits- und Diskussionsgemeinschaften auf globaler Ebene ermöglicht. Wenn solche Aktivitäten zu einem Prozess der Öffnung und des Austauschs führen, sprechen wir von Open Innovation.**



Dies trifft auf Projekte wie GNU, Linux, Mozilla, Apache, The GIMP, MySQL, AmaroK, OpenOffice, YAMPP oder jüngst OpenMoko zu. Sowohl bei Software als auch bei Schaltungen (Hardwarelösungen) sprechen wir von *Open Source*. Auch Künstler profitieren von dieser Entwicklung und diesen Kommunikationsmitteln. Werfen Sie einen Blick auf



das von Alex Ciccus und *The Sambuki Social Crew* lancierte *Wallpaper-Projekt*, in dem 100 Künstler ein Transparent mit einer Länge von 100 Fuß (ca. 32 Meter) geschaffen haben, das auf der Website [www.thespacejockeys.com/wallpaper](http://www.thespacejockeys.com/wallpaper) zu sehen ist.

Die unermessliche Menge der im Internet zu diesem Thema zur Verfügung stehenden Informationen macht daraus eine schwer zu greifende, unübersichtliche Angelegenheit. Dieser Artikel liefert zunächst einige historische Orientierungspunkte und eine Beschreibung der *Open Innovation*. In einem zweiten Schritt werden die Verbreitungswege und die Realisierbarkeit anhand einiger Beispiele beleuchtet. Dabei geht es auch um die spezifischen Merkmale von Hardwareprojekten und um den Entwicklungsprozess. Das Thema Patente und Urheberrecht lassen wir dabei außen vor. Insbesondere die Umsetzung der EUCED (European Union Copyright Directive) wäre noch einen separaten Artikel wert...

## Open Source

Die *Open-Source*-Bewegung nahm ihren Ursprung in den siebziger Jahren an der Universität von Berkeley in Kalifornien im Rahmen eines Entwicklungsprojekts für ein mit UNIX kompatibles Betriebssystem namens BSD (Berkeley Software Distribution). Durch die uneingeschränkte Offenlegung des Quellcodes zur gemeinsamen Nutzung konnte sich dieses Projekt als gemeinschaftliche Arbeit entwickeln. An solchen großen Projekten sind viele Entwickler betei-

ligt, die sich nicht notwendigerweise von Beginn an kennen. Sie arbeiten auf das gleiche Ziel hin, jedoch an verschiedenen Orten in der Welt und organisieren von Zeit zu Zeit Treffen, um den Entwicklungsplan festzulegen.

Im Jahr 1983 wurde von Richard Stallman die *Free Software Foundation* gegründet, um einen Rahmen für die Entwicklung des GNU (das berühmte rekursive Akronym *GNU is Not Unix*) zur Verfügung zu stellen. Er verfasste die GPL (*General Public Licence*) genannten Lizenzbedingungen, um die Früchte dieser Arbeit zu verbreiten und ihren Fortbestand zu sichern. Auf diese Weise entstand die freie Software oder *free software*. Im Englischen kann *free* sowohl frei als auch kostenlos bedeuten, was zu allerlei Missverständnissen geführt hat. **Freie Software ist nicht notwendigerweise kostenlos**; sie kann auch verkauft werden, was in der Regel jedoch nicht der Fall ist.

Mit der gemeinschaftlichen Entwicklung des Linux Kernels, die von einem finnischen Studenten ausging, und der darauf folgenden Integration dieses Kernels in GNU wurde der Grundstein für diese Form der offenen Innovation gelegt. Wenn es hierbei hauptsächlich um Software geht, spricht man von *Open Source*.

Es gibt keine eindeutige Definition von *Open Source*, die übrigens auch nicht gleichbedeutend mit einer Softwarelizenz ist. Die *Open Source* verfügt noch nicht einmal über einen eigenen formaljuristischen Wert. Außerdem sollte man aufhören, die Begriffe freie (kostenlose) Software und *Open-Source*-Software zu vermischen, wie dies häufig getan wird.



# offen, ungebunden?

## Die wichtigsten Lizenzen

Die in diesem Kasten enthaltenen Informationen sind in keiner Weise rechtlich verbindlich. Aufgrund der Fülle der in Lizenzen enthaltenen Informationen müssen wir uns damit begnügen, einige für den Einstieg in die Materie relevante Passagen herauszugreifen. Wir haben uns auf die hauptsächlich für die Verbreitung von Quellcodes oder Elektronik-Projekten verwendeten Lizenzen konzentriert. Der an der einen oder anderen Lizenz interessierte Leser sollte das entsprechende vollständige Dokument lesen. Sie finden die Texte der verschiedenen Lizenzen inklusive der diversen Neufassungen im Internet über die Website der FSF.

- Gemäß BSD-Lizenz (Berkeley Software Distribution) ist es gestattet, das durch die Lizenz geschützte Dokument in seiner Gesamtheit oder in Teilen ohne Einschränkung für eine öffentliche oder urheberrechtlich geschützte Arbeit weiter zu verwenden. Im Unterschied zum öffentlichen Bereich gibt es jedoch bei der Weiterverbreitung einige Regeln zu beachten.

Diese Lizenz schützt auch die Namen der Urheber in davon abgeleiteten Veröffentlichungen und entlastet sie im Hinblick auf etwaige mit der Nutzung der Arbeit verbundene Probleme. Sie ist eine der am wenigsten restriktiven Lizenzen.

- Die GNU GPL (GNU General Public Licence) ist sicherlich die radikalste. Sie schreibt die Veröffentlichung von Modifikationen für die Weiterverbreitung der Software zwingend vor. Sie basiert auf dem Prinzip des Copyleft, das die Verwertungsrechte der Gemeinschaft überlässt (im Gegensatz zum Copyright).

Jedoch gehört ein unter einer GPL veröffentlichtes Dokument faktisch dem Urheber, der über die Lizenz für die Verbreitung späterer Versionen des Dokuments entscheidet, was ihm erlaubt, diese Lizenz zu ändern.

Sie hat einen virusartigen Charakter, denn sie kontaminiert die (geschlossenen) Projekte, in denen sie verwendet wird.

Ein unter einer GPL veröffentlichtes Projekt muss in seiner Gesamtheit GPL sein.

Dies ist sicherlich die älteste und populärste Lizenz für freie Software.

- Die GNU LGPL Lizenz (GNU Lesser General Public Licence) stellt eine abgeschwächte Form der vorigen Lizenz dar. Sie gestattet es, den Code unter GNU GPL in seiner Gesamtheit an ein Mittel zu binden, das nicht frei ist. So erlaubt sie es, urheberrechtlich geschützte Software zu schreiben und dafür freie Software zu nutzen. Sie gilt hauptsächlich für Softwarebibliotheken. Ansonsten ist ihre Verwendung von geringem Interesse.

- Die MPL-Lizenz (Mozilla Public Licence) erlaubt die Kombination des geschützten (im Allgemeinen nicht veröffentlichten) Quellcodes mit dem freien Quellcode innerhalb eines Projekts. Sie garantiert also

die Offenlegung des als frei bezeichneten Quellcodes. Aufgrund ihres nicht virusartigen Charakters wird sie von vielen als freier als die GPL-Lizenz angesehen. Sie garantiert dem Urheber den Fortbestand des offenen Charakters seiner Arbeit ohne dabei die (derzeitigen oder künftigen) Projektmitarbeiter zu zwingen, dasselbe zu tun.

- Zeitgebundene Lizenzen

Die Offenlegung des Quellcodes erfolgt mit zeitlicher Verzögerung erst nach der Vermarktung der Software, normalerweise bei Veröffentlichung einer neuen kommerziellen Version. Dies ist zum Beispiel bei der Firma ID Software der Fall, die durch den Erfolg ihres ersten Spiels in 3D („Doom“) berühmt geworden ist. Dessen Quellcode wurde erst am Ende der kommerziellen Nutzungsdauer veröffentlicht.

- Die IBMPL-Lizenz (IBM's Public Licence), die von IBM für die Verbreitung mancher ihrer Quellcodes verwendet wurde, die als nicht strategisch galten, ist von der OSI anerkannt.

Sie weicht von der GPL in der Verwaltung und Verbreitung von Patenten ab. Sie macht eher Herausgeber und Verbreiter der Software verantwortlich als die am Projekt mitarbeitenden Personen.

- Die SCSL-Lizenz (Sun's Community Source Licence) ist die ursprünglich von SUN Microsystems zur Verbreitung ihrer Programmiersprache JAVA verwendete Lizenz.

Sie gestattet die Veröffentlichung des Quellcodes von JAVA nicht und wird daher nicht als offen betrachtet. Dafür erlaubt sie einerseits den Erwerb, die Nutzung und kostenlose Weiterverbreitung der JAVA virtual machine (ohne die der von Ihnen geschriebene Code nicht interpretiert werden kann) und andererseits die Verbreitung von in JAVA geschriebener freier Software, Tools und Bibliotheken. Seit 2005 hat SUN neue Lizenzen eingeführt, um die Öffnung voranzutreiben, ohne jedoch den Quellcode zu veröffentlichen.

- Die Shared-Source-Lizenz, die von Microsoft als Antwort auf den von der freien Software-Szene ausgeübten Druck eingeführt wurde, besteht aus mehreren Lizenzverträgen und ist den Großkunden von Microsoft und bestimmten einflussreichen Institutionen (Universitäten, chinesische Regierung...) vorbehalten. Sie erlaubt hauptsächlich die Einsicht in den Quellcode und die Fehlerbeseitigung. Nur sehr wenige haben ein Modifikationsrecht.

- Die Creative-Commons-Lizenz ist flexibel und personalisierbar. Sie bietet vier Verbreitungsoptionen: Urheberschaft, keine kommerzielle Verbreitung, keine Modifikation, Weitergabe unter gleichen Bedingungen. Auch in Kombination, um eine spezifische Verbreitungslizenz zu etablieren.

Ein innovatives und erfrischendes Konzept mit Zukunft.

Die *Open Source Initiative* OSI wurde 1998 als Antwort auf die ökonomischen und technischen Realitäten gegründet, um den freien Zugang zu den Quellcodes von Software zu verteidigen. Sie definiert einige Bedingungen, die eine Software erfüllen muss, um als *Open-Source-Software* bezeichnet zu werden; sie ist im übrigen auch das offizielle Zertifizierungsorgan der *Open-Source-Bewe-*

gung und ihrer verschiedenen Veröffentlichungslizenzen geworden.

## Vor- und Nachteile

In einem Artikel mit dem Titel „*Die Open Source Dynamik*“ hat Robert Visseur die Vor- und Nachteile von *Open Source* beschrieben. Wir fassen sie hier zusammen.

- Qualität: Stabilität und Leistung von Open-Source-Software wie Linux oder FreeBSD.

- Reaktivität: Die Aktualisierungen für Software wie zum Beispiel Apache oder Linux erscheinen sehr schnell.

- Beständigkeit (Legacy): Treue Nutzer haben zum Beispiel alte Betriebssysteme wie MS-DOS oder CP/M neu programmiert.

- Kosten



- **Freiheit:** Unabhängigkeit der Entwickler.
- **Konkurrenz:** Eine Monopolisierung ist durch die Offenlegung des Quellcodes und der Kommunikationsstandards unmöglich. Der Wettbewerb zwischen Unternehmen und sogar zwischen einzelnen Ländern wird stimuliert. Mit einem offenen Quellcode kann sich die Software frei entwickeln und profitiert von der Unterstützung der gesamten Gemeinschaft. Dasselbe gilt für die Dokumentation und Entwicklung wirklich offener und verlässlicher Standards, auf die jeder einen kritischen Blick werfen kann.
- **Schutz des geistigen Eigentums:** Der Hauptkritikpunkt der Gegner von *Open Source* ist, dass zur Förderung von Innovationen Geheimhaltung und eine erbitterte Konkurrenz unerlässlich seien. Dem halten die Befürworter von *Open Source* entgegen, dass der Austausch von Ideen und die gegenseitige Anregung Innovationen weit mehr fördern, als es Geheimhaltung je könnte.
- **Ausführung:** Berühmte klassische Interfaces.
- **Divergenzrisiko:** BSD hat sich zum Beispiel schrittweise in drei Projekte aufgespalten: FreeBSD (das am weitesten verbreitete und benutzerfreundlichste, das auf Leistung ausgerichtet ist), OpenBSD (das auf Sicherheit ausgelegt ist) und das ursprüngliche Projekt, NetBSD (das auf Anpassung an die Hardware ausgerichtet ist).
- **Markenimage:** Linux wird in hierarchisch strukturierten Organisationen oftmals als ungeeignet für die professionelle Nutzung angesehen.
- **Support :** Die Dokumentation ist oft schwach. Das Mitte der 90er Jahre aufgekommene *Wiki*-Konzept, das in letzter Zeit durch die freie Enzyklopädie Wikipedia populär wurde, versucht diese Lücken zu schließen. Es bietet dem Leser die Möglichkeit, online zur Dokumentation beizutragen. Englisch ist trotz lobenswerter Übersetzungsbemühungen obligatorisch.

### An einem Projekt mitarbeiten

Nichts ist einfacher als an einem *Open Source*-Projekt mitzuarbeiten. Kontaktieren Sie den Autor per Mail oder Instant Messaging, schreiben Sie einen Beitrag im Projektforum, sagen Sie Ihre Meinung oder fügen Sie eine Dokumentationsseite oder Anregungen zu *Wiki* hinzu. Sie können auf mehreren Ebenen einen Beitrag leisten, dies kann einfach durch Ihre Unterstützung geschehen oder auch durch eine kleine Spende.

Ihre allgemeinen Anmerkungen zu den Funktionen oder zum Interface sind gern gesehen, die Meldung von Bugs, ein Kommentar, die Durchsicht des Quellcodes sind willkommen. Und wenn Sie das literarische Fieber packt, dann zögern Sie nicht, an der Projektdokumentation mitzuarbeiten. Wenn Sie programmieren möchten, dann zeigen Sie, was Sie können und schicken *Patches* (Korrekturen, Verbesserungen), und Sie werden bald einen Schreib- oder Verwaltungszugang für die Version (**CVS**, **Subversion...**) erhalten

### Ein Projekt starten

Ein Projekt zu starten ist nicht viel schwieriger. Sie haben eine Idee von der Sie glauben, dass Sie für die Gemeinschaft nützlich ist und möchten sie mit anderen teilen. Wenn Sie nicht ihr eigener Webmaster sein möchten, dann melden Sie einen Account bei [www.sourceforge.net](http://www.sourceforge.net) an und veröffent-



lichen Ihr Projekt mit dem Quellcode und allem, was dazugehört. Wenn Sie für Ihr Projekt lieber selbst eine Website einrichten möchten, können Sie dafür gemeinschaftlich entwickelte Tools nutzen und mit Hilfe von gforce Ihre eigene Community Site aufbauen. Wenn nicht, genügt auch eine einfache Website ...

### Verbreitung eines Projekts

Wie können die Urheber freier Software davon leben oder zumindest etwas Geld verdienen? Bevor Sie Ihre Arbeit im Internet veröffentlichen und damit offen zugänglich machen, ist es wichtig, einen rechtlichen Rahmen festzulegen, der es Ihnen erlaubt, als Urheber anerkannt zu werden, sich gegen eine etwaige missbräuchliche Nutzung zu schützen und eine Nutzung durch andere mehr oder weniger zuzulassen. Die freie und kostenlose Verbreitung unter Beibehaltung der Urheberrechte

für Codes, Entwürfe, Bilder, Fotos, Graphiken und so weiter ist mit Hilfe einer Lizenz möglich (zum Beispiel *Creative Commons*). Die Übersicht im Kasten erleichtert Ihnen die Auswahl einer Ihren Bedürfnissen entsprechenden Lizenz. Solche Lizenzen reichen von der kompromisslosen *open source (freie Software)* über verschiedene moderatere Formen bis zur *Shared-Source-Lizenz*. Wenn es darum geht, Geld zu verdienen, muss man sich mehr oder minder den ökonomischen Modellen unserer Marktwirtschaft mit den üblichen Strategien und Marketingmaßnahmen anpassen. Die OSI arbeitet für eine Softwarewirtschaft ohne Nutzungslizenzen, jedoch basierend auf dem Verkauf von Angeboten wie Support und Hardware, die eine schnelle Nutzung ermöglichen. In Europa verdienen mehrere tausend Ingenieure dank dieses ökonomischen Modells ihr Geld.

### Open Hardware

Das Konzept von *Open Source Hardware* oder *Open Hardware* unterscheidet sich kaum hiervon. Es ist jedoch viel einfacher, Software zu kopieren als Hardware zu reproduzieren, für die genauere Kenntnisse und Kompetenzen erforderlich sind.



FreeBSD®

Während *Open Source* bei Software durch die zugehörigen Lizenzen klar definiert ist, existiert für das Konzept der *Open Source Hardware* noch keine weltweit anerkannte oder vollständig akzeptierte Definition.

Richard Stallman, der Gründer der FSF (*Free Software Foundation*) und Verfasser der Lizenz GPL (*General Public Licence*), bekräftigte 1999 in einem in „*Linux Today*“ veröffentlichten Artikel, dass sich die GPL für elektronische Schaltungen nicht eignet, weil diese schwieriger zu verändern oder zu kopieren (nachzubauen) sind. In den meisten Reaktionen darauf wurde dies auch für eingebaute Software (im Speicher von Chips) so gesehen.

Die Nutzer von *Open Source Hardware* müssen die Hardware modifizieren und frei weitergeben (vertreiben) können. Hierfür ist die uneingeschränkte Bereitstellung der Schaltpläne, aber auch des Quellcodes der einzelnen Kompo-



nenten erforderlich, insbesondere der Quellcodes der HDL (*Hardware Description Language*) der *Hardware* und der *Firmware* (meistens in C oder Assembler geschrieben).

In einem Gespräch mit Elektor hat Harald Kipp, der Vater des an anderer Stelle in dieser Ausgabe beschriebenen Ethernut-Projekts, nicht ohne Humor gesagt, dass *Open Source* für viele wie der Kommunismus ist, in dem Sinne, dass jeder von der Gemeinschaft alles bekommen kann, ohne irgend etwas dafür zu zahlen. Kipp fügte hinzu, dass *Open Source* für manche Entwickler ebenfalls mit dem Kommunismus gleichzusetzen ist, aber deswegen, weil sie die hierarchischen Strukturen als Diktatur empfinden!

Man könnte in der Tat versucht sein zu glauben, dass in der *Open-Source*-Gemeinde Anarchie herrscht, aber so ist es keineswegs. Die Organisation der *Open-Source*-Projekte ist solide, ja sogar stark hierarchisch aufgebaut, wobei der Eigentümer des Projekts die meiste Zeit der unangefochtene Chef an der Spitze des Gefüges ist. Wenn



ein Beitrag ihm nicht gefällt, kann er ihn herausnehmen, ohne irgend jemanden zu fragen. Gefällt ihm ein Entwickler nicht, so kann er ihn ohne weiters „entlassen“, indem er ihm jegliche Form der Mitarbeit verweigert.

Bei kleineren Projekten ist es ebenfalls der Eigentümer, der entscheidet, wann welche neue Version veröffentlicht wird. Die Quellcodes können (im Allgemeinen) frei weiterverbreitet werden, das heißt, dass jeder als Verteiler freier Software agieren kann (vorausgesetzt, er respektiert die Lizenzbedingungen). Wenn ein Teil der Gemeinschaft oder ein isolierter (und frustrierter) Entwickler glaubt, es besser machen zu können, steht es ihm frei, den Quellcode aufzugreifen und ein neues Projekt ins Leben zu rufen. Eine solche Abspaltung wird *fork* genannt (nach dem C-Befehl der POSIX-Norm, der es in C ermöglicht, einen Prozess zu duplizieren).

## Geistige Offenheit

Für ein Unternehmen ist es wichtig, die Berichte von Nutzern über Fehl-

funktionen (bugs) zu sammeln. Das Unternehmen kann sogar noch weiter gehen und ein öffentliches Forum zur Verfügung stellen, in dem jeder Nutzer sich äußern, etwaige Mängel kritisieren oder die Vorteile preisen kann. Eine solche Initiative kann sich auch als unangenehm erweisen. Jedermann zu erlauben, die Qualität des Quellcodes und der Hardwarekonzeption frei zu analysieren und diese offen (und oft auch unangemessen) zu kritisieren, erfordert ein gewisses Maß an Mut, Gewöhnung, Durchhaltevermögen und auch ein dickes Fell.

Die vernünftigen Kritiker sind wichtig, um das Projekt effektiv vorantreiben zu können. Ohne den Druck des kritischen Blicks von außen würden die meisten Konzeptionsmängel über die verschiedenen Entwicklungsstadien hinweg bestehen bleiben, sich festsetzen und schließlich nach einem anfänglich eleganten Start das ganze Projekt gefährden.

Während der gesamten Laufzeit eines Projekts, ob es dabei um Soft- oder Hardware geht, muss darauf geachtet werden, dass alle gesetzlichen Anforderungen erfüllt und Patentverletzungen, ob absichtlich oder nicht, vermieden werden. Während es in einem „geschlossenen“ Projekt schwierig ist, solche Verstöße aufzudecken, so treten sie in einem offenen Projekt umso schneller zutage, je größer und vielfältiger die Gemeinschaft ist. Es ist also notwendig, angemessene Maßnahmen zu ergreifen, um sich der Wahrung der Urheberrechte im Hinblick auf die verschiedenen Beiträge zu versichern, was eine schwierige Aufgabe ist. Der Gemeinschaftsgeist offener Projekte zwingt die Entwickler, solche Verstöße festzustellen und unverzüglich Maßnahmen zu ergreifen.

Zurzeit werden die meisten *Open Hardware*-Projekte und *Firmwares* mit BSD-Lizenzen verbreitet, die weniger restriktiv sind als die GPL-Lizenz und die insbesondere die Verbreitung verbesserter *Firmware* ohne den entsprechenden Quellcode erlauben. Auf diese Weise behalten die Unternehmen die Kontrolle über die Veröffentlichung bestimmter Verbesserungen, während andere geheim bleiben, um die Wertschöpfung des Produkts zu erhalten.

## Kommerzielle Beiträge

Worin besteht das Interesse eines Unternehmens, zu einem *Open Source*-Projekt beizutragen, wenn es nicht durch eine Lizenz dazu gezwungen ist? Warum sollte es das Risiko eingehen, dass einer

seiner Konkurrenten von den Früchten seiner Arbeit profitiert? Es geht zweifellos darum, von den Verbesserungen anderer Projektmitarbeiter zu profitieren. Statt darauf zu warten, dass man von Konkurrenten eingeholt wird, kann ein Unternehmen durchaus die Vorteile der Öffnung genießen, die es in die Lage versetzen, dank besserer Produkte, besserer Dienstleistungen und überlegener Qualität nicht ins Hintertreffen zu geraten, selbst wenn sämtliche Mitbewerber dieselbe Software nutzen. Manche Unternehmen leisten auch „zeitlich verzögerte Beiträge“ und halten anfangs ihre Verbesserungen geheim, um sie erst dann zu veröffentlichen, wenn ihr Produkt gut am Markt eingeführt ist (siehe Beispiel für eine zeitlich verzögert freigegebene Lizenz im Kasten).

Elektor veröffentlicht diesen Artikel unter der *Creative Commons* Lizenz, die eine ausdrückliche Erwähnung der Urheberschaft (Quellenangabe), den Abschluss jeglicher kommerziellen Nutzung und die Weitergabe unter gleichen Bedingungen (*share-alike*) verlangt.

Bei allen in Elektor veröffentlichten Projekten legen wir großen Wert darauf, dass der Quellcode wann immer möglich offen gelegt wird. Manchmal kommt es vor, dass Autoren damit nicht einverstanden sind. Die Entscheidung für oder gegen eine Veröffentlichung ist dann nicht einfach. Soll ein sehr interessantes Projekt aus diesem Grund abgelehnt werden, oder soll es trotzdem veröffentlicht werden, um möglichst vielen zur Verfügung zu stehen, auch wenn der Quellcode nicht offen gelegt wird? Wir sind fest davon überzeugt, dass sich die Offenlegung auf lange Sicht bezahlt macht.

(070948-1)

## Links

[www.fsf.org](http://www.fsf.org)  
[www.fsf.org/campaigns](http://www.fsf.org/campaigns)  
[www.opensource.org](http://www.opensource.org)  
<http://en.wikipedia.org/wiki/Copyleft>  
<http://opencollector.org/Whyfree/>  
<http://opencollector.org/Whyfree/vilbrandt.html>  
<http://en.wikipedia.org/wiki/FLOSS>  
[http://en.wikipedia.org/wiki/Open\\_Source\\_Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative)  
[http://en.wikipedia.org/wiki/GNU\\_Project](http://en.wikipedia.org/wiki/GNU_Project)  
[http://features.linuxtoday.com/news\\_story.php3?ltsn=1999-06-22-005-05-NW-LF](http://features.linuxtoday.com/news_story.php3?ltsn=1999-06-22-005-05-NW-LF)  
<http://en.wikipedia.org/wiki/Wiki>  
<http://en.wikipedia.org/wiki/Wikipedia>