

Kapitel 1

Grundlagen und Erfahrungen

Einleitung

SVETLANA KHARITONIOUK UND PATRICK STEWIN

Was in der großen Weltpolitik der Fall der Berliner Mauer war, das wird dieses Votum in unserer Branche sein.

Richard Seibt, Vorstandsvorsitzender der Suse Linux AG¹

Der Beschluss der Münchener Regierung im Mai letzten Jahres, 16000 EDV-Arbeitsplätze der Stadtverwaltung von der marktbeherrschenden Microsoft-Software auf Linux und andere Open-Source-Produkte umzustellen, hat für viel Aufregung in der Öffentlichkeit gesorgt. „Eine Entscheidung Münchens für Linux hätte einen besonderen Leuchtturmcharakter für andere Kommunen“, betonte Andreas Gebhard von der Initiative Bundestux (Bundestux 2003), „weil hier erstmals die gesamte Verwaltung einer Millionenstadt auf Freie Software umgestellt würde.“

Die Schlagzeilen um dieses Ereignis aus dem Sommer 2003² haben wieder einmal verdeutlicht, dass Open Source und Freie Software nicht nur ein Thema für Informatiker, sondern auch ein Gegenstand der Politik sind. Die politischen Aspekte von Open Source (OS) werden von Bundestux – der Initiative für die Einführung von Freier Software im Deutschen Bundestag – wie folgt formuliert: *Ordnungspolitik und Freier Wettbewerb, Demokratie und Standortvorteile* (Bundestux 2003).

Für den durchschnittlichen PC-Benutzer gehören solche Aspekte nicht zum Alltag. Open-Source-Software (OSS), wie z.B. Linux, Openoffice oder Mozilla, kennt er meistens nur vom Namen und nicht durch die Benutzung auf dem eigenen PC. In der Wirtschaft und im öffentlichen Bereich sieht das inzwischen anders aus. Dort wird das Interesse auf OSS unter anderem durch versprochene Kostenvorteile und Stabilität gelenkt.³ Auch in der Wissenschaft spielt das Phänomen Open Source eine

¹ In (Kerbusk 2003).

² Weitere Informationen zur Münchner Entscheidung unter: <http://www.lochner-fischer.de/frameset.html?themel/linux3.htm> oder <http://linux-muenchen.de/>.

³ Studien dazu sind unter <http://www.dbresearch.de/PROD/PROD0000000000047532.pdf>, http://www.it-surveys.de/itsurvey/pages/studie_oss_executive_summary.html oder <http://www.prolinux.de/news/2003/5265.html> zu finden.

wichtige Rolle. Für die aus den verschiedensten Bereichen stammenden Forscher wirft das Thema um die Open-Source-Bewegung viele Fragen auf:

- Was bringen offene Standards in der IT, und ist in diesem Zusammenhang OS sicher?
- Was motiviert viele Entwickler, sich unentgeltlich in ihrer Freizeit in der OS-Gemeinde zu engagieren?
- Welche Vorteile haben OS-Lizenzen für den Nutzer bzw. den Anbieter und Entwickler?
- Welche Vorteile haben öffentliche Institutionen bei der Nutzung von OSS?
- Soll OS-Nutzung und -Entwicklung (staatlich oder betrieblich) gefördert werden?
- Kann man mit OS Gewinne erwirtschaften?
- Können Information und Software Eigentum sein, oder sollten sie generell als „frei“ gelten?
- Ist OS-Entwicklung ein Vorbild für die Forschung?
- Ist Open Source ein besseres Gesellschaftsmodell?

Die vorliegende Publikation will – nicht zuletzt, um die Entscheidung des Lesers für oder wider Open-Source-Software zu erleichtern – Einblicke in derzeitige Diskussionen bieten. Aktuelle Entwicklungen werden in diesem ersten Band eines als Reihe angelegten Jahrbuches zunächst mit dem Focus auf die Bundesrepublik Deutschland in ausgesuchten Beiträgen verschiedener Autoren präsentiert. Die nach Themenschwerpunkten sortierten Kapitel dieses Bandes werden jeweils durch eine Einleitung ergänzt, die einen kurzen Überblick verschaffen soll.

Im ersten Kapitel wird auf die Grundlagen von Open Source und Freier Software eingegangen. Es werden die Geschichte, die Definition und die bekanntesten bzw. erfolgreichsten Projekte vorgestellt. In den anschließenden Artikeln berichten Dienstleister und Anwender über ihre Erfahrungen bei der Migration von Micro-soft-Produkten zu Linux und anderer Open-Source-Software. Den Erfahrungsberichten folgt eine Beschreibung der KDE-Gemeinschaft, die von der beteiligten Projektleiterin Eva Brucherseifer verfasst wurde. Seinen Abschluss findet dieses Kapitel in einer Rezension von Steven Webers grundlegendem Buch⁴ „Success of Open Source“.

Im Folgenden soll zunächst der Begriff Open Source bezogen auf Software eingeordnet werden.

Der Open-Source-Begriff

Der Begriff Open Source (OS) wird in diesem Buch, wie oft auch in anderer Fachliteratur und in der Presse, für Freie Software (*free software*) und *open source software* verwendet. Die Unterschiede dieser beiden Ausprägungen sollen im Laufe der folgenden Abschnitte geklärt werden.

Open Source heißt aus dem Englischen übersetzt „quelloffen“. Um das zu erläutern, muss zunächst näher auf die Entstehung von Software eingegangen werden. Jede Software wird von einem Programmierer in einer Programmiersprache ge-

⁴ Erscheint im April 2004.

geschrieben und kann anhand des so erzeugten Programmtextes vom Menschen nachvollzogen werden. Wird dieses als Quelltext, Quellcode oder Sourcecode bezeichnete Produkt mit oder ohne das dazugehörige ausführbare Programm veröffentlicht (und durch bestimmte Lizenzen in seiner freien Nutzung gesichert), spricht man von Open-Source-Software.

Damit ein Programm gestartet werden kann, wird der Programmtext in eine für den Computer verständliche Form (die für den Menschen nicht mehr lesbar ist) übersetzt (kompiliert), die dann von der Maschine ausgeführt werden kann. Bei handelsüblicher – also meist proprietärer – Software bleibt wegen nachvollziehbarer Interessen des anbietenden Unternehmens der Quellcode verborgen. Der Benutzer erhält ein Programm nur in der für den Computer ausführbaren Form, er kann damit zwar die Funktion, nicht aber die Art des Funktionierens seiner Erwerbung nachvollziehen. Entsprechende Lizenzbedingungen schützen das fertige Produkt vor dem Vervielfältigen, Verändern, Weiterverbreiten und Dekompilieren, also dem Rückübersetzen in eine zumindest dem Experten verständliche Form. Der Nutzer sieht nur das, was auf seinem Bildschirm passiert. Ein großer Nachteil für ihn ist, dass er diese Software bei Bedarf nicht an seine konkreten individuellen Bedürfnisse anpassen kann. Jede Weiterentwicklung und Fehlerbehebung liegt in der Hand des Herstellers.

Diese Nichtflexibilität proprietärer Software hat viele Entwickler bewegt, eigene Programme zu schreiben und diese mit ihrem Quellcode im Internet zu veröffentlichen. Als Alternative zur proprietären Software ist eine „freie“ bzw. „quelloffene“ Software entstanden, die jeder nach seinen Vorstellungen anpassen kann. Lizenzgebühren werden nicht erhoben, im Allgemeinen kann jeder Nutzer den Code aus dem Internet herunterladen. Verbesserungsvorschläge und Änderungen kann er ebenfalls im Internet, z.B. durch Mailinglisten, veröffentlichen. So wird freie bzw. offene Software durch Anregungen der Nutzer und weltweites Mitwirken anderer Entwickler ständig verbessert. Open-Source-Software wird deshalb inzwischen nicht nur als Produkt, sondern auch als Entwicklungsmodell begriffen. Beachtenswert ist dabei, dass die Entwicklergemeinde durch die schnellen Kommunikationswege und offen gelegte Ergebnisse oft viel schneller als ein Unternehmen reagieren kann.

Der Entwicklungsprozess wird von den Beteiligten selbst koordiniert, legt aber keine Beschränkungen für den Einzelnen fest. Das Phänomen der freien Software besteht auch darin, dass eine große Anzahl der Entwickler in ihrer Freizeit und meist kostenlos in den Projekten arbeitet. Linus Torvalds, der Initiator von Linux, verdiente seinen Unterhalt lange Zeit in einem Unternehmen, in dem er mit Linux nichts zu tun hatte. Über die Motivation von Entwicklern und die Abläufe in einer solchen Entwicklungsgemeinde erfährt der Leser mehr im Kapitel Technik und im Bericht der KDE-Projektleiterin Eva Brucherseifer in diesem Kapitel des Buches.

Entscheidende Einflüsse auf die Entstehung der Open-Source-Bewegung

Um die Entstehung der OS-Bewegung klären zu können, muss die Geschichte des Internets und der Software herangezogen werden.

Dieses Netz ist nicht nur Produkt einer von vielen Beteiligten geschaffenen wissenschaftlichen Arbeit, sondern auch die Entwicklungsumgebung für Informatiker und die Kommunikationsgrundlage vieler anderer Wissenschaften.⁵

Die Koordination der Entwicklungsarbeit für das Internet bzw. der technischen Standards hat 1986 die Internet Engineering Task Force (IETF) übernommen.⁶ Die Standards wurden von Anfang an offen gelegt. Jeder konnte sie einsehen, sich bei den Diskussionen auf den Mailinglisten beteiligen und somit Verbesserungsvorschläge und Kommentare einarbeiten. Es entstanden schnell weitere Diskussionsmöglichkeiten in öffentlichen Foren und durch synchrone Kommunikationsformate (wie z.B. Chat).

Das „Entwicklungsmodell“ des Internets ähnelt dem der OSS. Einmal vorhanden, ermöglichte das Internet dann eine weltweite, kostengünstige und verwaltungsarme Zusammenarbeit auf elektronischem Wege. Damit schafft es eine wichtige Grundlage für die OS-Gemeinschaft und für die Verbreitung Freier Software.

Wenn Open-Source-Software im großen Stil erst durch die weltweite Vernetzung möglich wurde, so hat sie wiederum auch die Entwicklung des Internets beeinflusst, wie beispielsweise durch die Integration der gängigen Internet-Protokolle in das freie Betriebssystem BSD-UNIX. Auch Projekte wie BIND und Sendmail sind in diesem Zusammenhang zu erwähnen. Die Entwicklung des Internets hat Open-Source-Software entscheidend vorangetrieben, während durch diese wiederum das weltweite Netz entscheidende Verbesserungen erfuhr.

Im Gegensatz zum Internet wurde die Geschichte von Open-Source-Software aber nicht nur durch die öffentliche Entwicklung (z.B. an Universitäten), sondern auch durch das Engagement einzelner Wissenschaftler und den kommerziellen Kommunikations- und Computermarkt geprägt.

Die ersten Computernutzer waren Fachleute, welche ihre Software selbst nachvollziehen und weiterentwickeln konnten. Software wurde in der Anfangszeit der Computer als „Bedienungsanleitung“ kostenlos mit der Hardware ausgeliefert. Ende der 60er Jahre begann der damalige Marktführer IBM, den zum Bedienen eines Computers erforderlichen Programmcode separat anzubieten, und schuf so nach und nach einen Markt für proprietäre Software.⁷ Die Standardisierung der Hardware – wie z.B. durch den IBM-PC, der bis heute die Mehrzahl der Schreibtische bevölkert – ließ den Computer dann schnell zu erschwinglicher Massenware werden und führte in der Folge zu einer enormen Nachfrage an Software. Immer mehr Unternehmen, Behörden und schließlich Privathaushalte nutzten die Computertechnik. Als der Benutzer nicht mehr automatisch gleichzeitig Computerfachmann war, hörte er auf, sich für die genauen Funktionen „seiner“ Programme zu interessieren. Er wollte sie benutzen. Andere Nutzer entwickelten hingegen ein starkes Interesse für Software und wurden zu Programmierern – gerade auch zu OS-Programmierern, weil ihnen die Funktionen der als „Blackbox“ ohne Quellcode ausgelieferten proprietären Software nicht ausreichten, weil sie verstehen wollten, was ihr Rechner ge-

⁵ Vgl. (Grassmuck 2002, S. 179 ff.).

⁶ Siehe <http://www.ietf.org>.

⁷ Vgl. http://www-1.ibm.com/ibm/history/history/decade_1960.html.

nau tut, weil sie die Programme verändern, verbessern und an spezielle Bedürfnisse anpassen wollten.

Von UNIX zu GNU/Linux

Als Ursprung der ersten OSS wird oft das Betriebssystem UNIX genannt. Die Begriffe „Free Software“ und „Open Source“ sind erst viel später entstanden. Im Jahr 1969 wurde die erste UNIX-Version bei AT&T Bell Telephone Laboratories entwickelt. UNIX konnte damals von AT&T auf Grund eines Kartellrechtsprozesses nicht kommerziell vertrieben werden. Es wurde gegen geringen Lizenzgebühren an Universitäten verkauft. Da kein Support von AT&T angeboten wurde, entwickelten die Nutzer das Betriebssystem selbst weiter. Dabei kommunizierten sie untereinander schon auf elektronischem Wege per E-Mail, Newsletter, Konferenzen und Dateiübertragung innerhalb einer universitätsinternen UNIX-*community*.

An der University of Berkeley in Kalifornien wurden mehrere weiterentwickelte Programme zur Berkeley Software Distribution (BSD) zusammengestellt, deren Nachfolger FreeBSD⁸ noch heute als OS-Projekt existiert. Ab 1982 wurde UNIX dann aber stärker kommerzialisiert. Der Lizenzdruck von AT&T erschwerte eine offene Weiterentwicklung.

Mitte der 80er Jahre wurde der Begriff „Freie Software“ von Richard Stallman geprägt. Die Ursprünge seiner Idee liegen allerdings in den 70er Jahren. Als Mitarbeiter am MIT (Massachusetts Institute of Technology) lernte er 1971 im Artificial Intelligence Lab (AI Lab) die „software sharing community“⁹ kennen. Diese *community* bestand aus Programmierern, die ihre Software offen gehalten, geteilt und gegenseitig verbessert haben. Sie gilt als erste Verkörperung der Philosophie der Freien Software. Als diese Gemeinschaft Anfang der 80er Jahre beinahe aufgelöst wurde, beschloss Stallman, die Voraussetzungen für eine neue Gemeinschaft zu schaffen:

Die Antwort war klar: zuerst wurde ein Betriebssystem gebraucht. Das war der entscheidende Punkt um anzufangen, einen Computer zu nutzen. Mit einem Betriebssystem kann man viele Dinge machen; ohne kann man den Computer überhaupt nicht nutzen. Mit einem freien Betriebssystem könnten wir wieder eine neue Gemeinschaft von zusammenarbeitenden Hackern haben – und jeden einladen, sich uns anzuschließen. (Stallman 2003)

Das neue Betriebssystem GNU (GNU's Not Unix) adaptierte die Funktionalität von UNIX, wurde aber von der ersten Codezeile an neu geschrieben. Im Gegensatz zu UNIX war es „frei“. Richard Stallman veröffentlichte diese Idee 1983 bei den UNIX-Newsgroups und lud zur Mitarbeit ein. Mit der Gründung der *Free Software Foundation*¹⁰ im Jahr 1985 schuf er eine Finanzierungsmöglichkeit und einen Organisationsrahmen für den GNU-Vertrieb sowie eine Plattform für die weitere Unterstützung von Freier Software.

⁸ Siehe auch <http://www.freebsd.org>.

⁹ Vgl. hierzu (Levy 1984).

¹⁰ Vgl. <http://www.fsf.org>.

„Freie Software‘ hat etwas mit Freiheit zu tun, nicht mit dem Preis. Um das Konzept zu verstehen, ist an ‚frei‘ wie in ‚freier Rede‘, und nicht wie in ‚Freibier‘ zu denken.“ (Free Software Foundation 2002). Die „Freiheit“ der Software wurde in der so genannten *GNU General Public License (GPL)*¹¹ manifestiert und setzt Folgendes voraus¹²:

- den Zugang zum Quellcode,
- die Freiheit, die Software zu kopieren und weiterzugeben,
- die Freiheit, das Programm zu ändern, und
- die Freiheit, das Programm unter denselben Bedingungen zu verbreiten.

Das GNU-Projekt hatte im Laufe der Zeit große Fortschritte gemacht und viele Programme als Freie Software in das zu vollendende Betriebssystem implementiert. Allerdings gelang es bis Anfang der 90er Jahre nicht, einen Kernel (essenzieller Kernteil eines Betriebssystems – verbirgt die direkte Ansteuerung der Hardware vor dem Benutzer) für GNU zu entwickeln.

Glücklicherweise kodierte 1991 der aus Finnland stammende Student Linus Torvalds den an UNIX angelehnten Kernel Linux, der von nun an zusammen mit der GNU-Software der FSF ein vollständiges, freies Betriebssystem (GNU/Linux) bildete. Mit diesem Zusammenschluss war ein ernst zu nehmender Konkurrent für Microsoft-Produkte und andere proprietäre Software geschaffen worden. GNU/Linux verkörpert all die Vorteile, die OSS zugeschrieben werden. Fehlende Lizenzgebühren, Sicherheit, Stabilität und Flexibilität erhöhten den Bekanntheitsgrad von OSS und haben somit zu einer weiten Verbreitung dieser alternativen Programme in der Wirtschaft und im öffentlichen Sektor beigetragen.

Zurzeit wird der Marktanteil von GNU/Linux im Serverbereich weltweit auf Platz zwei mit 26 %¹³ geschätzt – die Installationen nehmen weiter zu. Verschiedene Distributoren wie SuSE, Redhat, Debian, Mandrake u.a.¹⁴ vertreiben das Softwarepaket, sie verlangen Gebühren für Support, Schulungen, Handbücher etc., wodurch das GNU/Linux-Betriebssystem nicht mehr grundsätzlich kostenfrei ist. Die Möglichkeit, eine solche Distribution unter Verzicht auf weiteren Service gebührenfrei aus dem Internet herunterzuladen, geht dadurch aber nicht verloren.

Die Open-Source-Initiative

Im Laufe der Zeit haben als „Freie Software“ veröffentlichte Programme wie GNU/Linux immer mehr an Bedeutung gewonnen. Neben der *Free Software Foundation* etablierte sich Ende der neunziger Jahre eine zweite, von Bruce Perens und Eric S. Raymond gegründete Bewegung, die sich den Namen *Open Source Initiative (OSI)* gab.

Die Gründung dieser Initiative 1998 ist insbesondere auf zwei einschneidende Ereignisse zurückzuführen: Die Firma Netscape hatte den Quellcode seines Webrowsers zur offenen Programmierung freigegeben, damit war zum ersten Mal die offene Entwicklung eines auch bei Privatanwendern bekannten und weit verbreite-

¹¹ Nachzulesen hier: <http://www.gnu.org/licenses/gpl.html>.

¹² Vgl. (Grassmuck 2002).

¹³ Quelle: ICD, gefunden in (Kerbusk 2003).

¹⁴ Vgl. <http://www.linux.org/dist/list.html>.

ten Programms möglich. Weiterhin fand am 3. Februar 1998 das erste so genannte „Open-Source-Gipfeltreffen“ statt. Auf dieser Konferenz wurde das neue Label „Open Source“ gegründet (*Open Source Certification, Open Source Trademark*). Kurz nach diesem Treffen entstand die Webseite www.opensource.org, die bis heute den Internetauftritt der Open-Source-Initiative darstellt, die sich dort vorstellt und die aktuellsten Neuigkeiten zum Thema präsentiert. Weiterhin finden sich dort die Open-Source-Definition und andere interessante Dokumente wie z.B. der Aufsatz „Advocacy“¹⁵, der weitere Hintergründe zu erläutern versucht.

Ziel der Open-Source-Initiative ist es, Open-Source-Software der Wirtschaft näher zu bringen. Die Zusammenarbeit wird gesucht, um die OS-Bewegung zu stärken und um Geschäftsmodelle zu finden, mit denen es möglich wird, mit OSS Gewinne zu erwirtschaften. An dieser Stelle wird auch deutlich, warum man den Begriff „Open Source“ eingeführt und nicht bei „Free Software“ geblieben war. Die *Free Software Foundation* verfolgt einen eher idealistischen Ansatz. Sie verkörpert eine besondere Philosophie der Freiheit: Nicht nur der Quellcode einer Software muss frei sein, sondern auch an ihm getätigte Modifikationen. Mit diesem Ansatz blieb quelloffene Software von der Wirtschaft und von den Unternehmen weitestgehend ungenutzt. Zum einen kam es zu dem Missverständnis, „free“ stünde für kostenlos und es ließe sich so mit quelloffener Software in keiner Weise Gewinn erwirtschaften. Zum anderen hegten Unternehmen die Befürchtung, dass durch die Verwendung von quelloffener Software ihre eigene „infiziert“ würde, sodass diese dann auch „frei“ sein muss.

Durch die initiierten Veränderungen der OSI erlangte Open-Source-Software und die im folgenden Absatz erläuterte Open-Source-Definition weiter an Bedeutung – unter anderem auch für Wirtschaft und Unternehmen. Ein Jahr später kam es zu einem weiteren Gipfeltreffen. Dort waren nun auch namhafte Unternehmen wie Sun Microsystems, IBM und Oracle vertreten.

Die Open-Source-Definition

Die OSI entwarf die bereits oben genannte Open-Source-Definition¹⁶, die derzeit (24.1.2004) in der Version 1.9 existiert und die Grundlage für alle Open-Source-Lizenzen bildet.

Beim Entwurf der Definition hat man sich auf vier Schwerpunkte geeinigt, die eine konforme Lizenz beinhalten muss: *Verbreitung* (bezieht sich auf die Lizenz und die Software als solche; Punkt 1 und 7), *Schutz des Quellcodes* (Punkte 2, 3, 4), *Ausschlussmöglichkeiten* (z.B. von Personen oder Technologien; Punkte 5, 6, 8 und 10) und *Einflussnahme* (auf andere Software; Punkt 9).

Konkret beinhaltet die Definition Folgendes¹⁷:

1. Freie Weiterverbreitung

Es soll gewährleistet werden, dass beliebig viele Kopien der Software (auch Distributionen, also eine Zusammensetzung von mehreren Softwaremodulen) angefer-

¹⁵ Vgl. <http://www.opensource.org/advocacy/>.

¹⁶ Vgl. (Open Source Initiative 2004).

¹⁷ Vgl. (O'Reilly 1999).

tigt werden können. Diese Kopien dürfen weitergegeben und sogar verkauft werden. Das Verkaufen bzw. Kopieren ist dann allerdings als Dienstleistung zu betrachten, da für die Software selbst keine Lizenzgebühr erhoben werden darf.

2. *Quellcode*

Der Quellcode soll offen und in einer für den Menschen nachvollziehbaren Form verbreitet werden (ein Programmierer soll den Code entsprechend verändern können). Wird der Quellcode nicht zusammen mit der kompilierten Form verbreitet, dann muss klargestellt sein, dass er lizenzgebührenfrei aus dem Internet bezogen werden kann.

3. *Auf dem Programm basierende Werke*

Auf Open-Source-Programmen basierende Werke *sollen* unter den gleichen Lizenzbedingungen weiterverbreitet werden. Das heißt aber auch, dass eine Weiterverbreitung unter anderen Lizenzbedingungen nicht ausgeschlossen wird. Es ist also möglich, dass ein auf einem Open-Source-Programm nach diesen Regeln basierendes Werk nicht frei sein *muss*. Es gibt Open-Source-Lizenzen, die entsprechend ausformuliert wurden (wie die *Berkeley Software Distribution License*).

4. *Die Unversehrtheit des Originalcodes*

Originalcode und veränderter Code müssen unterscheidbar sein. Der Ruf des Autors soll dadurch erhalten bleiben, und die Nutzer sollen erkennen, wer welchen Code wirklich geschrieben hat.

5. *Keine Diskriminierung von einzelnen Personen oder Gruppen*

Niemand kann von der Nutzung oder Weiterentwicklung von Open-Source-Software ausgeschlossen werden. Dies würde der Idee von Open Source widersprechen. Weiterhin wird das Ziel verfolgt, möglichst viele Personen bzw. Gruppen für die Entwicklung von Open-Source-Software zu gewinnen.

6. *Keine Einschränkungen für bestimmte Anwendungsbereiche*

Dieser Punkt der Definition lehnt an Punkt 5 an. Allerdings zielt er auf die Anwendungsbereiche von Open-Source-Software ab. Eine mögliche Lizenz darf keine bestimmten Einsatzgebiete der Software verbieten (sonst ist sie nicht mehr Open-Source-konform). Dazu zählt auch die kommerzielle Nutzung.

7. *Verbreitung der Lizenz*

Es ist zu verhindern, dass Open-Source-Software auf indirektem Weg ihren Status verliert. Die Rechte des Nutzers müssen erhalten bleiben, ohne dass zusätzliche Bedingungen oder Einverständniserklärungen beachtet werden müssen.

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

Es soll ausgeschlossen werden, dass die betrachtete Software nur innerhalb einer bestimmten Software-Distribution als Open-Source-Software gelten darf. Wird die betrachtete Software aus einer Distribution herausgenommen, so gelten weiterhin die Rechte für den Benutzer, welche der Software innerhalb der Distribution gegeben waren.

9. Die Lizenz darf andere Software nicht beeinträchtigen

An dieser Stelle der Definition wird die Einflussnahme einer Lizenz für Open-Source-Software auf andere Software thematisiert. „Die Lizenz darf andere Software nicht beeinträchtigen“ besagt, dass es beispielsweise nicht möglich sein darf zu fordern, dass die gesamte Software, die auf einem Computer oder System genutzt wird, Open-Source-Software sein muss.

10. Die Lizenz muss technologisch neutral sein

Wenn eine Lizenz formuliert wird, dann muss auch darauf geachtet werden, dass die technologische Neutralität gewahrt wird. Es soll nicht möglich sein, innerhalb einer Lizenz eine bestimmte Technologie oder eine Schnittstelle vorzuschreiben und somit andere auszuschließen.

Eine Lizenz, welche als Open-Source-konform gelten soll, muss der Definition genügen. Die *GNU General Public License*, die *Berkeley Software Distribution License* und die *Mozilla Public License* sollen hier als die bekanntesten Beispiele für Open-Source-konforme Lizenzen genannt werden. Open-Source-Projekte, bei denen unter anderem diese Lizenzen angewendet werden, sind weiter unten in dieser Einleitung aufgezählt. Eine detailliertere Betrachtung von Open-Source-Lizenzen findet der Leser im Kapitel Recht und Politik.

Es ist zu beachten, dass einige dieser Lizenzen, wie die erste Lizenz für quelloffene Software, die *GNU General Public License* (GPL), schon vor der Einführung der Open-Source-Definition existierten. Die GPL und andere Lizenzen der *Free Software Foundation* entsprechen den Bedingungen des OSI, auf diesen beruhende neuere Lizenzen aber nicht unbedingt den Ansprüchen der FSF. Damit ist „Free Software“ zwar immer auch „Open-Source-Software“, andersherum gilt diese Beziehung aber nicht. Die Regeln für Free-Software-Lizenzen sind strenger als die für Open-Source-Software. So *müssen* beispielsweise Programme, die Code verwenden, der unter den Bedingungen der FSF lizenziert wurde, unbedingt wieder Free Software sein, wurde Code aus einem Open-Source-Produkt verwendet, *kann* das Endprodukt wiederum als Open Source vertrieben werden (dies wäre im Sinne des OSI sicher auch wünschenswert), es *muss* aber nicht. Wegen dieser asynchronen Wechselwirkung (und weil die offenen Quellen in jedem Falle der gemeinsame Nenner sind) hat sich im allgemeinen Sprachgebrauch der Begriff „Open-Source-Software“ für jede Art von freier oder quelloffener Software durchgesetzt.

Die Anhänger der Freien Software vertreten weit höhere Ideale, sie propagieren eine gesellschaftliche Utopie, während die Akteure der Open-Source-Initiative vor allem das technische Entwicklungsmodell von Software mit offenen Quellen praktizieren – nach Möglichkeit zusammen mit proprietären Unternehmen. Wegen ihrer schon in den Grundlagen verschiedenen Ansätze widersprechen sich diese beiden Ideen damit nicht so sehr, wie es auf den ersten Blick scheint.

Als Open Source werden inzwischen auch Projekte, Produkte und Vorgehensweisen bezeichnet, die nichts mehr mit Software zu tun haben, aber den Definitionen und dem OS-Geist entsprechen. Beispiele dafür sind die *Open Web School*¹⁸ – ein

¹⁸ Vgl. <http://www.openwebschool.de>.

Web-Portal, wo Unterrichtsmaterialien für Lehrer und Schüler unter der GPL angeboten werden – und die von interessierten Laien im Internet erstellte Enzyklopädie Wikipedia¹⁹, die unter der ebenfalls von der FSF erstellten *GNU Free Documentation License* erschienen ist. Es handelt sich in beiden Fällen nicht um Software im klassischen Sinne, sondern um vielfältige Inhalte, die von ihren Autoren zum Kopieren, zur Weitergabe, zum Verändern und zur Verbreitung unter den Bedingungen der „Offenheit“ zur Verfügung gestellt wurden.

Vergleich von Softwaremodellen

Damit quelloffene Software richtig eingeordnet werden kann, muss neben der Darstellung ihrer Definitionen ein Vergleich mit anderen Modellen vorgenommen werden. Dass Lizenzen der üblichen proprietären Software jeden erdenklichen Umgang mit ihr, außer der Benutzung nach dem Entrichten der Lizenzgebühr, meist verbieten, aus nahe liegenden Gründen aber zumindest das Weitergeben von Kopien und das Dekompilieren, also das Rückübersetzen des gelieferten Programmcodes in eine Programmiersprache, ausschließen, dürfte im Allgemeinen bekannt sein. Daneben gibt es aber noch die auch als proprietär zu bezeichnenden Modelle Freeware und Shareware sowie den Sonderfall Public Domain.

Zum Vergleich bzw. zur Abgrenzung soll die folgende Tabelle dienen²⁰. Sie enthält eine Kurzbeschreibung der Modelle und stellt wichtige Nutzungsrechte (I bis V) gegenüber:

I. Der Quellcode ist modifizierbar.

II. Modifikationen müssen immer unter denselben Bedingungen veröffentlicht werden.

III. Das Produkt ist lizenzgebührenfrei.

IV. Die Nutzung ist uneingeschränkt.

V. Die Weiterverteilung ist erlaubt.

	<i>Kurzbeschreibung</i>	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>
Free Software	Der Quellcode ist offen, und seine Modifikationen müssen auch offen bleiben.	x	x	x	x	x
Open-Source-Software	Quelloffene Software soll Unternehmen und Wirtschaft näher gebracht werden. Die kommerzielle Nutzung soll einfacher sein (im Vergleich zu Free Software).	x		x	x	x
Public Domain	Diese Software ist als ein Sonderfall zu betrachten: Der Urheber verzichtet komplett auf das <i>copyright</i> . Somit wird diese Software zum Gemeingut und kann uneingeschränkt genutzt werden. Sollte der Quellcode zur Verfügung stehen, liegt Open-Source-Software vor.	x		x	x	x

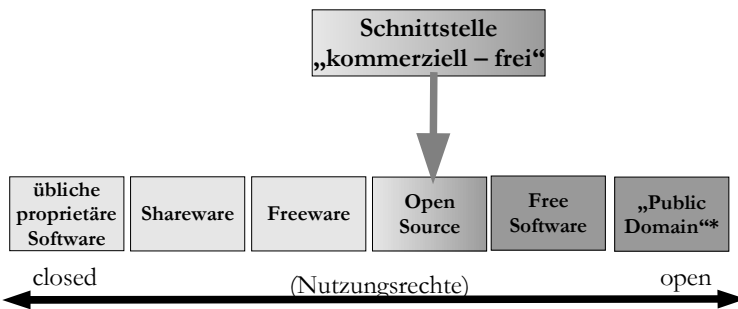
¹⁹ Vgl. <http://de.wikipedia.org>.

²⁰ In Anlehnung an (Ploch u.a. 2002, S. 20 ff.).

	<i>Kurzbeschreibung</i>	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>	<i>V</i>
Freeware	Diese Art der Software ist keine Free Software. Es werden zwar keine Lizenzgebühren erhoben, aber der Quellcode steht nicht zur Verfügung.			x	x	x
Shareware	Hierunter wird Software verstanden, die für eine vom Autor festgelegte Testphase genutzt werden darf. Ist die Testphase abgelaufen, so sind Lizenzgebühren zu bezahlen.					x
Übliche proprietäre Software	Proprietäre Software wird als „Gegenmodell“ zur Free bzw. Open-Source-Software gesehen: Die Weiterverteilung ist verboten, der Quellcode bleibt verborgen, sämtliche Modifikationen sind verboten, und es werden immer Lizenzgebühren verlangt. Es bleiben fast alle Rechte beim Anbieter.					

Tab. 1: Vergleich von Softwaremodellen

Die folgende Grafik²¹ verweist auf einen weiteren interessanten Aspekt bei der Einordnung von Open-Source-Software im Vergleich zu den oben genannten Modellen. Open-Source-Software kann als eine Art Schnittstelle zwischen Free Software und der kommerziellen bzw. geschlossenen, nicht freien Software betrachtet werden:



*Falls Quellcode vorhanden

Abb. 1: Schnittstelle „kommerziell – frei“

Der nächste Abschnitt wird zeigen, welche speziellen Open-Source-Projekte unter anderem diese Schnittstelle besetzen.

Erfolgreiche OS-Projekte

Auf Grund der vielen OS-Projekte ist es kaum möglich, eine genaue Angabe über die Anzahl der bis heute entwickelten OS-Software und deren Anwender abzu-

²¹ In Anlehnung an (Ploch u.a. 2002, S. 26).

geben²². Ein großer Teil der Anwendungen wurde für die Bereiche Internet, Betriebssystem, Emulatoren, Spiele, Multimedia und Entwicklungs- bzw. Programmierwerkzeuge geschrieben. OSS ist in fast allen Einsatzgebieten zu finden, in denen Software genutzt wird – von der Büroanwendung bis zur Bibel-Lern-Software.

Im folgenden Abschnitt soll eine Auswahl von interessanten und erfolgreichen OS-Projekten präsentiert und Software vorgestellt werden, die sich im privaten, öffentlichen wie unternehmerischen Bereich vielseitig bewährt hat.

Das wohl bekannteste Projekt, das immer wieder auch als Zugpferd der OS-Bewegung bezeichnet wird, ist wohl *GNU/Linux*²³. Es ist ein freies Betriebssystem, das in den letzten Jahren seinen kommerziellen Konkurrenten in vielen Bereichen Marktanteile abringen konnte und das eine hohe Anzahl an freien Softwarepaketen wie z. B. Emacs, Bash, GNU-Compiler-Sammlung, beinhaltet.

Der Name wird oft auf „Linux“ reduziert, was von der FSF-Gemeinde scharf kritisiert wird. Wenn von der Migration auf Linux die Rede ist, ist die Umstellung des Betriebssystems auf GNU(-Software) mit den Linux-Kernel gemeint.

Openoffice ist eine umfangreiche Büroanwendung, die Textverarbeitung, das Erstellen von Präsentationen, Zeichnungen und HTML-Dokumenten, Tabellenkalkulation und die Arbeit mit Datenbanken ermöglicht. Sie ähnelt damit bewusst dem Office-Paket von Microsoft. Etwa 16 Millionen Anwender weltweit nutzen inzwischen diese bislang in 30 Sprachen übersetzte Büro-Software, sie kann auf allen gängigen Betriebssystemen installiert werden. Als kleines Beispiel seiner Funktion mag dieses Buch dienen, das fast ausschließlich mit Openoffice gestaltet wurde.²⁴

Nachdem Netscape den Quellcode seines Browsers freigegeben hat, entstand in einer riesigen virtuellen Gemeinschaft dessen Weiterentwicklung *Mozilla*²⁵. Dieses Programmpaket kann zum Surfen im Internet, zum Schreiben und Lesen von E-Mails und News, zum Chatten, zur Erstellung von Webseiten (per WYSIWYG²⁶-Editor) und sogar zum Verwalten von Adressen und Terminen verwendet werden. Der Browser wurde vom PC-World-Magazin mit dem Titel „Best of 2003, Web Browser“ ausgezeichnet.

*KDE (K Desktop Environment)*²⁷ ist eine grafische Benutzeroberfläche und Arbeitsumgebung für UNIX-PCs (also auch für GNU/Linux). Es beinhaltet eine umfangreiche Sammlung an Anwendungen inklusive eines eigenen Office-Pakets, eines Webbrowsers und einer E-Mail-Anwendung. Das Projekt wurde 1996 in Deutschland gegründet – heute ist KDE die meistgenutzte grafische Benutzeroberfläche auf entsprechenden Desktopsystemen.

²² Eine Vielfalt von Projekten findet sich unter anderem auf den folgenden Webseiten: <http://freshmeat.net>, <http://sourceforge.net>, <http://OSDir.com>, <http://developer.berlios.de>, <http://bioinformatics.org>.

²³ Vgl. <http://www.gnu.org> bzw. <http://www.linux.org>.

²⁴ Vgl. <http://www.openoffice.org>.

²⁵ Vgl. <http://www.mozilla.org>.

²⁶ *what you see is what you get*

²⁷ Vgl. <http://www.kde.de>.

*GNOME (GNU Network Object Model Environment)*²⁸ stellt ebenfalls eine grafische Desktop-Umgebung dar. Das Projekt wurde 1997 als Konkurrent²⁹ zum KDE-Projekt gegründet. Es hat allerdings eine nicht so große Nutzer- und Entwicklergemeinschaft wie das KDE-Projekt.

Die nun folgenden Software-Projekte werden dem Endanwender vielleicht weniger geläufig sein, für Spezialisten sind sie aber heute nicht mehr wegzudenken.

*Samba*³⁰ ist ein File- und Druckserver. Er macht es möglich, auf andere im Netzwerk befindliche Computer (auch auf Windows-Rechner) zuzugreifen, um Dateien auszutauschen oder Druckaufträge abzuschicken.

Den heute verbreitetsten Webserver stellt *Apache*³¹. Ein Webserver ermöglicht es, Seiten der unterschiedlichsten Art im Internet zu veröffentlichen und anzuzeigen. Dieser Server ist ein Projekt der *Apache Software Foundation*, welche noch viele weitere erfolgreiche OS-Projekte betreut.

Zope und *OpenCms*³² sind zwei sehr erfolgreiche Content-Management-Systeme. Sie erleichtern die Verwaltung von Webinhalten (Content, wie z.B. Texte und Bilder für bestimmte Bereiche wie News oder Unternehmens- bzw. Produktinformationen) von entsprechenden Webseiten. *Zope* basiert auf der Programmiersprache Python und *OpenCms* auf Java.

Eines der heute erfolgreichsten Datenbanksysteme ist *MySQL*³³. Neben der Möglichkeit, diese Software unter der freien Lizenz GPL zu nutzen, kann auch, wenn beispielsweise ein kommerzielles Produkt auf Basis des MySQL-Quellcodes entwickelt werden soll, eine entsprechende kommerzielle Lizenz erworben werden.

Die folgenden beiden OS-Projekte haben die Entwicklung des Internets maßgeblich geprägt: So ist *BIND*³⁴ eine Software, die das Domain-Name-System (DNS) unterstützt, welches dazu dient, die vom Menschen lesbaren Netz-Adressen (wie z.B. www.opensource.org) in eine für den Computer nutzbare IP-Adresse (bestehend aus Zahlen) aufzulösen.

*Sendmail*³⁵ ist ein so genannter *Mail Transport Agent (MTA)*. Er dient zur Weiterleitung von E-Mails. Der Marktanteil dieses bereits 1981 entwickelten Programms lag im Jahre 1999 bei ca. 75 %.

Migrationen, Entwicklergemeinschaft und Buch-Rezension: Die Artikel in diesem Kapitel

In den nächsten Abschnitten des ersten Kapitels sollen Erfahrungsberichte über die Migration von Microsoft-Produkten zu Open-Source-Software, ein Bericht über

²⁸ Vgl. <http://www.gnome.org>.

²⁹ Vgl. auch den Artikel von Eva Brucherseifer: „Die KDE-Entwicklergemeinschaft – wer ist das?“, Abschnitt: „3. KDE und der Rest der Welt“ in diesem Band.

³⁰ Vgl. <http://samba.org>, vgl. auch die interessante Nutzerstudie, die unter <http://samba.anu.edu.au/pub/samba/survey> einsehbar ist.

³¹ Vgl. <http://www.apache.org>.

³² Vgl. <http://www.zope.org> und <http://www.opencms.org>.

³³ Vgl. <http://www.mysql.com>.

³⁴ Vgl. <http://www.isc.org>.

³⁵ Vgl. <http://www.sendmail.org>.

die KDE-Gemeinschaft und eine Rezension über Steven Webers demnächst erscheinende Darstellung „Success of Open Source“ vorgestellt werden.

Es wurden drei Migrationsprojekte für das Einleitungskapitel ausgewählt. Davon wurden zwei Projekte vom Bundesamt für Sicherheit in der Informationstechnik (BSI) bei den Bundesbehörden – nach einer Umfrage auf „Migrationsinteresse“ – zum Zweck der Evaluation von OSS europaweit ausgeschrieben und finanziert.

Eines dieser Projekte wird in dem Artikel „Migration auf Open-Source-Software beim Institut für Tierzucht der Bundesforschungsanstalt für Landwirtschaft“ aus der Sicht des mit der Migration beauftragten Unternehmens – der GONICUS GmbH – dargestellt. Der Autor Alfred Schröder beschreibt dabei nicht nur die Umstände des Projekts und inwieweit es Erfolg hatte. Er geht auch auf die vorgefundenen Benutzertypen und -profile ein und zeigt entsprechend gesammelte Erfahrungen. Weiterhin betont er die Wichtigkeit der Benutzer bei einer solch gravierenden Umstellung der Bildschirmarbeitsplätze.

Ein weiteres Projekt wurde bei der Monopolkommission durchgeführt. Für dieses Jahrbuch konnte hierzu ein Bericht aus Sicht des Dienstleisters („Beispiel einer Migration von Windows 2000 auf Open-Source-Software“ von Thomas Sprickmann Kerkerinck), der die entsprechende Open-Source-Software eingeführt hat, gewonnen werden.

Dieser Migrationsbericht ist in ähnlicher Form schon einmal veröffentlicht worden.³⁶ Da aber gerade das Thema Migration auf Open-Source-Software und der Rahmen des durchgeführten Pilotprojektes bzw. die daraus gesammelten Erfahrungen als wegweisend angesehen werden können, durfte er in diesem Jahrbuch nicht fehlen. Aus den gemachten Darstellungen kann der Leser sich ein Bild vom Umfang und von der Komplexität einer solchen Migration machen. Einige Ausführungen werden dem Laien sehr technisch erscheinen, sie stellen aber die Kernidee der Umstellung dar und zeigen außerdem, was mit OSS alles möglich ist.

Die bei der Umstellung der Monopolkommission gewonnenen Erfahrungen wurden beim Landesrechnungshof Mecklenburg-Vorpommern verwendet. Den Anwenderbericht aus diesem Projekt finden Sie im Artikel von Frank Müller: „Migration der Server- und Desktoplandschaft im Landesrechnungshof Mecklenburg-Vorpommern“.

Einen weiteren Bericht aus der Sicht des Benutzers konnte Kerstin Terhoeven beisteuern. Sie betrachtet das Migrationsprojekt in ihrem Aufsatz „Open-Source-Software am Büroarbeitsplatz: Erfahrungen der Endanwender aus der Migration der Geschäftsstelle der Monopolkommission“ eher kritisch. Sie schildert diverse im Vergleich zum vorher eingesetzten System als negativ empfundene Eindrücke. Weiterhin beschreibt sie die Vor- und Nachteile des neuen Systems. Es erschien der Redaktion dieses Buches wichtig, auch die vorhandene Kritik an dieser Thematik abzubilden, nicht zuletzt, um auch die Herausforderungen, die eine Migration mit sich bringt, darzustellen.

Um dem Leser eine OS-Entwicklergemeinschaft näher zu bringen, wurde die Projektleiterin und stellvertretende Vorsitzende des KDE e.V. Eva Brucherseifer

³⁶ S. z. B. <http://www.pl-forum.de/berichte/monopolkommission.html>.

gebeten, als Insiderin dieses Projekt vorzustellen. In dem Artikel „Die KDE-Entwicklergemeinschaft – wer ist das?“ beschreibt sie die Anfänge, überwundene Probleme und die Erfolge. Weiterhin berichtet Eva Brucherseifer sehr detailliert vom Leben in einer solchen Gemeinschaft – insbesondere von den Treffen, die auf Grund des sonst kaum vorhandenen persönlichen Kontakts sehr wichtig sind.

Beendet wird das erste in die Thematik einführende Kapitel durch den Hinweis auf weiterführende Literatur. Dazu dient die von Hendrik Scheider verfasste Rezension des im April dieses Jahres erscheinenden Buches von Steven Weber: „Success of Open Source“.

Literatur

- Kerbusk, Klaus-Peter (2003): *Mauerfall an der Isar*, in: Spiegel 23/2003,
online <http://www.spiegel.de/spiegel/0,1518,251077,00.html> (24.1.2004).
- Bundestux(2002): *Wir können uns das gut vorstellen!*,
online http://www2.bundestux.de/bundestux_alt/erklaerung.html
(24.1.2004).
- Bundestux (2003): *München als Vorhut der Informationsgesellschaft*,
online <http://www.bundestux.de/themen/inlkom/12914.html> (24.1.2004).
- Free Software Foundation (2002): *Die Definition Freier Software*,
online <http://www.gnu.org/philosophy/free-sw.de.html> (24.1.2004).
- Grassmuck, Volker (2002), *Freie Software. Zwischen Privat- und Gemeineigentum*, Bundeszentrale für politische Bildung, Bonn 2002.
- Levy, Steven (2001): *Hackers, Heroes of the Computer Revolution*, Penguin Books 2001.
- Open Source Initiative (2004): *The Open Source Definition – Version 1.9*,
online www.opensource.org/docs/definition.php (24.1.2004).
- O'Reilly (1999): *Open Source – kurz und gut*, O'Reilly 1999,
online http://www.oreilly.de/german/freebooks/os_tb/ (24.1.2004).
- Ploch, Danuta; Stewin, Patrick; Koch, Ramona: *Einführung in Open Source Software*, Technische Universität Berlin – Informatik und Gesellschaft 2002,
online <http://ig.cs.tu-berlin.de/w2002/ir1/uebref/PlochEtAl-Referat-G1-Referat-122002.pdf> (24.1.2004).
- Stallman, Richard (2001): *Das GNU Projekt*, Originalveröffentlichung in dem Buch „Open Sources“, übersetzt 2003 von Stephan Knuth, Free Software Foundation 2003,
online <http://www.gnu.org/gnu/thegnuproject.de.html> (24.1.2004).