

# Freie Software und Open Source Software

Seminararbeit im Rahmen des Seminars  
„Wissen in der modernen Gesellschaft“  
Universität Leipzig, Sommersemester 2009

Verfasser: Alexander Lust (1025950)  
Betreuer: Prof. Hans Gert Gräbe

**18.06.2009**

# Vorwort

Diese Ausarbeitung entstand im Rahmen der Veranstaltung

"Wissen in der modernen Gesellschaft" an der Universität Leipzig, unter der Leitung von Professor H.-G. Gräbe.

Alexander Lust

# Inhaltsverzeichnis

Einleitung .....	4
Kurze Vorgeschichte .....	7
Freie Software .....	7
Shareware .....	9
Public Domain .....	9
Unterschiede .....	9
Kritik des „Stallman’s Ideals“ .....	15
Entwicklungsmodell .....	16
Ergebnisse .....	18
Quellen .....	20
Internetverzeichnis .....	20
Literaturverzeichnis .....	22

# Einleitung

*„Ich glaube, daß es das Gebot der Nächstenliebe verlangt, daß ich ein Programm, das mir gefällt, mit anderen teile, denen es ebenfalls gefällt.“*

(Richard Stallman)

*“'Free software' is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer'.”*

(gnu.org)

Gegenwärtig ist es in den Informatikerkreisen sehr modisch geworden „Freie Software“ oder „Open-Source-Software“ anstatt der proprietären Software zu benutzen und zu entwickeln. Viele folgen der Mode ohne dabei zu wissen, was eigentlich im größeren Sinn dahinter steckt. Den meisten sind nur die wesentlichen Unterschiede zwischen „Freie Software“ und „Open-Source-Software“ bekannt. Diese bestehen darin, dass nichtkommerzielle Software „Freie Software“ oder „Open-Source-Software“ heißt und „proprietäre Software“ für kommerzielle Software steht. Die „Open-Source-Software“ (quelloffene Software) erlaubt den Quellcode anzusehen und zu verändern. Der Begriff „Freie Software“ wird oft als „nichtprofessionelle“ Bezeichnung von „Open-Source-Software“ betrachtet, wenn man diese der „proprietären Software“ gegenüberstellt. Mit anderen Worten, stehen sich eine kostenlose Software, die für alle offen ist, und kostenpflichtige Software, deren Quellcode aus kommerziellen Gründen selbstverständlich für alle geschlossen ist, gegenüber. Aus eigener Erfahrung im Austausch mit anderen Informatikstudenten kam mir zu Ohren, dass „Freie Software“ auch Demo-Versionen der kommerziellen Software sein können. Diese werden nur für einen begrenzten Zeitraum für Werbungszwecke freigegeben. Andererseits führt der Begriff „Freie Software“ oft zu Missverständnis. Wie die Open-Source-Bewegung im Jahr 1998 kritisiert hatte, assoziiert man oft den Begriff „Freie Software“ mit „freiem Bier“, das nichts kostet, was folglich für Unternehmer und Entwickler nicht besonders lukrativ klingt. So teilte sich 1998 die Freie-Software-Bewegung in zwei verschiedene Bewegungen, mit unterschiedlichen Ansichten, Werten und Philosophien.

Die Bewegungen waren nicht nur ein Ergebnis auf Grund von verschiedenen Annahmen hinsichtlich der Begrifflichkeit. Die Open-Source-Bewegung hatte ebenso versucht, die Freie Software für Unternehmen und Entwickler auf eine andere Art zu präsentieren. Es sollte darauf aufmerksam gemacht werden, dass Freie Software in der modernen „kapitalistischen“ Welt auch konkurrenzfähig zur proprietären Software ist. Außerdem sollten die beiden Richtungen Begründungen liefern, dass dieses neue wirtschaftliche Modell auch zum Nutzen der Unternehmen und Entwickler dienen kann und keine „kommunistische“ Utopie ist.

Natürlich brachte das neue Modell auch verschiedene Abweichungen in die Freie-Software-Bewegung.

Auf einem strategischen Treffen am 3. Februar 1998 in Palo Alto, California entschied sich die Open Source Initiative (OSI) das „Open-Source“-Label anstelle von Freie Software zu integrieren.

Richard Stallman, ein amerikanischer Aktivist, Hacker und Programmierer schreibt 2007 in seinem Artikel „Warum „Open Source“ am Kern der Sache vorbei geht“: *„Open Source ist ein Entwicklungsmodell. Freie Software ist eine soziale Bewegung*

....

*Womöglich ist diese Software "Open Source" und benutzt auch das Open-Source-Entwicklungsmodell. Aber es ist auf gar keinen Fall Freie Software, da es die Freiheit des Nutzers nicht respektiert, der die Software letztendlich einsetzt. Wenn das Entwicklungsmodell der Open-Source-Bewegung die betreffende Software erfolgreich verbessert und zuverlässiger macht, und zwar zur Beschneidung Ihrer Rechte, dann macht dies die Sache eigentlich nur schlimmer.“*

Richard Stallman legte den Grundstein für die Regeln zur Verwendung freier Software – die Gnu Public License und hatte damit eine neue Philosophie entwickelt: das Wissen mit anderen zu teilen und gemeinsam zu verbessern, statt es zu „verheimlichen“.

Software-Entwickler, die ihre Software unter dieser Lizenz veröffentlichen, erlauben anderen Nutzern ihre Software zu studieren, zu verändern und weiterzuverbreiten.

Die Aktivisten der OSI versuchten der Open-Source-Bewegung den Begriff „Freie Software“ an die kapitalistische Realität anzupassen und zu erweitern. Sie wollten, dass dabei Entwicklungsstrategie und nicht „freies Bier“ betont wird.

Richard Stallman äußerte die Meinung, dass die Werte der Bewegung dadurch verletzt wurden, indem die Open-Source-Bewegung auch nicht ganz „quelloffene“ Lizenzen zulässt. Der Begriff „Open Source“ sei nicht ausdrucksstark und stelle dadurch die ganze Ideologie in Schatten.

Aktivisten der Open-Source-Bewegung kritisierten ebenfalls den Begriff „Freie Software“ und fanden, dass nur die Ideologie nicht ausreichend konkurrenzfähig zur proprietären Software ist, weil es für Unternehmen und Firmen den Eindruck verschafft, dass außer Ideologie nichts dahinter steht. Da für die Firmen und Unternehmen der Begriff „Freie Software“ nach ihrer Meinung nicht besonders lukrativ klang, entschieden sie sich für „Open Source“.

Die Open Source Initiative versucht 1998 die Freie Software erneut zu präsentieren und für die Gesellschaft als perspektivvolle Alternative zu den kommerziellen Produkten darzustellen.

# Kurze Vorgeschichte

## ***Freie Software***

Als Richard Stallman 1983 das GNU-System zu entwickeln anfang, hatte er die Absicht nicht ein besseres System als das Betriebssystem UNIX zu entwickeln, sondern seine Ideen über Freies Nutzen und Teilen des Wissens zum Leben zu erwecken. Richard Stallman war der Meinung, dass es sich nicht nur um Software handelt, sondern überhaupt um das Wissen allgemein, da das Wissen jedem zugänglich gemacht werden muss. Ohne diese Voraussetzungen ist weitere Entwicklung der Wissenschaft kaum möglich. Wenn jedes Wissen sich patentieren lässt, wird die Entwicklung der Menschheit deutlich langsamer. Solch ein „geschlossenes“ Wissen verschärft und schafft Unfreiheit für die Menschheit. Da die Software bzw. der Quellcode eines Programms als reines Wissen angesehen werden kann, entsteht ein Widerspruch mit der Realität, wenn man z. B. die öffentlichen Bibliotheken kommerziellalisiert und man bezahlen muss um Bücher lesen zu können. Wenn eine Bibliothek eine Nutzungsgebühr verlangt und es ihr dabei um die eigene Existenz geht, resultiert dieses Verhalten aus wirtschaftlichen und auf keinen Fall aus kommerziellen Gründen. Richard Stallman sah in der proprietären Software nicht nur eine Einschränkung für die Entwicklung der Software allgemein, sondern den Versuch das Wissen zu kommerziellen Zwecken auszunutzen, also die Freiheit jedes Menschen zu kommerziellen Zwecken zu missbrauchen.

Schließlich hatte Richard Stallman den Begriff der “Freien Software” eingeführt. Der Begriff steht für die Idee, dass Software von ihren Benutzern zu jedem Zweck ausgeführt, verändert und weiterentwickelt werden kann und darf. Um die Idee der Freien Software zu kommunizieren und zu verbreiten gründete Richard Stallman 1985 die Free Software Foundation (FSF), die mittlerweile auch Schwesterorganisationen in Europa (FSFE), Indien (FSFI) und Lateinamerika (FSFLA) hat.

Er schrieb den leistungsstarken Editor Emacs, danach den GNU-C-Compiler (heute als GNU Compiler Collection GCC bekannt). Für alle diese Programme schuf er damals

eine eigene Lizenz: die GNU Emacs General Public Licence (GPL) und die GCC GPL. Die Lizenzen sollten dafür sorgen, dass seine Ideen über die Freiheit des Benutzers garantiert bleiben. Der wichtige Aspekt dabei war zu verhindern, dass der Quellcode einer solchen Software Teil einer proprietären Software wird. Dabei entstand aber ein Problem: Man konnte keinen Code aus einem Programm unter einer der verschiedenen GPL-Lizenzen in einem anderem wiederverwenden. Stallman schuf dann eine einzige Lizenz, die sich in jedes beliebiges Programm einzufügen erlaubte. Selbst hatte er sie eine "Lizenz als Subroutine" genannt.

Diese Lizenz heißt GNU GPL. Sie ist die meist verbreitete Lizenz für Freie Software.

Joachim Korb stellte in seinem Artikel „Freie Software für Freiheit und Gerechtigkeit“ die Geschichte der Entstehung grundlegender Ideen und Konzepte von freier Software in zwei Teilen dar: *„Der erste Teil schließt die Entwicklung bis zur Entstehung der ersten Betriebssysteme und die 1950er und 60er Jahre ein. Im zweiten Teil rücken die Zustände, die zur Entwicklung des GNU-Systems führten, also etwa die 1970er und die frühen 1980er Jahre, in den Mittelpunkt. Richard Stallman beschäftigt sich hauptsächlich mit der Entwicklung des UNIX-Systems und dessen BSD-Variante, die er bis zu ihrer endgültigen Lösung von der Universität im Jahr 1995 verfolgt.“*

Im Februar 1998 wurde in einer Brainstorming Session der Begriff „Open Source“ geprägt. Erst zwei Monate später beschloss die OSI den neuen Begriff „Open Source“ für „Freie Software“ zu verwenden. Die Einführung des neuen Begriffes war nicht als Marketingstrategie gedacht, es war der Versuch, einen Begriff zu etablieren.

Sie hatten versucht sowohl Entwicklern, als auch Firmen den Unterschied zwischen „Freier Software“ und „kostenloser Software“ und die Differenz zwischen verschiedenen Lizenzen klarer zu machen. Open Source umfasst eine Palette von Lizenzen und sollte dazu dienen, das Hauptprinzip der Bewegung auszuprägen.

Dabei setzte OSI das größte Gewicht auf das Recht, den Quellcode nicht nur zu sehen, sondern auch zu verändern und weiterleiten zu dürfen. Dabei versucht OSI dieses Konzept als Entwicklungsmodell für Firmen und Unternehmen darzustellen, welches nicht wie „freies Bier“ klingt, sondern wie „Vorgehensweise“.



## **Shareware**

Gleichzeitig entwickelte sich noch eine Form der Verteilung von Programmen. Die Programme wurden kostenlos verteilt und falls der Anwender mit dem Programm zufrieden blieb, konnte er eine geringe Gebühr dem Entwickler überweisen.

So entstand Shareware. Dabei handelt es sich nicht um eine Form der Softwareentwicklung, sondern um eine Form der Vertriebsart. Die Anwender bezahlen Gebühren z.B. bei der Registrierung oder bei der Zustimmung der Lizenz.

## **Public Domain**

Diese Art von Freier Software entstand an Universitäten in den USA. In den Universitäten wurde Software entwickelt. Bereits vom Staat finanziert wurde sie zum öffentlichen Eigentum. Bei Public Domain-Software (eng. gemeinfreie Werke) verzichtet der Autor auf sein Programm und gibt frei für absolut grenzlose Benutzung.

## **Unterschiede**

*„Bei Freeware wird nur die Benutzung ohne Lizenzgebühren gestattet, aber kein Quelltext offen gelegt. Änderungen sind nicht erlaubt. Bei Public-Domain-Software hingegen entäußert sich der Urheber seiner Rechte und stellt sein Programm in eine Art öffentliches Eigentum. Es kann in jeder Form genutzt werden, wobei keine die Nutzung beschränkenden Bedingungen möglich sind.“ [8]*

Bei Shareware ist Software kostenpflichtig, Quellcode muss aber nicht offen sein.

## Definition

Um besser zu verstehen, wie sich die zwei Begriffe „Freie Software“ und „Open-Source-Software“ unterscheiden, muss man die Definitionen von beiden vergleichen. Auf [www.gnu.org](http://www.gnu.org) steht folgende Definition für Freie Software:

*„Freie Software bedeutet die Freiheit des Benutzers, die Software zu benutzen, zu kopieren, sie zu vertreiben, zu studieren, zu verändern und zu verbessern. Genauer gesagt, bezieht sich der Begriff »Freie Software« auf vier Arten von Freiheit, die der Benutzer der Software hat:*

- 1) Die Freiheit, das Programm für jeden Zweck zu benutzen (Freiheit 0).*
- 2) Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.*
- 3) Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann (Freiheit 2).*
- 4) Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann (Freiheit 3). Der Zugang zum Quellcode ist dafür Voraussetzung.*

“

”

**Einführung**

“Quelloffen” (“open source”) bedeutet nicht nur freien Zugang zum Quellcode. Bei quelloffener Software müssen die Lizenzbestimmungen in Bezug auf die Weitergabe der Software folgenden Kriterien entsprechen:

**1. Freie Weitergabe**

Die Lizenz darf niemanden in seinem Recht einschränken, die Software als Teil eines Software-Paketes, das Programme unterschiedlichen Ursprungs enthält, zu verschenken oder zu verkaufen. Die Lizenz darf für den Fall eines solchen Verkaufs keine Lizenz- oder sonstigen Gebühren festschreiben.

**2. Quellcode**

Das Programm muss den Quellcode beinhalten. Die Weitergabe muss sowohl für den Quellcode als auch für die kompilierte Form zulässig sein. Wenn das Programm in irgendeiner Form ohne Quellcode weitergegeben wird, so muss es eine allgemein bekannte Möglichkeit geben, den Quellcode zum Selbstkostenpreis zu bekommen, vorzugsweise als gebührenfreien Download aus dem Internet. Der Quellcode soll die Form eines Programms sein, die ein Programmierer vorzugsweise bearbeitet. Absichtlich unverständlich geschriebener Quellcode ist daher nicht zulässig. Zwischenformen des Codes, so wie sie etwa ein Präprozessor oder ein Konverter (“Translator”) erzeugt, sind unzulässig.

### **3. Abgeleitete Software**

*Die Lizenz muss Veränderungen und Derivate zulassen. Außerdem muss sie es zulassen, dass die solcherart entstandenen Programme unter denselben Lizenzbestimmungen weitervertrieben werden können wie die Ausgangssoftware.*

### **4. Unversehrtheit des Quellcodes des Autors**

*Die Lizenz darf die Möglichkeit, den Quellcode in veränderter Form weiterzugeben, nur dann einschränken, wenn sie vorsieht, dass zusammen mit dem Quellcode so genannte "Patch files" weitergegeben werden dürfen, die den Programmcode bei der Kompilierung verändern. Die Lizenz muss die Weitergabe von Software, die aus verändertem Quellcode entstanden ist, ausdrücklich erlauben. Die Lizenz kann verlangen, dass die abgeleiteten Programme einen anderen Namen oder eine andere Versionsnummer als die Ausgangssoftware tragen.*

### **5. Keine Diskriminierung von Personen oder Gruppen**

*Die Lizenz darf niemanden benachteiligen.*

### **6. Keine Einschränkungen bezüglich des Einsatzfeldes**

*Die Lizenz darf niemanden daran hindern, das Programm in einem bestimmten Bereich einzusetzen. Beispielsweise darf sie den Einsatz des Programms in einem Geschäft oder in der Genforschung nicht ausschließen.*

### **7. Weitergabe der Lizenz**

*Die Rechte an einem Programm müssen auf alle Personen übergehen, die diese Software erhalten, ohne dass für diese die Notwendigkeit bestünde, eine eigene, zusätzliche Lizenz zu erwerben.*

### **8. Die Lizenz darf nicht auf ein bestimmtes Produktpaket beschränkt sein**

*Die Rechte an dem Programm dürfen nicht davon abhängig sein, ob das Programm Teil eines bestimmten Software-Paketes ist. Wenn das Programm aus dem Paket herausgenommen und im Rahmen der zu diesem Programm gehörenden Lizenz benutzt oder weitergegeben wird, so sollen alle Personen, die dieses Programm dann erhalten, alle Rechte daran haben, die auch in Verbindung mit dem ursprünglichen Software-Paket gewährt wurden.*

### **9. Die Lizenz darf die Weitergabe zusammen mit anderer Software nicht einschränken**

*Die Lizenz darf keine Einschränkungen enthalten bezüglich anderer Software, die zusammen mit der lizenzierten Software weitergegeben wird. So darf die Lizenz z. B. nicht verlangen, dass alle anderen Programme, die auf dem gleichen Medium weitergegeben werden, auch quelloffen sein müssen.*

“ [3]

Wie leicht zu sehen ist, ist die zweite Definition eine Erweiterung der ersten, indem die Definition von Open Source starken Akzent auf Verfügbarkeit des Quellcodes setzt. Der Stützpunkt ist dann der neunte Punkt in der Definition von Open Source: „Die Lizenz darf die Weitergabe zusammen mit anderer Software nicht einschränken“. Stallman schreibt in seinem Artikel „Warum "Open Source" am Kern der Sache vorbei geht“ folgendes: „Die offizielle Definition von „Open-Source-Software“, wie sie von

*der Open Source Initiative veröffentlicht wird, kommt unserer Definition Freier Software sehr nah. Sie ist jedoch in mancherlei Hinsicht ein wenig lockerer und die Open Source Initiative hat ein paar Lizenzen akzeptiert, von denen wir glauben, dass sie die Benutzer auf unzumutbare Weise einschränken.*

.....

*Es beinhaltet Freie Software, aber auch halbfreie Programme wie zum Beispiel Xv und sogar einige proprietäre Programme, wie zum Beispiel Qt unter seiner ursprünglichen Lizenz (vor der QPL).“*

## Kritik des „Stallman's Ideals“

*„Verbindung des Linux-Betriebssystemkernels mit Stallmans GNU-System wurde nach und nach auch für den Einsatz in Unternehmen brauchbar. Jedoch stellte Stallman Open Source auf eine Weise dar, welche die Akzeptanz der Werte gewisser Freiheiten a priori voraussetzte. Stallman ist Programmierer, und er entschied sich daher für eine philosophische Darstellung, die Programmierer ansprach.*

*Geschäftsleute hingegen sind Pragmatiker und lassen sich von wirtschaftlichen Vorteilen beeindrucken. Da die Art seiner Darstellung Stallmans Zielgruppe von vorneherein begrenzte, hatte seine Kampagne nicht die wirtschaftliche Schlagkraft, die sich heute abzeichnet.“*

(Bruce Perens)

*“ Englisch scheint die einzige Sprache zu sein, in der eine solch große Doppeldeutigkeit zwischen Freiheit und Preis besteht. Wenn man den Begriff in andere Sprachen übersetzt, wird aus Freier Software "logiciel libre" im Französischen, "software libre" im Spanischen, "software libero" im Portugiesischen, "Fri Software" im Dänischen oder was auch immer der vergleichbare Begriff in der jeweiligen Sprache für Freiheit sein mag.“ [5]*

Das führt nach Meinung der OSI zu einem Missverständnis, da das englische Wort „free“ als „gratis“ verstanden werden kann.

Aber mit free ist nicht nur *„die Freiheit von Kosten im Sinne von gratis“* gemeint, sondern auch die *„Freiheit im Sinne der Freiheit des Geistes und der Wissenschaft“*. [1]

Richard Stallman: *„Frei wie in freier Rede und nicht wie in Freibier!“*

Richard Stallman sieht in der neu präsentierten Form der Freien Software eine Verletzung der Ideale, für die er solange gekämpft hatte. Er bleibt mit rein technischer Betrachtung unzufrieden und will, dass es nicht nur um die Zugänglichkeit des Quellcodes geht, sondern um etwas größeres, um das Wissen für alle, um die Möglichkeit mit dem Programm alles zu machen, was man sich nur wünschen kann.

# Entwicklungsmodell

Die Open Source als Entwicklungsmodell wurde vom Eric Raymond in seinem Aufsatz „The Cathedral and the Bazaar“ beschrieben. Der wichtige Aspekt ist dabei, dass der Nutzer des Programms selbst mitwirken, alle für ihn notwendigen Änderungen durchführen, testen und dann an die Entwickler weiterleiten kann. Die zahlreichen Nutzer sind in der Lage Fehler zu finden, zu beschreiben, vom Entwickler beheben zu lassen und dann korrigierte Version wieder zu testen. Die Software-Entwicklung nach dem Basarprinzip gibt den Quellcode frei und jeder kann seine Meinung äußern, was eine schnellere Fehlerbehebung liefert als bei der kommerziellen Software-Entwicklung. Die kommerzielle Software-Entwicklung basiert sich auf Bausteinen, mit den zuerst eine komplette Software entwickelt wird und erst danach, wenn man ein Fehler schon in der verkauften Software findet, wird in dem von vorne entwickelten Programm behoben.

Außerdem durch Freigabe des Quellcodes werden freiwillige Mitwirkende gewonnen, was auch zur Verbesserung der Software wird. Raymond versuchte damit die Hackerkultur zu erwecken und diese von einer besseren Seite zu zeigen..

Bezahlt werden die Entwickler entweder direkt vom Unternehmen, das die Open Source Software einsetzt oder es können freiwillige Entwickler sein, die Freude daran finden und keine Kosten für ihre Arbeit verlangen.

Da die meisten Open-Source-Projekte eine modulare Architektur besitzen, können die Nutzer weitere Funktionen einführen, ohne dabei Kernfunktionen ändern zu müssen.

Die Organisation der Projekte erfolgt über das Internet. Neue Versionen, Änderungen und Vorschläge kann man auf entsprechenden Seiten finden.

*"Der wichtigste Motivationsfaktor ist natürlich der gemeinsame Erfolg, aber die Mitglieder des Kern-Teams müssen auch sicher stellen, dass die Arbeit jedes einzelnen entsprechend gewürdigt wird, und dass die Leute, deren Kenntnisse und Qualifikationen gut zusammenpassen, zueinander gebracht werden, damit sie am gleichen Teilbereich des Projektes arbeiten können." [10]*

Bruce Perens beschreibt 2007 in seinem Artikel „Open Source - ein aufstrebendes



ökonomisches Modell“ wie und warum Open Source lebens- und konkurrenzfähig ist. Er beschreibt und analysiert von allen Seiten die Vorgehensweise der Herstellung von Open-Source-Software und erarbeitet das Konzept der risikolosen Softwareentwicklung ohne große Investitionen zu machen, was OSI bereits 1998 in der „Revolution“ in der Bewegung bereits angekündigt hatte. Er zeigt, dass Open Source sich selbst finanziert und wirtschaftlich gesehen nach dem kapitalistischen Prinzip funktioniert.

Die Kosten und Risiken der Entwicklung werden unter den Programmierern aufgeteilt und der Betrieb selbst muss keine großen Kosten tragen. Beim Kauf einer der fertigen kommerziellen Software ist das im Gegenteil: man bezahlt viel und die Behebung von Fehlern, Wartung und Pflege der Software ist kaum möglich.

# Ergebnisse

Freie Software ist neben der proprietären weit verbreitet und man merkt, wie das in Informatikerkreisen zur Mode geworden ist. Nicht nur bei den kommerziellen Organisationen wird die Freie Software eingesetzt, sondern auch viele Softwarehersteller geben ihre Produkte frei, Hardwarehersteller verkaufen ihre Produktion nicht nur mit Treiber für Windows, sondern auch für Linux.

Freie Software hat riesige Vorteile im Gegensatz zur nichtfreier. Das sind unter anderem die geringen Kosten, hohe Sicherheit, Stabilität und natürlich über die Zeit gewonnenes Vertrauen.

Viele Unternehmen und Behörden in Europa steigen auf Open-Source-Software um, viele Schulen und Universitäten setzen diese ein. Die Bundesregierung hatte im Jahr 2000 allen Bundesbehörden den Einsatz freier Software empfohlen. Der Deutsche Bundestag hatte am 9. November 2001 beschlossen, die Freie Software in der Bundesverwaltung zu fördern.

*„Darüber hinaus hat der Deutsche Bundestag am 9. November 2001 den Beschluss gefasst, offene Software in der Bundesverwaltung zu fördern. (Ausführliche Darstellung)*

*Ebenso hat der Bayerische Rechnungshof in seinem Abschlussbericht für das Jahr 2001 (ab Seite 63) auf insgesamt 11 Seiten (!) dringend den bevorzugten Einsatz von OSS in der bayerischen Landesverwaltung empfohlen.*

*Mit der Umstellung ist zwischenzeitlich in vielen Behörden begonnen worden, durchweg sind vorliegende Erfahrungsberichte unter Kosten-, Sicherheits- und Stabilitätsaspekten positiv.*

*Einige der in allen Dokumenten genannten Gründe für die Umstellung sind:*

*Kosten: Open-Source-Software (OSS) ist regelmässig lizenzkostenfrei erhältlich. Das Bundesinnenministerium ermittelte in o.g. Empfehlung eine Kostenersparnis von über DM 800 pro Arbeitsplatz!*

*Sicherheit: Bestandteil von OSS ist ein Betriebssystem (z.B. Linux, OpenBSD, etc.), das u.a. aufgrund seiner Mehrbenutzermöglichkeiten als ausgesprochen sicher gilt.*

*Stabilität: OSS und insbesondere das zugrunde liegende Betriebssystem gelten als sehr stabil.*

*Vertrauen: Von (fast) jedem Programm sind die Quellen offen gelegt. Unerwünschte "Nebeneffekte", die häufig bei nicht offengelegter Software zu beobachten sind, können so wesentlich einfacher ausgeschlossen werden. „ [9]*

# Quellen

## *Internetverzeichnis:*

[1]

Open source - kurz & gut 1. Auflage, Mai 1999 O'Reilly Verlag

[http://www.oreilly.de/german/freebooks/os\\_tb/toc.html](http://www.oreilly.de/german/freebooks/os_tb/toc.html)

[2]

Stallman, Richard: Warum "Open Source" am Kern der Sache vorbei geht

URL: <http://www.gnu.org/philosophy/>

[3]

[www.opensource.org](http://www.opensource.org)

[4]

„Software-Lizenzen von Richard Stallman über duale Lizenzmodelle bis zum GPL-Java Lizenz zum Geldverdienen“ von Glyn Moody

<http://www.linux-magazin.de/Online-Artikel/Lizenz-zum-Geldverdienen/%28kategorie%29/392>

[5]

Die offizielle Seite von Der Free Software Foundation Europe (FSFE)

<http://fsfe.org/>

[6]

Wikipedia - Die freie Enzyklopädie,

URL: <http://de.wikipedia.org/wiki> (Stand: 18.08.2009)

[7]

Joachim Korb's blog

Geschichte der Softwareprogrammierung: „Freie Software für Freiheit und Gerechtigkeit“

[http://www.perspektive89.com/2006/12/21/geschichte\\_der\\_softwareprogrammierung\\_fur\\_eie\\_software\\_fur\\_freiheit\\_und\\_gerechtigkeit#\\_ftn1](http://www.perspektive89.com/2006/12/21/geschichte_der_softwareprogrammierung_fur_eie_software_fur_freiheit_und_gerechtigkeit#_ftn1)

[8]

Lizenzen bestimmen, was Open-Source-Software ausmacht

Copyright oder Copyleft

Von Till Jaeger\*

[http://www.ifross.de/ifross\\_html/art6.html](http://www.ifross.de/ifross_html/art6.html)

[9]

Open Source Software - warum?

<http://www.bettercom.de/de/opensource>

**Literaturverzeichnis:**

[10]

Dalheimer, Kalle: Organisation eines Open-Source-Projekts

Hannover: Verlag Heinz Heise. 1999

[11]

Perens, Bruce: Open Source - ein aufstrebendes ökonomisches Modell, Open Source Jahrbuch 2007.

[12]

Freyermuth, Gundolf: Offene Geheimnisse - Die Ausbildung der Open-Source-Praxis, Open Source Jahrbuch 2007.

[13]

Stallman, Richard: Warum "Open Source" das Wesentliche von "Freier Software" verdeckt, Open Source Jahrbuch 2007.