



# **Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte**

Eine Studie im Rahmen des Projektes NOW: Nutzung des Open Source Konzepts in  
Wirtschaft und Industrie

Förderkennzeichen: 01 ISB04D

Jiayin Hang und Dr. Heidi Hohensohn

## **C-LAB Report**

Vol. 2 (2003) No. 2

Cooperative Computing & Communication Laboratory

**ISSN 1619-7879**

C-LAB ist eine Kooperation  
der Universität Paderborn und der Siemens Business Services GmbH & Co OHG  
[www.c-lab.de](http://www.c-lab.de)  
[info@c-lab.de](mailto:info@c-lab.de)

## **C-LAB Report**

Herausgegeben von  
Published by

Dr. Wolfgang Kern, Siemens Business Services GmbH & Co OHG  
Prof. Dr. Franz-Josef Rammig, Universität Paderborn

Das C-LAB - Cooperative Computing & Communication Laboratory - leistet Forschungs- und Entwicklungsarbeiten und gewährleistet deren Transfer an den Markt. Es wurde 1985 von den Partnern Nixdorf Computer AG (nun Siemens Business Services GmbH & Co OHG) und der Universität Paderborn im Einvernehmen mit dem Land Nordrhein-Westfalen gegründet.

Die Vision, die dem C-LAB zugrunde liegt, geht davon aus, dass die gewaltigen Herausforderungen beim Übergang in die kommende Informationsgesellschaft nur durch globale Kooperation und in tiefer Verzahnung von Theorie und Praxis gelöst werden können. Im C-LAB arbeiten deshalb Mitarbeiter von Hochschule und Industrie unter einem Dach in einer gemeinsamen Organisation an gemeinsamen Projekten mit internationalen Partnern eng zusammen.

C-LAB - the Cooperative Computing & Cooperation Laboratory - works in the area of research and development and safeguards its transfer into the market. It was founded in 1985 by Nixdorf Computer AG (now Siemens Business Services GmbH & Co OHG) and the University of Paderborn under the auspices of the State of North-Rhine Westphalia.

C-LAB's vision is based on the fundamental premise that the gargantuan challenges thrown up by the transition to a future information society can only be met through global cooperation and deep interworking of theory and practice. This is why, under one roof, staff from the university and from industry cooperate closely on joint projects within a common research and development organization together with international partners. In doing so, C-LAB concentrates on those innovative subject areas in which cooperation is expected to bear particular fruit for the partners and their general well-being.

**ISSN 1619-7879**

C-LAB  
Fürstenallee 11  
33102 Paderborn  
fon: +49 5251 60 60 60  
fax: +49 5251 60 60 66  
email: [info@c-lab.de](mailto:info@c-lab.de)  
Internet: [www.c-lab.de](http://www.c-lab.de)

© Siemens Business Services GmbH & Co. OHG und Universität Paderborn 2003  
Alle Rechte sind vorbehalten.

Insbesondere ist die Übernahme in maschinenlesbare Form sowie das Speichern in Informationssystemen, auch auszugsweise nur mit schriftlicher Genehmigung der Siemens Business Services GmbH & Co. OHG und der Universität Paderborn gestattet.  
All rights reserved.

In particular transfer of data into machine readable form as well as storage into information systems, (even extracts) is only permitted prior to written consent by Siemens Business Services GmbH & Co. OHG and Universität Paderborn.

# **Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte**

## **Eine Studie im Rahmen des Projektes NOW: Nutzung des Open Source Konzepts in Wirtschaft und Industrie**

*Jiayin Hang und Dr. Heidi Hohensohn*

*jiayin.hang/heidi.hohensohn@c-lab.de*

*Siemens Business Services GmbH & Co. OHG, C-LAB*

*Fürstenallee 11, 33102 Paderborn*

## Inhaltsverzeichnis

1	Einleitung.....	6
2	Verschiedene Softwarevarianten.....	7
2.1	Public Domain.....	7
2.2	Freeware.....	8
2.3	Shareware.....	8
2.4	Kommerzielle Software .....	9
2.5	Proprietäre Software .....	9
3	Open Source Software Definition .....	10
4	Die rechtlichen Aspekte von Open Source Software.....	14
4.1	Gewährleistung/Haftung .....	14
4.2	Patentrecht.....	16
4.3	Urheberrecht .....	17
4.4	Geografische Eingrenzung der Rechtsgrundlagen .....	18
4.5	Allgemeine Rechtsfragen .....	18
5	Verschiedene Lizenzarten .....	20
5.1	Was ist eine Lizenz? .....	20
5.2	Welche Lizenzen sind verbreitet? .....	20
5.3	Copyleft-Lizenzen .....	21
5.4	Verschiedene Kategorisierungen von Lizenzen .....	22
5.5	Übersicht der Open Source Lizenzen .....	30
6	Vorteile/ Stärken von Open Source Software.....	34
6.1	Höhere Qualität.....	34
6.2	Wiederverwendbarkeit von OSS .....	34
6.3	Höhere Reife der Software.....	35
6.4	Performance .....	35
6.5	Offenheit/Flexibilität .....	37

6.6	Stabilität .....	38
6.7	Preisvorteil .....	39
7	Nachteile/Schwächen von OSS.....	40
7.1	Gewährleistung/ Haftung.....	40
7.2	Dauerhafte Nutzungsmöglichkeit und regelmäßiges Update .....	41
7.3	Dokumentation.....	41
7.4	Bedienbarkeit .....	42
7.5	Probleme beim Datenaustausch .....	42
7.6	Mangel an Applikationen.....	43
7.7	Investitionsschutz.....	44
7.8	Administrierbarkeit .....	45
8	Open Source Software Produkte/ Projekte.....	47
8.1	GNU .....	47
8.2	FreeBSD .....	48
8.3	Linux .....	49
8.4	DNS und BIND.....	50
8.5	Sendmail.....	51
8.6	Apache.....	53
8.7	Samba .....	54
8.8	Perl .....	55
8.9	Mozilla.....	55
8.10	StarOffice .....	57
8.11	PHP .....	57
8.12	Python.....	57
8.13	Tcl/Tk .....	58
8.14	Das K Desktop System .....	58
8.15	GNOME .....	59
8.16	GIMP.....	59
8.17	PGP .....	59
8.18	Groff, SP und Jade .....	59

9	Open-Source-Portale.....	60
9.1	Der OSS Mediator SourceForge .....	60
9.2	Das OSS-Portal BerliOS .....	62
10	Ergebnisse aus vorhandenen Studien.....	67
10.1	Ergebnisse aus der FLOSS-Studie .....	67
10.1.1	Nutzungsbereitschaft von OSS in verschiedenen Ländern .....	67
10.1.2	Motivationen für den Einsatz von Open Source Software .....	68
10.1.3	Vorteile von Open Source Software in verschiedenen Anwendungsbereichen.....	71
10.2	Studie über die Kosten von Open Source Software.....	73
10.2.1	Studie der Robert Frances Group (RFG) .....	74
10.2.2	Studie im Auftrag der BMWi.....	76
10.2.3	Studie von IDC .....	80
10.3	Studie über die Integrationshäuser bezüglich ihre Aktivitäten im Bereich Open Souce Software.....	81
10.4	Studie über die Software- und Hardwareanbieter bezüglich ihre Aktivitäten im Bereich Open Source Software .....	83
11	Wertschöpfungsprozesse bei IT-Unternehmen .....	87
12	Die Prozess-Beteiligten des OSS Marktes .....	89

## Abbildungsverzeichnis:

Abbildung 1: Beschreibung der verschiedenen Softwarevarianten .....	10
Abbildung 2: Copyleft-Lizenzen und Non-Copyleft-Lizenzen .....	22
Abbildung 3: Darstellung der Eigenschaften der verschiedenen Lizenzen .....	30
Abbildung 4: Vergleich der verschiedenen Open Source Lizenzen .....	33
Abbildung 5: Datendurchsatz von Linux/Samba im Vergleich zu Windows .....	37
Abbildung 6: Vergangene und zukünftige Entwicklung von Linux.....	49
Abbildung 7: Entwicklungsprozess bei SourceForge .....	61
Abbildung 8: Allgemeine Software Value Chain.....	87
Abbildung 9: Die Beziehungen zwischen den Parteien im Open Source Prozess .....	91

## 1 Einleitung

In dieser Studie wird ein Überblick über eine Reihe relevanter Aspekte des kommerziellen Einsatzes von Open Source gegeben. Die Übersicht richtet sich insbesondere an Interessierte, die einen einfachen und schnellen Überblick über wesentliche Eigenschaften und Restriktionen des populären Begriffes Open Source suchen.

Ziel im Rahmen des Projektes NOW ist es Geschäftsmodelle für Open Source Software (OSS) zu entwickeln, die die hier vorgestellten Chance und Risiken berücksichtigen. In diesem ersten Teil werden hierzu die wirtschaftlichen, technischen, rechtlichen und organisatorischen Aspekte des Open Source-Ansatzes diskutiert. Ziel der Studie ist es, heraus zu finden, wie alle Beteiligten durch den Open Source Ansatz profitieren können. Dazu werden Geschäftsmodelle für OSS erarbeitet, die auch für die Unternehmen Erfolg versprechend und Gewinn bringend sind.

In dieser Studie werden zunächst die verschiedenen Softwarevarianten vorgestellt, um die Besonderheiten der Open Source Software hervorzuheben und von anderen Softwarearten abzugrenzen. Anschließend werden die rechtlichen Aspekte der OSS diskutiert, wobei in dieser Studie keine fertige Lösungskonzepte zu den rechtlichen Fragestellungen erarbeitet werden, sondern lediglich ein Problemaufriss der rechtlichen Aspekten, also eine Aufzählung der noch offenen, zu klärenden Problemstellungen. Im nächsten Kapitel wird versucht die verschiedenen OS-Lizenzen anhand ihrer besonderen Eigenschaften zu kategorisieren. Die fünf wichtigsten und weitverbreitesten Lizenzen werden näher untersucht, das heißt welche Konsequenzen sie für die Software hat und welche Rechte und Verpflichtungen sich daraus für den Nutzer ergeben. Danach werden die herausragenden Vorteile bzw. Nachteile der OSS gegenübergestellt. Im achten Kapitel werden exemplarisch ein paar bekannte OS-Produkte bzw. Projekte vorgestellt, bei diesen Projekten werden die jeweiligen Entstehung und Vorgehensweise beschrieben sowie die weitere Entwicklung prognostiziert. Die OS-Portale SourceForge und BerliOS werden



analysiert. Im Kapitel zehn werden die Ergebnisse von verschiedenen bereits durchgeführten Studien zu OSS vorgestellt. In der FLOSS-Studie wurde auch eine Umfrage zu OSS durchgeführt, in dieser Umfrage wurden verschiedene Länder hinsichtlich ihrer Anwendungsbereiche und Motivationen für den Einsatz von OSS befragt. Außerdem wurden die Kosten von OSS analysiert, dabei wurden die Total Costs of Ownership (TCO) von OSS mit denen der proprietären Software gegenübergestellt. Zuletzt werden die Integrationshäuser und Software- und Hardwareanbieter auf ihrer Aktivitäten im OS-Bereich hin untersucht. In Kapitel 11 und 12 werden die Elemente des Wertschöpfungsprozesses und die verschiedenen Rollen im Open Source Prozess vorgestellt, als Basis zu den konkreten Geschäftsmodellen der OS-Marktbeteiligten, die in der nachfolgenden zweiten Studie im Herbst 2003 erscheint.

## 2 Verschiedene Softwarevarianten

Im folgenden Kapitel werden die verschiedenen Softwarevarianten vorgestellt. Um die teilweise vorhandene Unklarheit zwischen den verschiedenen Softwarearten zu klären, wird in diesem Kapitel die Eigenschaften der verschiedenen Arten von Software vorgestellt und von denen der freien Software/ Open Source Software abgegrenzt.

### 2.1 Public Domain

Entscheidet sich der Autor dazu, sein Werk als Public Domain zu erklären, so tritt er all seine Rechte und Ansprüche darauf an die Allgemeinheit ab. Sie unterliegt nun keinem Copyright oder anderen rechtlich bindenden (Lizenz-)Bedingungen mehr; sie ist Gemeingut. Die Software kann beliebig modifiziert, weiterverbreitet, verkauft und auch in nicht-freie Software umgewandelt werden. Allerdings wird der Quellcode in den seltensten Fällen mitgeliefert, so dass Änderungen nicht möglich sind.<sup>1</sup>

---

<sup>1</sup> Vgl. n.n., Open Source kurz & gut, 1999, S. 4

## 2.2 Freeware

Zwar ist Freeware kein eindeutig definierter Begriff, meint aber im allgemeinen proprietäre Software von Herstellern, die ihr Produkt kostenlos verbreiten. Das Kopier- und Weiterverbreitungsverbot ist somit nicht mehr gültig, wohl aber alle anderen einschränkenden Lizenzbedingungen. Die Veränderung ist nicht erlaubt und auch nicht möglich, da der Quellcode nicht zur Verfügung steht. Freeware ist somit keine freie Software.

Softwarehersteller vertreiben häufig ihr Produkt als Freeware, wenn sie auf Rückflüsse verzichten, andere Prioritäten setzen oder die Verbreitung einen strategischen Marktvorteil erwirkt. Bekannte Beispiele hierfür sind der Internet Explorer von Microsoft oder der Acrobat Reader von Adobe.<sup>2</sup>

## 2.3 Shareware

Als Shareware wird diejenige Software bezeichnet, die zwar frei kopiert und verbreitet, aber nur zu einem bestimmten Zweck (beispielsweise für den privaten Einsatz) oder über einem gewissen Zeitraum hinweg kostenlos genutzt werden kann. Nach dieser Testperiode muß der Benutzer, falls er die Software weiterhin nutzen möchte, eine Lizenzgebühr an den Autor zahlen. Die sonstigen Lizenzbedingungen entsprechen weitgehend der von proprietärer Software. Shareware wird in den meisten Fällen im Internet zur Verfügung gestellt oder als Sammlung auf CD-ROMs gegen eine geringe Gebühr verkauft. Er hat somit keine Möglichkeit, die Anwender seines Programms zu identifizieren und eine Zahlung nach Ablauf der Testzeit zu erzwingen. Um einer geringen Zahlungsmoral der Anwender vorzubeugen, ist Shareware oft so modifiziert, dass sie während der Testzeit nur eingeschränkten Funktionsumfang bietet und/oder danach ihren Dienst von selbst einstellt.

Des Shareware-Modells bedienen sich häufig Privatleute oder kleinere Firmen, um auf sich aufmerksam zu machen oder einen kleinen Nebenverdienst zu erwirken. Einige Programme, die ihr Dasein als Shareware begonnen haben

---

<sup>2</sup> Vgl. n.n., Open Source kurz & gut, 1999, S. 5

(und teilweise immer noch sind), sind heute weltbekannt, beispielsweise WinZIP, das Zeichenprogramm Paint Shop Pro oder das Spiel Doom.<sup>3</sup>

## 2.4 Kommerzielle Software

Kommerzielle Software wird meist in Unternehmen entwickelt, um einen Profit zu erwirtschaften. Fast jede kommerzielle Software ist auch proprietär. Ausnahmen bestätigen die Regel. Ein Beispiel ist das Linux Betriebssystem von Red Hat. Das Linux Paket ist kommerziell, da die Firma Red Hat es mit dem Ziel vertreibt, Gewinn zu erwirtschaften. Andererseits ist es freie Software, da dessen Quellcode frei verfügbar ist.<sup>4</sup>

## 2.5 Proprietäre Software

Proprietäre (herstellerspezifische) Software kann als das Gegenteil von freier Software bezeichnet werden. Ihre Vervielfältigung, Weiterverbreitung und Modifizierung ist untersagt. Der Gebrauch ist ebenfalls verboten, solange ein Kunde die Software nicht lizenziert. Eigentümer der Software ist dann aber immer noch nicht der Anwender – er ist „nur“ Eigentümer einer Lizenz, sondern weiterhin der Hersteller, denn er hält darauf das Urheberrecht. Ihm obliegt die vollständige Kontrolle über das Produkt.

Oft werden zusätzlich zur eigentlichen Software Support- oder andere Serviceverträge abgeschlossen, die eine technische Unterstützung für den Kunden bieten und eventuell Erweiterungen und Verbesserungen des Produkts ermöglichen.

Dem Hersteller proprietärer Software stehen Mittel zur Verfügung, den widerrechtlichen Gebrauch seiner Software (z.B. „Raubkopieren“) zu unterbinden. Er kann digitale Signaturen verwenden, um die Authentizität des Produkts zu garantieren oder Kopierschutzmechanismen einbauen, um der Vervielfältigung Einhalt zu gebieten. Beispiele aus der Musikindustrie (siehe Napster oder MP3.com), zeigen (auch wenn die Märkte sich nicht direkt

---

<sup>3</sup> Vgl. n.n., Open Source kurz & gut, 1999, S. 5

<sup>4</sup> Vgl. n.n., Open Source kurz & gut, 1999, S. 6

vergleichen lassen), dass Lizenzierung und Urheberrechte ein Milliardengeschäft sind und mit viel Aufwand versucht wird, dieses Geschäft zu wahren.<sup>5</sup>

Die folgende Tabelle veranschaulicht die einzelnen Kriterien der verschiedenen Softwarevarianten.

	Kostenlos <sup>6</sup>	Weiterverbreitbar	Uneingeschränkt nutzbar	Quelltext (modifizierbar)	Lizenzierung
Public Domain	x	x	x	x <sup>7</sup>	
Freie Software/ OS	x	x	x	x	x
Freeware	x	x			x
Shareware		x			x
Proprietäre Software					x

Quelle: n.n., Open Source kurz & gut, 1999, S. 7

Abbildung 1: Beschreibung der verschiedenen Softwarevarianten

### 3 Open Source Software Definition

Open Source Software (OSS) sind Softwareprogramme, die kostenlos der Allgemeinheit zur Verfügung gestellt werden. Der Quellcode der Softwareprogramme wird offen gelegt, und der Benutzer darf den Quelltext modifizieren, vervielfältigen und kostenlos weiterverbreiten. Die neue modifizierte Version der Software sollte wieder der Öffentlichkeit zur Verfügung gestellt werden, allerdings hängt das auch von den verschiedenen Lizenzbedingungen ab. Bei der GNU General Public License (GPL) muss die veränderte Software

---

<sup>5</sup> Vgl. n.n., Open Source kurz & gut, 1999, S. 6

<sup>6</sup> Kostenlos wird hier auf die Kostenfreiheit der Lizenz bezogen. Kosten für Versionierung, Handling, Dokumentation oder ergänzende Arbeiten können durchaus entstehen (a.A).

<sup>7</sup> Im allg. Ist hier der Quellcode aber nicht verfügbar (a.A.)

wieder unter der GPL gestellt werden<sup>8</sup>. Andere Lizenzen erlauben auch eine kommerzielle Weiterverwendung der veränderten Software, wie z. B. die Mozilla Public License (MPL). Unter kostenlos wird verstanden, dass für die Lizenzen keine Gebühren erhoben werden dürfen, aber für das Kopieren, Packaging, Dokumentation und sonstige Leistungen können Einnahmen erzielt werden, deshalb kann OSS durchaus kommerziell genutzt werden.

Das Phänomen Open Source Software hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Open Source Software wird sowohl im Unternehmen als auch im privaten Gebrauch immer öfter eingesetzt. Durch die große Beliebtheit des Open Source Konzepts beim Endanwender stellt sie eine immer größere Konkurrenz für proprietäre Software dar. Insbesondere von Seiten großer Softwarefirmen (wie z. B. Microsoft) entwickelt sich aus Gründen der Marktanteilssicherung zum Teil ein besonderer Widerstand gegen das OS-Konzept.

Die Einsatzgebiete von Open Source Software im Unternehmen und im privaten Gebrauch sind sehr groß, sie reichen vom Front-end bis zum Back-end. Mit OSS können alle Bereiche abgedeckt werden, sie umfassen den Desktop-Bereich, Server-Bereich und den Embedded-Bereich. Im Server-Bereich werden OSS als Betriebssysteme (Linux), Web Server, Application Server und Datenbanken eingesetzt. Im Desktop-Bereich wird OSS als Betriebssystem (Linux), Office-Anwendung (StarOffice), Bildbearbeitung, Datenbank, Internet-Anwendung und Entwicklungsumgebung angewandt. Auch im Embedded-Bereich wird immer mehr OSS eingeführt, z. B. als Betriebssysteme, Anwendungsprogramme und Bedieneroberflächen.

---

<sup>8</sup> Was hierbei eine Ergänzung – und damit zwingend unter GPL - darstellt und was ggf. eine selbständige Software ist, die unter eine frei zu wählende Lizenz gestellt werden kann, ist in der GPL definiert. Im Falle einer Weitergabe der modifizierten Software ist auch der Sourcecode weiterzugeben oder dies anzubieten. Eine Beschränkung der Weitergabe durch Dritte ist unzulässig. Unternehmen können dabei aber u.U. als eine einheitliche Rechtsperson verstanden werden.

## Die Open Source Definition

Mit dem Erfolg von Open Source Software Produkten veröffentlichen immer mehr Software-Firmen einen Teil ihres bisher proprietären Quellcodes. Allerdings nutzen sie dafür nicht immer die General Public Licence, sondern entwickeln verstärkt eigene Lizenzen (z. B. Netscape Public Licence, Sun Community Source Licence etc.). Um entscheiden zu können, welche Software tatsächlich frei ist, entwickelte die Open-Source-Initiative einen Katalog von rechtlichen Elementen, die eine Lizenz enthalten muss, damit die betreffende Software gemäß der Open-Source Definition als frei gilt. Wesentlich ist natürlich die Frage, nach der alle neuen Lizenzen darauf hin untersucht werden, ob der einmal veröffentlichte Quellcode und alle an ihm verrichtete Arbeit auf Dauer zugänglich bleiben werden.

Im Einzelnen sind folgende Lizenzpunkte der „Open Source Definition“ festgelegt:

### 1. Freie Weiterverbreitung

Die Lizenz darf niemanden im Verkauf oder in der Weitergabe der Software als Teil einer aus verschiedenen Quellen zusammengesetzten Softwaredistribution einschränken. Die Lizenz darf keinerlei Lizenzgebühren oder andersartige Beiträge verlangen.<sup>9</sup>

### 2. Quellcode

Das Programm muss den Quellcode beinhalten und die Verbreitung sowohl als Quellcode als auch in kompilierter Form gestatten. Wird ein Teil des Produkts nicht mit Quellcode verbreitet, so muss auf eine Möglichkeit, den Quellcode gebührenfrei aus dem Internet downzuloaden, ausdrücklich hingewiesen werden. Der Quellcode muss in einer Form zur Verfügung gestellt werden, in der ein Programmierer ihn verändern kann. Absichtlich verwirrend geschriebener Quellcode ist nicht erlaubt. Ebenso sind Zwischenformen, wie die Ausgabe eines Präprozessors oder eines Übersetzers, verboten.<sup>10</sup>

---

<sup>9</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>10</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

### **3. Auf dem Programm basierende Werke**

Die Lizenz muss die Veränderung des Programms, auf dem Programm basierende Werke sowie deren Verbreitung unter den gleichen Lizenzbedingungen gestatten.<sup>11</sup>

### **4. Die Unversehrtheit des Originalcodes**

Die Lizenz darf die Verbreitung von modifiziertem Quellcode nur dann einschränken, wenn sie die Verbreitung von so genannten Patchdateien in Verbindung mit dem Originalcode gestattet, damit das Programm vor der Benutzung verändert werden kann. Die Lizenz muss ausdrücklich die Verbreitung von Software erlauben, die mit verändertem Quellcode erstellt wurde. Die Lizenz darf allerdings von auf dem Programm basierenden Werken verlangen, einen von der Originalsoftware verschiedenen Namen oder eine andere Versionsnummer zu tragen.<sup>12</sup>

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

Die Lizenz darf keinerlei Personen oder Personengruppen diskriminieren.<sup>13</sup>

### **6. Keine Einschränkungen für bestimmte Anwendungsbereiche**

Die Lizenz darf niemanden in der Benutzung des Programms in einem bestimmten Einsatzgebiet einschränken. Sie darf beispielsweise nicht die kommerzielle Nutzung oder die Benutzung in der Genforschung verbieten.<sup>14</sup>

### **7. Verbreitung der Lizenz**

Die zum Programm gehörigen Rechte müssen für jeden gelten, der das Programm erhalten hat, ohne dass eine weitere Lizenz beachtet werden muss.<sup>15</sup>

### **8. Die Lizenz darf nicht für ein bestimmtes Produkt gelten**

Die zum Programm gehörigen Rechte dürfen nicht davon abhängen, dass das Programm Teil einer bestimmten Softwaredistribution ist. Wird das Programm

---

<sup>11</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>12</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>13</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>14</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>15</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

außerhalb einer solchen Distribution genutzt oder verbreitet, so gelten für den Benutzer dieselben Rechte, die in der Originaldistribution gewährt werden.<sup>16</sup>

### **9. Die Lizenz darf andere Software nicht beeinträchtigen**

Die Lizenz darf keine andere Software einschränken, die zusammen mit der lizenzierten Software verbreitet wird. Die Lizenz darf beispielsweise nicht verlangen, dass jegliche Software, die auf demselben Datenträger verbreitet wird, Open Source-Software sein muss.<sup>17</sup>

Damit sich eine Software nun „Open Source“ nennen darf, muss die Lizenz, unter der sie veröffentlicht wird, alle Forderungen dieser Open Source Definition erfüllen. Zusammengefasst bedeutet das für Open Source-Software:

- Die Software muss unter einer Lizenz veröffentlicht werden, welche die Open Source Forderungen erfüllt.
- Jeder hat das Recht, die Software nach eigenem Ermessen zu nutzen, das schließt kommerzielle Nutzung ausdrücklich ein.
- Jeder hat das Recht, die Software zu modifizieren und in modifizierter Form weiterzuverteilen.

## **4 Die rechtlichen Aspekte von Open Source Software**

### **4.1 Gewährleistung/Haftung**

Problematisch ist die Frage der Haftung bei Open Source Software.

Grundsätzlich ist sie nach den gesetzlichen Vorschriften und durch die General Public Licence (GPL) geregelt. Nach BGB wird beim Herunterladen von Software aus dem Internet von einer Schenkung ausgegangen, wobei der Schenker außer durch grobe Fahrlässigkeit oder arglistiges Verschweigen eines Fehlers nicht haftbar gemacht werden kann.

In der GPL ist die Frage der Gewährleistung aber auch noch einmal explizit geregelt:

---

<sup>16</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition

<sup>17</sup> Vgl. Open Source Initiative (OSI): The Open Source Definition



„§11. Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, „wie es ist“, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich - aber nicht begrenzt auf - Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.“<sup>18</sup>

§12. In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar...“<sup>19</sup>

Falls also ein Programm fehlerhaft ist, können für den Nutzer erhebliche Kosten entstehen, die er selbst zu tragen hat.

Gewährleistung bedeutet nach deutschem Zivilrecht: Es wird eine vertragliche Haftung dafür übernommen, dass eine Ware oder eine Dienstleistung ohne Mängel erbracht wird. Für Software heißt dies, dass das jeweilige Programm ordnungsgemäß funktionieren muss.

Solange Software, die unter einer Open-Source-Lizenz steht, kostenlos zum Download angeboten wird, ist die Rechtslage in den meisten Fällen klar.

Schwieriger stellt sich die Sachlage für Distributoren dar. Wenn sie Software-Programme auf Datenträgern, eventuell noch zusammen mit Handbüchern und/oder Support verkaufen, so ist eine Gewährleistung nach deutschem Recht möglich.

Der Begriff „Haftung“ geht noch weiter: Prinzipiell haftet jeder für sein Tun und Lassen, um Haftungsfragen geht es etwa, wenn eine Software Viren enthält.

---

<sup>18</sup> Die General Public License (GPL) Definition, Abschnitt Gewährleistung und Haftung

<sup>19</sup> Die General Public License (GPL) Definition, Abschnitt Gewährleistung und Haftung

Auch hier gilt das Schenkungsrecht: Derjenige, der ein Programm verschenkt, haftet nur, wenn er grob fahrlässig gehandelt hat.<sup>20</sup>

Die GPL sieht eine Haftungs- und Gewährleistungsausschluss vor. Es stellt sich die Frage, ob sie auch auf das deutsche Recht anwendbar ist und Gültigkeit hat, da die GPL nach dem amerikanischen Recht konzipiert wurde.

Für die Beteiligten der „Open-Source-Kette“, zu der der Urheber des Quellcodes gehört, der sie unentgeltlich zur Verfügung stellt und somit Dritten die Möglichkeit der Modifikation des Quellcodes einräumt, stellt sich die Frage, in welchem Umfang der Urheber bzw. der Dritte für die Schäden haftet bzw.

Gewährleistungsansprüchen ausgesetzt ist, die durch eine fehlerhafte Software verursacht werden. Es ist vor allem interessant, ob der Urheber auch für Schäden haftet, die durch Dritte verursacht werden.

### 4.2 Patentrecht

Software ist in Deutschland urheberrechtlich geschützt, aber als solche bisher nicht patentierbar. Dieses Prinzip kann allerdings umgangen werden, wenn zum Beispiel durch Software ein technisches Verfahren beschrieben wird.

Ein wichtiger Aspekt bei der Herstellung von Software ist der Schutz geistigen Eigentums. Nicht nur bei kommerziellen Programmen haben Entwickler ihre Software patentieren lassen, auch im Open-Source-Bereich wurden die ersten Patente angemeldet.

Das Patentrecht hat das Ziel, die Offenlegung von Erfindungen und Entwicklungen zu bewirken, um den technischen Fortschritt zu fördern.

Diese Regelung, dass Software in Deutschland eigentlich nicht patentierbar ist, wird aber seit einiger Zeit von Firmen umgangen, die ihre Software zum Beispiel zusammen mit Hardware als technische Einheit patentieren lassen.<sup>21</sup>

Die Software-Patente bergen einige Risiken, wie das Beispiel USA zeigt. Die immer ungenauere Handhabung des Patentrechtes führte in den USA zu so genannten Trivialpatenten. Durch die Patentierung immer einfacherer

---

<sup>20</sup> Vgl. Marx, Ben, Linux Manager Guide, 2001, S. 55

<sup>21</sup> Vgl. Marx, Ben, Linux Manager Guide, 2001, S. 62

Algorithmen und immer kleinerer Softwaremodule steigt die Gefahr für Entwickler, unwissentlich Patente zu verletzen und dafür gerichtlich belangt zu werden. Dies hemmt die Entwicklung von Open-Source-Software, sodass einige Programme erst gar nicht veröffentlicht werden. Dadurch wird das vorhandene Innovationspotenzial nicht ausgeschöpft und der Sinn von Patenten verfehlt. Programme und Algorithmen patentieren zu können, ist eine Seite, auf anderer Seite muss man aber auch in der Lage sein, diese Patente zu nutzen. Und an dieser Stelle hapert es noch, denn in den großen Softwarehäusern sind Patentverletzungen an der Tagesordnung, da niemand mehr einen Überblick hat. Wenn eine Firma anfangen würde ihre Patente einzuklagen, würde wahrscheinlich die gesamte Wirtschaft zusammenbrechen. Und das ist die einzige Hoffnung für OSS, denn solange es eine relative Patentgleichheit gibt, wird niemand anfangen seine Rechte einzuklagen.

Was würde aber passieren, wenn die großen Softwarekonzerne IBM, Microsoft usw. sich auf einmal einigen? Dann würden Klagen wegen Patentverletzung eigentlich nichts mehr im Wege stehen und die Zeit der Open Source Software wäre möglicherweise endgültig vorbei.

Dem könnte man möglicherweise mit einem Patentpool für OSS-Projekte entgegenwirken. Allerdings müsste ein solcher Pool erst einmal finanziert werden, denn was nützt es wenn die Patente dann nach 3 Jahren wieder verfallen, weil die Gebühren nicht aufgebracht werden können?<sup>22</sup>

### 4.3 Urheberrecht

Das Urheberrecht ist das eigentumsähnliche Recht des Urhebers an seinem Werk. Als „Werk“ wird danach nur eine persönliche geistige Schöpfung der Literatur, Wissenschaft oder Kunst geschützt. Als Urheber bezeichnet man den Schöpfer des Werkes. Das Urheberrecht umfasst die Urheberpersönlichkeitsrechte (z.B. das Veröffentlichungsrecht) und die Verwertungsrechte (z.B. das Vervielfältigungsrecht). Das

---

<sup>22</sup> Vgl. Weber, N., Suhr, G., Ausarbeitung: Rechtliche Aspekte von OSS, 2001, S. 4

Urheberpersönlichkeitsrecht räumt dem Urheber das Bestimmungsrecht darüber ein, ob und wie sein Werk veröffentlicht wird, d.h. unter welchem Titel und unter welchem Namen. So hat der Urheber eines Werkes zum Beispiel das Recht, die Herausgabe oder Vernichtung von nicht genehmigten Kopien zu verlangen. Außerdem stellt es sicher, dass sein Werk nicht verfälscht wird und dass er immer als der Autor genannt wird.

Das Verwertungsrecht stellt die wirtschaftliche Seite des Urheberrechts dar, es umfasst das Vervielfältigungs-, Verbreitungs- und Ausstellungsrecht.

Das Urheberpersönlichkeitsrecht, kann im Gegensatz zum Verwertungsrecht nicht an Dritte übertragen werden.<sup>23</sup>

Es stellt sich die Frage, ob und inwieweit die OSS unter das Urhebergesetz fällt und wer eigentlich der Urheber eine Open Source Software ist, da oft mehrere Programmierer an der Entwicklung einer Software beteiligt sind.

Da jetzt immer mehr Unternehmen auf das Open Source Phänomen aufmerksam werden und eigene Mitarbeiter mit der Entwicklung von OSS beauftragen, stellt sich die Frage, ob in dem Fall das Unternehmen oder die angestellten Mitarbeiter die Urheber der Software sind.

#### 4.4 Geografische Eingrenzung der Rechtsgrundlagen

Bezüglich der geographischen Einschränkung der Rechtsgrundlagen wird nur das deutsche Recht behandelt. Das ausländische Recht, das ebenfalls die Open Source Software tangieren kann, wird nicht betrachtet.

Man muss auch bedenken, dass in dem Entwicklungsprozess einer Open Source Software oft eine Vielzahl von Personen involviert sind – manchmal hunderte von Programmierer - die oft in verschiedenen Ländern ansässig sind, so dass bei Rechtsstreitigkeiten auch ausländische Rechtsordnungen berührt und berücksichtigt werden müssen.

#### 4.5 Allgemeine Rechtsfragen

---

<sup>23</sup> Vgl. Marx, Ben, Linux Manager Guide, 2001, S. 49

Zu untersuchende Rechtsgebiete: Urheberrecht, Vertrags- und Produkthaftungsrecht, Kartell- und Wettbewerbsrecht.

Es gibt kaum Literatur oder Veröffentlichungen zum Thema rechtliche Aspekte von OSS, und eine entsprechende spezifische Rechtsprechung ist ebenfalls nicht vorhanden.

Es ist zu klären, welchem Vertragstypus eine Open Source Software zuzuordnen ist. Handelt es sich bei OSS wegen der unentgeltlichen Weitergabe des Quellcodes um eine Schenkung, um einen Auftrag oder – aufgrund der Beteiligung von mehreren Personen – eine Gesellschaft bürgerlichen Rechts (GbR).

Die Beantwortung der Vertragstypen hat auch einen wesentlichen Einfluss auf die Haftungsfragen, da die drei verschiedenen Vertragstypen unterschiedliche Regelungen bezüglich Gewährleistung und Haftung beinhalten und damit wird die Frage beantwortet, ob ein Haftungsausschluss überhaupt möglich ist.

Die GPL soll im weiteren Projektverlauf anhand der Vorschriften über Allgemeine Geschäftsbedingungen überprüft werden.

Es bedarf auch der Klärung, inwieweit in einem Open Source Modell die Produzentenhaftung oder das Produkthaftungsgesetz greift. Man muss bei der Haftung zwischen der verschuldensabhängigen Haftung (Produzentenhaftung) und der Garantiehafung (Produkthaftungsgesetz) unterscheiden.

Es stellt sich die Frage, ob eine Open Source Software ein Produkt ist, und ob es von der Haftung ausgeschlossen ist, weil der Hersteller des Produktes nicht die Absicht hat, mit dem Verkauf des Produktes Gewinne zu erwirtschaften.

Allerdings könnte auch ein „Vertrieb mit wirtschaftlichem Zweck“ vorliegen, wenn die Software an sich zwar unentgeltlich zur Verfügung gestellt wird, aber dadurch neue kostenpflichtige Geschäftsfelder erschlossen werden können. Dazu gehören Dienstleistungen wie Consulting, Implementation usw., die in Verbindung mit OSS erbracht werden.

Ebenso müssen kartellrechtliche bzw. wettbewerbsrechtliche Fragestellungen betrachtet werden. Durch den Vergütungsverzicht des Open Source Modells

könnte Open Source Programme in eine Wettbewerbssituation zu den kommerziellen proprietären Programmen geraten.

## 5 Verschiedene Lizenzarten

### 5.1 Was ist eine Lizenz?

Eine Software-Lizenz ist zunächst nichts anderes als ein Vertrag, in dem eine Partei (Lizenzgeber) einer anderen (Lizenznehmer) bestimmte Nutzungsrechte an einer urheberrechtlich geschützten Software überlässt oder beschränkt. Diese Rechte können beinhalten:

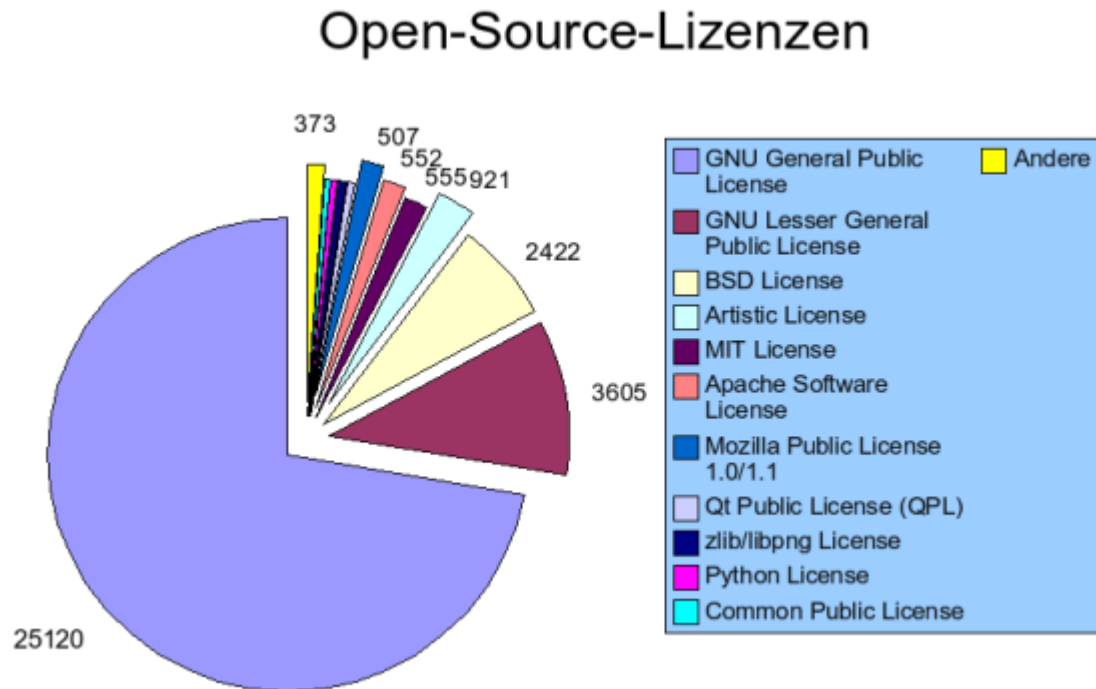
- Allgemeines Nutzungsrecht an der Software
  - o Zahl der Nutzer, Art der Nutzung usw.
- Recht auf Weiterverbreitung der Software
- Recht auf Veränderung der Binärdateien
- Recht auf Veränderung des Quellcodes, sofern vorhanden
- Recht auf Weiterverbreitung veränderter Versionen der Binärdateien oder des Quellcodes
- Recht auf Verbindung ("Linking") der Binärdateien oder des Quellcodes mit anderer Software
  - o statisches Linking vs. dynamisches Linking

Während bei kommerzieller Software Rechte üblicherweise eingeschränkt werden, geht es bei Open Source darum, möglichst vielen Menschen eine Veränderung des Codes zu gestatten und damit eine langfristige Verbesserung der Software zu ermöglichen.

### 5.2 Welche Lizenzen sind verbreitet?

Das SourceForge-Projekt ist mit über 50.000 Projekten der größte Hosting-Anbieter für Open-Source-Software (daneben gibt es andere Anbieter wie GNU

Savannah und BerliOS). Im Januar 2003 sah die Verteilung der Projekte nach Lizenzen wie folgt aus:



Quelle: BerliOS, Openfacts, Open Source Lizenzen

Abbildung: Aufteilung der Open Source Lizenzen

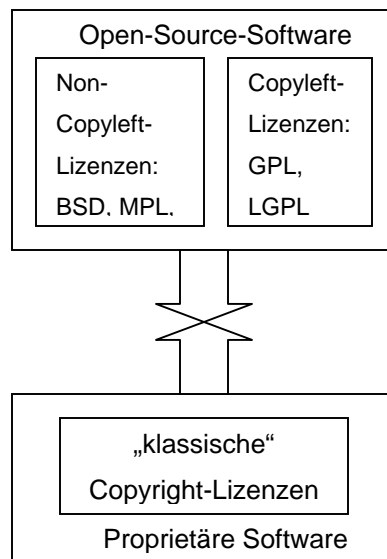
Deutlich erkennbar ist die massive Dominanz der GNU General Public License und ihrer Schwesterlizenz, der GNU Lesser General Public License (LGPL). Die BSD-Lizenz gehört zu den beliebtesten Alternativen, aber auch die Artistic License, die Apache Software License und die noch junge Mozilla-Lizenz sollten nicht unterschlagen werden.

Im Folgenden werden einige ausgewählte Lizenzen näher erläutert.

### 5.3 Copyleft-Lizenzen

Man unterscheidet zwischen zwei Arten von Lizenzen: Copyright-Lizenzen und Copyleft-Lizenzen. Die Copyright-Lizenzen werden bei proprietäre Software angewandt, wodurch die Software urheberrechtlich geschützt wird. Copyleft-Lizenzen beziehen sich auf Open-Source-Software, wobei man eine Unterscheidung macht zwischen Non-Copyleft-Lizenzen und Copyleft-Lizenzen.

Copyleft-Lizenz bedeutet, dass die Modifikationen einer Open Source Software wieder unter dieselbe Lizenz gestellt werden muss, bzw. dass der Quellcode wieder veröffentlicht werden muss. Diese Bedingung wird bei „Non-Copyleft“-Lizenz nicht gestellt.<sup>24</sup>



**Quelle: Marx, Ben, Linux Manager Guide, 2001, S. 47**

**Abbildung 2: Copyleft-Lizenzen und Non-Copyleft-Lizenzen**

### 5.4 Verschiedene Kategorisierungen von Lizenzen

Die Lizenzen können entsprechend ihrem lizenzrechtlichem Charakteristika in fünf verschiedene Gruppen eingeteilt werden:

- Lizenzen ohne Copyleft-Effekt
- Lizenzen mit strengem Copyleft-Effekt
- Lizenzen mit beschränktem Copyleft-Effekt
- Lizenzen mit Wahlmöglichkeiten
- Lizenzen mit Sonderrechten

Im Folgenden werden die Lizenzarten näher erläutert und Beispiel-

---

<sup>24</sup> Vgl. Marx, Ben, Linux Manager Guide, 2001, S. 47



Lizenzen genannt.<sup>25</sup>

### 1. Lizenzen ohne Copyleft-Effekt

Lizenzen ohne Copyleft-Effekt zeichnen sich dadurch aus, dass sie dem Lizenznehmer alle Freiheiten einer Open Source Lizenz einräumen und für Veränderungen der Software keine Bedingungen hinsichtlich des zu verwendenden Lizenztyps enthalten. Damit kann der Lizenznehmer veränderte Versionen der Software unter beliebigen Lizenzbedingungen weiterverbreiten, also auch in proprietäre Software überführen.

Beispiellizenzen dafür sind:

- BSD License (original)
- Apache License (v. 1.1)
- MIT License
- OpenLDAP Public License (v. 2.7)
- Zope Public License (v. 2.0)

### 2. Lizenzen mit strengem Copyleft-Effekt

Bei Lizenzen mit einem strengen Copyleft-Effekt wird der Lizenznehmer verpflichtet, von der ursprünglichen Software abgeleitete Werke ebenfalls nur unter den Bedingungen der Ursprungslizenz weiterzuverbreiten. Die hier aufgeführten Lizenzen sind deswegen aber nicht schon unbedingt "GPL-kompatibel".

Beispiele dafür sind:

- GNU General Public License (GPL) (v.2.0)
- IBM Public License
- Common Public License
- Red Hat eCos Public License (v. 1.1)
- Open Software License

### 3. Lizenzen mit beschränktem Copyleft-Effekt

---

<sup>25</sup> Lizenz Center, Auflistung der Lizenzen [http://www.ifross.de/ifross\\_html/lizenzcenter.html](http://www.ifross.de/ifross_html/lizenzcenter.html)

Lizenzen mit beschränktem Copyleft-Effekt gleichen denen mit strengem Copyleft-Effekt. Sofern Modifikationen der Software in eigenen Dateien realisiert werden, können diese Dateien auch unter anderen, z.B. proprietären Lizenzbedingungen, weiterverbreitet werden. Damit soll die Kombination von Software unter verschiedenen Lizenztypen erleichtert werden.

Beispiele dafür sind:

- Mozilla Public License
- GNU Lesser General Public License (LGPL) (v. 2.1)
- Nokia Open Source License
- Sun Public License
- Interbase Public License

#### 4. Lizenzen mit Wahlmöglichkeiten

Diese Lizenzen sehen unterschiedliche rechtliche Folgen vor, je nachdem wie umfangreich eine Modifikation ist. Zudem werden dem Lizenznehmer verschiedene Wahlmöglichkeiten eingeräumt, wie Weiterentwicklungen weiterverbreitet werden können.

Beispiel:

- Artistic License (v. 2.0)

#### 5. Lizenzen mit Sonderrechten

Die Lizenzen mit Sonderrechten gewähren den Lizenznehmern zwar alle diejenigen Rechte, die Freie Software ausmachen, sehen aber zugleich besondere Privilegien für den Lizenzgeber bei Weiterentwicklungen durch den Lizenznehmer vor. Diese Lizenzen werden zumeist bei Programmen verwendet, die ursprünglich proprietär vertrieben wurden.

Beispiele:

- Netscape Public License (NPL) (v. 1.1)
- Q Public License (QPL)
- Apple Public Source License (v. 1.2)
- OCLC Research Public License (v. 2.0)

### **Die GNU General Public License (GPL)**

Die GNU General Public License (GPL) ist sicherlich die wichtigste und weitverbreitetste Open Source Lizenz. Mit dieser Lizenz fing der Gedanke von freier Software an, eine rechtliche Absicherung zu erhalten. Geschaffen wurde sie 1984 von Richard Stallmann als Lizenz für die Software des GNU Projekts. Sie enthält nicht nur die Bestimmungen zur Benutzung der Software, sondern auch eine lange Präambel, die den Hintergrund freier Software erklärt. Ein wichtiger Punkt der GPL, der allen anderen Open Source Lizenzen fehlt, ist die Idee des "Copyleft". Hierbei wird durch Ausnutzung des Copyrights dafür gesorgt, dass ein Programm, welches unter der GPL steht, auch frei bleibt. Dies wird dadurch erreicht, dass jedes Programm, welches eine Ableitung eines GPL-Programmes ist, automatisch wieder unter die GPL fällt. Zu beachten ist auch, dass es die GPL - im Gegensatz zu einigen anderen OSS-Lizenzen - nicht verbietet, mit Software unter dieser Lizenz Geld zu verdienen. Es ist also durchaus erlaubt, auch deutlich mehr als nur die Kopierkosten zu verlangen, solange die Bedingungen der Lizenz erfüllt werden. Einen Aspekt des Selbstschutzes bietet die GPL dadurch, dass ein Verstoß gegen eine der Auflagen der Lizenz automatisch zum Verlust dieser Lizenz für die entsprechende Software führt.<sup>26</sup>

Die GPL hat sozusagen politische Ziele, da sie neben den schon bekannten Haftungsausschlüssen auch noch die Forderung enthält, dass alle Weiterentwicklungen und alle Programme, die in irgendeiner Form unter der GPL lizenzierten Code enthalten, wiederum unter der GPL veröffentlicht werden müssen. Diese Eigenschaft hat ihr den Spitznamen GNU „GNU is not Unix“ eingetragen. Dahinter steht nicht nur der Nutzen für die freie Software, sondern auch eine politische Motivation im Umgang mit dem Copyright. Aus Sicht der GPL verhindert das Copyright den natürlichen Umgang mit anderen Computernutzern, da es einigen Privilegien einräumt (denjenigen, die eine Software lizenziert haben). Ferner verbietet die GPL auch das Hinzufügen

---

<sup>26</sup> Vgl. Marx, Ben, Linux Manager Guide, 2001, S. 49

weiterer Einschränkungen, die sich auf den Quellcode beziehen. Diese Einschränkung macht GNU-Software, insbesondere Bibliotheken, im wesentlichen für eine Verwendung im kommerziellen Umfeld unbrauchbar.

### **Die GNU Library General Public License (LGPL)**

Die LGPL entspricht im wesentlichen der GPL, mit der Ausnahme, dass Programme, die lediglich mit einer der LGPL unterliegenden Bibliothek verbunden werden, nicht als abgeleitete Arbeit im Sinne der GPL betrachtet werden. So wird die Verwendung von durch LGPL geschützten Bibliotheken zur Entwicklung kommerzieller bzw. nicht der GPL unterliegender Software ermöglicht. Änderungen an der Bibliothek selbst müssen allerdings wieder der LGPL unterliegen. Software, die der LGPL unterliegt, bietet damit bessere Möglichkeiten für eine Verbindung zwischen freien und kommerziellen Entwicklungen.<sup>27</sup>

### **Die BSD License**

Die BSD Lizenz ist eine der ältesten Lizenzen und schränkt sowohl Programmierer als auch Anwender nur insofern ein, dass die ursprünglichen Autoren genannt werden müssen. Hauptpunkt der Lizenz ist der Ausschluss von Haftungsansprüchen gegenüber den Programmierern. Sourcecode, der der BSD Lizenz unterliegt, kann in eigenen Entwicklungen benutzt werden, ohne dass diese wiederum freie Software sein müssen. Die BSD Lizenz stammt von der Universität von Berkeley. BSD steht für Berkeley Software Distribution.<sup>28</sup>

Die BSD-Lizenz hat eine andere historische Tradition als die GPL bzw. LGPL. Der Name stammt von der Berkeley Software Distribution, einer seit den 1970er Jahren entwickelten UNIX-Distribution, die Anfang der 1990er als Open-Source-Software zur Verfügung gestellt wurde (daraus entwickelten sich verschiedene

---

<sup>27</sup> Vgl. n. n., Open Source – kurz & gut, 1999, S. 14

<sup>28</sup> Vgl. n. n., Open Source – kurz & gut, 1999, S. 11

freie Unices: OpenBSD, NetBSD, FreeBSD usw.). Die Lizenzbedingungen sind einfach:

- Das Programm darf in jeder Form, auch in Binärform, weitergegeben werden. Eine Pflicht zur Überlassung des Quellcodes besteht nicht.
- Bei der Weitergabe in Binärform muss die Lizenz den Dateien beigefügt werden.
- Bei Derivaten darf der Name der Autoren/des Herstellers nicht ohne Erlaubnis für Werbezwecke verwendet werden.

Zusätzlich enthielt die erste Version der BSD-Lizenz eine so genannte Werbeklausel, die es erforderlich machte, in Werbematerialien auf die Universität Berkeley hinzuweisen. Diese Klausel wurde aufgrund von Beschwerden 1999 entfernt.

## **Die NPL (Netscape Public License) und MPL (Mozilla Public License)**

### **Lizenzen**

Die Netscape Public Licence (NPL) ist die Lizenz, unter der am 1. April 1998 der Quelltext des Netscape Communicators veröffentlicht wurde. Eric Raymond und Bruce Perens von der Open Source Initiative halfen, die einzelnen Paragraphen der Lizenz zu formulieren, aber die zentralen Aspekte steuerte Netscape selber bei.

Teile des Codes, insbesondere die später entwickelten, unterliegen einer zweiten Lizenz, der Mozilla Public Licence (MPL). Sie unterscheidet sich nur insofern von der NPL, dass sie keine Klauseln enthält, die Netscape besondere Rechte gewähren, wie zum Beispiel die Nutzung von Erweiterungen in anderen, nicht-freien Netscape-Produkten, und sich nicht im Besonderen auf den Netscape Communicator bezieht.

Modifikationen des vorhandenen Codes müssen wiederum der NPL/MPL unterliegen, wohingegen Erweiterungen (neue Routinen in separaten Dateien) ein anderes Copyright (oder gar keines) tragen können.

Obwohl die NPL im Falle Netscapes auf eine spezielle Geschäftssituation zugeschnitten wurde, nehmen einige Unternehmen sie oder eine Ableitung ihrer

als Lizenz für ihre Produkte, die sie im Zuge der Open-Source-Euphorie veröffentlichen. Schließlich handelt es sich beim Netscape Communicator um ein ehemals proprietäres Produkt, dessen Umwandlung in freie Software einer besonderen Lizenz bedurfte.<sup>29</sup>

Bei MPL wird unterschieden zwischen Datei-Derivaten und Werkderivaten. Datei-Derivate sind Änderungen an einzelnen MPL-lizenzierten Dateien, ihre Zusammenführung oder Inklusion in anderen Dateien. Werkderivate sind Werke, die Funktionen aus den MPL-Lizenzierten Dateien aufrufen oder von ihnen aufgerufen werden. Der Source-Code von Datei-Derivaten muss auf Anfrage ausgehändigt werden; Datei-Derivate müssen ebenfalls unter der MPL lizenziert werden. Werkderivate können dagegen beliebig lizenziert werden.

Damit wird sichergestellt, dass die Schnittstellen der Applikation offen bleiben, einzelne Erweiterungen jedoch proprietär sein können. Diese Vorgehensweise ist sicherlich besonders bei einem Web-Browser nachvollziehbar, aber auch für andere modulare Systeme geeignet.

Zwar schreibt die MPL für Datei-Derivate die Aushändigung des Quellcodes vor, erlaubt aber für die Binärdateien beliebige Lizenzierung, sofern die Lizenzbedingungen nicht mit der MPL in Konflikt stehen. So kann z.B. die Verbreitung von bestimmten Binärdateien untersagt werden, die Herstellung und freie Verbreitung von Binärdateien durch Dritte jedoch nicht.

Die MPL enthält auch eine salvatorische Klausel, was bedeutet, dass die verbleibenden Bedingungen auch rechtskräftig sind, wenn es einzelne Bedingungen auf Grund im Lande des Rechtsvorgangs geltender Bestimmungen nicht sind. So könnten z.B. Kryptographie-Exportverbote eine Verbreitung bestimmten Quellcodes verhindern, dies würde jedoch die übrigen MPL-Bedingungen nicht tangieren, sie müssen soweit möglich erfüllt werden. Seit Version 1.1 erlaubt es die MPL, einzelne Dateien des Werkes unter mehreren Lizenzen zur Verfügung zu stellen, was bei Wahl einer GPL-

---

<sup>29</sup> Vgl. n. n., Open Source – kurz & gut, 1999, S. 15

kompatiblen Lizenz auch die Verbindung derartig designierter Dateien mit GPL-Code ermöglicht.<sup>30</sup>

### **Die Apache Software License**

Die Apache Software Foundation, die von namhaften IT-Unternehmen unterstützt wird, hat für ihre Open-Source-Serverprodukte (am bekanntesten davon der Apache Webserver) eine eigene Lizenz entwickelt. Diese Lizenz erlaubt Veränderungen am Quellcode und die ausschließliche Weitergabe in Binärform, sofern die Lizenzbedingungen beigefügt werden. Sie enthält eine BSD-ähnliche Werbeklausel, die in Version 1.1 auf Dokumentation beschränkt wurde. Sie verbietet außerdem den Gebrauch des Namens "Apache" für Derivate.

Die Apache Software License ist sehr stark auf Apache-Produkte spezialisiert und empfiehlt sich kaum für allgemeine Open-Source-Applikationen.<sup>31</sup>

### **Die Artistic License**

Die Artistic License wird für die Programmiersprache Perl verwendet. Perl selbst ist parallel unter der GPL lizenziert, das gleiche gilt für viele Perl-Programme und Bibliotheken. Aufgrund von juristischen Unklarheiten bei bestimmten Formulierungen wurde die Artistic License durch die Clarified Artistic License abgelöst.

Die Artistic License gestattet kostenlose Verbreitung und die Weitergabe von Veränderungen, unterscheidet dabei aber zwischen der sog. Standard-Version und Derivaten von dieser. Sie erlaubt es, proprietäre Derivate zu erstellen, diese müssen aber eindeutig gekennzeichnet sein (andere Dateinamen, Dokumentation der Unterschiede). Die freie Bereitstellung des Quellcodes eigener Veränderungen kann einmalig erfolgen, z.B. über das Posting im Usenet oder den Upload in ein öffentliches Datei-Archiv. Es dürfen Distributionsgebühren gefordert werden (also Gebühren für die bloße Bereitstellung der Software), aber

---

<sup>30</sup> Vgl. BerliOS, Openfacts, Open Source Lizenzen

<sup>31</sup> Vgl. BerliOS, Openfacts, Open Source Lizenzen

keine Lizenzgebühren (also Gebühren, die z.B. an die Zahl der Nutzer gebunden sind).

Die Vielzahl von Fallunterscheidungen macht die Artistic License zu einer recht komplizierten Lizenz.<sup>32</sup>

### Die QPL und andere Lizenzen

Die QPL und andere Lizenzen sind meistens Derivate der oben genannten Lizenzvereinbarungen. Sie folgen dem Geist der GPL mit der Ausnahme, dass sie den Lizenzgebern andere Bedingungen als allen anderen Nutzern gewähren. Die folgende Tabelle stellt die individuellen Eigenschaften der verschiedenen Lizenzen dar.

Lizenz	Kann mit kommerzieller Software verwendet werden	Eigene Veränderungen müssen wieder frei sein	Kann unter anderen Bedingungen veröffentlicht werden	Enthält besondere Rechte für den Lizenzinhaber
GPL	Nein	Ja	Nein	Nein
LGPL	Ja	Ja	Nein	Nein
BSD	Ja	Nein	Nein	Nein
NPL	Ja	Nein	Nein	Ja
Public Domain	Ja	Nein	Ja	Nein

Quelle: Vgl. n. n., Open Source – kurz & gut, 1999, S. 16

Abbildung 3: Darstellung der Eigenschaften der verschiedenen Lizenzen

## 5.5 Übersicht der Open Source Lizenzen

Es gibt inzwischen eine recht große Anzahl an Lizenzen, die für sich in Anspruch nehmen, Open Source zu sein. Nicht alle davon entsprechen der Open Source Definition, noch weniger sind kompatibel zur GNU General Public License. Es gibt mehrere Versuche, eine Übersicht über die Lizenzen zu schaffen, der

---

<sup>32</sup> Vgl. BerliOS, Openfacts, Open Source Lizenzen



folgende Überblick richtet sich weitgehend nach der FSF (Free Software Foundation).

Lizenz	Copy-left	GPL komp.	frei	OSS (nach OSD)	Besonderheiten
<b>GPL</b>	X	X	X	X	Siehe Abschnitt über GPL.
<b>LGPL</b>	-	X	X	X	Diese Lizenz ist im Gegensatz zur GPL nicht Copyleft, da sie ein Verlinken mit unfreien Modulen gestattet.
<b>X11-Lizenz</b>	-	X	X	X	Eine sehr einfache Lizenz, benutzt für XFree86.
<b>modifizierte BSD-Lizenz</b>	-	X	X	X	Eine sehr einfache Lizenz, die Advertising-Klausel der originalen BSD-Lizenz wurde entfernt.
<b>Perl-Lizenz</b>	-	X	X	X	Dies ist die Disjunktion von GPL und Artistic-Lizenz, nicht Copyleft, da Artistic nicht Copyleft ist, benutzt fast nur für Perl-Pakete.
<b>Netscape Javascript Lizenz</b>	-	X	X	X	Dies ist die Disjunktion von GPL und NPL, nicht Copyleft, da NPL nicht Copyleft ist.
<b>originale BSD-Lizenz</b>	-	-	X	X	Eine sehr einfache Lizenz, inkompatibel zur GPL durch die "obnoxious BSD advertising clause".
<b>Apache-Lizenz</b>	-	-	X	X	Eine sehr einfache Lizenz, inkompatibel zur GPL (sowohl Version 1.0 als auch Version 1.1).
<b>Python-Lizenz</b>	-	-	X	X	Lizenz für Python 1.6b1 und später (ab Version 2.0 gilt allerdings wiederum eine andere Lizenz). Inkompatibel zur GPL da die Gesetze von Virginia (USA) dafür gelten.
<b>LaTeX Project Public License</b>	-	-	X	X	Unvollständige Lizenz, inkompatibel zur GPL, jede veränderte Datei muss einen neuen Dateinamen bekommen.

					merkwürdige Definition von "Distribution".
<b>MPL</b>	-	-	X	X	Mozilla Public License, inkompatibel zur GPL.
<b>NPL</b>	-	-	X	X	Netscape Public License, Netscape behält sich besondere Rechte an dem Code vor (Möglichkeit, diesen in proprietären Programmen zu verwenden), inkompatibel zur GPL.
<b>Sun Public License</b>	-	-	X	X	Ähnlich zur MPL, nicht zu Verwechseln mit Sun Community Source License, inkompatibel zur GPL.
<b>Qt Public License</b>	-	-	X	X	unpraktisch, da modifizierte Sourcen nur als Patches verbreitet werden dürfen, wurde für die Qt-Library verwendet (steht jetzt unter GPL), inkompatibel zur GPL.
<b>PHP Lizenz</b>	-	-	X	X	PHP 3 dual-licensed unter PHP-Lizenz und GPL, PHP 4 nur noch unter PHP-Lizenz, Source-Split droht, da Version 3 und 4 parallel weiterentwickelt werden, inkompatibel zur GPL.
<b>Artistic License</b>	-	-	-	X	Ungenaue und vage Formulierungen.
<b>APSL</b>	-	-	-	X	Apple Public Source License, nicht frei, da Apple jederzeit die Erlaubnis zur Verbreitung der Software zurückziehen kann. Ab der Version 1.1 eine Open Source Lizenz nach der OSD. In der Version 1.2 erfolgte eine weitere Anpassung an die OSD, jedoch weiterhin nicht ausreichend für die FSF.
<b>Sun Community Source License</b>	-	-	-	X	Keine freie Lizenz, da man veränderte Versionen der Software nicht veröffentlichen darf.

<b>Plan 9 Lizenz</b>	-	-	-	-	Keine freie Lizenz, da man veränderte Versionen der Software nicht für sich behalten darf. Diese Lizenz ist auch von der OSI nicht als Open Source anerkannt.
<b>Open Public License</b>	-	-	-	X	Keine freie Lizenz, da man veränderte und veröffentlichte Versionen der Software immer an einen "initial developer" senden muss.

Quelle: Vgl. Weber, N., Suhr, G., Ausarbeitung: Rechtliche Aspekte von OSS, 2001, S. 11,12

Abbildung 4: Vergleich der verschiedenen Open Source Lizenzen

### Kompatibilitätsfragen

Bei der Wahl einer Lizenz sollte man sich zunächst überlegen, ob man Code aus anderen Produkten in der eigenen Software verwenden möchte. Das betrifft Bibliotheken oder auch komplette Programmbestandteile. Solch integrierter Quellcode darf nicht einzelne Lizenzbedingungen des Hauptwerks, welches man darum erweitert, zurücknehmen. Enthält z.B. die Lizenz des Hauptwerks eine Werbeklausel, so müssen auch alle Werkderivate diese Werbeklausel enthalten. Die Kompatibilitätsproblematik ist mitunter hochkomplex, da nicht klar definiert ist, was Hauptwerk und was Derivat ist und welche Lizenzbedingungen dominieren. Entscheidet man sich für die "Standardlizenz", die GPL, wird man naturgemäß die wenigsten Probleme haben: Man kann ohne Schwierigkeiten Quellcode aus der großen Mehrzahl von Open-Source-Applikationen verwenden, und viele andere Lizenzen wurden angepasst, um mit der GPL kompatibel zu sein. So können GPL-Programme Quellcode, der unter der neuen BSD-Lizenz steht, nach Belieben verwenden. Der umgekehrte Weg ist nicht möglich. Der Hauptunterschied zwischen der GPL und anderen Lizenzen ist das Verbot der Weitergabe in Binärform ohne Bereitstellung des Quellcodes. Ist diese Einschränkung (oder Freiheit, je nach Definition) gewünscht, stellt die GPL ohne Zweifel fast immer die beste (weil kompatibelste) Wahl dar.

Neben diesen Modellen gibt es noch die Option der Doppel-Lizenzierung. Dieses Modell eignet sich vor allem für Programm-Bibliotheken und Plugins. Es beruht darauf, dass die GPL im Gegensatz zur LGPL auch die Verlinkung der Software mit proprietären Applikationen nicht gestattet. Der Urheber hat jedoch grundsätzlich das Recht, seine Software unter mehreren Lizenzen zu verbreiten. So kann z.B. ein Programm sowohl unter der GPL vertrieben als auch kommerziell verkauft werden, wobei die kommerzielle Lizenz die Nutzung in proprietärer Software gestattet, die GPL aber nicht. Um es einfacher auszudrücken: Wer die Programmbibliothek verwenden will, ohne seine eigene Software unter die GPL zu stellen, muss zahlen.<sup>33</sup>

## 6 Vorteile/ Stärken von Open Source Software

Open Source-Entwicklung bietet im Gegensatz zu geschlossenem Code eine Reihe von nicht zu unterschätzenden Vorteilen, die im Folgenden näher untersucht werden soll.

### 6.1 Höhere Qualität

Durch das „Peer-Review“ Prinzip kann die Open Source Software eine höhere Qualität und Sicherheit erlangen als die proprietäre Software, unter der Voraussetzung, dass sich auch fähige und professionelle Entwickler an dem OS-Projekt beteiligen. Aber auch hier ist ein erfolgreiches Projektmanagement des Koordinators entscheidend. Beim „Peer-Review“ Prinzip wird der Quellcode von hunderten von Programmierern überprüft. Dadurch können Fehler schneller entdeckt und behoben werden. Proprietäre Software wird dagegen nur von einigen wenigen Experten durchgesehen, wenn die Überprüfung überhaupt stattfindet.<sup>34</sup>

### 6.2 Wiederverwendbarkeit von OSS

---

<sup>33</sup> Schönfeldt, Rene; von Raison, Andre; Linux im Unternehmen, 2. Business-Kongress auf dem LinuxTag, dpunkt.verlag, 1. Auflage Heidelberg 2001, S. 51

<sup>34</sup> Vgl. Raymond, Eric S., The Cathedral and the Bazaar, 1999, S. 138

Eine der großen Innovationen von OSS ist sein Wiederverwendungscharakter, der durch die Offenlegung seines Quellcodes erreicht wird. Denn „das Rad muss nicht jedes Mal neu erfunden werden“. Das heisst, dass Module von OSS-Programmen auch auf andere Programme übertragen werden können, und ein Programmierer muss nicht nach neue Lösungen suchen, sondern kann von bereits vorhandenen OS-Lösungen lernen und seine Programme darauf aufbauen. Bei der Softwareentwicklung kann mit Hilfe von Open Source Software eine Effizienzsteigerung durch Reduzierung redundanter Arbeiten erreicht werden.<sup>35</sup>

### 6.3 Höhere Reife der Software

Open Source Software hat in der Regel eine höhere Reife als proprietäre Software, weil bei klassischer Software der Verkaufserfolg oft von einem günstigen Zeitpunkt der Markteinführung abhängt. Daher kam es in der Vergangenheit schon vor, dass Software auf dem Markt vertrieben wurde, obwohl sie noch nicht ausgereift und nicht fehlerfrei war. Da bei OSS nicht die wirtschaftlichen Aspekte im Vordergrund stehen, sondern ideelle wie Spaß und Neugier, stehen die OS-Entwickler nicht unter dem Druck zu einem bestimmten Zeitpunkt mit der Programmierung fertig zu werden. Sie sind in der Lage die Software erst dann auf dem Markt zu bringen, wenn sie die Entwicklung vollständig abgeschlossen haben, das heißt, dass die Software keine Fehler mehr enthält und vollen Funktions- und Leistungsumfang bietet. Dies kann durchaus ebenso zu regelmäßigen Releases führen, die Zeitschritte sind jedoch eher vom Projektfortschritt der Entwicklung geprägt.

### 6.4 Performance

OSS ist nach Ansicht vieler Analysten (im Server-Bereich) schneller als proprietäre Produkte. Oliver Kaven hat einen Vergleichstest zwischen Linux und Windows 2000 auf drei Hardwareplattformen durchgeführt. Getestet wurde der

---

<sup>35</sup> Vgl. Raymond, Eric S., The Cathedral and the Bazaar, 1999, S. 140

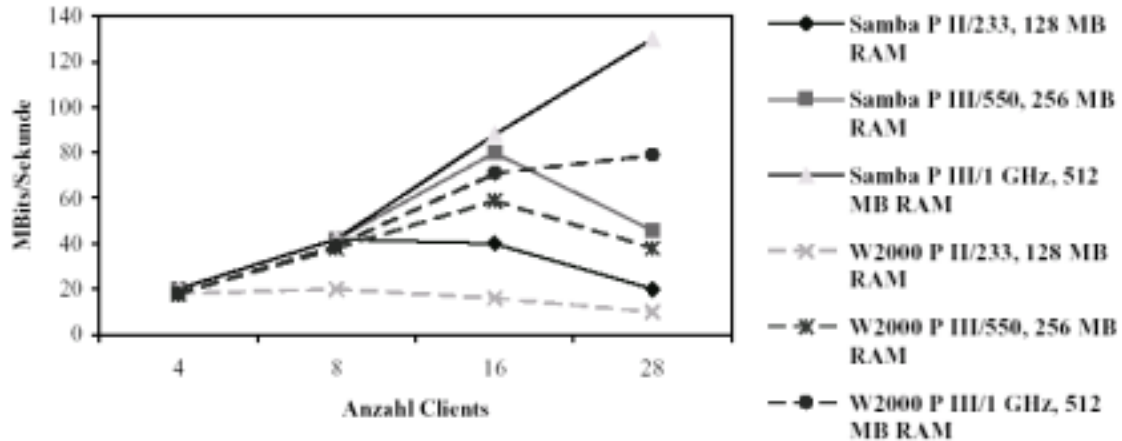
Datendurchsatz bei der Anwendung als Dateiserver, die als am zweithäufigsten genutzte Anwendung im kommerziellen Bereich gilt.<sup>36</sup>

Die Grafik zeigt, dass Samba (eine Open Source Software, um Windows-Clients an einen Dateiserver anzubinden) Windows-Server im Datendurchsatz schlägt. Ähnlich positive Ergebnisse wurden auch für den Parameter „Reaktionszeit“ erzielt.

Der oben durchgeführte Performance-Test bezieht sich auf einen Vergleich von Linux mit Samba auf der einen und Windows 2000 mit dem integrierten Dateiserver auf der anderen Seite auf verschiedenen Rechnersystemen. Um allgemeingültige Aussagen zur Performance von OSS gegenüber proprietärer Software zu machen, müsste man alle gängigen Betriebssysteme (Linux, BSD, Unix, Windows 2000) und alle gängigen Anwendungen (Dateidienste, Webserver, Email etc.) in einer Matrixform vergleichen. Diese Vorgehensweise kann im Rahmen dieser Studie nicht geleistet werden. Es sei an dieser Stelle lediglich darauf hingewiesen, dass in der Literatur beschriebene Performancevergleiche das Bild ergeben, dass OSS gegenüber Windows-Systemen in der Regel Leistungsvorteile bietet. Diese variieren je nach Anwendung und Hardwareausstattung von schlechterer Performance über vergleichbarer Performance bis hin zu deutlichen Performancevorteilen. Es gibt allerdings auch gegensätzliche Studien, die belegen, dass Windows Server in bestimmten Einsatzbereichen leistungstärker ist als Linux Server.

---

<sup>36</sup> Vgl. Kaven, Oliver, Performance Tests: File Server Throughput and Response Times, 2001



Quelle: Kaven, Oliver, Performance Tests: File Server Throughput and Response Times, 2001

Abbildung 5: Datendurchsatz von Linux/Samba im Vergleich zu Windows

## 6.5 Offenheit/Flexibilität

Hersteller von proprietärer Software bedienen oft eine bestimmte Zielgruppe, insbesondere im Applikationsbereich. Für diese Zielgruppe erstellen sie ein Produkt, das eine gewisse Individualität (für eine bestimmte Zielgruppe entwickelt), aber gleichzeitig auch einen gewissen Grad an Standardisierung bietet. Je individueller ein Produkt ist, desto teurer ist es (weil es nur an eine begrenzte Anzahl an Kunden verkauft werden kann). Je standardisierter ein Produkt ist, desto preiswerter ist es (da die Software öfter verkauft werden kann). Der Kunde muss folglich eine Abwägung zwischen den beiden Parametern Individualität und Preis treffen. Des weiteren muss Software in der Regel konfiguriert, also auf die konkreten Einsatzzwecke beim Kunden angepasst werden. Die Konfigurierbarkeit von proprietärer Software ist dabei auf den Rahmen beschränkt, den der Hersteller vorgibt. Entscheidend ist hierbei die Mächtigkeit der Schnittstellen und die Anpassbarkeit (z. B. die Verbindung zu anderen Softwaresystemen oder das Ein- und Ausgabeformat von Daten).

Manchmal ist der vom Hersteller vorgegebene Anpassungsrahmen nicht ausreichend. So kann z. B. das Format eines alten Buchhaltungssystems nicht lesbar sein, oder ein Netzwerkprotokoll ist für eine Software nicht geeignet. Ein besonders unangenehmer Fall liegt dann vor, wenn ein Kunde ein Update einer Software einspielen muss (z. B. aus Sicherheitsgründen), dort aber eine bestimmte Funktion in einer Softwareversion nicht mehr vorhanden ist, oder ein Softwareanbieter wird aufgekauft oder ist insolvent, und deswegen wird die Pflege der Software eingestellt. In all diesen Fällen ist der Kunde auf die vom Hersteller vorgesehenen Möglichkeiten beschränkt. Eine Anpassung oder Erweiterung darüber hinaus ist nicht möglich.<sup>37</sup>

Das Argument der „Interoperabilität“, ein technischer Begriff für einen Aspekt der Flexibilität und Offenheit von OSS, stellt einen wesentlichen Vorteil von OSS gegenüber proprietärer Software dar. Allerdings muss erwähnt werden, dass eine Veränderung einer installierten Software, sei sie OSS oder proprietär, zwar ein aktuelles Problem löst, aber auch ein zukünftiges Problem schafft: Jede Abweichung vom Standardsystem erzwingt bei der Umstellung auf neue Softwareversionen eine erneute Anpassung und damit erneuten Zeit- und Kostenaufwand. Dies gilt auch für individuelle Abweichungen von Standard-OSS-Installationen, wenn sich zwischen zwei Softwareversionen wesentliche Teile der Software verändern (z. B. Datenstrukturen, Format, Schnittstellen).

### 6.6 Stabilität

Ein weiteres, wichtiges Argument für den Einsatz von OSS ist die hohe Stabilität. Diese ist ein kritischer Faktor für den Betrieb von IT-Systemen, denn mangelnde Stabilität äußert sich in Abstürzen, Ausfallzeiten und erhöhtem Wartungsaufwand.

---

<sup>37</sup> Dibona, Chris; Ockman, Sam; Stone, Mark; Open Sources, Voices from the Open Source Revolution; O'Reilly Verlag, Sebastopol, CA, 1999



Als Argumente werden angeführt, dass OSS in der Regel besser getestet und kommentiert wird und der Entwicklungsprozess im Gegensatz zu kommerziellen Produkten ohne Zeitdruck erfolgt.

### 6.7 Preisvorteil

OSS ist in der Regel kostenfrei erhältlich. Für den Nutzer entfällt damit der Kaufpreis für Softwarelizenzen für Betriebssystem und Applikationen. Allerdings ist der Kaufpreis als Maßstab im kommerziellen Bereich nicht ausreichend. Eine Software muss nicht nur erworben, sondern auch installiert, konfiguriert, geschult und gewartet werden. Das Marktforschungsunternehmen IDC definiert den Begriff „Total Cost of Ownership“ (TCO) als umfassendes Kostenkonzept wie folgt: Das TCO-Modell kann unterteilt werden in drei Phasen: Beschaffung, Betrieb und Außerstandsetzung. Dabei enthält die Beschaffung neben den Kosten für Hard- und Software auch die Kosten für die Finanzierungsplanung (Leasing, Kauf), die Erhebung der Anforderungen und die Lieferkosten. Der Betrieb beinhaltet Inbetriebnahme, Schulung, technischen Support und Disposition. Die Außerstandsetzung beinhaltet den physischen Abtransport der Hardware und die Überspielung von Daten und Applikationen auf Nachfolgesysteme.

Weitere Kosten für den Betrieb von Rechnersystemen wie z. B. externe Kommunikationskosten, Medien, Strom und Platzbedarf werden im folgenden ausgeklammert, da sich verschiedene IT-Systeme in diesen Parametern nicht wesentlich unterscheiden.

Die TCO-Untersuchung von IDC/Red Hat<sup>38</sup> untersucht die Kosten von Linux auf einer PC-Plattform im Vergleich mit Unix-Systemen auf RISC-Prozessoren (Unix-Workstations). Dabei wurden zwei Einsatzzwecke getrennt untersucht. Erstens die Verwendung als Internet/Intranet/Extranet-Plattform und zweitens als

---

<sup>38</sup> Vgl. IDC Studie, „Linux: What’s the use“, 2000

„Collaborative“-Plattform (u.a. Nutzung gemeinsamer Verzeichnisse, Email, Kalender).

IDC kommt zu dem Ergebnis, dass Linux als Plattform deutliches Einsparpotenzial gegenüber Unix-Systemen ermöglicht. Die Einsparung bezieht sich dabei vor allem auf die Kostenbereiche Hard-/Software („harte Kosten“) und den Bereich Personalaufwand („weiche Kosten“). Die ausgewiesenen Kennzahlen beziehen sich dabei auf die Kosten, die pro Jahr und Anwender auf der Basis 1.000 unterstützter Anwender anfallen. Im Internet/Intranet/Extranet-Bereich unterbietet Linux die Unix-Systeme mit 377 US-Dollar gegenüber 684 US-Dollar. Noch erheblicher fällt der Unterschied im Collaborative Computing aus. Hier schlägt laut IDC ein Linux-User mit 255 Dollar zu Buche, während unter Unix pro Anwender 1.407 Dollar anfallen.

Begründet wird das Einsparpotential im Bereich der harten Kosten mit niedrigeren Preisen für die Anschaffung der Software und der niedrigeren Hardwarekosten. Die Einsparungen im Bereich Personal resultieren vor allem aus der niedrigeren Komplexität der unter Linux verwendeten Applikationen.

## 7 Nachteile/Schwächen von OSS

Bei allen Vorteilen hat Open-Source-Software aber auch Schwächen und birgt Probleme.

### 7.1 Gewährleistung/ Haftung

Problematisch ist die Frage der Haftung bei Open Source Software. Grundsätzlich ist sie nach den gesetzlichen Vorschriften und durch die General Public Licence geregelt. Nach BGB wird beim Herunterladen von Software aus dem Internet von einer Schenkung ausgegangen, wobei der Schenker außer durch grobe Fahrlässigkeit oder arglistiges Verschweigen eines Fehlers nicht haftbar gemacht werden kann.

In der GPL ist die Frage der Gewährleistung aber auch noch einmal explizit geregelt. In der GPL steht, dass Open Source Software von jeglicher Gewährleistung/Haftung ausgeschlossen wird. Durch den Haftungsausschluss hat der Benutzer keinerlei Anspruch auf Schadensersatz. Falls also ein Programm fehlerhaft ist, können für den Nutzer erhebliche Kosten entstehen, die er selbst zu tragen hat.<sup>39</sup>

### 7.2 Dauerhafte Nutzungsmöglichkeit und regelmäßiges Update

Anders als bei kommerziellen Programmen kann bei Open Source nicht zwingend von einer garantierten dauerhaften Nutzungsmöglichkeit ausgegangen werden. Dieses Risiko sollten User abwägen, wenn sie planen, Computersysteme auf Open Source Programme umzustellen. Auch das regelmäßige Update und die Anpassung an die neuesten Hardware-Systeme kann nicht garantiert werden. Dieses hängt völlig von den Entwicklern in der Open Source Community ab. Es werden aber immer mehr Bemühungen angestoßen, auch dieses Risiko für die User zu reduzieren. Z. B. wurde die KDE Free QT Foundation gegründet, um zu garantieren, dass die QT-Bibliothek auch in Zukunft für die Entwicklung freier Software unentgeltlich verfügbar bleibt.<sup>40</sup> Diese Aussage ist allerdings zu relativieren, da hier der rechtliche Anspruch auf dauerhafte Verfügbarkeit im Mittelpunkt steht. Die reale Verfügbarkeit ist durchaus unterschiedlich gegeben<sup>41</sup>.

### 7.3 Dokumentation

Rein intuitiv lassen sich die Möglichkeiten der meisten Programme kaum benutzen, daher ist eine allgemeinverständliche Anleitung nötig. Bei kommerzieller Software ist eine ausführliche Anleitung selbstverständlich. Freie

---

<sup>39</sup> von Raison, Andre; Schönfeldt, Rene; Linux im Unternehmen, 2. Business-Kongress auf dem LinuxTag, dpunkt.verlag GmbH, 1. Auflage Heidelberg 2001, S. 60

<sup>40</sup> Dibona, Chris; Ockman, Sam; Stone, Mark; Open Sources, Voices from the Open Source Revolution; O'Reilly Verlag, Sebastopol, CA, 1999

<sup>41</sup> Anpassung und Updates sind auch bei kommerzieller Software nicht endgültig gewährleistet. Zunehmend ist festzustellen, dass Unternehmen den Support eigener Software schon nach wenigen Jahren einstellen, durch neue Produkte ersetzen, oder durch Insolvenzen nicht mehr zur Verfügung stehen. Hingegen werden verschiedene OS-Projekte schon seit mehr als 15 Jahren kontinuierlich gepflegt (z.B. GNU Emacs)

Software wurde oft von Programmierern für Programmierer geschrieben, wobei auf die Dokumentation oftmals kein großer Wert gelegt wurde. Die frei vorhandene Dokumentation wird allerdings zunehmend umfangreicher und verständlicher. Da sie jedoch nicht ausreicht und der Bedarf hoch ist, existieren auf dem Büchermarkt viele Angebote.<sup>42</sup>

### 7.4 Bedienbarkeit

Um einen möglichst großen Umsatz zu erzielen, versuchen Firmen kommerzielle Produkte für möglichst alle potenziellen Anwender leicht bedienbar zu machen. Das wird von Open-Source-Anhänger manchmal als überflüssig abgetan, ist aber für normale Anwender ein entscheidendes Problem. OSS hat seine Ursprünge in einem technischen Milieu. Bei der Entwicklung von OSS werden technische Maßstäbe angelegt, weniger anwenderorientierte Kriterien wie z. B.

Benutzerfreundlichkeit und Bedienbarkeit.

Deshalb gibt es trotz vieler Verbesserungen in den letzten Jahren immer noch Probleme mit Inkompatibilitäten und mangelnder Bedienbarkeit. Für technisch versierte Personen mag dies selten ein Problem darstellen; für an Windows gewohnte Benutzer kann es eine große Hürde bedeuten.

Trotz der Tatsache, dass in den letzten Jahren einige Initiativen<sup>43</sup> wie z. B. KDE und Gnome für die Verbesserung der Qualität von OSS (insbesondere Linux) auf dem Desktop gestartet wurden, stagniert der Marktanteil von Linux deshalb bei wenigen Prozentpunkten.

### 7.5 Probleme beim Datenaustausch

Auch der Datenaustausch mit Open-Source-Software-Programmen läuft nicht immer reibungslos. Die Open-Source-Office-Programme verfügen zwar über Filter, mit denen Dokumente, die beispielsweise mit Microsoft-Office-Paketen erstellt wurden, eingelesen und anschließend wieder in Microsoft-Dateiformaten

---

<sup>42</sup> Raymond, Eric S.; The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999

<sup>43</sup> Raymond, Eric S.; The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999

abgespeichert werden können, allerdings funktioniert der Im- und Export von Textdokumenten mitunter nicht zuverlässig oder nur eingeschränkt.

Schwieriger wird der Dokumentenaustausch bei komplexeren Tabellenkalkulationen. Diese enthalten häufig Makros, das sind Sammlungen von Befehlen, die eine bestimmte Aufgabe automatisieren. Sie sind meistens einem Dokument zugeordnet. Oft scheitern die Filter an diesen Makros. Allerdings muss man berücksichtigen, dass Probleme beim Datenaustausch auch bei kommerziellen Produkten auftauchen, deren Ursache oft allgemein in fehlender Offenheit oder in der mangelnden bzw. fehlenden Dokumentation der Dateiformate liegt. Bei offenen oder standardisierten Formaten treten solcher Probleme nicht auf.

## 7.6 Mangel an Applikationen

Die Gründe für die mangelnde Verbreitung von OSS auf dem Desktop sind die üblicherweise mit Windows vorinstallierten PC-Systeme, die bevorzugte Entwicklung von Windows-Treibern durch Hardwarehersteller, das größere Angebot von Applikationen für Windows, die bessere und einfachere Bedienbarkeit von Windows und das Fehlen von defacto-Standardapplikationen wie z. B. Microsoft Office. Entscheidungen für Betriebssysteme werden in den seltensten Fällen als Selbstzweck getroffen. In der Regel ist die Entscheidung für ein Betriebssystem sekundär und leitet sich aus der Entscheidung für bestimmte Applikationen ab (z. B. Buchhaltungssoftware, Internetserver, Office-Produkt). Deshalb zeigen nur wenige Unternehmen oder Behörden Bereitschaft, Linux auf dem Desktop einzuführen. Auch in der Diskussion über den Einsatz von Linux in der deutschen Verwaltung und im Bundestag zeigt sich die Tendenz zu einer gemischten Umgebung von Linux auf dem Server und Windows auf den Arbeitsplätzen.<sup>44</sup>

---

<sup>44</sup> BMWi; Open-Source-Software, Ein Leitfaden für kleine und mittlere Unternehmen; BMWi Bundesministerium für Wirtschaft und Technologie; 1. Auflage, Stand März 2001

Das Angebot von Applikationen ist deshalb ein kritischer Faktor für den Erfolg eines Betriebssystems. OSS und speziell Linux haben durchaus ein reichhaltiges Angebot an Applikationen. Allerdings zeichnen sich diese durch bestimmte Charakteristika aus:

- *Applikationen haben Infrastrukturcharakter.* Typische Applikationen für den Einsatz von Linux im Netzwerk sind z. B. Firewalls, Webserver, Newsserver, Emailserver etc. Diese sind eher Basislösungen, als dass sie eine spezielle Lösung im Sinne einer Unternehmens- oder Verwaltungsanwendung darstellen.
- *Mangel an Unternehmens- und Verwaltungsapplikationen:* Es gibt zwar eine Reihe von „prominenten“ Linux-Unterstützern aus diesem Bereich (z. B. SAP oder Oracle), dennoch gibt es eine Reihe von Aufgaben, die unter Linux bisher nur am Rande adressiert wurden. Beispiele hierfür sind z. B. Management Information Systems, CRM-Systeme, Projektmanagement-, Collaborative- und Workflowsysteme.
- *Mangel an Clientapplikationen:* Die wichtigsten OSS-Applikationen sind Serverapplikationen. Echte „Killerapplikationen“ im Desktop-Bereich wie z. B. eine Portierung von Microsoft Office oder komfortable Produktivitäts-, Groupware-, Kalender- und Workflow-Applikationen aus dem Windows-Bereich fehlen. Alternative Produkte erzeugen Probleme mit dem Datenaustausch (Filter- und Konvertierungsproblematik).

### 7.7 Investitionsschutz

Der Open Source-Entwicklungsprozess zeichnet sich durch hohe Parallelität und Komplexität aus. Zum Beispiel wird bei Linux der Kernel und die einzelnen Softwarekomponenten mit hoher Frequenz aktualisiert. Auf der einen Seite stellt dies einen Vorteil dar (Qualität, Sicherheit), auf der anderen Seite kann dies für Unternehmen auch Unsicherheit und höheren Kostenaufwand bedeuten.

Das Fehlen einer Koordinationsstelle im Sinne des „bottom-up“-Prozesses der OSS Entwicklung bedeutet, dass für die Unternehmen kein zentraler Ansprechpartner „auf Augenhöhe“ existiert. Auch wenn zunehmend

Unternehmen wie IBM, HP, Compaq und viele kleine und mittlere Unternehmen OSS-Kompetenz aufbauen, kann für Entscheider das Gefühl eines „Abenteurers“ verbleiben. Hier liegen die Aufgabenfelder der Distributoren.

Dieser Vertrauensaspekt äußert sich auch in Haftungsfragen. Da es keinen Verkäufer und Lizenzgeber gibt, kann keine definierte Instanz im Schadensfall angesprochen werden. Dieses Argument verliert allerdings an Bedeutung, je mehr garantierte Serviceleistungen in Form von Wartungsverträgen und Service Level Agreements angeboten werden. Außerdem wird auch bei proprietärer Software regelmäßig jede Haftung durch die Lizenzbestimmungen ausgeschlossen.<sup>45</sup>

### 7.8 Administrierbarkeit

OSS-Betriebssysteme und insbesondere Linux sind aufgrund der Abstammung von Unix sehr mächtige Werkzeuge. Das Bedienungskonzept und die Mächtigkeit der zahllosen Unix-Administrations- und Verwaltungstools sind für ausgebildete und erfahrene Administratoren sehr hilfreich. Auf der anderen Seite sind die hinter Unix, Linux und den meisten anderen OSS-Betriebssystemen liegenden Konzepte jahrzehnte alt und haben sich in dieser Zeit wenig verändert. Die Bedienung ist für ungeübte und ungeschulte Benutzer schwierig. So hat auch eine repräsentative Befragung des Marktforschungsunternehmens Emnid ergeben, dass (private) Linux-Benutzer die „umständliche Installation“ und die „umständliche Bedienung“ mit 18% bzw. 16% als die meistgenannten Nachteile von Linux einschätzen.<sup>46</sup> Viele der Innovationen von Microsoft dagegen kommen aus dem Bereich einfache Bedienung und sinnvolle Reduktion der Komplexität. Das sind wichtige Gründe für die Entwicklung von Windows zum de-facto-Standard auf dem Desktop.

#### **Fazit:**

---

<sup>45</sup> Marx, Ben; Linux Manager Guide: Was Entscheider über das Betriebssystem und Open Source wissen müssen; SuSE-Press, Nürnberg, 2001

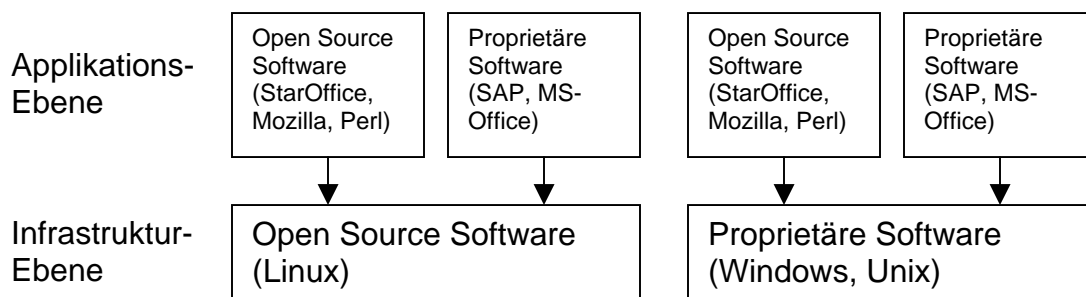
<sup>46</sup> Vgl. Emnid: „Bevölkerungsbefragung über Nutzung und Akzeptanz des Betriebssystems Linux“, 2001

OSS hat sich fest in Unternehmen und Verwaltungen etabliert. OSS ist dabei am stärksten im Bereich der Basistechnologien/Infrastrukturen auf Servern vertreten. Typische Einsatzszenarien sind z. B. der Einsatz im Intranet, als Virtual Private Network, als Internetserver, in heterogene Netzen und zur Datenarchivierung in kleinen und mittleren Unternehmen und in Abteilungen. OSS und speziell Linux konkurriert dabei hauptsächlich mit klassischen Unix-Systemen, nur bedingt mit Windows.

Die Schwächen von OSS sind der Mangel an Applikationen, das Desktop-Einsatzgebiet, mangelndes Vertrauen/ Investitionssicherheit und die Administrierbarkeit.

Diese Nachteile werden aller Voraussicht nach aber im Laufe der Zeit verschwinden, da bereits einige Software- und Hardwarehersteller bemerken, wie wichtig Open-Source-Systeme werden. Ein gutes Beispiel ist Corel, welches sein berühmtes Grafikprogramm Corel Draw bereits auf Linux portiert hat (allerdings ist es hier nicht gratis).

Mit Linux oder FreeBSD als Betriebssystem sowie ergänzender Open Source Software und kommerzieller Software auf OSS-Betriebssystemen kann heutzutage ein stabiles, preiswertes, ressourcenschonendes, sichereres und von ausreichend vielen Beratungsfirmen unterstütztes Rechnersystem auch für die professionelle Nutzung zusammengestellt werden. Dies gilt sowohl für den Client- als auch für den Serverbereich. Insbesondere durch die große Resonanz im Umfeld der IT-Industrie bietet Linux heutzutage einen guten Investitionsschutz.



Quelle: Eigene Darstellung

Abbildung: Zusammenhang zwischen der Applikationsebene und der Infrastrukturebene.



## 8 Open Source Software Produkte/ Projekte

Der Softwareentwicklungsprozess von Open Source Software unterscheidet sich wesentlich von dem der proprietären Software. Klassische kommerzielle Software wird hierarchisch im Wasserfallmodell erstellt, während eine Open Source Software von mehreren Entwicklern verstreut auf der ganzen Welt gleichzeitig entwickelt wird. Hierarchische Vorgehen würde erfordern bzw. erfordert hier entsprechende geregelte Strukturen in OS-Projekten. Die genaue Vorgehensweise beim Softwareentwicklungsprozess der Open Source Software wird im Arbeitspaket 2.1 und 2.2 von der Technischen Universität München (TUM) untersucht.

Im Folgenden werden ein paar Open Source Software Projekte vorgestellt, um zu zeigen, in welchen Bereichen der Wirtschaft und Industrie und in Form von welchen Produkten Open Source Software heute bereits eingesetzt wird.

### 8.1 GNU

Als AT&T 1983 anfang, in Bezug auf Lizenzen für UNIX restriktiver zu werden, begann Richard Stallman vom MIT mit einem umfassenden Projekt eine komplett freie Alternative namens GNU zu schaffen. Stallman gründete dafür auch die sogenannte Free Software Foundation, um die Idee zu unterstützen, dass Quellcodes für alle Programme immer frei verfügbar sein sollten. Er entwickelte dazu eine Lizenz namens GNU Public License (GPL), die festlegt, unter welchen Bedingungen Quellcodes verfügbar sein sollen, und dass jedes Programm, das GPL-Code beinhaltet, der GPL unterliegt.

Hunderte von Programmierer entwickelten neue Open Source-Versionen der meisten UNIX-Utilities. Allerdings schlug der Versuch fehl, einen UNIX-Kernel zu entwickeln.

Einige der GNU-Utilities waren so leistungsfähig, dass sie de facto Standards auf fast allen UNIX-Systemen geworden sind. Besonders der GNU C-Compiler gcc wurde der dominierende C-Compiler und GNU Emacs der führende Editor.

Die GPL erlaubt es sogar, die Software zu verkaufen, solange die Quellcodes verfügbar sind. Allerdings können Vertreiber in der Praxis nur geringe Gebühren für Vervielfältigungskosten berechnen. Zudem führte der Zusatz, dass jedes Programm, das GPL-Code enthält (inkl. Libraries), ebenfalls im Quellcode verfügbar sein muss, zu dem Ergebnis, dass der gcc Compiler nicht eingesetzt werden durfte, um kommerzielle Applikationen zu erstellen. Es gab daraufhin eine modifizierte GPL (die LGPL), die dieses Problem löste, aber der zu dogmatische Ansatz der FSF (Free Software Foundation) führte dazu, dass immer mehr Entwickler freier Software liberalere Lizenzen entwarfen, um eine kommerzielle Verwertung zu vereinfachen.

Die Firma Cygnus Support verkauft kommerzielle Support-Dienstleistungen für GNU-Software. GNU-Utilities sind integraler Bestandteil aller Linux-Distributionen.

Von vielen wird Richard Stallman als der Vater der Freien Software betrachtet. Er ist Präsident der Free Software Foundation, gründete 1984 das GNU-Projekt und war der Hauptentwickler von GNU Emacs, dem GNU C-Compiler, dem GNU Debugger und Teilen anderer Software-Pakete. Er ist weiterhin für GNU Emacs verantwortlich.

Andere, wie zum Beispiel Eric S. Raymond, haben sich auch am GNU-Projekt beteiligt.<sup>47</sup>

## 8.2 FreeBSD

FreeBSD stammt ab von der Berkeley Software Distribution UNIX. Es begann im Jahr 1993 als eine Portierung des BSD 4.3-Lite Release auf eine Intel-Plattform. Das FreeBSD-Projekt war dazu gezwungen, jedes Aufsetzen auf das 4.3 BSD Lite (Net/2 Release) zu umgehen und vollständig das letzte BSD-Release (4.4 Lite2) zu benutzen. Dies wurde mit dem FreeBSD 2.0 Release erreicht. Das FreeBSD-Projekt besteht aus einem harten Kern von 16 Entwicklern, die den Vorstand bilden. Es gibt mehr als 120 Entwickler, die an dem Projekt

---

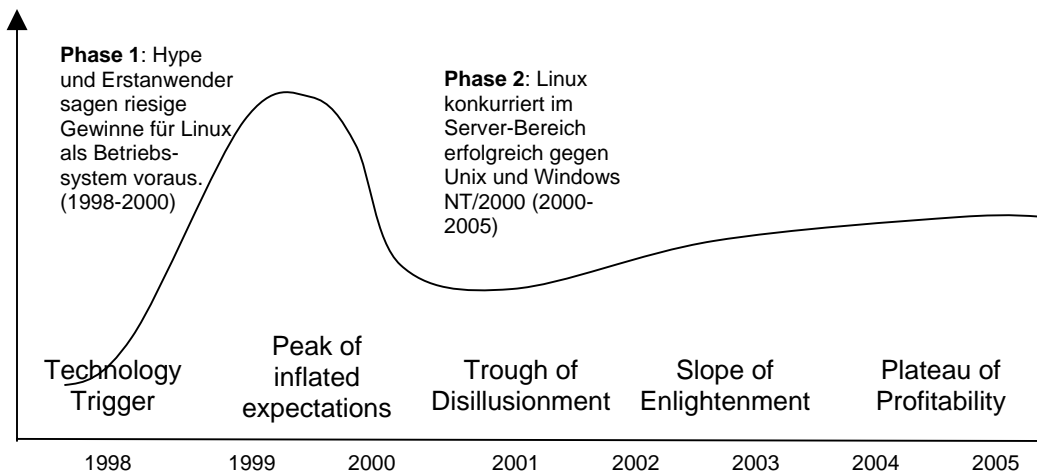
<sup>47</sup> Dibona, Chris; Ockman, Sam; Stone, Mark; Open Sources, Voices from the Open Source Revolution; O'Reilly Verlag, Sebastopol, CA, 1999

arbeiten, die meisten davon unentgeltlich. Einige werden von ihren Arbeitgebern, die ein eigenes Interesse an FreeBSD haben, bezuschusst.

Das Ziel von FreeBSD ist es, Quellcodes mit den geringst möglichen Restriktionen zu verteilen. Der Großteil des Systems wird unter der Original-BSD-Lizenz vertrieben, die sowohl den kommerziellen und als auch den freien Einsatz der Software erlaubt. Die verbleibenden Teile sind unter der GNU Public License, der Artistic License und anderen Lizenzen erhältlich.

Hubbard ist der Ansicht, dass es die größte Herausforderung für FreeBSD ist, langsam zu wachsen, ohne die Infrastruktur des Projekts zu überlasten, und mehr auf die Bedürfnisse der Kunden einzugehen, so wie es eine kommerzielle Firma machen würde. Er sagt: Wir wollten immer kommerzieller sein als die richtig kommerziellen Betriebssysteme, ohne allerdings Geld ins Spiel zu bringen. Das ist eine sehr interessante Herausforderung. Hubbard schätzt konservativ, dass FreeBSD mindestens 1,5 Millionen Benutzer hat. Japan ist mit großem Abstand der größte Markt.<sup>48</sup>

### 8.3 Linux



**Quelle: Gartner Research 2001: The future of Linux and Open Source**

**Abbildung 6: Vergangene und zukünftige Entwicklung von Linux**

---

<sup>48</sup> Raymond, Eric S.; The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999

Das Betriebssystem wurde von Linus Torvalds entwickelt, auch um anderen Programmierern zu beweisen, dass es durchaus sinnvoll ist, ein Betriebssystem von Grund auf neu zu schreiben und dass manchmal der eigene Weg der einzig richtige Weg ist.

Linux ist so bemerkenswert, weil seine Entwicklergemeinde international zusammengesetzt ist und trotzdem ein sehr hohes Niveau der Kooperation aufweist. Im Mittelpunkt steht Linus Torvalds, vormals aus Helsinki, Finnland. Während sich UNIX schnell in mehrere Dutzend Versionen aufteilte, was zu Verwirrung bei Managern und Programmierern führte, hat es Linux geschafft, als ein einheitliches System auf mehreren Architekturen zu bestehen. Es bietet eine UNIX-ähnliche Plattform, die es den Programmierern erlaubt, die beste Maschine auszusuchen, um produktiver zu arbeiten.

Die kommerzielle Reichweite von Linux ist schwer einzuschätzen, aber man kann zweifellos feststellen, dass viele, die früher auf UNIX setzten, nun Linux einsetzen. Viele sind der Ansicht, dass Linux die einzige Alternative zu Microsofts NT darstellt. In der Tat fangen immer mehr Hersteller von Multi-User Software an, ihre Software auf Linux zu portieren, wie z. B. Oracle, Informix, SAP R/3, Corel, Netscape etc.

Einige Hardware-Hersteller, wie z. B. SUN und IBM, liefern ihre Maschinen auf Wunsch mit Linux aus.

Die Firmen Red Hat und Caldera vertreiben verschiedene Versionen von Linux. In Deutschland ist die S.u.S.E.-Distribution sehr populär. Diese Distributionen enthalten teilweise auch kommerzielle Programme, die nicht frei sind. Die Debian Linux-Distribution enthält ausschließlich Programme, die unter einer Open Source-Lizenz veröffentlicht werden.<sup>49</sup>

### 8.4 DNS und BIND

Der BIND Daemon (Berkeley Internet Name Daemon) ist außerhalb der Technikerelite fast völlig unbekannt, aber jeder kennt die Dienstleistung, die er

---

<sup>49</sup> Sieckmann, Jens; Bravehack: Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source, 06.03.2001, <http://www.bravehack.de/html/node1.html>

zur Verfügung stellt: das Domain Name System (DNS). Der Verbund von Name-Servern ermöglicht die Umsetzung von Adressen wie 207.25.98.191 in eine Form, die jeder versteht, wie beispielsweise ibm.com, harvard.edu oder whitehouse.gov. Ursprünglich von Paul Mockapetris im Jahr 1984 entwickelt, wird die Entwicklung von DNS zur Zeit von Paul Vixie vom ISC (Internet Software Consortium) gesteuert. Das ISC wurde 1993 von Rick Adams gegründet, als er die Firma UUNET von einer Non-profit-Organisation in eine kommerzielle Firma umwandelte.

Das ISC ist ein gutes Beispiel dafür, was mit ausgereifter freier Software passiert: Sie trägt dann einen Non-profit-Mantel - den gleichen Effekt kann man bei HTML und HTTP erkennen.

DNS und der Rest der Berkeley TCP/IP Software Suite ist die Grundlage für die gesamte Internet-Industrie, einschließlich aller Internet Service-Provider, Produzenten von Web Sites und vieler Softwarefirmen. Die erste Geschäftsidee in Bezug auf das Internet entstand in Form von Firmen wie UUnet, den ersten kommerziellen Internet Service Providern. UUnet wurde 1997 an WorldCom in einer Multimilliarden-Dollar-Transaktion verkauft. BIND ist fester Bestandteil aller UNIX-Systeme und auch Microsoft liefert eine NT-Version von BIND aus.

Es gibt zur Zeit 29 Millionen registrierte Sites in DNS. Zwar hat nicht jede Site einen Domain Name-Server, aber es gibt sicherlich mehrere Millionen aktive Hosts im Internet.<sup>50</sup>

### 8.5 Sendmail

Sendmail wurde ursprünglich 1981 von Eric Allman entwickelt und ist mit ca. 75-80% Marktanteil der dominierende Mail Transport Agent (MTA) im Internet.

Unabhängig davon, welches Programm genutzt wird, um die Mail zu erstellen, wird fast jede Mail, die außerhalb der Firma zugestellt werden muss, mit Hilfe von sendmail weitergeleitet.

---

<sup>50</sup> Sieckmann, Jens; Bravehack: Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source, 06.03.2001, <http://www.bravehack.de/html/node1.html>

Im November 1997 gründeten Eric Allman und Greg Olson mit einigen privaten Geldgebern Sendmail Inc., um eine kommerzielle Version von sendmail herzustellen. Die Firma wird weiterhin die kostenlose Freeware MTA herstellen. Die kommerziellen Teile werden eine Web-basierte Administrationsoberfläche und andere Management-Tools sein. Sendmail 8.9 beinhaltet ein umfassendes Tool, um Spam-Mails zu kontrollieren.

Eric Allman selbst ist überrascht vom Erfolg von sendmail. Die Ursache ist nicht eine massive Marketingorganisation oder ein großes Budget. Dafür gibt es drei Gründe.

Erstens: Sendmail ist immer davon ausgegangen, daß es auch Mail-Nachrichten akzeptieren, bereinigen und zustellen sollte, die in Bezug auf die Einhaltung der Protokolle nicht ganz einwandfrei sind.

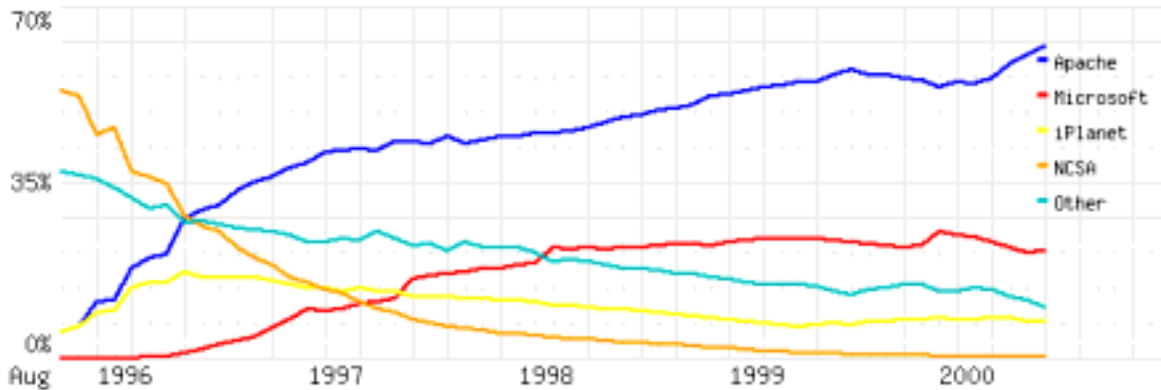
Zweitens wurde sendmail auf die Routing-Funktion beschränkt. Dies war eine Abweichung vom damals gängigen Modell, wo das Mail Routing, die Zustellung und auch oft der Netzwerkcode im gleichen Benutzeragenten implementiert waren. Es war auch entscheidend, daß es sendmail unentgeltlich gab, daß es zur richtigen Zeit erhältlich war, und daß es so funktionierte, wie es sollte.

Drittens war die sendmail-Konfigurationsdatei flexibel genug, um sich an eine rapide verändernde Welt anzupassen.<sup>51</sup>

---

<sup>51</sup> Raymond, Eric S.; The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999

## 8.6 Apache



Quelle: Wheeler, David A., Why Open Source Software / Free Software (OSS/FS)? Look at the numbers, 2003

Abbildung: Apache Marktanteile im Vergleich zu anderen Web Server

Apache ist eines der überraschendsten und erfolgreichsten freien Software-Projekte. Als die Produktpläne von Netscape und Microsoft die industriellen Schlagzeilen beherrschten, wurde Apache durch simple Mundpropaganda und durch die Verbreitung über das Internet zu dem am weitesten verbreiteten Webserver. Als bescheidenes Unternehmen ohne einen zentralen Visionär bricht Apache mit den Regeln, die für normale Projekte freier Software gelten. Das Apache-Projekt entstand aus der Einsicht, dass die Kontrolle über die Quellcodes für Web-Server-Software entscheidend ist, um mit den schnellen Änderungen im Web Schritt zu halten.

Während kommerzielle Firmen heutzutage immer wieder ihre Richtung ändern und ihre Prioritäten neu bestimmen, hat das Apache-Projekt bemerkenswerte Stabilität gezeigt und treu die Bedürfnisse der Benutzer erfüllt; deshalb hält es nach wie vor die Spitzenposition als der meist benutzte Web-Server.

Im Juni 1998 hat IBM ihre offizielle Unterstützung für die Apache-Gruppe angekündigt. IBM liefert Apache im Rahmen ihres Web Sphere-Produktes aus. Die Ankündigung von IBM war ein wichtiger Schritt, mit dem die freie Software-

Bewegung viel an Akzeptanz in der kommerziellen Entwicklungsumgebung gewonnen hat.

Brian Behlendorf ist der Pressesprecher der Apache-Gruppe, und obwohl er nie eine einzige Zeile Code geschrieben hat, gilt er als einer der führenden Köpfe des Projektes. Schon weniger als ein Jahr nach dem ersten Apache-Release im Jahr 1995 hatte Apache den NCSA-httpd-Server vom ersten Platz verdrängt.<sup>52</sup>

## 8.7 Samba

Ein erstes Release von Samba war 1993 verfügbar. Samba ermöglicht es einer UNIX-Maschine, als File- und Druck-Server für Windows NT- und Windows 95/98-Clients auf einem Netz tätig zu werden. Es ersetzt somit einen NT-Server und macht PC-NFS auf den Clients überflüssig. Samba wurde von Andrew Tridgell an der Australian National University entwickelt, und es stießen andere Entwickler dazu, als er den Code auf dem Internet verfügbar machte. Zur Zeit ist Jeremy Allison, der bei SGI arbeitet, das einzige bezahlte Mitglied im Team der Samba-Entwickler.

Ähnlich wie Linux wird Samba zur Zeit von der sogenannten Samba-Gruppe weiterentwickelt. Samba wird unter der GNU GPL vertrieben und es gibt immer zwei Releases: ein stabiles Release und eine Alpha-Version. Zur Zeit gibt es das Samba-Release 2.03 - in der Tat zeigte dieses Release, wie schnell die Entwicklergemeinde auf Probleme reagieren kann. Samba 2.0 wurde nämlich mit einem schlimmen Bug im Signal-Handling ausgeliefert. Innerhalb von drei Tagen konnte dieser Bug und dazu noch einige andere repariert werden. Auf diese Weise machte Samba innerhalb von drei Tagen den Sprung von 2.0 auf 2.02. Diese schnelle Reaktionszeit wird im Netz als massiver Vorsprung gegenüber Firmen wie Microsoft oder Netscape gewertet, die sicherlich kaum innerhalb so kurzer Zeit auf das Problem hätten reagieren können.

Das 2.02er Release von Samba unterstützt erstmalig Samba in der Rolle als Microsoft Primary Domain Controller: Diese undokumentierten Protokolle werden

---

<sup>52</sup> Raymond, Eric S.; The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999



von Microsoft bewusst zurückgehalten. Somit gibt es keinen Grund mehr, für solche Dienste einen NT-Server zu beschaffen.<sup>53</sup>

## 8.8 Perl

Perl (Practical Extraction and Report Language) wurde 1986 von Larry Wall entwickelt und ist ein wichtiges Werkzeug für System- und Netzwerk-Administratoren sowie zur CGI-Programmierung. Perl wird oft als das Klebeband des Internets bezeichnet, weil es sozusagen als Klebstoff mehrere Prozesse zusammenfügt. Große Sites wie Yahoo, Netscape, CNET, Amazon und Excite benutzen Perl, um ihre Web Sites zu managen und interaktive Dienste zur Verfügung zu stellen.

Perl wird von einer Gruppe von ca. 100 Programmierern weiterentwickelt, die über die perl5porters Mailing-Liste miteinander kommunizieren.

Perl wird mit vielen UNIX-Systemen ausgeliefert, zudem ist es Teil des Microsoft NT Resource Kit. Perl ist die am häufigsten benutzte Sprache zur Entwicklung von Internet-Diensten und interaktiven Datenbankabfragen. O'Reilly publiziert eine Vielzahl von Perl-Büchern und veranstaltet jährlich eine große Perl-Konferenz. ActiveState Tool Corp. vertreibt die Standarddistribution sowie Profi-Tools für Perl für Win32-Systeme.

Es gibt etwa 500.000 Perl-Programmierer und viele Millionen Perl-Anwender. Aufgrund der Buchverkäufe geht O'Reilly davon aus, dass Perl mindestens ebenso oft genutzt wird wie Java.<sup>54</sup>

## 8.9 Mozilla

Mozilla ist ein Web Browser. Die Mutterfirma von Netscape (AOL) gab den größten Teil des Quellcode ihres Browsers am 23. Januar 1998 als Open-Source Code frei. Der Quellcode wurde am 31. März 1998 auf der Seite [www.mozilla.org](http://www.mozilla.org) zur freien Verfügung gestellt. Der Code umfasste 1,5 Millionen Zeilen und wurde

---

<sup>53</sup> Sieckmann, Jens; Bravehack: Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source, 06.03.2001, <http://www.bravehack.de/html/node1.html>

<sup>54</sup> Sieckmann, Jens; Bravehack: Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source, 06.03.2001, <http://www.bravehack.de/html/node1.html>

unter 2 unterschiedlichen Lizenzen freigegeben: Die Netscape Public License (NPL) für Entwickler, die Änderungen vornehmen wollen und die Mozilla Public License (MPL) für die, die nur den neuen Code hinzufügen wollen. Im August 2000 wurde Mozilla auch unter die GPL (General Public License) gestellt. Der Navigator und die Communicator Standard Edition sind für alle User frei erhältlich. Durch die Veröffentlichung des Source Codes des Communicators erhoffen sich die Verantwortlichen von Netscape, dass durch die Vielzahl von Hackern der Source Code verbessert und die Fähigkeiten des Netscape Communicator erweitert werden.

Ein weiterer Grund für diesen Schritt war auch der Zugang zum Potential der Programmierer in aller Welt, um Feedback zu erhalten und den Code testen zu lassen.

Mit diesem Schritt wollte man seitens Netscape Standards in der Internet Technologie setzen, die Zahl der User im Internet maximieren und die Gemeinschaft der Unternehmen und Produkte vergrößern, die von der Netscape Software Plattform profitieren.

Das alles sollte zu einem besseren Produkt führen, um so die führende Stellung am Browser Markt gegenüber Microsoft verteidigen zu können.

Zur Koordination der Änderungen arbeitet innerhalb Netscapes eine Gruppe namens mozilla.org . "Mozilla" war der ursprüngliche Name für das Produkt, das später als Netscape Navigator bekannt wurde und heute als Netscape's Communicator bekannt ist. Dort werden die Änderungen gesammelt, Autoren bei der Abstimmung ihrer Arbeit unterstützt und regelmäßig werden neue Releases publiziert, die die besten Änderungen beinhalten. Die mozilla Gruppe gewährt Zugriff zum Source Code mittels eines Versions-Kontrollsystems und durch regelmäßige Veröffentlichung von Code Snapshots. Designentscheidungen werden diskutiert, entworfen und gemeinsam von allen Teilnehmern dieser Foren wird ein Konsens gefunden und danach auf der Website der mozilla Organisation veröffentlicht.

## 8.10 StarOffice

StarOffice ist eine Open Source Software, die die ähnlichen Funktionalitäten und Applikationen bietet wie das MS-Office Paket von Microsoft. StarOffice enthält alle Desktop-Anwendungen, die für den Geschäftsalltag benötigt werden, sie beinhalten Programme für die Textverarbeitung, Tabellenkalkulation, Präsentationen und Folien, Datenbanken, Internet-Browser, Terminverwaltung und Mail-Dienst. Die StarOffice-Anwendungen sind bezüglich ihrer Funktionalitäten und Leistungsfähigkeit denen der Windows-Anwendungen gleichgestellt, in manchen Fällen sogar vorteilhafter. Sie sind auch kompatibel mit anderen Software und ein Datenaustausch mit Windows-Anwendungen ist ebenfalls möglich. Sun Microsystems engagiert sich sehr stark mit StarOffice und hat auch schon eine Version auf dem Markt gebracht. Sie will mit dieser Absatzstrategie vor allem den Vorreiter im Desktop-Bereich Microsoft schwächen, und dadurch ein Eindringen des Open Source Modells im Desktop-Bereich beschleunigen.

## 8.11 PHP

PHP Hypertext Preprocessor (PHP) ist eine serverseitig interpretierte, in HTML-Code eingebettete Sprache zur dynamischen Erzeugung von Web-Inhalten. Das bedeutet, dass spezifische PHP-Anweisungen (PHP-Tags) in HTML eingebunden werden, die, wenn die entsprechende HTML-Seite angefordert wird, vom Web-Server bzw. vom PHP-Modul im Web-Server gelesen und ausgeführt werden. Durch die Beschränkung auf Web-Funktionalitäten ist die Sprache PHP leicht zu erlernen und nach nur kurzer Einarbeitung effizient anwendbar.

## 8.12 Python

Python ist eine interpretierte, objektorientierte Sprache, die ein weiteres Beispiel der typischen Open Source-Entwicklung ist. Python fing als ein kleines Tool an und wurde recht schnell zu einem universellen Werkzeug. Guido von Rossum, der Chefentwickler von Python, sieht es als eine verbesserte Perl-Version.

Vor kurzem wurde Jpython 1.0, eine vollständige Integration von Python mit Java, herausgebracht. Mit Jpython können Python-Programme überall dort verwendet werden, wo ein Java-Interpreter läuft.

### 8.13 Tcl/Tk

John Ousterhout begann seine Arbeit an Tcl/Tk, als er Professor an der University of California in Berkeley war. Er wechselte dann zu Sun Microsystems, die ein Interesse an Sun Script, einer kommerziellen Scripting-Sprache, hatten. Zur Zeit hat Ousterhout seine eigene Firma namens Scriptics, um ähnlich wie Sendmail Inc. die Entwicklung der Sprache voranzutreiben, während er auch kommerzielle Aspekte abdecken will.

Wie Perl und Python ist Tcl eine Script-Sprache - die Tool Command Language. Von Anfang an wurde Tcl mit Tk verbunden, um die Programmierung graphischer Oberflächen zu erleichtern. Ursprünglich kommt Tk aus der X-Windows-Welt (X11), aber in der Zwischenzeit wird auch das Windows- und Macintosh-Interface unterstützt.

1997 gewann Tcl/Tk den Software Systems Award, der von der Association for Computing Machinery (ACM) vergeben wird.

Es gibt zur Zeit zwischen 500.000 und einer Million Tcl/Tk-Entwickler. Pro Woche finden ca. 12.000 Downloads von Tcl/Tk vom Scriptics FTP-Rechner statt, wobei die Plattform-Verteilung wie folgt aussieht: 65% Windows, 30% UNIX und 5% Macintosh.

Von den drei Skript-Sprachen scheint Tcl/Tk der designierte Nachfolger der Windows-Entwicklungsumgebungen zu werden. Die hervorragende Unterstützung von graphischen Benutzeroberflächen könnte auch Visual Basic-Anhänger ansprechen.

Zudem bietet eine solche Sprache viele nützliche Erweiterungen, die jedem Entwickler enorm viel Arbeit und Zeit sparen können.

### 8.14 Das K Desktop System

<http://www.kde.org> ist die Web Site des KDE (K Desktop Environment) für UNIX Workstations. Es verbindet einfache Benutzung und hervorragendes graphisches

Design mit der technologischen Überlegenheit des UNIX-Systems und wird inzwischen mit fast allen Linux-Distributionen ausgeliefert.

### 8.15 GNOME

Unter <http://www.gnome.org> findet man das GNU-Projekt, das das Ziel hat, einen Desktop zu entwickeln, der ausschließlich auf freier Software basiert. GNOME benutzt das GTK (GNU GUI Toolkit) für alle GNOME-kompatiblen Programme. Dieser Desktop wird von Red Hat unterstützt.

### 8.16 GIMP

The Gimp (GNU Image Manipulation Program) wird häufig als das freie Photoshop bezeichnet. Es ist Bestandteil der meisten Linux-Distributionen. Das leistungsfähige Werkzeug zur Bildbearbeitung kann für die Fotoretouche, Bildkomposition und Bildmanipulation eingesetzt werden. Es wurde unter der GNU General Public License (GPL) veröffentlicht.

### 8.17 PGP

PGP (Pretty Good Privacy) ist de facto der Standard in Sachen Verschlüsselungssoftware. Der Autor, Phil Zimmermann, wählte das Modell der freien Software, um eine weite Verbreitung seiner Software sicherzustellen. Er hielt dies für wichtig, um Regierungskontrolleure daran zu hindern, diese Software einzuschränken. Er gründete die Firma PGP Inc., um kommerzielle PGP-Versionen zu vermarkten. McAfee, Network General, PGP und Helix fusionierten dann zu Network Associates.

### 8.18 Groff, SP und Jade

James Clark hat eine beeindruckende Sammlung von Werkzeugen zur Textverarbeitung und Formatierung zusammengestellt, wobei er mit groff begonnen hat. Innerhalb der SGML-Gemeinde hat er kostenlose Parser-Programme wie sgmls und sp zusammen mit der DSSSL-Formatierungsmaschine Jade entwickelt. In letzter Zeit hat er sich mehr um zwei XML-Parser gekümmert, einer davon in Java geschrieben, der andere in C.

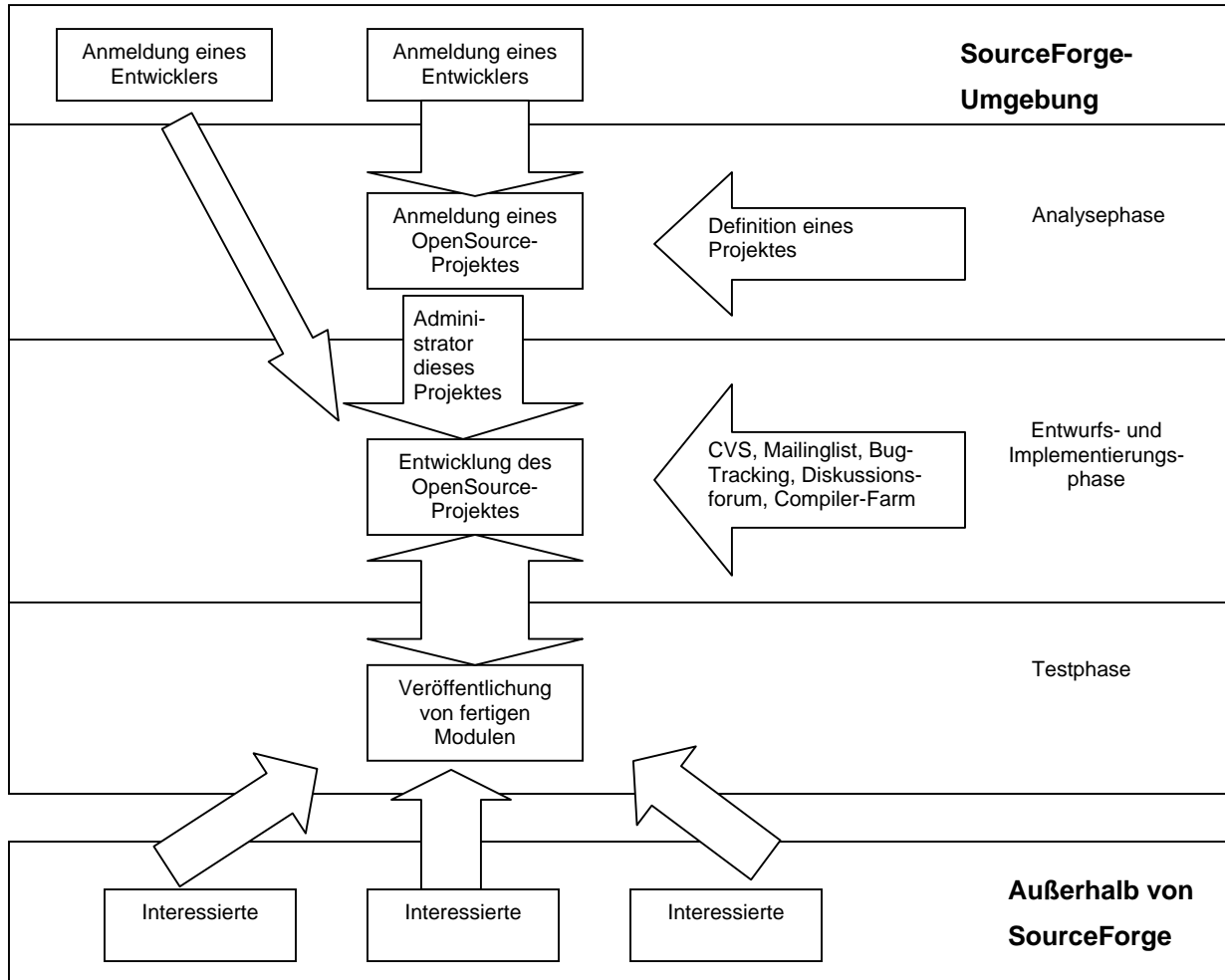
Obwohl er die Software selbst entwickelt hat, arbeitet er mit diversen Standardisierungsgruppen zusammen, um ihre Spezifikationen zu standardisieren.

## 9 Open-Source-Portale

Es gibt zur Zeit eine Menge Open-Source Portale im Internet. Durch das stets größer werdende Angebot von Open Source Software wird eine Art Marktplatz für den Austausch zwischen Anbietern und Nachfragenden dringend benötigt. Dies ist das Tätigkeitsfeld der sogenannten Mediatoren, deren Bezeichnung im Informationsmanagement „Intermediaries“ lautet. Sie dienen als Vermittler zwischen Angebot und Nachfrage und versuchen zwischen Entwicklern, Anwendern, Distributoren und kommerziellen Softwareanbietern zu vermitteln. Außerdem fungieren die Mediatoren auch als Entwicklungsplattform, die für die Entwicklung von Open Source Software zur Verfügung stehen. Mediatoren müssen ein klar definiertes Geschäftsmodell haben und finanzieren sich durch Sponsoren. Das bekannteste Portal im Open Source Bereich ist SourceForge, das im Folgenden beschrieben wird. Da das Projekt „NOW“ ein deutsches Projekt ist und in diesem Projekt auch ein Portal gebaut wird für die Entwicklung von verschiedenen Open Source Projekten, ist es interessant auch ein bestehendes deutsches Portal anzuschauen und daraus evtl. Rückschlüsse und Erkenntnisse für das NOW Portal zu ziehen. Von den deutschen Open Source Portalen ist das BerliOS Portal das bekannteste, deshalb wird es anschliessend vorgestellt.

### 9.1 Der OSS Mediator SourceForge

Derzeit bietet SourceForge das umfangreichste Spektrum für private und kommerzielle Open Source Projekte an. Der Schwerpunkt liegt im Open Source Softwareentwicklungsprozess, d.h. die Analyse-, Entwurfs-, Implementierungs- und Testphase sowie deren gesamte Koordination. SourceForge kann aus einem service-orientierten Blickwinkel als Application Service Provider (ASP) kategorisiert werden.



Quelle: Bensberg, Frank, 2001; Open Source Java-Enterprise: Plattformen, Mediatoren und Communities; Java Magazin

Abbildung 7: Entwicklungsprozess bei SourceForge

Die Dienste, die von SourceForge als „Team Development Environment (TDE)“ bezeichnet und angeboten werden, sehen folgendermaßen aus:

- Ein webbasiertes Administrationstool für das Management des Open Source Projektes steht jedem Entwickler zur Verfügung. Als Web-Server wird Apache verwendet. Die Nutzung von PHP3/PERL/CGI ist ebenfalls möglich. Jedes Projekt wird als Apache Virtual-Hosts mit einem separaten Verzeichnis von /cgi-bin verwirklicht. Insgesamt steht jedem Nutzer 100 MB Speicherplatz automatisch zur Verfügung.

- Falls eine Datenbank benötigt wird, bietet SourceForge jedem Entwickler ein MySQL-Datenbanksystem an.
- Eine Mailinglist wird durch „The Mailman“ und „majordomo“ zur Verfügung gestellt. Zudem können Diskussionsforen auf SourceForge verwaltet werden. Ein Bug-Tracking-System gehört ebenfalls dazu.
- Für eine erfolgreiche Zusammenarbeit mehrerer Entwickler kann auf die Verwendung von CVS nicht mehr verzichtet werden. SourceForge bietet daher ein CVS Repository für jedes Projekt an.
- Mehrere File-Server mit einer riesigen Kapazität stehen für jedes Open Source Projekt zur Verfügung. FTP-Verzeichnisse können ebenfalls genutzt werden. Für den Zugriff auf den SourceForge-Server wird anstelle einer telnet- eine ssh-Verbindung verwendet.
- Compile-Farm: Hier hat der Entwickler eines Open Source Projektes die Möglichkeit, seinen Quellcode auf dem SourceForge-Server kompilieren zu lassen. Für die Verwendung des Itanium-Prozessors muss ein separater Antrag eingereicht werden.

Um das Auffinden eines Projektes zu vereinfachen, können die Entwicklungsdaten des Projektes wie beispielsweise der Entwicklungsstatus in einer Projekt-Management-Software eingegeben werden. Hierfür existieren ebenfalls viele andere Kriterien, die für eine spätere Suche konzipiert sind. Zusätzlich können Hilfen für bestimmte Tätigkeiten angeboten bzw. aufgesucht werden.

### 9.2 Das OSS-Portal BerliOS

Der Mediator BerliOS hat sich zum Ziel gesetzt, die unterschiedlichen Interessengruppen im Umfeld der Open-Source-Software (OSS) zu unterstützen und dabei eine neutrale Vermittlerfunktion anzubieten. Die Zielgruppen von BerliOS sind einerseits die Entwickler und Anwender von Open-Source-Software und andererseits kommerzielle Hersteller von OSS-Betriebssystemen und Anwendungen sowie Support-Firmen. Im Folgenden werden die drei Zielgruppen näher beschrieben.



### **Anwender**

Neben den privaten Anwendern von Freier bzw. Open-Source-Software (FS/OSS) sind besonders kleine und mittlere Unternehmen (KMU) sowie die öffentlichen Verwaltungen und Einrichtungen die Zielgruppe von BerliOS. Bereits heute existieren für viele Anwendungsbereiche durchgängige FS/OSS-Lösungen, die in einem Produktionsumfeld einsetzbar sind. Die bei der Realisierung gewonnenen Erfahrungen und die sich daraus ergebenden Chancen und Risiken dienen als Beispiel für den Einsatz in anderen Unternehmen der gleichen Branche.

### **Entwickler**

Das Developer-Portal unterstützt Entwickler bei der koordinierten Realisierung, Präsentation und Verbreitung von Freier bzw. Open-Source-Software und bietet im Einzelnen folgende Dienste:

- Software-Repositoryen mit Versionsverwaltung
- Web-Server zur Projektpräsentation
- FTP-Server zur Softwaredistribution
- Datenbank als Basis für dynamische Web-Präsentationen
- Mailing-Listen mit Archivierungsfunktion zur Unterstützung von Gruppenkommunikation
- Fehlerverfolgung
- Umfragen und Bewertung
- Projektverzeichnis mit Suchfunktionen

Ein weiteres Ziel von BerliOS ist die finanzielle Unterstützung von Entwicklern mittels der Ausschreibung von Projekten durch Unternehmenskonsortien. Ein Entwickler hat damit die Möglichkeit sich an der Realisierung von bedarfsorientierter Software zu beteiligen oder sich selbst um eine Finanzierung von eigenen Softwareentwicklungsprojekten zu bemühen. In beiden Fällen entsteht Open-Source-Software, wobei die Auswahl einer spezifischen Lizenz den jeweiligen Projekten überlassen bleibt.

### **Hersteller und Dienstleister**

Das SourceBiz-Portal unterstützt Hersteller und Dienstleister bei der Präsentation ihrer angebotenen Produkte und Dienste und hilft Anwendern bei der Suche nach regionalen oder überregionalen Unternehmen mit entsprechendem Fachwissen und Erfahrungen.

## Die einzelnen Portale im BerliOS

### **SourceBiz**

Das SourceBiz-Portal hilft Anwendern bei der Suche nach regionalen oder überregionalen Unternehmen, die auf der Basis von FS/OSS Produkte und Dienstleistungen anbieten.

### **SourceWell**

Mit SourceWell unterstützt BerliOS Anwender bei der Suche nach existierenden oder neu erscheinenden Softwarekomponenten. Das Verzeichnis enthält Komponenten aus ca. 100 Kategorien.

Entwickler können ihre neuesten Softwarekomponenten bzw. Versionen im SourceWell-Portal ankündigen, wobei dieser Dienst nicht nur auf Komponenten beschränkt ist, die mit Hilfe der BerliOS-Plattform entwickelt wurden.

Entwickler veröffentlichen neueste Entwicklungen und Versionen auf dem SourceWell-Portal. SourceWell bietet somit ein aktuelles Verzeichnis von FS/OSS-Komponenten, wobei dieser Dienst nicht nur auf Komponenten beschränkt ist, die mit Hilfe der BerliOS-Plattform entwickelt wurden.

### **SourceLines**

Mit SourceLines ermöglicht BerliOS die Dokumentation von Lösungen und den Austausch von Erfahrungen zwischen Anwendern mit vergleichbaren Problemstellungen.

Dienstleister sind weiterhin aufgefordert Informationen über ihre beispielhaften Kundenlösungen im Rahmen des SourceLines-Portals bereitzustellen. "Best Practice"-Beispiele zeigen die Vielfältigkeit der bereits heute existierenden Lösungen und deren Übertragbarkeit für Unternehmen des gleichen Wirtschaftszweigs oder mit vergleichbaren Problemstellungen auf.

Bereits heute existieren für viele Anwendungsbereiche durchgängige FS/OSS-Lösungen, die in einem Produktionsumfeld einsetzbar sind. Die bei der

Realisierung gewonnenen Erfahrungen und die sich daraus ergebenden Chancen und Risiken dienen als Beispiel für den Einsatz in anderen Unternehmen der gleichen Branche. Mit SourceLines ermöglicht BerliOS die Präsentation von Lösungen und den Austausch von Erfahrungen zwischen Anwendern mit vergleichbaren Problemstellungen.

### **SourceAgency**

Begleitend dazu werden Bedarfsanalysen im Dialog mit den Anwendern einzelner Branchen durchgeführt, wodurch fehlende Komponenten identifiziert werden, die den Einsatz von FS/OSS-Lösungen heute noch verhindern. Die Realisierung der fehlenden Komponenten werden dann von Unternehmenskonsortien in der Projekt- und Ideen-Börse (SourceAgency) ausgeschrieben und die Entwicklung gemeinsam finanziert. Daneben können auch Entwicklergruppen Projektvorschläge erstellen, um damit potentielle Anwender für eine Finanzierung zu gewinnen.

Projektausschreibungen und -Ideen können in der Projektbörse SourceAgency präsentiert werden. Sie unterstützt die Bildung von Unternehmens- sowie Entwickler-Konsortien, die Diskussion und Festlegung von Projekthinhalten, bis hin zur Durchführung und Überwachung des Projekts.

SourceAgency richtet sich an Unternehmen, die existierende Open-Source-Entwicklungsprojekte finanziell unterstützen oder neue Entwicklungen initiieren wollen. SourceAgency unterstützt dabei die Abläufe, angefangen von der Vorstellung und Diskussion einer Projektidee, über die Suche nach weiteren finanziellen Beteiligungen und damit die Bildung von Unternehmenskonsortien, die Diskussion und Festlegung von Projekthinhalten mit potenziellen Projektnehmern, die Auswahl der Projektnehmer, bis hin zur Durchführung und Überwachung des Projekts.

### **OSS-Broschüre**

Die Broschüre "Open-Source-Software" richtet sich an mittelständische Unternehmen und Verwaltungen. Sie ist ein Wegweiser für potenzielle Anwender und soll vor allem über die Vorteile und Chancen, aber auch über Nachteile und Risiken von Open-Source-Software aufklären.

### **DocsWell**

Mit DocsWell bietet BerliOS eine zentrale Stelle für die Durchführung von Online-Recherchen nach genereller Dokumentation über OSS. Das Portal enthält Bücher, Handbücher, Einführungen und Hinweise, sowie FAQ, Fachzeitschriften und Artikel.

### **News-Dienst**

Der News-Dienst von BerliOS präsentiert Neuigkeiten und Veranstaltungshinweise rund um die deutschsprachige Open-Source-Gemeinde.

### **Wiki**

Wiki unterstützt den offenen Austausch von Informationen auf der Basis der Wiki-Prinzipien und stellt damit eine Alternative zu Mailing-Listen und Diskussionsforen dar.

### **Forum**

Forum ist ein Informationsdienst, der die Diskussion von Themen rund um Freie Software bzw. Open-Source-Software, sowie Lösungen und Produkte ermöglicht.

### **Entwickler-Portal**

Das Developer-Portal unterstützt Entwickler bei der koordinierten Realisierung, Präsentation und Verbreitung von Freier bzw. Open-Source-Software.

### **DevCounter**

DevCounter befragt Open-Source-Software-Entwickler nach ihren Kenntnissen und Erfahrungen sowie der Mitarbeit an Open-Source-Projekten. Dieser Dienst unterstützt insbesondere die Bildung von Entwickler-Konsortien und erlaubt die zielgerichtete Suche nach Entwicklern mit definiertem Know-How.

Ein ähnlicher Dienst steht auch für Software-Entwickler zur Verfügung. Auf dem DevCounter-Portal beschreiben Entwickler ihr Fachwissen und ihre Erfahrungen und erleichtern damit die Bildung von Entwickler-Konsortien und ermöglichen gleichzeitig Anwendern und Unternehmen die zielgerichtete Suche nach Entwicklern mit definiertem Know-How.

## 10 Ergebnisse aus vorhandenen Studien

### 10.1 Ergebnisse aus der FLOSS-Studie

Das FLOSS-Projekt<sup>55</sup> wurde von 06/2001 bis 06/2002 vom Berlecon Research und der Universität von Maastricht durchgeführt. Das Projekt wurde von der europäischen Kommission finanziert. Dieser Bericht stellt Daten zur Verfügung, die aus den Ergebnissen einer Erhebung gewonnen wurden. In der Erhebung geht es um die Benutzung der Open Source Software in europäischen Unternehmen und öffentlichen Einrichtungen, und um die Motivationen, OSS zu benutzen, sowie die Vorzüge, die die Unternehmen und öffentlichen Einrichtungen daraus ableiten. Insgesamt wurden 1452 Firmen und öffentliche Institutionen in Deutschland, Schweden und Großbritannien mit mindestens 100 Mitarbeitern telefonisch befragt, ob sie Open Source Software benutzen. 395 Firmen haben tatsächlich OSS benutzt oder planen dies in den nächsten Jahren zu tun. Diese wurden daraufhin noch näher interviewet. Im Folgenden werden die Ergebnisse kurz erläutert.

#### 10.1.1 Nutzungsbereitschaft von OSS in verschiedenen Ländern

1. Benutzung von Open Source Software nach Länder:

Länder	Deutschland	Schweden	Großbritannien
Prozent	43,7%	17,7%	31,5%

2. Gegenwärtige und geplante professionelle Nutzung von OSS in verschiedene Anwendungsbereichen.

	Großbritannien		Schweden		Deutschland	
	Klein	Groß	Klein	Groß	Klein	Groß
OSS als Server Betriebssystem	8,1%	3,7%	9,8%	11%	30,7%	30,6%

---

<sup>55</sup> FLOSS Final Report – Part 1, FLOSS (Free/Libre Open Source Software): Survey and Study, Use of Open Source Software in Firms and Public Institutions, Evidence from Germany, Sweden and UK; Berlecon Research, Berlin, Juli 2002

OSS für Datenbanken	13,3%	4,6%	7,5%	8,2%	14,1%	20,8%
OSS auf dem Desktop	7,6%	2%	3,4%	3,2%	13,7%	6,5%
OSS für Webseiten	7,9%	4,3%	7,5%	8,7%	15,8%	17,3%

Aus dieser Befragung kann man erkennen, dass in Deutschland OSS größtenteils im Bereich Server Betriebssysteme eingesetzt werden, was fast ein Drittel des Gesamtvolumens ausmacht. Als zweites werden OSS für Datenbanken, danach OSS für Webseiten und als letztes wird OSS auf dem Desktop angewandt. Eine ähnliche Konstellation hat auch Schweden, nur mit einem geringeren Prozentsatz. In Großbritannien wird am meisten OSS für Datenbanken eingesetzt, danach kommen Server Betriebssysteme, Webseiten und Desktopanwendungen.

### 10.1.2 Motivationen für den Einsatz von Open Source Software

3. Wir benutzen Open Source Software, weil wir mehr Unabhängigkeit wollen von der Preis- und Lizenzpolitik der großen Softwarefirmen.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
30%	26%	12%	10%	19%	4%

	Insgesamt	GB	Deutschland	Klein	Groß
Stimme vollkommen zu	30%	46%	25%	28%	35%
Stimme teilweise zu	26%	16%	30%	25%	27%

4. Durch den Einsatz von Open Source Software wollen wir den Open Source Community unterstützen.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
14%	21%	18%	12%	29%	7%

	Insgesamt	GB	Schweden	Deutschland
Stimme vollkommen zu	14%	26%	14%	9%
Stimme teilweise zu	21%	18%	22%	20%

5. Wir setzen Open Source Software ein, weil IT Spezialisten für diese Art von Software auf dem Arbeitsmarkt leichter verfügbar sind als Spezialisten für proprietäre Software.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
7%	14%	15%	10%	45%	8%

	Insgesamt	GB	Schweden	Deutschland
Stimme überhaupt nicht zu	45%	42%	32%	60%
Stimme teilweise nicht zu	10%	12%	9%	11%

6. Wir befürworten den Einsatz von Open Source Software – das ist ein Teil unserer Firmen-Politik.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
8%	11%	15%	13%	48%	6%

	Insgesamt	GB	Schweden	Deutschland
Stimme überhaupt nicht zu	48%	44%	47%	51%
Stimme teilweise nicht zu	13%	15%	9%	16%

7. Unsere Software Entwickler sind freigestellt an Open Source Projekte zu arbeiten während ihrer Arbeitszeit.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
19%	17%	8%	9%	37%	11%

	Insgesamt	Schweden	Deutschland	Niedrige IT Intensität	Hohe IT Intensität
Stimme überhaupt nicht zu	37%	25%	49%	47%	21%
Stimme teilweise nicht zu	9%	9%	10%	8%	9%

8. Wir arbeiten absichtlich mit Integrationshäuser/ Distributoren zusammen, um die Entwicklung der Open Source Software zu unterstützen.

Stimme vollkommen zu	Stimme teilweise zu	Weder noch	Stimme teilweise nicht zu	Stimme überhaupt nicht zu	Keine Antwort/ weiß nicht
6%	8%	11%	15%	52%	7%

Zusammenfassung:

Abschließend kann man sagen, dass (abgesehen von der ersten Aussage, die bei den Befragten größtenteils Zustimmung fand), die anderen 5 Aussagen überwiegend keine Zustimmung fanden.

Die Hauptmotivation für Firmen Open Source Software einzusetzen ist das Streben nach Unabhängigkeit von der Preis- und Lizenzpolitik der großen Softwarefirmen. Die anderen Gründe für den Einsatz von OSS, wie Unterstützung des OS Communitys, einfacher verfügbarer IT Spezialisten, OSS als Firmenpolitik, OSS Entwicklung während der Arbeitszeit und Zusammenarbeit mit Distributoren zur Unterstützung der OS Entwicklung, wurden als unwichtig eingestuft.

9. Aus Ihrer Sichtweise, wie groß ist die Wichtigkeit von Open Source Software für die IT-Infrastruktur deiner Firma?



Sehr hoch	Hoch	Mittel	Niedrig	Sehr niedrig	Weiß nicht
10%	23%	21%	26%	18%	1%

### 10.1.3 Vorteile von Open Source Software in verschiedenen Anwendungsbereichen

#### 10.1.3.1 OSS als Server Betriebssystem

220 von 395 befragten Einrichtungen (56%) setzen OSS als Server Betriebssystem ein.

	Linux	Free/ Open BSD	Andere
Im Einsatz	78%	9%	11%
Geplanter Einsatz	12%	2%	1%

Kriterien für die Entscheidung zugunsten von OSS als Server Betriebssystem

	Sehr wichtig	wichtig
Höhere Stabilität	44%	39%
Bessere Zugangsschutz	40%	35%
Niedrigere Lizenzkosten	32%	39%
Besseres Preis/ Leistungsverhältnis	27%	44%
Höhere Leistung	20%	53%
Bessere Funktionalität	19%	47%
Operations und Administrations Kosteneinsparungen	16%	44%
Installations und Integrations Kosteneinsparungen	16%	37%
Offener und/oder modifizierbarer Quellcode	12%	33%
Existierende Lösungen und Know-How	12%	35%
Hardware Kostenersparnisse	12%	29%
Höhere Anzahl von potentiellen Anwendungen	11%	29%
Trainings Kostenersparnisse	6%	21%
IT Dienstleistungsanbieter Empfehlungen	6%	18%
Integration in erworbene Produkte	3%	16%

#### 10.1.3.2 OSS als Datenbank

167 von 395 befragten Institutionen (42%) benutzen OSS als Datenbanken.

	MySQL	SAP-DB	PostgreSQL	Interbase	Andere
Im Einsatz	54%	12%	11%	4%	10%
Geplanter Einsatz	17%	2%	3%	2%	3%

Kriterien für die Entscheidung zugunsten von OSS als Datenbank

	Sehr wichtig	wichtig
Bessere Zugangsschutz	34%	37%
Höhere Stabilität	31%	46%
Niedrigere Lizenzkosten	28%	43%
Besseres Preis/ Leistungsverhältnis	21%	50%
Höhere Leistung	20%	49%
Operations und Administrations Kosteneinsparungen	17%	43%
Installations und Integrations Kosteneinsparungen	16%	46%
Bessere Funktionalität	15%	50%
Offener und/oder modifizierbarer Quellcode	12%	38%
Existierende Lösungen und Know-How	11%	34%
Hardware Kostenersparnisse	10%	31%
Höhere Anzahl von potentiellen Anwendungen	9%	37%
Trainings Kostenerspanisse	7%	34%
IT Dienstleistungsanbieter Empfehlungen	7%	26%
Integration in erworbene Produkte	7%	33%

**10.1.3.3 OSS auf Desktop Computer**

80 von 395 befragten Einrichtungen (20%) benutzen OSS auf ihren Desktop Computern.

	Linux	StarOffice/ OpenOffice	KDE	Gnome	Mozilla	Andere
Im Einsatz	49%	33%	31%	23%	23%	8%
Geplanter Einsatz	19%	15%	8%	4%	5%	3%

Kriterien für die Entscheidung zugunsten von OSS auf Desktops.

	Sehr wichtig	wichtig
Bessere Zugangsschutz	41%	30%
Höhere Stabilität	35%	45%
Niedrigere Lizenzkosten	34%	44%
Höhere Leistung	25%	40%
Besseres Preis/ Leistungsverhältnis	20%	49%
Operations und Administrations Kosteneinsparungen	17%	43%
Installations und Integrations Kosteneinsparungen	16%	46%
Bessere Funktionalität	15%	50%
Offener und/oder modifizierbarer Quellcode	12%	38%
Hardware Kostenersparnisse	11%	34%
Höhere Anzahl von potentiellen Anwendungen	10%	31%

Trainings Kostenerspanisse	9%	37%
Existierende Lösungen und Know-How	7%	34%
IT Dienstleistungsanbieter Empfehlungen	7%	26%
Integration in erworbene Produkte	7%	33%

#### 10.1.3.4 OSS für die Erstellung und Betreiben von Webseiten

155 von 395 befragten Einrichtungen (39%) benutzen verschiedene Open Source Komponenten für Webseiten.

	Apache	Perl	PHP	Squid	OS Content Management	Python	Andere
Im Einsatz	72%	48%	32%	26%	10%	8%	8%
Geplanter Einsatz	10%	6%	8%	3%	6%	1%	2%

Kriterien für die Entscheidung zugunsten von OSS für Webseiten.

	Sehr wichtig	wichtig
Höhere Stabilität	48%	35%
Bessere Zugangsschutz	44%	30%
Niedrigere Lizenzkosten	32%	41%
Höhere Leistung	32%	41%
Besseres Preis/ Leistungsverhältnis	30%	47%
Bessere Funktionalität	30%	41%
Operations und Administrations Kosteneinsparungen	20%	35%
Offener und/oder modifizierbarer Quellcode	19%	37%
Installations und Integrations Kosteneinsparungen	16%	37%
Existierende Lösungen und Know-How	12%	38%
Hardware Kostenersparnisse	11%	27%
Höhere Anzahl von potentiellen Anwendungen	10%	38%
Trainings Kostenerspanisse	10%	23%
IT Dienstleistungsanbieter Empfehlungen	6%	21%
Integration in erworbene Produkte	6%	19%

## 10.2 Studie über die Kosten von Open Source Software

Die Kosteneinsparungen, die mit OSS verwirklicht werden können, werden als eine der wichtigsten Vorteile betrachtet, die viele Anwender dazu bewegen zugunsten von OSS zu entscheiden. Es wurden bereits eine Menge Studien bezüglich Kostenvergleiche von OSS und proprietäre Software von vielen

renommierten Marktforschungs- und Technologieunternehmen durchgeführt. Die Ergebnisse der Studien waren nicht einheitlich, so dass keine eindeutige Aussage gemacht werden kann zugunsten einer Softwarevariante. Man sollte die Resultate immer kritisch beleuchten, und nicht bedenkenlos und zweifelsfrei akzeptieren, da die Studien oft im Auftrag von kommerziellen Unternehmen durchgeführt wurden oder von ihnen unterstützt werden, so dass in den Studien auch ihre Interessen vertreten sind. Im folgenden werden drei Studien von verschiedenen Marktforschungsunternehmen vorgestellt, die ein Vergleich der Kostenaufwände von OSS und proprietäre Software darstellen. Die Ergebnisse der ersten beiden Studien fielen zugunsten von OSS aus. Die letzte Studie bewertete proprietäre Software als die kostengünstigere Variante.

### **10.2.1 Studie der Robert Frances Group (RFG)**

Eine Studie der amerikanischen Robert Frances Group (RFG)<sup>56</sup>, die von IBM unterstützt wurde, sieht Linux als Preisbrecher im gewerblichen Einsatz auf Servern. Erstmals haben die Marktforscher aus Connecticut, die auf die IT-Beratung von Konzernen und Unternehmen mittlerer Größe spezialisiert sind, dabei nicht nur Lizenzgebühren verglichen. Über einen Zeitraum von drei Jahren haben sie auch die Kosten für Implementierung, Betrieb und Support, die so genannten Total Costs of Ownership (TCO), der Plattformen Linux (mit Apache-Server), Solaris auf einer SPARC-Architektur und Windows in Verknüpfung mit Microsofts Internet Information Server (IIS) unter die Lupe genommen. Bei den Gesamtkosten lag die Open-Source-Variante dabei deutlich günstiger als ihre Konkurrenten: Im ersten Jahr schlug demnach Linux durchschnittlich mit knapp 50.000 US-Dollar zu Buche, Windows mit rund 92.000 US-Dollar und die Highend-Lösung Solaris mit etwa 422.000 US-Dollar. Im dritten Jahr mussten die Firmen für Linux 74.475 US-Dollar, für Windows 190.662 US-Dollar und für Solaris 561.520 US-Dollar ausgeben.

---

<sup>56</sup> Robert Frances Group (RFG), Marktforschungsunternehmen für IT-Beratung, Studie: Total Cost of Ownership for Linux in the Enterprise, Juli 2002

Als Maßstab legte RFG das Konzept einer "Prozesseinheit" zu Grunde, die 100.000 Zugriffe auf eine Website pro Tag bearbeiten musste. Sie wurde auf die tatsächlich installierten Server umgerechnet. Am besten schnitt Linux naturgemäß bei den Lizenzkosten ab. Hier kamen die Analysten auf einen Preis von 400 US-Dollar pro Prozesseinheit, was an der Red-Hat-Distribution ausgerichtet ist. Diese wurde von den 14 Unternehmen, die Daten für die Studie zur Verfügung stellten, leicht bevorzugt. Bei den Anschaffungskosten für Windows stehen dem von Anfang an 5.320 US-Dollar gegenüber. Diese erhöhen sich auf Grund des neuen, heftig umstrittenen Microsoft-Lizenzprogramms "Software Assurance", das RFG bereits zu Grunde legt, um 1.330 US-Dollar pro Jahr. Solaris wird mit einmalig 27.500 US-Dollar berechnet.

Bei den Hardwarekosten schlagen die SPARC-Workstations bei Solaris mit 387.566 US-Dollar zu, während Linux und Windows mit 38.015 beziehungsweise 39.042 US-Dollar fast gleichauf liegen. Bezahlt macht sich Linux dagegen wieder voll bei den Support- und Administrationskosten. Überraschenderweise ergab die Umfrage von RFG, dass die beteiligten Unternehmen bei Linux weniger als 10 Dollar im Gesamtzeitraum der drei Jahre für externe Beratung bei Migration und Implementierung pro Prozesseinheit ausgaben. "Viele Administratoren bedienten sich der kostenlosen Support-Ressourcen, die Mailinglisten, Newsgruppen und Websites einschließen", erklären die Marktforscher die geringe Summe. Firmen mit Windows-Ausrüstung mussten 1.520 US-Dollar, Solaris-Kunden 19.309 US-Dollar für Beratung und Hilfe ausgeben. Die gesamten Administrations- und Supportkosten liegen laut RFG für Linux bei 36.060 US-Dollar, bei Windows bei 143.640 US-Dollar und bei Solaris bei 146.454 US-Dollar. Das Server-Personal sei bei Linux und Solaris zwar zunächst teurer zu entlohnen, dafür könnten die Admins aber mehr Server in gleicher Zeit warten als ihre Windows-Kollegen.

Gravierende Vorteile für Linux und Solaris sehen die Analysten auch im Bereich Sicherheit. Microsoft ist ihrer Ansicht nach momentan "ein politisches Ziel für Hacker." Das mache die Windows-Produkte daher zu einem "höherem Risikosystem für die Kunden". Die Umfragedaten hätten gezeigt, dass Windows-

Installationen doppelt so viele Administratorstunden für das Einspielen von Sicherheits-Updates und das Abdichten der Server erforderten als die Konkurrenzsysteme. Microsoft hat zwar inzwischen eine hoch aufgehängte Sicherheitsinitiative mit dem Programm "Trustworthy Computing" gestartet. Doch die Macher der Studie glauben, dass der Vorstoß frühestens in einigen Jahren greifen wird. Eine Warnung spricht RFG auch vor den kommenden .NET-Webservices aus, da deren Sicherheitsimplikationen noch größtenteils unerforscht seien.

RFG empfiehlt dagegen nun klar, "angesichts der tiefpreisigen und flexiblen Lizenzanforderungen, dem Verzicht auf proprietäre Verkaufsziele, dem hohen Sicherheitsgrad und der generellen Stabilität und Nutzbarkeit den Umstieg auf Linux für alle Sorten von Server-Anwendungen zu erwägen."

### **10.2.2 Studie im Auftrag der BMWi**

Die BMWi (Bundesministerium für Wirtschaft und Technologie) hat im März 2001 eine Studie<sup>57</sup> zur Untersuchung des Einsatzes von Open Source Software im kleinen und mittleren Unternehmen herausgebracht. In einem Teil der Studie wurden die Kostenaufwände von OSS und proprietäre Software miteinander verglichen. Dabei wurde die Kostenberechnung an zwei Fallbeispielen durchgeführt, im ersten Beispiel handelt es sich um einen firmeninternen Web-Server und im zweiten ein Routersystem mit Internetzugang.

Bei Open Source Produkten müssen die Anschaffungskosten berücksichtigt werden, diese lassen sich unterteilen in:

- Planungs- und Entwicklungskosten
- Systemkosten (Hardware, Software, Anpassung) und
- Kosten für die Systemeinführung (Installation, Integration)

---

<sup>57</sup> BMWi, Open-Source-Software, Ein Leitfaden für kleine und mittlere Unternehmen; BMWi Bundesministerium für Wirtschaft und Technologie; 1. Auflage, Stand März 2001

In den meisten Unternehmen und Behörden kommen Serverdienste im Internet- und Intranet-Bereich zum Einsatz. Im Folgenden werden an einigen Beispielen die Kosten aufgezeigt, die entstehen, wenn:

1. Open-Source-Software
2. eine Kombination aus Software mit offenem und geschlossenem Quellcode oder
3. proprietäre Software ohne offenen Quellcode eingesetzt wird.

### 10.2.2.1 Beispiel 1: Firmeninterner Web-Server

Im ersten Beispiel werden die Kosten für einen Intranet-Server mit einem datenbankgestützten Informationssystem aufgeschlüsselt.

#### 1. Open-Source-Software

Die erste Lösung setzt an Open-Source-Software an. Die genannten Komponenten sind in den gängigen Linux-Distributionen enthalten.

#### Rechenbeispiel 1

Komponente	Produkt/Aufwand	Einzelpreis
Betriebssystem	Linux-Distribution	50 €
Web-Server	z.B. Apache	0 €
System für dynamische Webseiten	PHP	0 €
Datenbanksystem	Postgres	0 €
Installation und Konfiguration	8 Stunden	800 €
<b>Gesamtpreis</b>		<b>850 €</b>

#### 2. Gemischte Lösung

Die zweite Lösung setzt auf eine Mischung aus proprietärer Software und Open-Source-Software. Der Einsatz kommt dann in Betracht, wenn ein proprietäres Betriebssystem verwendet werden soll.

#### Rechenbeispiel 2

Komponente	Produkt/Aufwand	Einzelpreis
Betriebssystem	Proprietäres	1250 €

	Betriebssystem	
Web-Server	z.B. Apache	0 €
System für dynamische Webseiten	PHP	0 €
Datenbanksystem	MySQL Server	0 €
Installation und Konfiguration	8 Stunden	800 €
<b>Gesamtpreis</b>		<b>2050 €</b>

### 3. Proprietäre Software

Die dritte Lösung basiert vollständig auf proprietärer Software.

#### Rechenbeispiel 3

Komponente	Produkt/Aufwand	Einzelpreis
Betriebssystem	Proprietäres Betriebssystem, inkl. 5 Client	1250 €
Web-Server	Im Betriebssystem enthalten	0 €
System für dynamische Webseiten	Proprietäres System	0 €
Datenbanksystem	Proprietäre Datenbank	2145 €
Installation und Konfiguration	8 Stunden	800 €
<b>Gesamtpreis</b>		<b>4195 €</b>

#### 10.2.2.2 Beispiel 2: Router-System mit Internetzugang

Als zweites Beispiel dient ein Router-System, das die Verbindung zum Internet über eine ISDN-Standleitung herstellt. Dabei wird von zwanzig Benutzern ausgegangen.

##### 1. Open-Source-Software

#### Rechenbeispiel 1

Komponente	Produkt/Aufwand	Einzelpreis
Routing-System	Linux-Distribution	50 €
ISDN	ISDN-Paket	0 €



Firewall	Ipchains	0 €
Proxys	Squid	0 €
Messaging	Postfix	0 €
Mail-Client	Diverse	0 €
Installation und Konfiguration	20 Stunden	2000 €
<b>Gesamtpreis</b>		<b>2050 €</b>

## 2. Gemischte Lösung

### Rechenbeispiel 2

Komponente	Produkt/Aufwand	Einzelpreis
Routing-System	Red Hat Linux	50 €
ISDN	ISDN-Paket	0 €
Firewall	Proprietäre Firewall, max. 25 IP inkl. 1 Jahr SW-Update	2700 €
Proxys	Squid	0 €
Messaging	Postfix	0 €
Mail-Client	Diverse	0 €
Installation und Konfiguration	20 Stunden	2000 €
<b>Gesamtpreis</b>		<b>4750 €</b>

## 3. Proprietäre Software

### Rechenbeispiel 3

Komponente	Produkt/Aufwand	Einzelpreis
Routing-System	Proprietäres Betriebs- system, inkl. 5 Clients	1250 €
15 Clients	Proprietäre Client- Lizenz	715 €
ISDN	ISDN-Paket	0 €

Firewall	Proprietäre Firewall	2700 €
Proxys	Proprietärer Proxy-Server	1215 €
Messaging	Proprietärer Messaging-Server	2700 €
Mail-Client	Netscape	0 €
Installation und Konfiguration	20 Stunden	2000 €
<b>Gesamtpreis</b>		<b>10580 €</b>

### 10.2.3 Studie von IDC

In einer Studie, die von der Unternehmensberatung IDC<sup>58</sup> im Auftrag der Microsoft Corporation erstellt wurde, wurden die beiden Betriebssysteme Linux und Windows 2000 bezüglich ihres betriebswirtschaftlichen Nutzens für ausgewählte Einsatzbereiche untersucht. In dieser Studie wurde der Einsatz von Linux und Windows 2000 in fünf gängigen Unternehmensanwendungen untersucht. Diese fünf Einsatzbereiche waren: Netzwerkinfrastruktur, Print Serving, File Serving, Sicherheitsanwendungen und Web Serving. Die Untersuchung ergab, dass über einen Zeitraum von fünf Jahren Windows 2000 einen Kostenvorteil von 11%-22% gegenüber Linux verwirklichen konnte. Lediglich im Bereich Web Serving konnte Linux Kosteneinsparungen von 6% gegenüber Windows erreichen. Die Ergebnisse dieser Studie widerlegt die gängige Meinung, dass Windows teurer wäre als Linux. Bei der Berechnung der Total Costs of Ownership (TCO) zeigt sich, dass die Anschaffungskosten, die bei OSS aufgrund des Wegfalls der Lizenzgebühren geringer ausfallen, nur einen kleinen Anteil am Gesamtkosten ausmachen, und dass die Personalkosten den größten Teil des TCOs beanspruchen.

Bei allen untersuchten Einsatzbereichen – mit Ausnahme des Bereiches Web Serving – waren die signifikantesten Kostenbereiche im Zusammenhang mit dem

---

<sup>58</sup> IDC, Windows 2000 vs. Linux für Unternehmensanwendungen, ein IDC White Paper im Auftrag der Microsoft Corporation

Personal zu finden. Der größte Anteil an den Gesamtkosten resultiert nicht aus den Anschaffungskosten von Hardware und Software, sondern aus laufenden arbeitsintensiven Supportleistungen und damit verbundenen Kosten. Die Aufschlüsselung der durchschnittlichen Kosten über einen Zeitraum von fünf Jahren ergab, dass Personalkosten einen Anteil von 62,2% an den Gesamtkosten ausmachten, während Ausfallzeiten des Betriebssystems mit 23,1% auf dem zweiten Platz landeten. Auf Schulungen, die Anschaffung von Software und Upgrades sowie die Anschaffung von Hardware und Upgrades entfielen je 5%, während der Anteil des Outsourcings bei 0,4% lag. Diese Studie widerspricht den anderen Studien, die bisher belegen, dass Linux für den Kunden preisgünstiger sei als Windows. Laut dieser Studie ist Linux teurer als Windows 2000, weil der Hauptanteil am Gesamtkosten die Personalkosten sind, und die liegen bei Linux höher, da dort mehr Supportleistungen anfallen. Allerdings sollte man die Ergebnisse dieser Studie nicht zweifellos akzeptieren, sondern kritisch überprüfen, weil die Studie im Auftrag von Microsoft erstellt wurde und deshalb auch deren Interessen berücksichtigt wurden.

### 10.3 Studie über die Integrationshäuser bezüglich ihre Aktivitäten im Bereich Open Souce Software

Über die Aktivitäten der Integrationshäuser und ihr Engagement im Open Source Bereich gibt es nur wenig Anhaltspunkte und Informationsquellen. Auch auf den Internetseiten der IT-Dienstleister sind wenig bis gar keine Informationen zu ihren Open Source-Aktivitäten zu finden. Daraus darf man nicht die Schlussfolgerung ziehen, dass die Systemintegratoren deshalb auf dem Feld nicht aktiv sind und OSS nicht Bestandteil ihres Produktportfolios ist. Denn viele Integrationshäuser sehen in dem Open Source Modell ein zukunftssträchtiges Geschäftsfeld mit hohem Entwicklungspotenzial und beabsichtigen, in Zukunft stärker in dem Bereich zu investieren. Bei manchen IT-Dienstleistern ist OSS bereits im Produktportfolio integriert und sie wollen dies noch weiter ausbauen. Viele Firmen halten sich aus firmenpolitischen Gründen mit öffentlichen Bekundungen

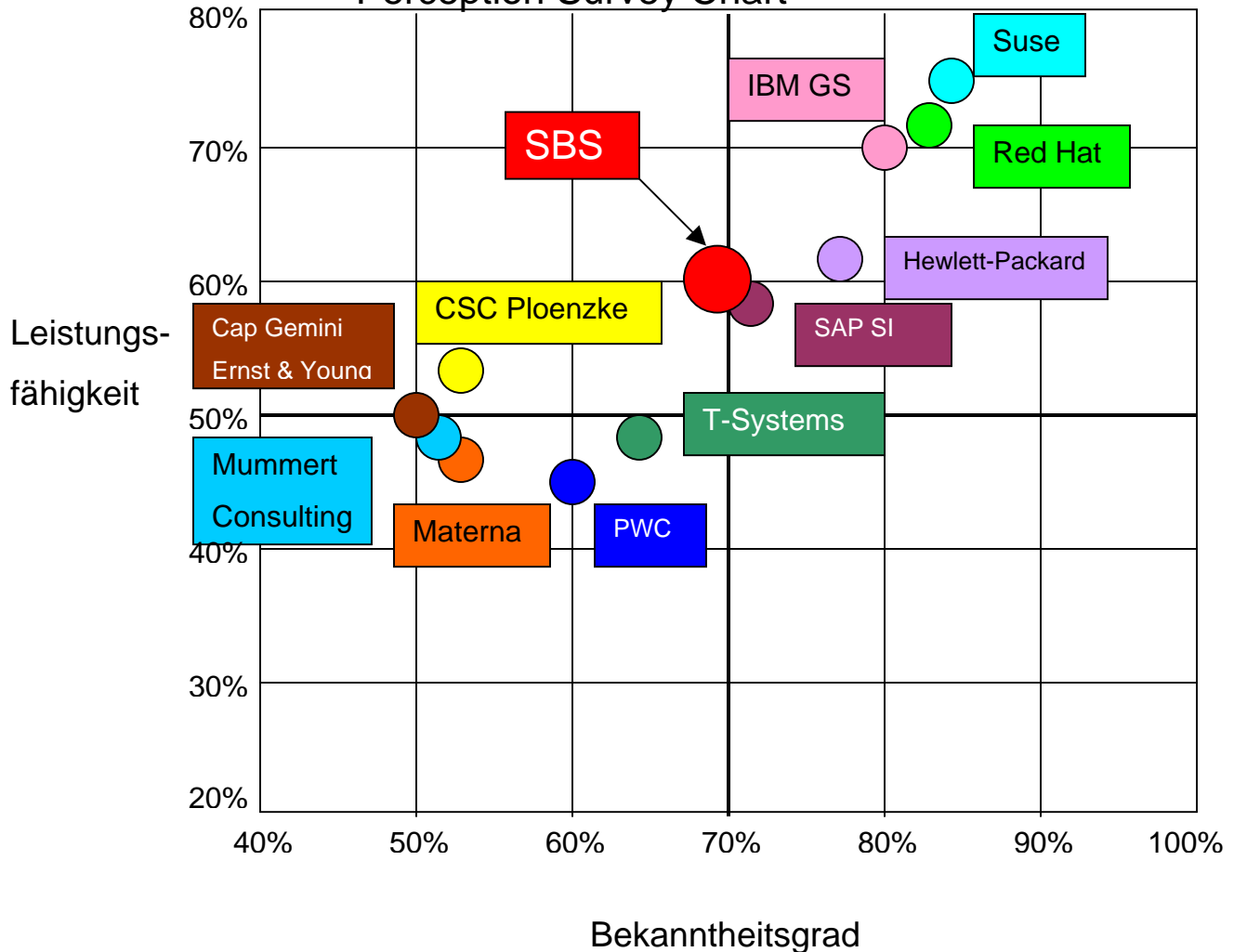
zu OSS noch zurück oder wollen mit ihrer Geschäftsstrategie zu OS nicht in die Öffentlichkeit treten. Manche Firmen haben auch Kooperationsverträge mit großen Softwarefirmen geschlossen, die gegen die Open Source Bewegung sind und eine Partizipation ihrer Partnerunternehmen an OSS missbilligen würden. Dennoch gibt es viele IT-Dienstleister, die sich stark im OS Bereich engagieren. Das Marktforschungsunternehmen META Group hat dazu eine Studie<sup>59</sup> durchgeführt, worin die OS-Aktivitäten der IT-Dienstleister miteinander verglichen wurden. Basis waren 188 Unternehmen. Die Integrationshäuser wurden nach den Kriterien Bekanntheitsgrad und Leistungsfähigkeit im Open Source Bereich untersucht und miteinander verglichen. Danach kann man sie in drei Gruppen einteilen. Die Firmen SuSE, Red Hat und IBM GS haben einen hohen Bekanntheitsgrad bezüglich OSS (80%-85%) und ebenfalls eine hohe Leistungsfähigkeit in der Open Source Technologie (70%-75%). SuSE und Red Hat sind Vorreiter in der Open Source Bewegung und sind auch die bisher größten und erfolgreichsten Firmen in dieser Branche. Ihr Hauptgeschäftsfeld besteht in der Distribution von Open Source Betriebssystemen (in dem Fall Linux, da es das weitverbreitetste und erfolgreichste OS-Betriebssystem darstellt). IBM GS ist die Beratungsfirma von IBM und ist ebenfalls sehr erfolgreich mit OSS, da IBM selber auch sehr stark in OSS involviert ist. Eine mittlere Position nehmen Hewlett-Packard, SBS und SAP SI ein. Sie erreichten beim Bekanntheitsgrad einen Wert von 70%-78% und bei der Leistungsfähigkeit 58%-62%. Einen niedrigeren Bekanntheitsgrad (50%-65%) und eine schwächere Leistungsfähigkeit (45%-55%) besaßen die Firmen T-Systems, PWC, CSC Ploenzke, Mummert Consulting, Cap Gemini Ernst & Young und Materna.

---

<sup>59</sup> O'Neill, Peter; Linux – Betriebssystemlandschaft im Wandel, Ergebnisse der META Group Studie Deutschland Pressekonferenz 2003

## Leistungsfähigkeit von Linux IT-Dienstleistern

### – Perception Survey Chart



Quelle: META Group Deutschland

Basis: 188 Unternehmen

#### 10.4 Studie über die Software- und Hardwareanbieter bezüglich ihre Aktivitäten im Bereich Open Source Software<sup>60</sup>

Es wurden die weltweit 25 größten Softwarefirmen bezüglich ihres Engagement in Open Source Aktivitäten untersucht. Dabei konnten sie in 3 Kategorien eingeteilt werden:

<sup>60</sup> Wichmann, Thorsten; FLOSS Final Report – Part 2, Free/Libre Open Source Software: Survey and Study, Firms' Open Source Activities: Motivations and Policy Implications; Berlecon Research, Berlin, Juli 2002

1. Firmen, die viele Open Source Aktivitäten haben (8 Firmen, 32%)
2. Firmen, die gelegentlich Open Source Aktivitäten haben (3 Firmen, 12%)
3. Firmen, die keine Open Source Aktivitäten haben (14 Firmen, 56%)

1. Firmen mit vielen OS Aktivitäten sind:

- IBM
- Hewlett-Packard
- Compaq
- SAP
- Computer Associates
- Hitachi
- Sun Microsystems
- Cadence Design Systems

2. Firmen mit gelegentlichen Open Source Aktivitäten:

- EDS
- BMC Software
- Adobe

3. Firmen ohne OS Aktivitäten:

- Microsoft
- Price WaterhouseCoopers
- Oracle
- Accenture
- Cap Gemini Ernst & Young
- Unisys
- NCR
- Compuware
- Siebel Systems
- PeopleSoft
- SunGard Data Systems
- Fiserv
- Computer Sciences Corporation

- EMC

Im Folgenden werden die Firmen mit vielen Open Source Aktivitäten näher untersucht:

**IBM**

IBM ist in vielen Open Source Projekten aktiv. Allein im Linux Technologie Zentrum hat IBM zu fast 70 Linux-bezogenen Projekten beigetragen. Das zeigt IBMs starke Commitment zu Linux: Im Dezember 2000 hat der damalige Geschäftsführer Louis Gerstner verkündet, dass IBM beabsichtigt, 1 Milliarde Dollar in Linux zu investieren. Die Hauptmotivation darin war der Glaube, dass demnächst eine standardisierte Version von Linux das Unix System ablösen wird. Obwohl es zur Zeit noch kein standardisiertes Linux gibt, da die verschiedenen Distributoren unterschiedliche Module in ihre Softwarepakete einbauen. Angeblich hat IBM mehr Vorteile durch Linux und der Open Source Bewegung gewonnen als ihre Wettbewerber. IBM behauptet schon im ersten Jahr ihre Investitionen durch den vermehrten Verkauf von Software und Systemen wieder hereinzubekommen. Aber IBM ist auch in dutzend andere Open Source Projekte involviert. Viele von diesen konzentrieren sich auf neue Technologie Trends wie Java, XML oder Web Services. IBM hat kürzlich mit Software im Wert von 40 Mio. Dollar die eigene Open Source Entwicklungsplattform „Eclipse“ unterstützt.

**Hewlett-Packard**

HPs Hauptaktivitäten in Open Source Projekten liegen im Linux Bereich. HP ist ein Sponsor der Linux Standard Base (LSB). Hewlett Packards Open Source Strategie konzentriert sich auf Linux. HP hält Linux unter bestimmten Umständen für die optimale Wahl des Betriebssystems.

**Compaq**

Compaq hostet eine Vielzahl von Open Source Projekten für Software, die auf Compaq Computern laufen. Von den 19 Projekten beziehen sich 14 auf Linux Software und sie bieten oft spezielle Treiber oder Werkzeuge für Compaq Hardware.

**SAP**

SAP hat seine Datenbank, SAP DB unter die GPL Lizenz gestellt. SAP DB ist ein offenes, SQL-basiertes und relationales Datenbanksystem, das hohe Verfügbarkeit und Leistung bietet und von kleinen bis großen Implementationen reicht. Dieser Schritt kann als Strategie gesehen werden, um die Datenbank Marktanteile unter den SAP Nutzern zu vergrößern. Die meisten Unternehmen, die SAP einsetzen, benutzen nicht SAP DB als Datenbank, sondern eher Produkte von Oracle, IBM und Microsoft. Mit SAP DB als Open Source haben die Kunden einen zusätzlichen Anreiz SAP DB als Datenbank zu nutzen.

URL von der SAP DB: <http://www.sapdb.org>

URL vom Linux: <http://www.sap.com/solutions/technology/linux>

### **Computer Associates**

Computer Associates ist einer der Sponsoren des Open Source Entwicklungslabors (OSDL).

### **Hitachi**

Hitachi ist hauptsächlich in der Linux Entwicklung involviert. Sechs von sieben Projekten befassen sich mit Linux.

### **Sun Microsystems**

Sun trägt aktiv an einer Vielzahl von freien und Open Source Projekten bei, einschließlich OpenOffice, GNOME, Mozilla, Apache, NetBeans, X Window. Diese Projekte sind ziemlich verschieden und so sind auch die Motivationen hinter Suns Engagement für Open Source Projekte. Zuerst die kommerzielle Software StarOffice zu kaufen und sie dann als Open Source Software OpenOffice zu vertreiben, ist hauptsächlich eine Marketing Strategie, um Suns Ruf in der Open Source Gemeinde zu steigern und gleichzeitig die Marktstellung von Microsoft zu schwächen.

### **Cadence Design Systems**

Cadence Design Systems produziert Werkzeuge für das Entwerfen von Chips. Die Firma sponsort zwei Open Source Projekte. Es bietet Testbuilder, eine Klassen Bibliothek, die durch eine Open Source Lizenz ermöglicht, C/C++ als eine effektive Testbench Sprache zu benutzen.



## 11 Wertschöpfungsprozesse bei IT-Unternehmen

Auf Basis der bis hier hin vorgestellten Eigenschaften und Erkenntnisse werden in einem zweiten Arbeitsschritt zum September 2003 potentielle Geschäftsmodelle analysiert. Als einen ersten Einblick werden an dieser Stelle die einzelnen Elemente des Wertschöpfungsprozesses kurz vorgestellt.

Die Prozesse in einem Unternehmen können anhand einer Wertschöpfungskette gegliedert werden. Wie sehen die Schritte der Wertschöpfungskette aus Sicht der Anbieter genau aus?



**Abbildung 8: Allgemeine Software Value Chain**

*Forschung* ist der erste Schritt in der Software Wertschöpfungskette. Hier forschen Softwareentwickler nach neue Programme, die besser die Kundenbedürfnisse befriedigen und die Benutzerfreundlichkeit steigern. Es wird auch nach Programme mit neue bessere Funktionalitäten geforscht.

*Entwicklung* ist der Abschnitt der Wertschöpfungskette, in der Software entwickelt wird. Basierend auf einem Entwicklungsplan verfassen Softwareautoren, meist im Team, den Quelltext der Software. Durch Compiler wird der Quelltext in Maschinencode übersetzt und dadurch ausführbar. Der Entwicklungsprozess kann mit Entwicklungswerkzeugen technisch unterstützt werden. Dadurch müssen Teams nicht räumlich verbunden sein. Man kann dabei unterscheiden zwischen dem „Cathedral“ – und dem „Bazaar“ - Prinzip. Nach dem „Cathedral“ – Prinzip werden die herkömmlichen proprietären Software entwickelt. Die Entwicklung erfolgt dabei nach dem Top-Down Prinzip entsprechend einer strengen hierarchischen Struktur. Das „Bazaar“ – Prinzip wird bei OSS angewandt, hier wird die Software dezentralisiert entwickelt. Die

Entwicklung erfolgt in einer Art Netzwerksystem, wo Programmierer auf der ganzen Welt gleichzeitig an der Software arbeiten können.

Bei der *Dokumentation* wird Information zur Software und zum Einsatz der Software angefertigt. Dokumentation entsteht dabei erstens im Rahmen des Quellcodes und seiner Handhabung, zweitens wird die Dokumentation über den Betrieb der Software in Form von Handbüchern (selten bei OSS) oder elektronischen Hilfsdokumenten erstellt.

*Packaging* bezeichnet den Vorgang, in dem die Einzelteile eines Produktes (Software, Dokumentation usw.) in ein Paket zusammengefasst werden. Bei kommerzieller Software ist dies üblicherweise ein Regalprodukt (commercial off the shelf), es kann dies auch ein Softwarepaket sein, das im Internet bereitgestellt wird.

Im Bereich *Marketing/Vertrieb* sind die Vermarktungs- und Absatzaktivitäten eines Unternehmens zusammengefasst. Im Marketing definiert das Unternehmen, welche Produkte wie an wen verkauft werden.

Marketingstrategien werden nach den „4P“ unterschieden: Produkt (welches Produkt, Gestaltung, Umfang), Price (Preis des Produktes), Place (wo wird das Produkt verkauft) und Promotion (wie wird das Produkt beworben). Von besonderer Bedeutung sind Standards. Wer eine Software als Standard am Markt durchsetzt, hat einen hohen Wettbewerbsvorteil. Im Desktop-Bereich ist Microsoft diese Standardsetzung gelungen. Bei Massenprodukten wird ein Partnermodell mit indirektem Vertrieb über Händler oder andere Hersteller (Bundling) verfolgt. Bei individuelleren Produkten mit hohem Dienstleistungsanteil werden Produkte über Dienstleistungspartner vertrieben.

*Beratung* beschreibt die Unterstützung des Kunden vor dem eigentlichen Softwareeinsatz. Dazu gehört das Anfertigen von Studien, Analysen und

Konzepten und eventuell die Anpassung der Unternehmensprozesse an die implementierte Software.

*Implementation/Integration* ist der Abschnitt, bei dem die Software vor Ort installiert wird. Wenn bereits IT-Systeme vorhanden sind, wird das neue System auf die Zusammenarbeit mit diesen Systemen konfiguriert und Schnittstellen angepasst.

*Training* beschreibt die Schulung des Kunden durch den Dienstleister. Schulung kann sich dabei erstens auf die technischen Aspekte der Software beziehen, dabei wird das IT Fachpersonal des Kunden geschult. Zweitens kann sich Schulung auf Nutzung und Anwendung der Software beziehen, dabei werden die Anwender der Software geschult.

Im Bereich *Support* werden die installierten IT-Systeme im laufenden Betrieb betreut. Dies kann im einfachsten Fall die Unterstützung der Mitarbeiter vor Ort bedeuten, aber auch bis hin zum vollständigen Betrieb der IT-Systeme durch den Dienstleister (outsourcing) reichen.

Ein Unternehmen muss nicht alle Teile der Wertschöpfungskette selber abdecken. Viele Softwareunternehmen beschränken sich auf Forschung, Entwicklung, Dokumentation und Packaging. Marketing und Vertrieb werden von Partnerunternehmen übernommen. Auch die Integration/Implementation kann im Rahmen von Partnerkonzepten ausgelagert werden.

## 12 Die Prozess-Beteiligten des OSS Marktes

Um die verschiedenen Beteiligten an Geschäftsprozessen und ihre Beziehungen untereinander hinsichtlich ihrer Motive, Anforderungen und derzeitigen Aktionsräume zu analysieren, wurde das folgende Rollenmodell entwickelt. Es besteht aus fünf Basisrollen. In der Realität sind durchaus Personen und Unternehmen zu finden, die mehrere der Rollenprofile abdecken. Dies ist für die

Analyse und Ableitung von Handlungsempfehlungen im weiteren Projektverlauf aber keineswegs hinderlich, sondern hilft die hieraus entstehende durchaus komplexe Motivations- und Zielstruktur zu gliedern.

### **Software-Entwickler**

Als Entwickler tragen unterschiedliche Rollen zur Entstehung und Weiterentwicklung bei. In der Open Source Software – Community sind „freiwilligen“ Entwickler zu finden, die aus eigener Motivation und Neugierde Software Programme entwickeln, ohne Profit-Absichten. Hiervon zu unterscheiden sind die akademischen Entwickler insb. aus dem Bereich der Hochschulen und die beruflichen Entwickler, die beide im Umfang ihrer beruflichen Aufgaben Beiträge leisten

### **Distributoren**

Distributoren schlagen eine Art Brücke zwischen der Open Source Software – Community und den Anwendern. Beispiele für Distributoren sind: SuSE, Red Hat, Debian, Mandrake. Distributoren sind kommerziell arbeitende Firmen, die Open Source Software auf Datenträgern verkaufen und dazu Handbücher und einen Kundenservice anbieten. Leistungen im Umfeld von Open Source Softwareprodukten sind ihre Einnahmequelle.

### **Integrationshäuser**

Integrationshäuser (Dienstleistungsanbieter/Unternehmensberatungen): Sie bieten Dienstleistungen an. Sie entwickeln selber keine Software, sondern nehmen die Software von anderen Softwarehäusern und führen bei den Kunden die Beratung und Implementation durch. (SBS, Accenture)

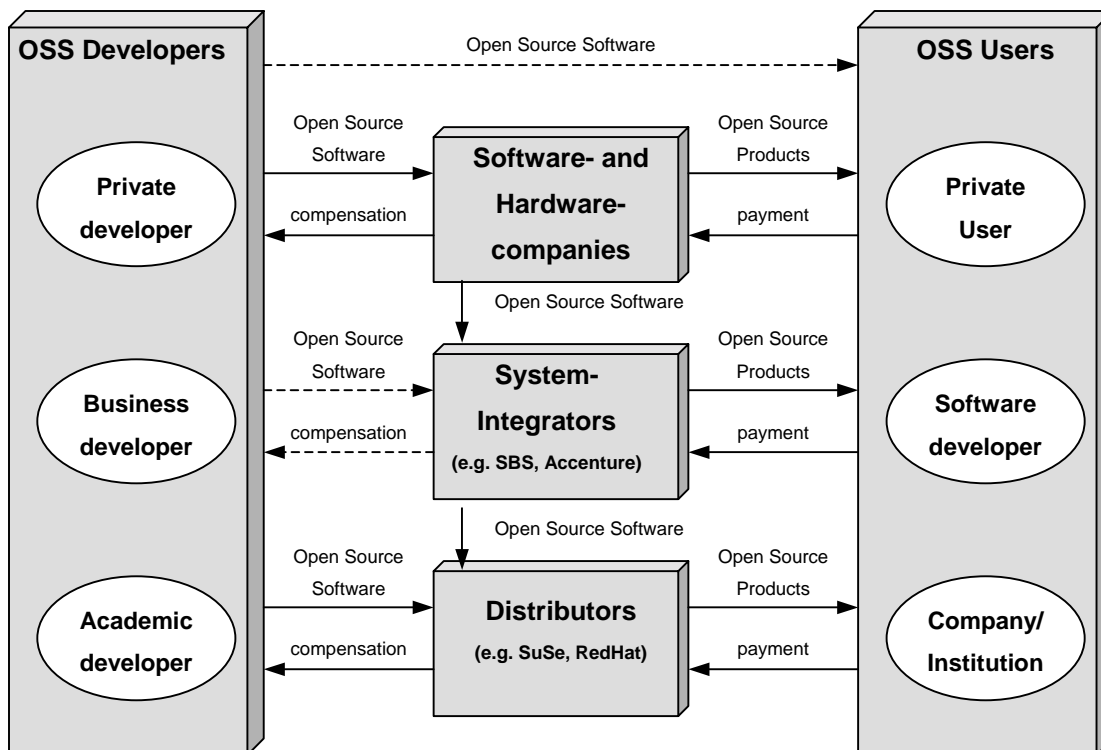
### **Software-, Hardwareanbieter**

Software- und Hardwareanbieter: Ihr Hauptgeschäft ist der Verkauf von kommerzieller Software bzw. Hardware, das ist ihre Kernkompetenz. Damit

machen sie den größten Umsatz und erwirtschaften am meisten Gewinn. Open Source ist nur ein Nebengeschäft. (IBM, Hewlett Packard)

### Anwender

Benutzer der Open Source Software Produkte. Diejenigen, die so entstandene Lösungen einsetzen. Hier sind wiederum private Nutzer und Unternehmen bzw. Institutionen zu unterscheiden. Ergänzend wurden explizit die Softwareentwickler aufgenommen, die OS-Produkte nutzen, z.B. als Entwicklungsumgebungen, da hier ein eigenes Anspruchsprofil zu vermuten ist.



Quelle: Eigene Darstellung

Abbildung 9: Die Beziehungen zwischen den Parteien im Open Source Prozess

Im zweiten Teil der Studie werden für die Intermediäre - die Distributoren und Software- und Hardwareanbieter und Integratoren - die eine Vermittlerfunktion zwischen OS-Entwickler und OS-Anwender besitzen, mögliche Geschäftsmodelle aufgezeigt und vor dem Hintergrund der in dieser Studie aufgezeigten Anforderungen und Restriktionen des OS-Marktes analysiert.

**Literaturverzeichnis:**

**Monographien:**

Autor	Titel
Behlendorf, Brian	Open Source as a Business Strategy. In: Dibona, Chris; Ockman, Sam; Stone, Mark. Hrsg.: Open Sources, Voices from the Open Source Revolution; O'Reilly Verlag, Sebastopol, CA, 1999.
Bensberg, Frank; Dewanto, Lofi	Java und Open Source Geschäftsmodelle, Open-Source-Software: Wirtschaftliche Aspekte unter der Lupe, und wo bleiben die Gewinne? Java Magazin, 2001
Bensberg, Frank; Dewanto, Lofi	Open Source: Hintergründe, Projekte und die Beziehung zu Java, Be Open! Java Magazin, 2001
Bensberg, Frank; Dewanto, Lofi	Open Source Java-Enterprise: Plattformen, Mediatoren und Communities; Java Magazin 2001
BerliOS	BerliOS, Wikipedia, Openfacts, Open Source Lizenzen <a href="http://openfacts.berlios.de/index.phtml?title=Open-Source-Lizenzen">http://openfacts.berlios.de/index.phtml?title=Open-Source-Lizenzen</a>
BMWi	Open-Source-Software, Ein Leitfaden für kleine und mittlere Unternehmen; BMWi Bundesministerium für Wirtschaft und Technologie; 1. Auflage, Stand März 2001
Carbone, Gunnison; Stoddard, Duane	Open Source Enterprise Solutions: Developing an E-Business Strategy, Wiley computer Publishing John Wiley & Sons, Inc., New York, Toronto 2001
Dibona, Chris; Ockman, Sam; Stone, Mark	Open Sources, Voices from the Open Source Revolution; O'Reilly Verlag, Sebastopol, CA, 1999.
Diedrich, Oliver	Sprung auf den Desktop für Linux zu hoch?; Heise Online News, 15. Mai, 2001a
Diedrich, Oliver	Schnellster Webserver unter Linux; Heise Online News, 4. Juli 2000
Duncan, Andy; Hull, Sean	Oracle & Open Source, Tools and Applications, O'Reilly & Associates, Inc., USA 2001
Emnid	Emnid: „Bevölkerungsbefragung über Nutzung und Akzeptanz des

	Betriebssystems Linux“. Eine repräsentative Befragung von TNS EMNID Telecommunication & IT im Auftrag der SuSE Linux AG, 24. März 2001 <a href="http://www.suse.de/de/press/press_releases/archive01/misc/emnid_studie/index.html">www.suse.de/de/press/press_releases/archive01/misc/emnid_studie/index.html</a>
Forster, Frank; Susanek, Alexander	Open Source Projekte und Ihre Bedeutung für das Internet, Teledienstleistungen: Open Source, München 30. Januar 2001
FSF/GNU	Categories of Free and Non-Free Software; gnu.org
Fuggetta, Alfonso	Open Source Software: An Evaluation, 23. April 2002
Gertz, Winfried	Das Consulting-Geschäft um Open Source; Computerwoche, April 27, 2001
Hecker, Frank	„Setting up Shop: The Business of Open Source Software“. 1999 <a href="http://www.hecker.org/writings/setting-up-shop.html">www.hecker.org/writings/setting-up-shop.html</a>
IDC	„Linux: What´s the use? Western Europe 1999-2004“. IDC; 2000
IDC	Windows 2000 vs. Linux für Unternehmensanwendungen, ein IDC White Paper im Auftrag der Microsoft Corporation
Kaven, Oliver	„Performance Tests: File Server Throughput and Response Times“. PC Magazine, November 13, 2001 <a href="http://www.pcmag.com/print_article/0,3048,a%253D16554,00.asp">www.pcmag.com/print_article/0,3048,a%253D16554,00.asp</a>
Kenwood, Carolyn A.	A Business Case Study of open Source Software, The MITRE Corporation, 2001
Köppen, A.; Nüttgens, M.	Open Source: Strategien für die Beratung. In: Scheer, A.-W.; Consulting – Wissen für die Strategie-, Prozess- und IT-Beratung, Saarbrücken, pp. 231-242, 2000 <a href="http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/Consulting/OpenSourceStrategien.pdf">www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/Consulting/OpenSourceStrategien.pdf</a>
Lizenz Center	Lizenz Center, Auflistung der Lizenzen <a href="http://www.ifross.de/ifross_html/lizenzcenter.html">http://www.ifross.de/ifross_html/lizenzcenter.html</a>
Marx, Ben	Linux Manager Guide: Was Entscheider über das Betriebssystem und Open Source wissen müssen; SuSE-Press, Nürnberg, 2001
MATERNA	Studie: Unternehmerische Beurteilung von Open Source Software, eine

GmbH	Online-Befragung, die von der Universität Dortmund in Kooperation mit MATERNA GmbH durchgeführt wurde
Microsoft 2002	A Microsoft Perspective on Linux for Decision Makers, White Paper, Microsoft, 6. September 2002
O'Neill, Peter (META Group)	Linux – Betriebssystemlandschaft im Wandel, Ergebnisse der META Group Studie Deutschland Pressekonferenz 2003
OSI	Open Source Initiative (OSI), The Open Source Definition <a href="http://www.opensource.org/docs/definition.php">http://www.opensource.org/docs/definition.php</a>
O'Reilly (Autor N.N.)	Open Source - kurz & gut, O'Reilly & Associates, 1999.
Raymond, Eric S.	The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999a
Raymond, Eric S.	The Magic Cauldron. In: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999b
Raymond, Eric S.	A Brief History of Hackerdom, In: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA 1999c
Robert Frances Group (RFG)	Robert Frances Group, Marktforschungsunternehmen für IT-Beratung, Studie: Total Cost of Ownership for Linux in the Enterprise, Juli 2002
Schönfeldt, Rene; von Raison, Andre	Linux im Unternehmen, 2. Business-Kongress auf dem LinuxTag, dpunkt.verlag GmbH, 1. Auflage Heidelberg 2001
Sieckmann, Jens	"Bravehack: Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source; ihr Wesen, ihre Geschichte, ihre Organisationen und Projekte", 06.03.2001, <a href="http://www.bravehack.de/html/node1.html">http://www.bravehack.de/html/node1.html</a>
Stallman, Richard	The GNU Operating System and the Free Software Movement. In: Raymond, Eric S. Hrsg.:The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary; O'Reilly Verlag, CA



	1999
Weber, Niels; Suhr, Gregor	Seminararbeit „Ausarbeitung: Rechtliche Aspekte von OSS“ von der Technischen Universität Berlin.
Weiss, George	Studie: The Future of Linux and Open Source, Gartner Group 2001
Wheeler, David A.	Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers, 2003, <a href="http://www.dwheeler.com/oss_fs_why.html">http://www.dwheeler.com/oss_fs_why.html</a>
Wichmann, Thorsten	FLOSS Final Report – Part 1, Free/Libre Open Source Software: Survey and Study, Use of Open Source Software in Firms and Public Institutions, Evidence from Germany, Sweden and UK; Berlecon Research, Berlin, Juli 2002
Wichmann, Thorsten	FLOSS Final Report – Part 2, Free/Libre Open Source Software: Survey and Study, Firms’ Open Source Activities: Motivations and Policy Implications; Berlecon Research, Berlin, Juli 2002
Wichmann, Thorsten	FLOSS Final Report – Part 3, Free/Libre Open Source Software: Survey and Study, Basics of Open Source Software Markets and Business Models; Berlecon Research, Berlin, Juli 2002
Wieland, Thomas	Entwicklungen und aktuelle Einsatzgebiete, Siemens AG, Corporate Technology, München 2001
Wieland, Thomas	Open Source im Unternehmen, Siemens AG, Corporate Technology, München 2001
Wieland, Thomas	Open Source aus Sicht des Anwenders, Siemens AG, Corporate Technology, München 2001
Wieland, Thomas	Open Source aus Sicht des Herstellers, Siemens AG, Corporate Technology, München 2001
Wieland, Thomas	Linux als Geschäftsfaktor, Chefredakteur „Linux Enterprise“, Businesskongress des Linuxtages 2000, Stuttgart, 29. Juni 2000



Cooperative Computing & Communication Laboratory

C-LAB
Marketing
Fürstenallee 11
D-33102 Paderborn

Telephone +49-5251-60-6060
Telefax +49-5251-60-6066
E-Mail marketing@c-lab.de
URL http://www.c-lab.de

Befragung über Ihre Zufriedenheit mit dem Report „Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte“

Wir bitten Sie, sich einen kurzen Moment Zeit zu nehmen, um uns ein paar Fragen über Ihre Einschätzung dieses Reports zu beantworten. Damit helfen Sie uns, Ihre Bedürfnisse besser zu verstehen. Wir möchten unsere Reports stärker nach Ihrem Interesse ausrichten, um so einen größeren Mehrwert bieten zu können. Vielen Dank für Ihre Mitarbeit.

Wie bewerten Sie das Thema dieses Reports?

trifft voll zu (checkboxes) trifft überhaupt nicht zu (checkboxes)
Aktuell
Interessant

Wie bewerten Sie den Inhalt dieses Reports?

trifft voll zu (checkboxes) trifft überhaupt nicht zu (checkboxes)
Aktuell
Interessant
Verständlich
Praxisrelevant
Informativ
Innovativ

Weitere Kommentare:

Horizontal lines for additional comments

Freiwillige Angaben:

Name, Vorname:
Telefon:
E-Mail:

Bitte senden Sie das ausgefüllte Formular per Post, Fax oder E-Mail an die jeweilige Adresse (s. o.).